Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Máster en Ingeniería Industrial

MÁSTER EN INGENIERÍA INDUSTRIAL

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Diseño e Implementación en FPGA de un sistema con Sensores Integrados

Autor: D. Kevin André Gutiérrez Yangali
Tutor: D. Francisco José de Andrés Rodríguez-Trelles

Valladolid, mayo, 2019

RESUMEN

El objetivo de este proyecto es el diseño y la implementación de sistemas que sean capaces de realizar lecturas de sensores integrados y posteriormente realizar la visualización de la medida a través de displays y LEDs. El usuario podrá controlar el sistema por medio de pulsadores y selectores.

Para la realización de este proyecto se utilizará un dispositivo FPGA Lattice modelo MachX02-1200ZE, un sensor de ultrasonidos modelo LV-MaxSonar-EZ3 y un convertidor de luz-frecuencia modelo TCS3210.

Por una parte, se realiza la configuración necesaria para la medición de distancias con el sensor de ultrasonidos. La medida se presenta mediante tres dígitos, expresada en centímetros y con una actualización de un segundo.

Por otra parte, se realiza la configuración para la medición del nivel de iluminación a través del convertidor de luz-frecuencia. La medida obtenida, que se expresa con tres dígitos, varía en función de la luz incidente y se actualiza cada segundo.

Palabras clave: FPGA, VHDL, sensor, ultrasonidos, luz-frecuencia.

ABSTRACT

The objective of this project is the design and implementation of systems that are capable of performing readings of integrated sensors and then make the visualization of the measurement through displays and LEDs. The user can control the system by means of pushbuttons and switches.

For the realization of this project will be used a FPGA Lattice device model MachX02-1200ZE, an ultrasonic sensor model MaxSonar-LV-EZ3, and a light-frequency converter model TCS3210.

On the one hand, the configuration for distance measurement with the ultrasonic sensor is carried out. The measurement is presented by three digits, expressed in centimeters and with an update of one second.

On the other hand, the configuration is made for the measurement of the level of illumination through the light-frequency converter. The measurement obtained, which is expressed with three digits, varies depending on the incident light and is updated every second.

Keywords: FPGA, VHDL, sensor, ultrasonic, light-frequency.

CONTENIDO

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

Actualmente hay tres alternativas en la realización de sistemas digitales: sistemas basados en microcontroladores, utilización de dispositivos lógicos reconfigurables (FPGA) y desarrollo de circuitos integrados de aplicación específica (ASICs). La última alternativa (ASICs) solo tiene sentido para volúmenes de fabricación muy grandes.

Si se realiza un sistema basado en microcontroladores, éste se deberá encargar de realizar tanto el control de la adquisición de la información proporcionada por los sensores, como de su tratamiento. En cambio, si se emplea un dispositivo tipo FPGA, se puede dedicar una parte de su lógica, de propósito general, al control de la interface con los sensores, y otra parte a la implantación de un procesador dedicado, cuya misión será, exclusivamente, el tratamiento de dicha información.

1.2. Objetivos

El objetivo fundamental de este trabajo es realizar el diseño y la implantación de la interface de comunicación y un pretratamiento de la información suministrada por diversos sensores, en la lógica de propósito general de un dispositivo FPGA.

Para conseguir este objetivo, se deben alcanzar una serie de objetivos parciales como los mostrados a continuación:

- ✓ Selección de un dispositivo tipo FPGA.
- ✓ Selección y aprendizaje de un software de diseño.
- ✓ Profundizar en el conocimiento del lenguaje de descripción de hardware VHDL.
- ✓ Emplear técnicas de diseño mixtas, utilizando lenguajes de descripción de hardware y esquemas.
- ✓ Empleo de herramientas hardware y software para la puesta a punto del sistema (analizador lógico).
- ✓ Selección de los sensores.
- ✓ Analizar los resultados y sacar conclusiones del trabajo realizado.

1.3. Organización de la memoria

En el CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS se realiza una introducción general del trabajo y se plantean los objetivos a alcanzar en el desarrollo del mismo.

En el CAPÍTULO 2: SELECCIÓN Y DESCRIPCIÓN DE LOS DISPOSITIVOS se realiza la selección de los dispositivos, tanto de la FPGA como de los sensores, y se explican las principales características de los modelos elegidos.

En el CAPÍTULO 3: IMPLEMENTACIÓN se desarrolla el proceso de configuración de la FPGA para que se alcancen los objetivos planteados.

En el CAPÍTULO 4: RESULTADOS se presentan los resultados de la implementación a través del empleo de un analizador lógico y por medio de una serie de pruebas de mediciones.

En el CAPÍTULO 5: ESTUDIO ECONÓMICO se realiza un análisis de costes asociados con la realización del proyecto.

En el CAPÍTULO 6: CONCLUSIONES se extraen las conclusiones a las que se llegan tras la realización del trabajo y se presentan algunas líneas futuras de estudio.

En el CAPÍTULO 7: BIBLIOGRAFÍA se proporciona la fuente de información que se ha utilizado para la elaboración del presente trabajo.

En el CAPÍTULO 8: ANEXOS se adjunta algunos documentos con información adicional sobre los dispositivos utilizados.

2. SELECCIÓN Y DESCRIPCIÓN DE LOS DISPOSITIVOS

En este apartado se pretende presentar los dispositivos que se han seleccionado para la realización de este proyecto, así como sus características más importantes.

2.1. FPGA (Field Programmable Gate Array)

Las FPGA fueron inventadas en el año 1984 por Ross Freeman y Bernard Vonderschmitt, cofundadores de Xilinx.

Son el resultado de la convergencia de dos tecnologías diferentes, los dispositivos lógicos programables (PLD Programmable Logic Devices) y los circuitos integrados de aplicación específica (ASIC Application-Specific Integrated Circuit).

Una FPGA (acrónimo en inglés que significa matriz de puertas reprogramable) es un circuito integrado que contiene matrices con bloques de lógica y se compone principalmente de cables, puertas lógicas, biestables, y puertos de entrada y salida.

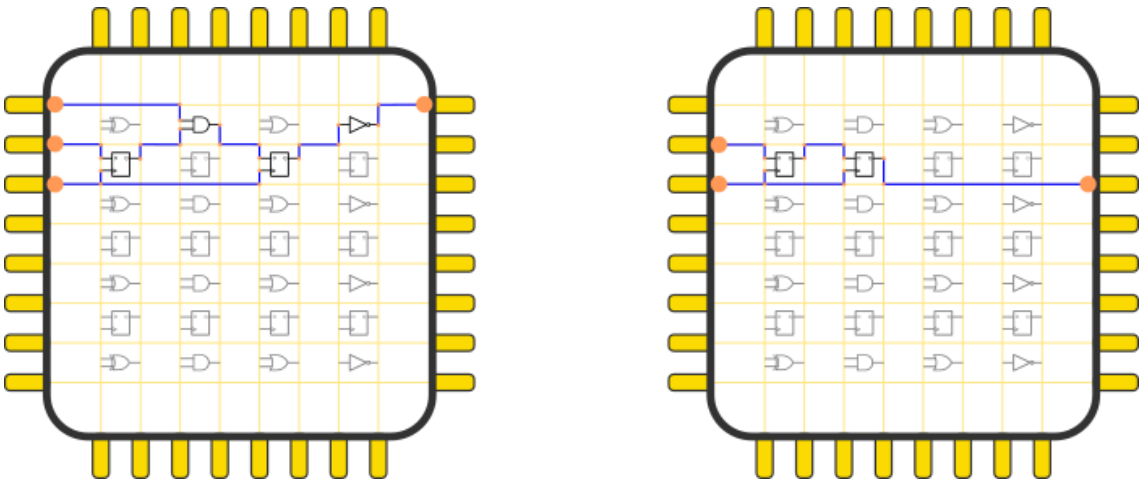


Figura 1. Estructura interna de una FPGA

Las conexiones son programables, esto quiere decir que se puede configurar qué elementos se unen a cuáles, formando así el circuito digital deseado y si se establecieran otras conexiones diferentes, se obtendría un circuito diferente. También se pueden configurar las conexiones con los pines de la FPGA, determinando por qué pines entra y por cuáles sale la información.

Cada una de las conexiones tiene asociado un bit de configuración que determina su estado: Conectado (1) o No Conectado (0). Todos los valores para los bits de configuración se agrupan en una tira de bits llamada bitstream, que se carga desde el exterior (ordenador).

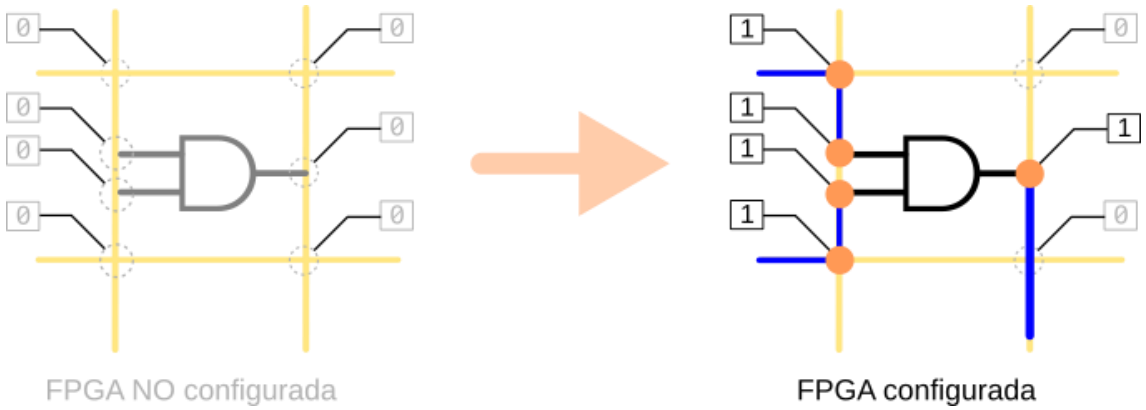


Figura 2. Ejemplo de conexión mediante bits de configuración

La mayor parte de las FPGA son volátiles, de modo que al quitar la alimentación pierden su configuración y cuando se vuelven a conectar, están en blanco y se quedan esperando a recibir un bitstream para reconfigurarse. Por eso, junto a la FPGA se sitúa una memoria externa no volátil, llamada memoria de configuración, que almacena el bitstream. Así, al alimentar la FPGA lo primero que hace es reconfigurarse con el bitstream de la memoria de configuración.

En el caso de este proyecto, se utiliza una FPGA con una configuración no volátil.

Para implementar un diseño en un dispositivo FPGA se hace uso de un lenguaje de programación especial conocido como lenguaje de descripción de hardware (HDL Hardware Description Language).

Los lenguajes HDL más conocidos y utilizados son:

- VHDL (Very High Speed Integrated Circuit Hardware Description Language)
- Verilog
- ABEL (Advanced Boolean Expression Language)

2.1.1. Fabricantes

En el mercado de las FPGA, los fabricantes más importantes actualmente son Xilinx, Altera (Intel) y Lattice Semiconductor.

Xilinx:

Dispone de 8 series de productos: Spartan, Artix, Kintex, Virtex, Kintex UltraSCALE, Virtex UltraSCALE, Kintex UltraSCALE+ y Virtex UltraSCALE+.

La gama Artix es la de menor rendimiento, pero también menor coste, Kintex tiene prestaciones y precio intermedios, y por último Virtex es la de mejores características y mayor precio.

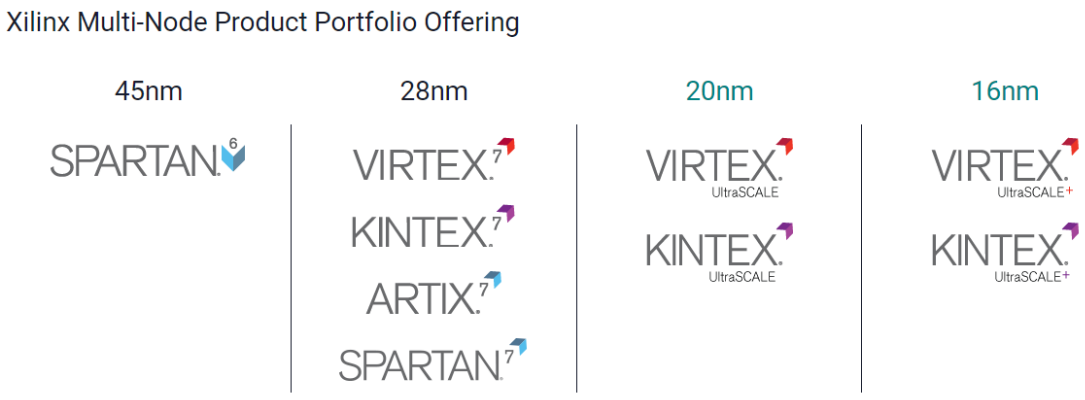


Figura 3. Clasificación de productos Xilinx en función de la densidad de puertas lógicas

Xilinx dispone de una herramienta de software de configuración y diseño electrónico propio llamado ISE Design Suite, utilizado para el análisis y síntesis de diseños realizados en HDL.

Altera (Intel):

Intel adquirió Altera el 1 de junio de 2015. Dispone de tres grandes familias de FPGA:

- *Cyclone Series:* El modelo más básico de FPGA de Altera. Precio reducido y poder de cálculo moderado.
- *Arria Series:* Modelo intermedio. Proporciona un balance óptimo entre precio y prestaciones.
- *Stratix Series:* Modelo de gama alta de Altera. Con gran poder de computación, pero con elevado precio.

Altera dispone de una herramienta de software de configuración propio llamado Quartus Prime, utilizado para el análisis y síntesis de diseños realizados en HDL.

Lattice Semiconductor:

Empresa líder en diseño de circuitos integrados programables de ultra bajo consumo. Entre sus familias de productos de FPGA, existe multitud de opciones dependiendo del nivel de prestaciones y precios. Las principales gamas de productos son:

- *ECP5:* Dispositivo de baja densidad, poco consumo y precio asequible. Ideal para aplicaciones con gran nivel de producción donde el coste es un factor determinante.
- *iCE40:* Dispositivo con rendimiento y precio moderados. Disponible en tres tipos: Bajo consumo (LP), bajo consumo con IP embebido (LM) y alto rendimiento (HX).
- *MachX03:* Dispositivos con memoria no volátil. Poseen la mayor densidad de puertos E/S de los dispositivos de bajo coste.
- *MachX02:* Dispositivo perfecto para la implementación rápida de sistemas de control. Ofrece fiabilidad a bajo precio.
- *LatticeECP3:* Solución eficiente para aplicaciones con requisitos muy altos.
- *ispMACH 4000ZE:* Diseñada para aplicaciones móviles de ultra-bajo consumo.

Lattice Semiconductor dispone de una herramienta de software de configuración propio llamado Lattice Diamond, utilizado para el análisis y síntesis de diseños realizados en HDL.

2.1.2. MachXO2-1200ZE

Para el desarrollo del presente proyecto se ha elegido una FPGA de la marca Lattice Semiconductor, concretamente el modelo MachXO2-1200ZE.

A continuación, se explica el significado de las letras que dan nombre al dispositivo.

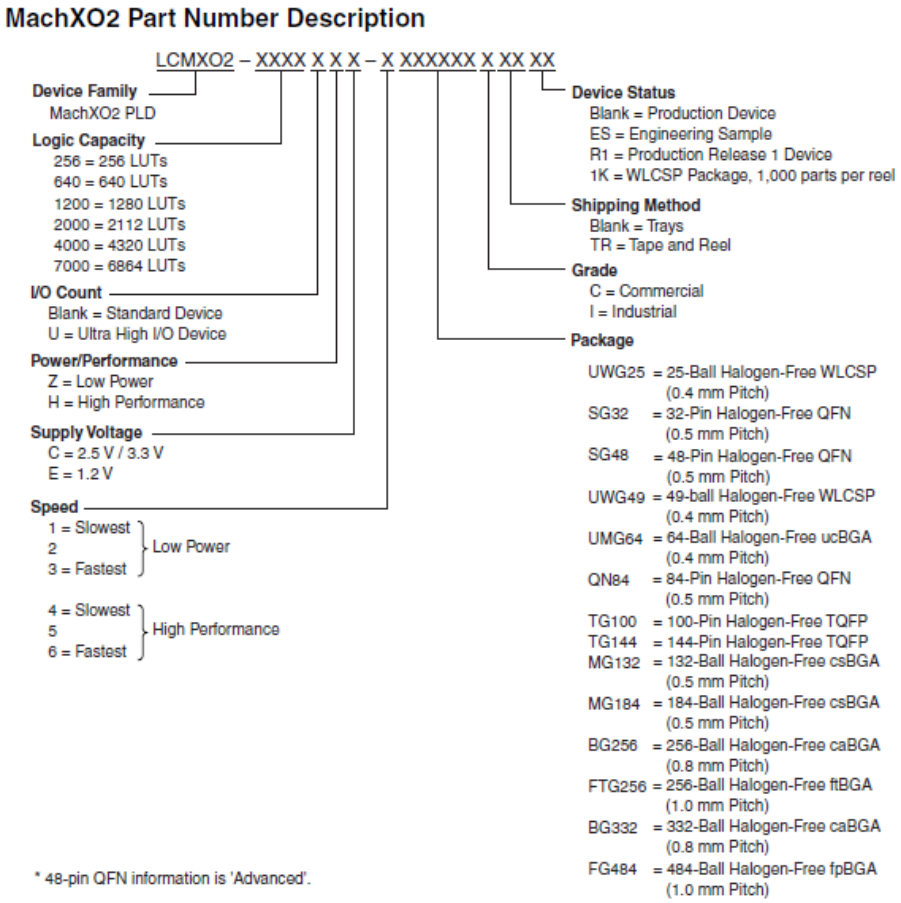


Figura 4. Nomenclatura de los equipos de la gama MachXO2

Según la figura anterior, las características del dispositivo son las siguientes:

- *MachOX2:* Familia a la que pertenece la FPGA.
- *1200:* Capacidad lógica de 1280 LUTs.
- *Z:* Dispositivo de bajo consumo.
- *E:* Voltaje de alimentación de 1,2 V.

La FPGA elegida es no volátil y no precisa de elementos adicionales para su arranque, como memorias ROM de apoyo. Posee diversidad de memorias integradas, integra 138 kbits de memoria distribuidos de la siguiente forma:

- 64 kbits de memoria Flash. Memoria no volátil, de propósito general, que permite al usuario, poder almacenar información propia.
- 64 kbits de memoria RAM dedicada. La memoria RAM dedicada sólo puede ser usada como almacenamiento volátil de datos. Dicho almacenamiento puede ser empleado, cuando se implementa, por ejemplo, un microprocesador en la FPGA.
- 10 kbits de memoria RAM distribuida. La memoria RAM distribuida puede ser empleada, tanto para memoria RAM, como en el caso de la memoria RAM dedicada, como para la implementación de lógica.

En cuanto a la arquitectura, la familia MachX02 se compone de una matriz de bloques lógicos rodeados por E/S programables (PIO). Contienen también bloques sysCLOCK PLLs y bloques sysMEM RAM embebidos (EBRs).

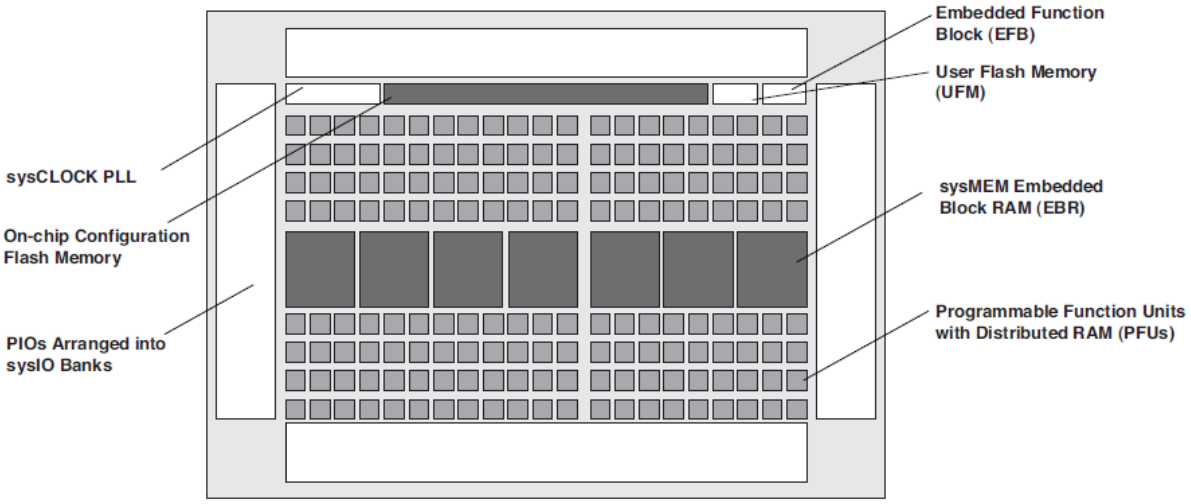


Figura 5. Arquitectura básica de la familia de dispositivos MachX02-1200

Los bloques lógicos, Unidad Funcional Programable (UFP) y sysMEM EBR, están situados en una cuadrícula de dos dimensiones (filas y columnas). Cada fila puede tener bloques lógicos o bloques de EBR. Las células PIO están situadas en la periferia del dispositivo. La UFP contiene los bloques correspondientes a la lógica, aritmética, RAM, ROM y registros. Las PIO utilizan un buffer de E/S flexible compatible con varios interfaces estándar. Los bloques se interconectan entre sí mediante pistas horizontales y verticales. La herramienta de diseño software se encarga de asignar automáticamente estos recursos de enrutamiento.

Todos los dispositivos MachX02 incluyen un módulo EFB (Embedded Function Block). Este bloque permite implementar funciones de control que facilitan el diseño y permiten ahorrar recursos de uso general como LUTs, registros, señales de reloj y rutado. Contiene los siguientes elementos:

- Dos núcleos I2C
- Un núcleo SPI
- Un timer/counter de 16 bits
- Interfaz para configurar las características dinámicas PLL
- Interfaz para configurar la lógica
- Interfaz para controlar la potencia del chip
- Memoria flash de usuario (UFM)

Una ventaja que presenta este dispositivo es que hay disponible un software gratuito de la propia empresa Lattice Semiconductor para programar sus dispositivos. El programa es Lattice Diamond, que permite el diseño, síntesis, análisis, configuración, debugger, simulación e implementación de esquemas y circuitos electrónicos.

Breakout Board:

El dispositivo FPGA está implementado dentro de una Breakout Board MACHX02 del fabricante Lattice.

Cada pin de E/S del dispositivo FPGA está conectado a un orificio de 100 milésimas de pulgada para poder conectar los diseños propios del usuario, que suelen estar en PCB externas.

El tamaño de la Breakout Board es de 7,5 x 7,5 cm. Cuenta en la parte superior con una matriz de diodos led, una matriz de prototipos para el desarrollo de pruebas y un puerto USB Mini-B por el que se conectará al ordenador y se podrá configurar la FPGA desde un software de diseño externo.

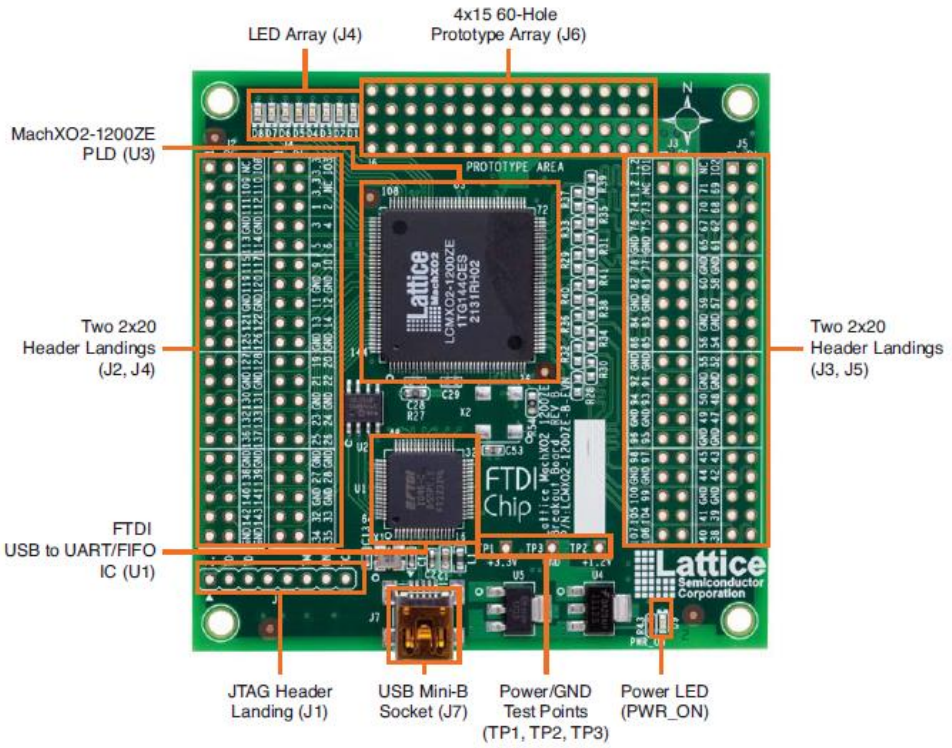


Figura 6. MachXO2-1200ZE Breakout Board

A esta Breakout Board se le ha añadido 2 bloques adicionales para lo comunicación con el usuario, que son la placa de displays y la placa de switch y pulsadores.

Placa PCB Display:

Esta placa contiene dos displays de siete segmentos, con 16 resistencias que limitan la intensidad que circula por cada segmento.

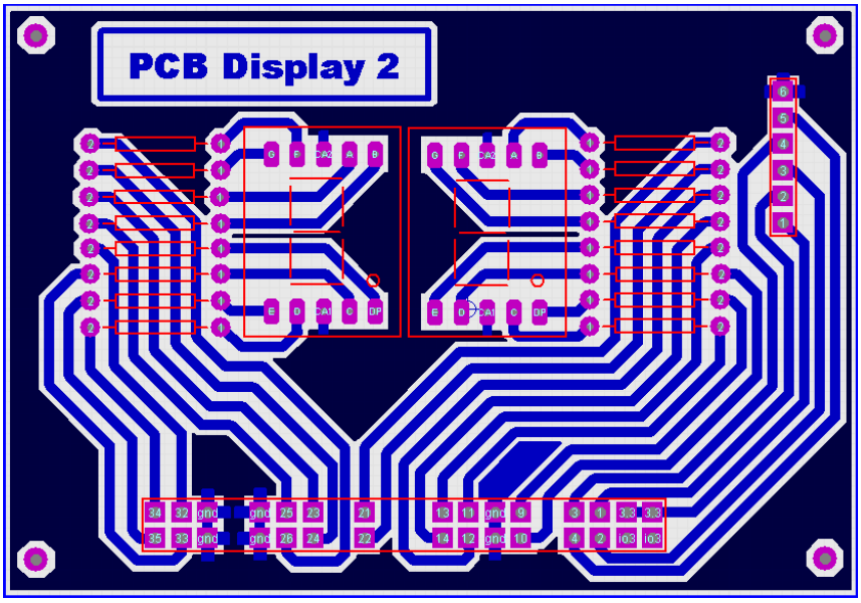


Figura 7. Conexión de placa PCB Display

Tiene un modelo de conexión de cátodo común.

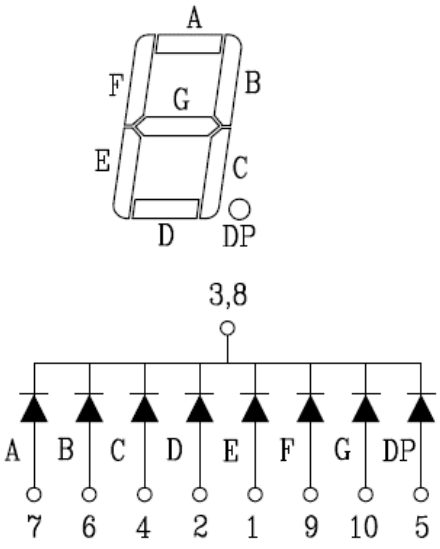


Figura 8. Tipo de conexión del display de 7 segmentos

La asignación de los pines de la FPGA para los dos displays es la siguiente:

Segmentos	Pin de la FPGA	
	Display1 (Derecha)	Display2 (Izquierda)
A	21	23
B	22	25
C	11	33
D	10	34
E	13	35
F	12	26
G	14	24
DP	9	32

Figura 9. Asignación de pines para los displays

Placa PCB Switch & Pulsadores:

Esta placa contiene ocho micro-interruptores y ocho micro-pulsadores para poder aplicar opciones de control de una manera simple al dispositivo FPGA.

Al subir a 'ON' un microswitch o presionar un pulsador el estado de la entrada correspondiente en la FPGA se pone a nivel alto (3,3 V), de lo contrario se pone a nivel bajo (0 V).

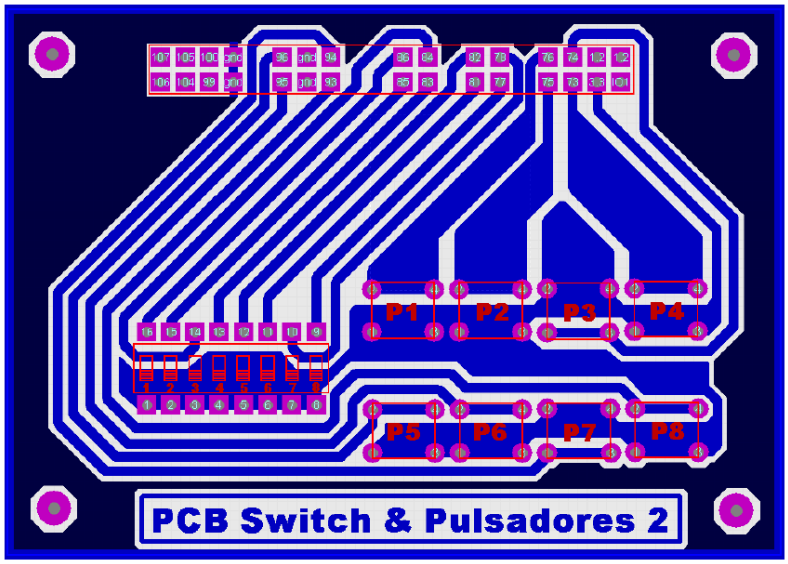


Figura 10. Conexión de placa PCB Switch & Pulsadores

La asignación de los pines de la FPGA para los interruptores y pulsadores es la siguiente:

Interruptor	Pin de la FPGA	Pulsador	Pin de la FPGA
SW1	86	PULS1	76
SW2	85	PULS2	75
SW3	84	PULS3	74
SW4	83	PULS4	73
SW5	82	PULS5	96
SW6	81	PULS6	95
SW7	78	PULS7	94
SW8	77	PULS8	93

Figura 11. Asignación de pines para los interruptores y pulsadores

Reloj Externo y LEDs:

Se ha agregado a la Breakout Board un reloj externo de 50 MHz. El reloj dispone de una patilla de habilitación CLK_EN que va al PIN 32 de la FPGA y la salida del reloj CLK_OUT va al PIN 27 de la FPGA.

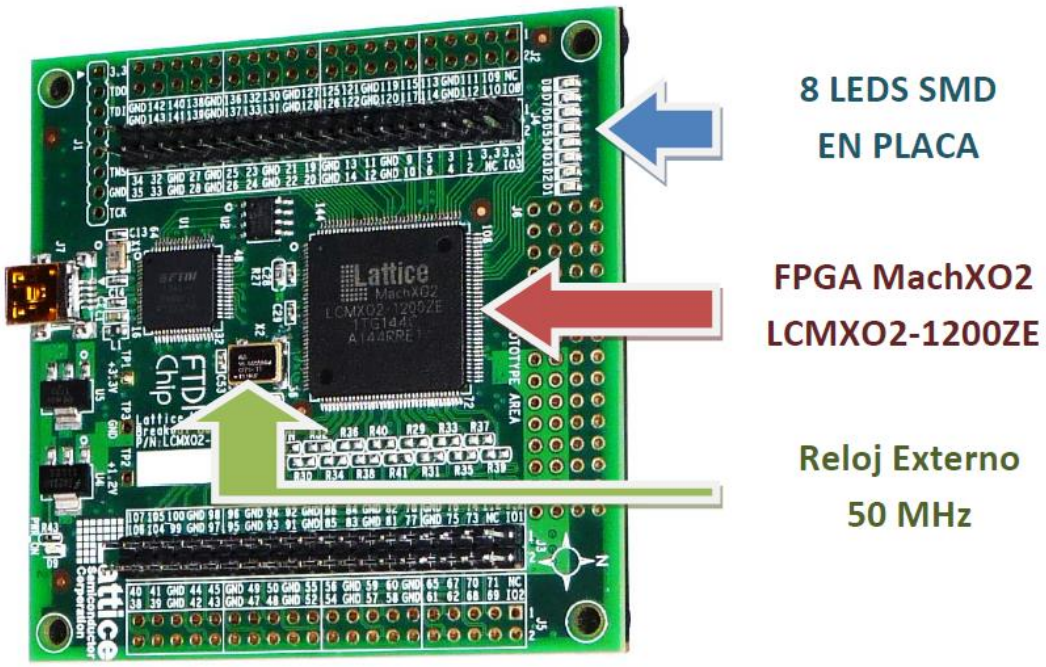


Figura 12. Reloj externo y LEDs en la Breakout Board

La asignación de los pines de la FPGA para los LEDs es la siguiente:

LEDs	Pin de la FPGA
D1	97
D2	98
D3	99
D4	100
D5	104
D6	105
D7	106
D8	107

Figura 13. Asignación de pines para los LEDs

2.2. Sensores

En este trabajo se buscan sensores con interface "no estándar" y que no necesiten un convertidor A/D. De esta forma existe la posibilidad de extenderlo a otros sensores con una interface semejante.

Se utilizan dos tipos de sensores: un sensor de ultrasonidos para medir distancias y un convertidor de luz-frecuencia para medir el nivel de iluminación.

2.2.1. LV-MaxSonar-EZ3

El sensor de ultrasonidos elegido es el LV-MaxSonar-EZ3, fabricado por la empresa MaxBotix Inc.

Figura 14. Sensor LV-MaxSonar-EZ3

Se trata de un módulo de 19,9 x 22,1 mm que contiene integrado emisor y receptor. Sus principales características son las siguientes:

- Rango de detección: Desde 6 hasta 254 pulgadas (15,2 cm hasta 6,45 metros) de distancia.
- Resolución: 1 pulgada (2,54 cm).
- Modos de funcionamiento: Se puede utilizar tres formatos de salidas (ancho de pulso PW, salida en tensión analógica AN y salida digital en serie TX).
- Rango de alimentación: Desde 2,5 V hasta 5 V.
- Consumo: 2 mA
- Frecuencia máxima de funcionamiento: 20 Hz (se pueden tomar medidas cada 50 ms).

El sensor viene soldado a una PCB con siete pines, que se explican a continuación:

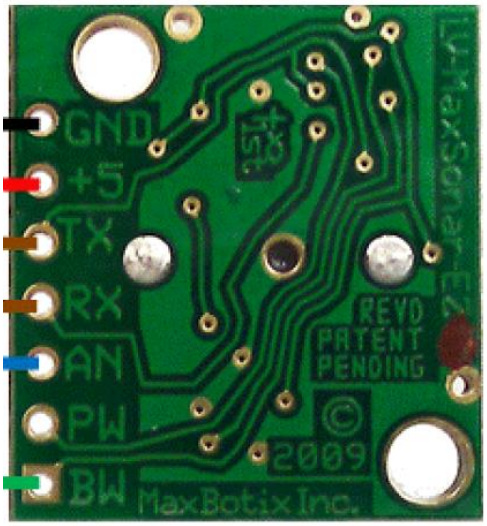


Figura 15. Vista posterior de la placa del sensor LV-MaxSonar-EZ3

- **Pin 1 (BW):** Sirve para la salida digital en serie.
- **Pin 2 (PW):** Este pin emite una representación del rango del ancho de pulso. La distancia se puede calcular utilizando el factor de escala de 147 µs/pulgada.
- **Pin 3 (AN):** Produce como salida un voltaje analógico, con un factor de escala de (Vcc/512) por pulgada. Con una fuente de 5 V produce ~9,8 mV/pulgada mientras que con una de 3,3 V produce ~6,4 mV/pulgada.
- **Pin 4 (RX):** Este pin está internamente forzado a 1. El sensor medirá continuamente distancias, y dará una salida si RX no se cambia de valor o está a 1. Si se deshabilita esta señal, el sensor dejará de medir y emitir salidas. Hay que habilitarlo durante al menos 20 µs para que el sensor comience a medir.
- **Pin 5 (TX):** Este pin es una salida del sensor, comunica el valor de la medición mediante un protocolo serie RS232.
- **Pin 6 (+5V o Vcc):** Pin de alimentación del sensor. Funciona entre 2,5 V y 5,5 V. Capacidad de corriente recomendada de 3 mA para 5 V, y 2 mA para 3 V.
- **Pin 7 (GND):** Conexión de retorno a tierra. Sirve tanto para retornar el voltaje como para eliminar el ruido. El fabricante recomienda que sea libre de ruido y no se conecte con otras tierras para una operación óptima.

En la placa PCB Display se han colocado 6 conectores para la conexión de placas adicionales. El sensor se conectará a esta placa de ampliación.

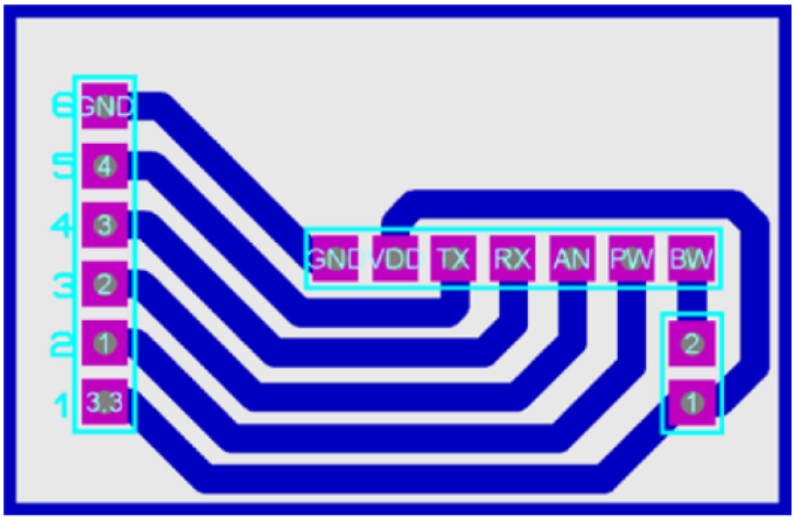


Figura 16. Conexión de la placa PCB del sensor LV-MaxSonar-EZ3

La asignación de los pines de la FPGA para el sensor de ultrasonidos es la siguiente:

Pin del sensor	Pin de la FPGA
PW	1
AN	2
RX	3
TX	4

Figura 17. Asignación de pines para el sensor LV-MaxSonar-EZ3

2.2.2. TCS3210

El convertidor de luz-frecuencia elegido es el TCS3210, fabricado por la empresa TAOS (Texas Advanced Optoelectronic Solutions). Se ha elegido este sensor porque se busca un dispositivo que dé como salida una frecuencia variable.

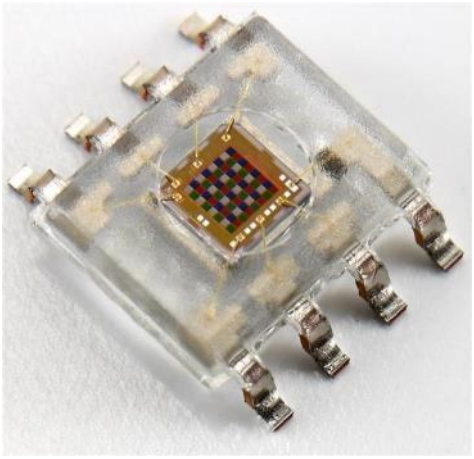


Figura 18. Sensor TCS3210

Es un dispositivo basado en un oscilador controlado por tensión que proporciona una señal de onda cuadrada cuya frecuencia varía dependiendo de la cantidad de luz que incide sobre él.

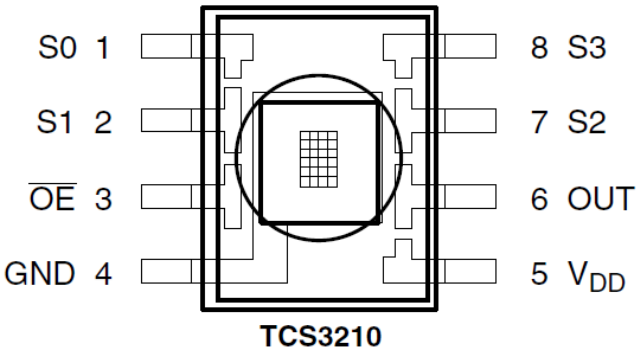


Figura 19. Disposición de los pines del sensor TCS3210

Este sensor posee una matriz de 24 fotodiodos, de los cuales 6 son sensibles a la luz roja, 6 a la luz azul, 6 a la luz verde y 6 a cualquier tipo de luz. Esta matriz genera una corriente u otra dependiendo de la intensidad luminosa incidente sobre ella. Esta corriente pasará por un bloque convertidor corriente frecuencia que generará la salida final del sensor.

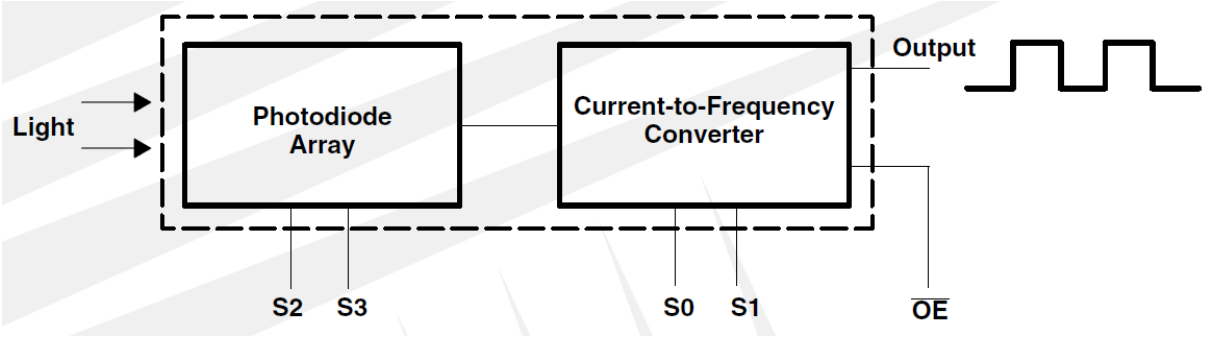


Figura 20. Esquema de funcionamiento del sensor TCS3210

Este chip cuenta con cuatro entradas de control (S3, S2, S1 y S0) y una señal de habilitación (OE).

Con los pines S0 y S1 se configura la frecuencia y con los pines S2 y S3 el color. A continuación, se muestran las tablas que determinan el comportamiento del convertidor.

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Figura 21. Valores de los pines de control del sensor TCS3210

Este sensor se conectará en la ampliación que se encuentra en la placa PCB Display.

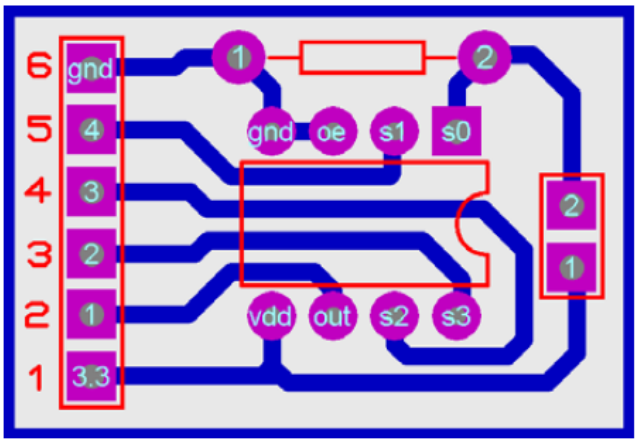


Figura 22. Conexión de la placa PCB del sensor TCS3210

La asignación de los pines de la FPGA para el convertidor luz-frecuencia es la siguiente:

Pin del sensor	Pin de la FPGA
OUT	1
S3	2
S2	3
S1	4

Figura 23. Asignación de pines para el sensor TCS3210

Para definir el valor del pin S0 se dispone de un jumper, de modo que si el jumper está cerrado S0 vale 1 y si se retira el jumper entonces S0 vale 0.

3. IMPLEMENTACIÓN

En este apartado se hablará de los pasos seguidos para realizar la configuración utilizando el software Lattice Diamond 3.10.3.144.

Este software posee algunas herramientas que se han utilizado durante la realización de este proyecto. Las herramientas que se han utilizado son IPexpress, Spreadsheet View, Programmer y Active-HDL Lattice Edition.

- **IPexpress:** Esta herramienta permite generar bloques Verilog o VHDL personalizados. Para incluir un bloque en un esquema, sólo es necesario generar el fichero IPX y añadirlo.
- **Spreadsheet View:** Permite la edición de diferentes parámetros de las señales del circuito. A través de esta herramienta se realiza la asignación de pines.
- **Active-HDL Lattice Edition:** Este módulo permite realizar simulaciones para comprobar el correcto funcionamiento de los programas.
- **Programmer:** Una vez finalizado el diseño y generado el fichero JEDEC, esta herramienta permite transferir la información del programa al dispositivo FPGA.

Existen tres alternativas para realizar el diseño: utilizar sólo descripción VHDL, utilizar sólo esquemas y una mezcla de VHDL con esquemas. Para este proyecto se ha optado por la tercera opción que mezcla la configuración en lenguaje VHDL con el diseño a través de esquemas. De esta forma, algunos elementos se describen en código VHDL y se genera un símbolo para poder añadirlos posteriormente en la configuración por esquemas.

A modo de visión general, se va a construir un contador BCD de 3 dígitos con una entrada de habilitación, una entrada de reloj y una entrada de reset. A este elemento se le asignará diferentes entradas en función del sensor utilizado, para cumplir con el objetivo de realizar la lectura de los mismos.

Para el sensor de ultrasonidos, se realiza un reset al contador al principio del ciclo de medida. Inmediatamente después se lanza la medición, es decir, se habilita el contador para que empiece a contar con una frecuencia que se calcula específicamente para que un pulso equivalga a un centímetro. De esta forma, el valor que devuelve el contador es equivalente a la distancia entre el sensor y el objeto a medir.

En el caso del convertidor de luz-frecuencia, el reset y la habilitación del contador se realizan alternativamente cada segundo, es decir, en cada ciclo de medida, la mitad de tiempo se realiza el reset y la otra mitad de tiempo se habilita el contador. En este caso la entrada de reloj que recibe el contador es la onda cuadrada que proporciona como salida el convertidor de luz-frecuencia. La frecuencia de esta onda es función de la luz incidente en la matriz de fotodiodos y, por lo tanto, el valor que devuelve el contador es función de esta luz incidente.

3.1. Lectura del sensor de ultrasonidos

Se pretende realizar una lectura de la distancia medida por el sensor de ultrasonidos y mostrar el resultado a través de los leds de la placa y los displays disponibles. La información de distancia se actualizará cada segundo y se mostrará en centímetros.

Para la lectura del sensor de ultrasonidos se ha utilizado la siguiente organización de carpetas:

- Lectura_Sensor_Ultrasonidos (Creado como Proyecto General)
  - └ Conv_BCD_7SEG (Creado como Proyecto)
  - └ Conv_BCD_Leds (Creado como Proyecto)
  - └ Generador_Eventos (Creado como Proyecto)
  - └ Memoria (Creado como Proyecto)

3.1.1. Convertidor BCD-7segmentos

Para poder visualizar datos numéricos por los displays que hay disponibles, es necesario crear un elemento capaz de convertir los números que están en formato BCD al formato adecuado para su posterior muestra por los displays de 7 segmentos.

Esto se logra con la realización de una descripción en lenguaje VHDL que tendrá como entrada la codificación BCD y como salida los valores que debe tener cada segmento para que se forme el número correspondiente y sea comprensible por el usuario.

El código VHDL generado es el siguiente:

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity Conv_BCD_7SEG is
5  port(
6       ENT: in std_logic_vector (3 downto 0);
7       SAL: out std_logic_vector (6 downto 0)
8       );
9  end Conv_BCD_7SEG;
10
11 architecture Conv_BCD_7SEG_arch of Conv_BCD_7SEG is
12 begin
13      SAL <=  "1111110" when (ENT="0000") else    --0
14              "0110000" when (ENT="0001") else    --1
15              "1101101" when (ENT="0010") else    --2
16              "1111001" when (ENT="0011") else    --3
17              "0110011" when (ENT="0100") else    --4
18              "1011011" when (ENT="0101") else    --5
19              "1011111" when (ENT="0110") else    --6
20              "1110000" when (ENT="0111") else    --7
21              "1111111" when (ENT="1000") else    --8
22              "1111011" when (ENT="1001");        --9
23 end Conv_BCD_7SEG_arch;
```

Figura 24. Descripción VHDL del Convertidor BCD-7segmentos

Se realiza una simulación con la ayuda del módulo Active-HDL Lattice Edition para comprobar el funcionamiento del código generado.

Se introduce como entradas los valores 0, 1 y 2 y se observa que las salidas son las que corresponden con los segmentos que se deben activar para la muestra por pantalla de los números introducidos.

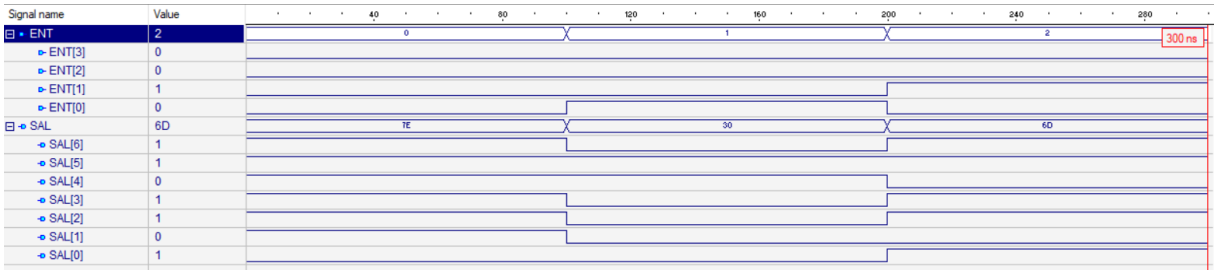


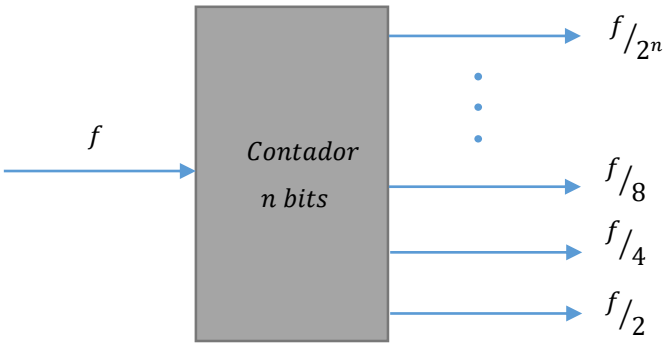
Figura 25. Simulación del convertidor BCD-7segmentos

Una vez comprobado el correcto funcionamiento del convertidor BCD-7segmentos, se genera el símbolo correspondiente para poder utilizarlo en el esquema.

3.1.2. Generación del reloj de 1 Hz

El siguiente paso es generar un reloj que oscile a una frecuencia de 1 Hz. En este paso no es necesario una precisión alta para la obtención de la frecuencia deseada, esto significa que la frecuencia que se va a obtener no será exactamente 1 Hz, sino que será aproximado. Debido a esto se puede utilizar un contador que realice la función de divisor de frecuencia tomando como fuente el oscilador interno de la FPGA (2,08 MHz) y así obtener el reloj de aproximadamente 1 Hz de frecuencia.

Las frecuencias de las múltiples salidas de los contadores tienen una relación con la frecuencia de la señal de entrada y el tamaño de estos contadores, lo que permite poder utilizarlos como divisores de frecuencia.



De modo que el cálculo del tamaño del contador necesario se realiza de la siguiente forma:

$$f = \frac{2{,}08 \cdot 10^6\ Hz}{2^n} \cong 1\ Hz\ \rightarrow \boldsymbol{n = 21\ bits}$$

Con la ayuda de la herramienta IPexpress se genera un contador de 21 bits, del cual se utilizará la salida correspondiente al bit número 20, ya que esta señal es la que tiene una frecuencia aproximada de 1 Hz.

Como entrada al contador de 21 bits se utiliza la señal proveniente del oscilador interno de la FPGA que se habilita a nivel bajo con una señal permanente. Las entradas de habilitación y reset de este contador se gobernarán desde los interruptores de la placa. Finalmente, la salida se sacará por el led D8 de la Breakout Board para comprobar el correcto funcionamiento.

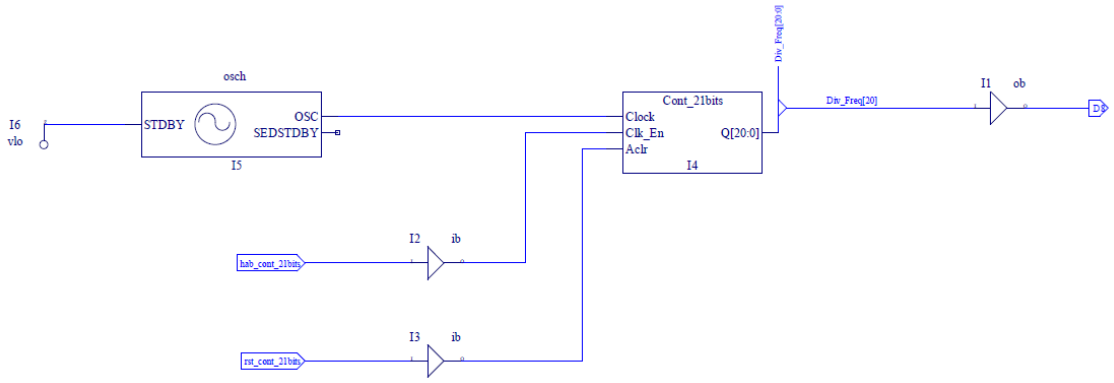


Figura 26. Esquema de generación del reloj de 1 Hz

3.1.3. Generación del contador BCD

Contador BCD de 1 dígito:

El objetivo es utilizar la señal de 1 Hz, generada anteriormente, como entrada de reloj del contador BCD, que se creará con la herramienta IPexpress. Este contador será de 4 bits, contará de forma ascendente de 0 a 9 y tendrá las entradas de habilitación y reset gobernadas por los interruptores de la placa.

La salida del contador se llevará al convertidor BCD-7segmentos y después se mostrará por el display de la derecha.

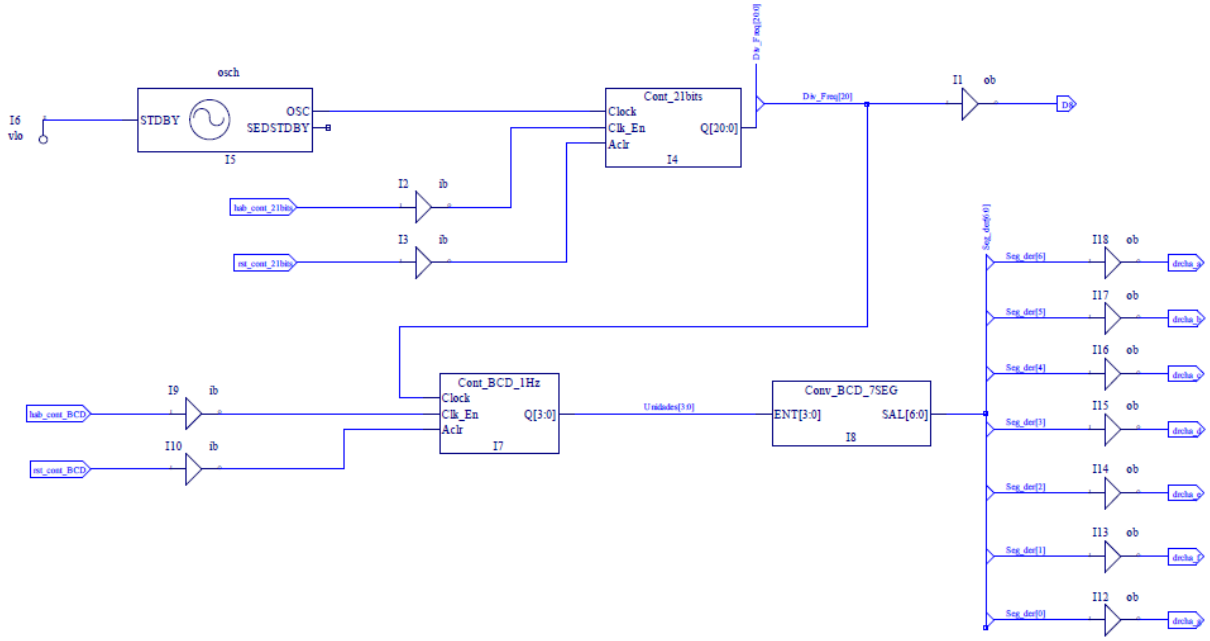


Figura 27. Esquema del contador BCD de 1 dígito

De esta forma, se ha configurado el contador de 1 dígito (unidades) que cuenta de 0 a 9 con una frecuencia de 1 Hz y se muestra la salida por uno de los displays disponibles.

Contador BCD de 2 dígitos:

El siguiente paso es modificar el diseño para lograr un contador de 2 dígitos (unidades y decenas).

Se añade otro contador que recibirá las mismas señales de reloj y reset que el primero, pero la entrada de habilitación será función de la salida del primer contador. Se debe cumplir que el segundo contador (decenas) debe habilitarse cuando el primer contador (unidades) llegue al límite de la cuenta, es decir, cuando éste llegue al número 9. Para lograr esto se hace uso de puertas lógicas, que habilitan el segundo contador cuando la salida del primer contador es 1001, que equivale al número 9 en binario.

Al igual que antes, la salida de este segundo contador se lleva a un convertidor BCD-7segmentos y en este caso se muestra la salida por el display de la izquierda.

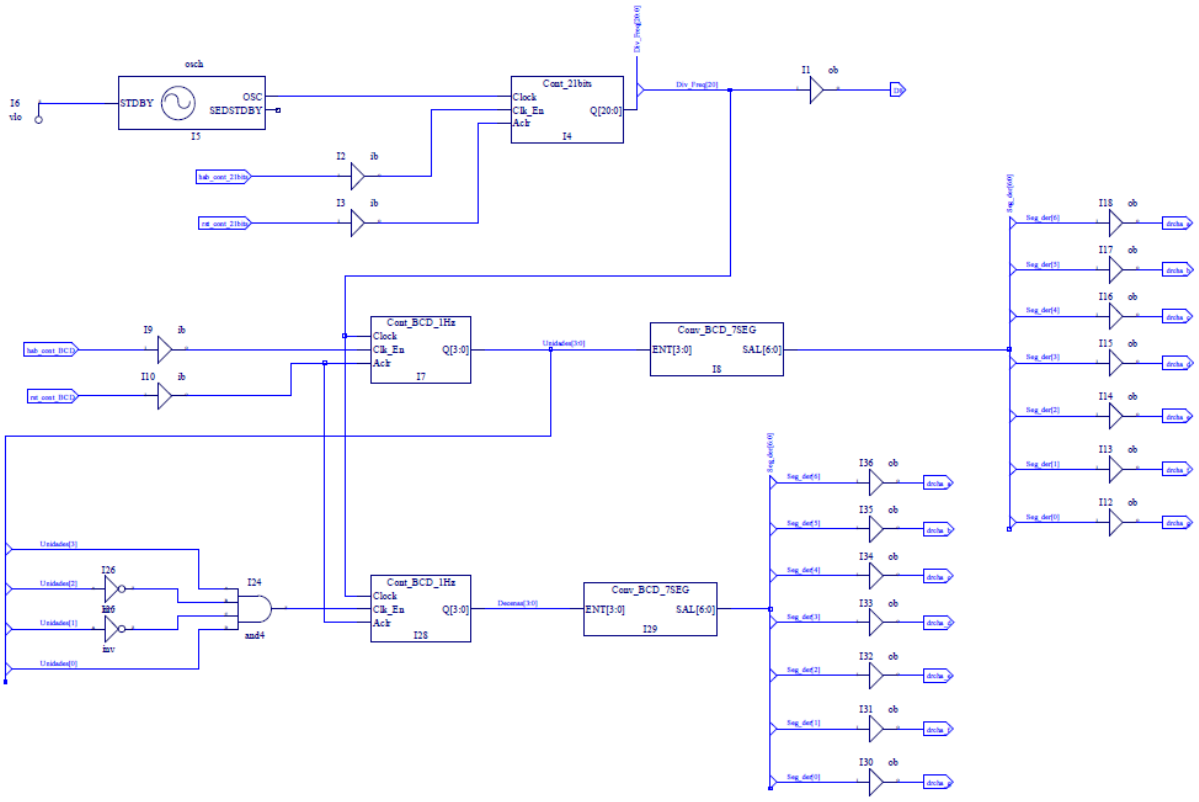


Figura 28. Esquema del contador BCD de 2 dígitos

Ahora se tiene un contador de 2 dígitos (unidades y decenas) que cuenta de 0 a 99 con una frecuencia de 1 Hz y se muestra la salida por los dos displays disponibles.

Contador BCD de 3 dígitos:

Finalmente se termina el diseño del contador BCD convirtiéndolo en un contador de 3 dígitos (unidades, decenas y centenas).

El tercer contador (centenas) recibe las mismas entradas que el segundo, excepto la entrada de habilitación que ahora será función de la salida de los dos primeros contadores (unidades y decenas). Sólo en el caso de que estos dos contadores lleguen a 9 se habilitará el tercer contador.

La salida del tercer contador se mostrará por los leds de la placa porque no existe un tercer display para esta función.

Cuando se llegue a la primera centena se encenderá el led D1, cuando se llegue a la segunda centena se encenderán los leds D1 y D2 conjuntamente, y así sucesivamente hasta llegar a las 6 centenas.

Para lograr la correcta salida de información por los leds de la placa se ha realizado una descripción VHDL que se muestra a continuación:

```
1   library IEEE;
2   use IEEE.std_logic_1164.all;
3
4   entity Conv_BCD_Leds is
5   port(
6       ENT: in std_logic_vector (3 downto 0);
7       SAL: out std_logic_vector (5 downto 0)
8       );
9   end Conv_BCD_Leds;
10
11  architecture Conv_BCD_Leds_arch of Conv_BCD_Leds is
12  begin
13      SAL <=  "111111" when (ENT="0000") else     --Todos los leds apagados
14              "011111" when (ENT="0001") else     --1 led encendido
15              "001111" when (ENT="0010") else     --2 leds encendidos
16              "000111" when (ENT="0011") else     --3 leds encendidos
17              "000011" when (ENT="0100") else     --4 leds encendidos
18              "000001" when (ENT="0101") else     --5 leds encendidos
19              "000000" when (ENT="0110");         --6 leds encendidos
20  end Conv_BCD_Leds_arch;
```

Figura 29. Descripción VHDL del Convertidor BCD-Leds

Estos leds se activan a nivel bajo, debido a esto, para que se encienda un led se le debe dar el valor de 0.

Se realiza una simulación con la ayuda del programa Active-HDL Lattice Edition para comprobar el funcionamiento del código generado.

Se introduce como entradas los valores 0, 1 y 2 y se observa que las salidas son las que corresponden con los leds que se deben encender.

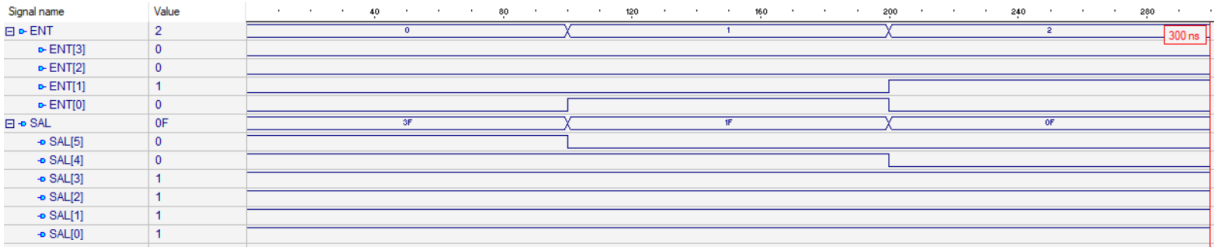


Figura 30. Simulación del convertidor BCD-Leds

Una vez comprobado el correcto funcionamiento del convertidor BCD-Leds, se genera un símbolo para poder utilizarlo en el esquema.

Con esta configuración final se habría generado un contador BCD de 3 dígitos que cuenta desde 0 hasta 699 con una frecuencia de 1 Hz, con entradas de habilitación y reset controladas por el usuario a través de los interruptores de la placa y que muestra los valores de las unidades y decenas por los displays y el valor de las centenas por los leds de la placa.

El tamaño del contador (desde 0 hasta 699) es suficiente para su posterior utilización en la lectura del sensor de ultrasonidos ya que este último tiene un rango máximo de medida de 645 cm.

El diseño final del esquema del contador BCD de 3 dígitos se muestra en la siguiente figura.

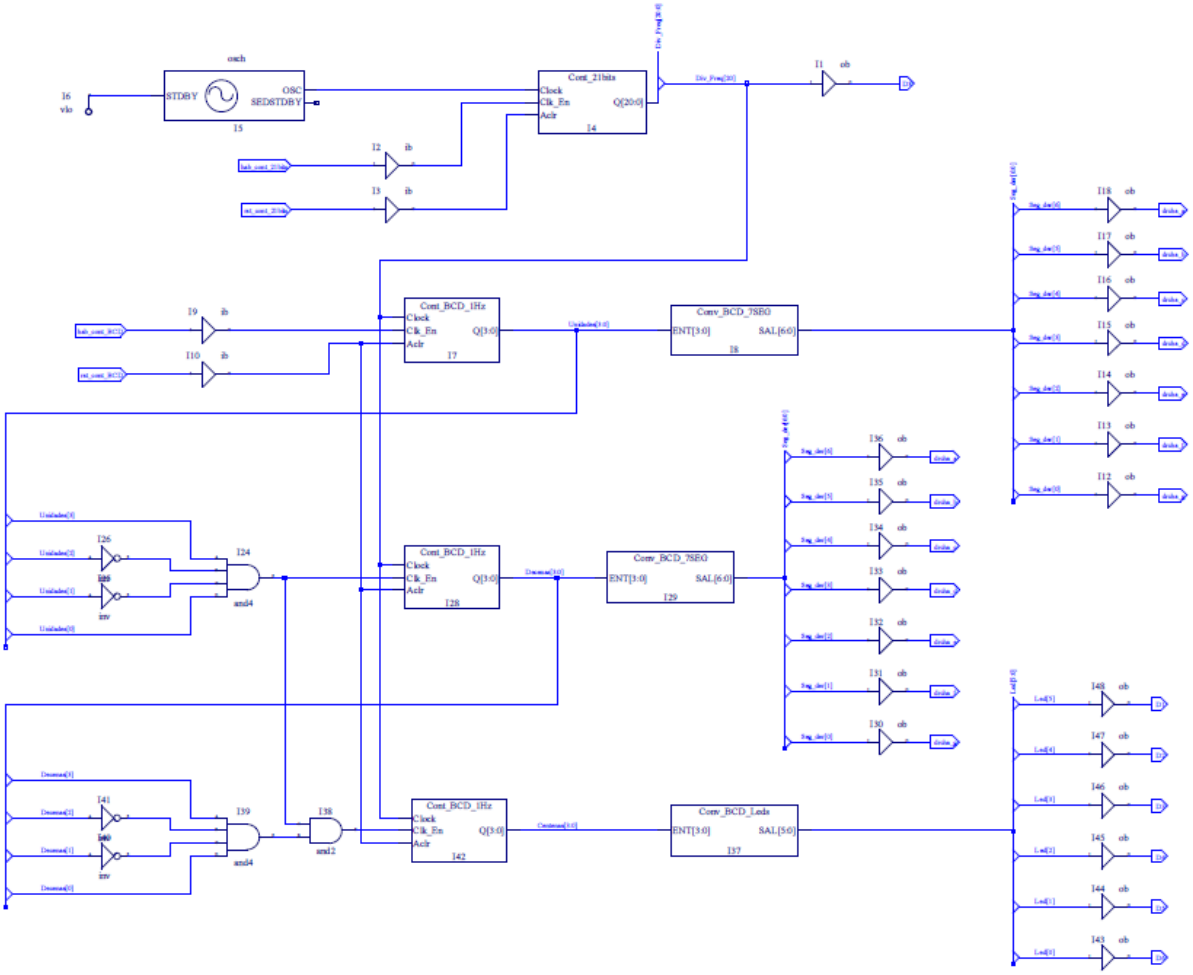


Figura 31. Esquema del contador BCD de 3 dígitos

3.1.4. Generador de eventos

Para la lectura de una medida, se debe hacer un reset al principio de cada ciclo para realizar una nueva medida. La señal de reset que se ha utilizado es un pulso de 2 µs que se repite periódicamente con una frecuencia de 1 Hz.

Para lanzar la medición del sensor de ultrasonidos se debe generar un pulso de una duración superior a 20 µs (según las características del sensor descritas en la página 16).

Como señal de lanzar medición (RX) se ha utilizado un pulso de 30 µs que se activa cuando la señal de reset se desactiva y también se repite periódicamente con una frecuencia de 1 Hz.

La generación cíclica de estos pulsos necesarios para la medición, se ha programado en una descripción VHDL. Esta descripción tiene como entrada el vector de salida del contador de 21 bits que se creó en el apartado 3.1.2.

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity Gen_Event is
5  port(
6          Q: in std_logic_vector(20 downto 0);
7          Rst, Rx: out std_logic
8          );
9  end Gen_Event;
10
11 architecture Gen_Event_arch of Gen_Event is
12 begin
13         Rst <= (not Q(20)) and (not Q(19)) and (not Q(18)) and (not Q(17)) and (not Q(16)) and (not Q(15)) and      --Reset activo los primeros 2 µs
14                 (not Q(14)) and (not Q(13)) and (not Q(12)) and (not Q(11)) and (not Q(10)) and (not Q(9)) and
15                 (not Q(8)) and (not Q(7)) and (not Q(6)) and (not Q(5)) and (not Q(4)) and (not Q(3)) and (not Q(2));
16
17         Rx  <= (not Q(20)) and (not Q(19)) and (not Q(18)) and (not Q(17)) and (not Q(16)) and (not Q(15)) and      --RX activo desde los 2 µs hasta 32 µs
18                 (not Q(14)) and (not Q(13)) and (not Q(12)) and (not Q(11)) and (not Q(10)) and (not Q(9)) and
19                 (not Q(8)) and (not Q(7)) and (not Q(6)) and
20                 (not ((not Q(20)) and (not Q(19)) and (not Q(18)) and (not Q(17)) and (not Q(16)) and (not Q(15)) and
21                 (not Q(14)) and (not Q(13)) and (not Q(12)) and (not Q(11)) and (not Q(10)) and (not Q(9)) and
22                 (not Q(8)) and (not Q(7)) and (not Q(6)) and (not Q(5)) and (not Q(4)) and (not Q(3)) and (not Q(2))));
23 end Gen_Event_arch;
```

Figura 32. Descripción VHDL del Generador de Eventos de 21 bits

Debido al gran tamaño de este generador de eventos, no se observaría correctamente el funcionamiento en una simulación. Para poder comprobar el funcionamiento del generador de eventos se ha diseñado otro más pequeño (10 bits) para poder realizar la simulación.

```
1   library IEEE;
2   use IEEE.std_logic_1164.all;
3
4   entity Gen_Event is
5   port(
6           Q: in std_logic_vector(9 downto 0);
7           Rst, Rx: out std_logic
8           );
9   end Gen_Event;
10
11  architecture Gen_Event_arch of Gen_Event is
12  begin
13          Rst <= (not Q(9)) and (not Q(8)) and (not Q(7)) and (not Q(6)) and          --Reset activo los primeros 2 µs
14                  (not Q(5)) and (not Q(4)) and (not Q(3)) and (not Q(2));
15
16          Rx <= (not Q(9)) and (not Q(8)) and (not Q(7)) and (not Q(6)) and           --RX activo desde los 2 µs hasta 32 µs
17                  (not((not Q(9)) and (not Q(8)) and (not Q(7)) and (not Q(6)) and
18                  (not Q(5)) and (not Q(4)) and (not Q(3)) and (not Q(2))));
19  end Gen_Event_arch;
```

Figura 33. Descripción VHDL del Generador de Eventos de 10 bits

Se realiza la simulación y se observa lo siguiente:

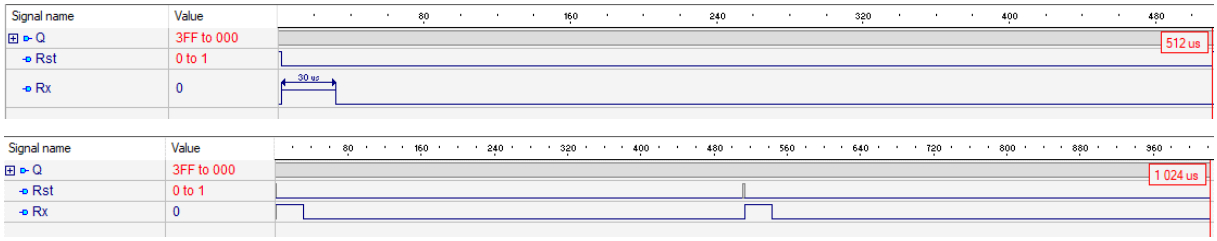


Figura 34. Simulación del Generador de Eventos

Los pulsos de las señales de reset y RX se generan de manera cíclica y en la forma correcta. Una vez comprobado esto, se puede generar el símbolo correspondiente para poder utilizar el generador de eventos en el esquema.

3.1.5. Generación del reloj fino

La salida del sensor de ultrasonidos proporciona un pulso de una anchura proporcional a la distancia con un factor de escala de 147 µs/pulgada. Para poder medir distancias se va a utilizar un reloj "fino" con una frecuencia de oscilación determinada que permita que 1 pulso equivalga a 1 cm.

De esta forma, se evita la circuitería de conversiones, se busca que el factor de proporcionalidad sea la unidad.

$$147\,\frac{\mu s}{pulgada} = \frac{147\ \mu s}{2{,}54\ cm} = 57{,}874\,\frac{\mu s}{cm}$$

$$57{,}874\,\frac{\mu s}{cm} \rightarrow \boldsymbol{f} = \frac{1}{57{,}874 \cdot 10^{-6}} = \mathbf{17{,}28\ kHz}$$

Según los cálculos realizados, se necesita un reloj con una frecuencia de oscilación de 17,28 kHz.

Para obtener esta frecuencia en concreto se configurará el sintetizador de frecuencia (PLL) que incorpora la FPGA con la ayuda de la herramienta IPexpress.

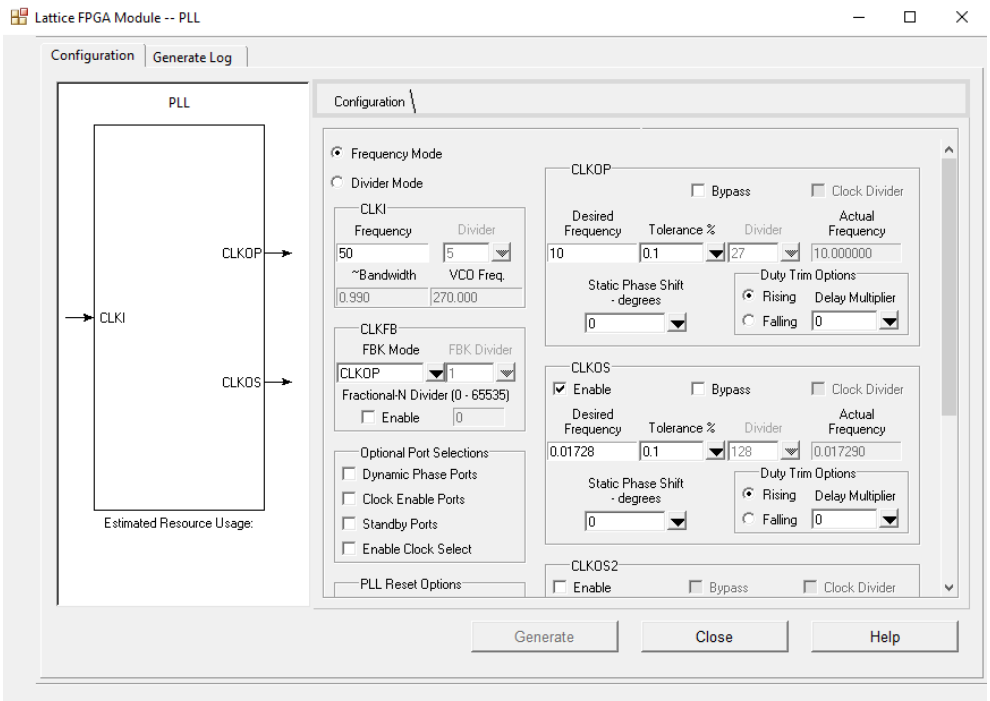


Figura 35. Configuración del sintetizador de frecuencia PLL

El reloj fino de 17,28 kHz se obtiene a partir del reloj externo de 50 MHz debido a la necesidad de una mayor precisión que el oscilador interno de 2,08 MHz no podría dar. El oscilador interno tiene una precisión del orden del 5%, mientras que el reloj externo es extremadamente preciso.

Una vez creados todos los elementos necesarios para la realización de las medidas, se interconectan todos estos elementos, modificando el esquema anterior.

El reloj fino de 17,28 kHz se utilizará como entrada de reloj al contador BCD de 3 dígitos. Debido a que este reloj fino se obtiene a partir del reloj externo de 50 MHz, es necesario habilitar este último, por lo que se le asigna el valor 1 a la entrada de habilitación del reloj externo que corresponde con el pin 32 de la FPGA.

El generador de eventos tiene como salidas la señal de reset y la señal Rx de lanzar la medición. La señal de reset se utiliza directamente en el contador BCD de 3 dígitos y la señal Rx se manda al sensor de ultrasonidos que proporcionará la salida llamada PW que se utiliza como entrada de habilitación del contador BCD.

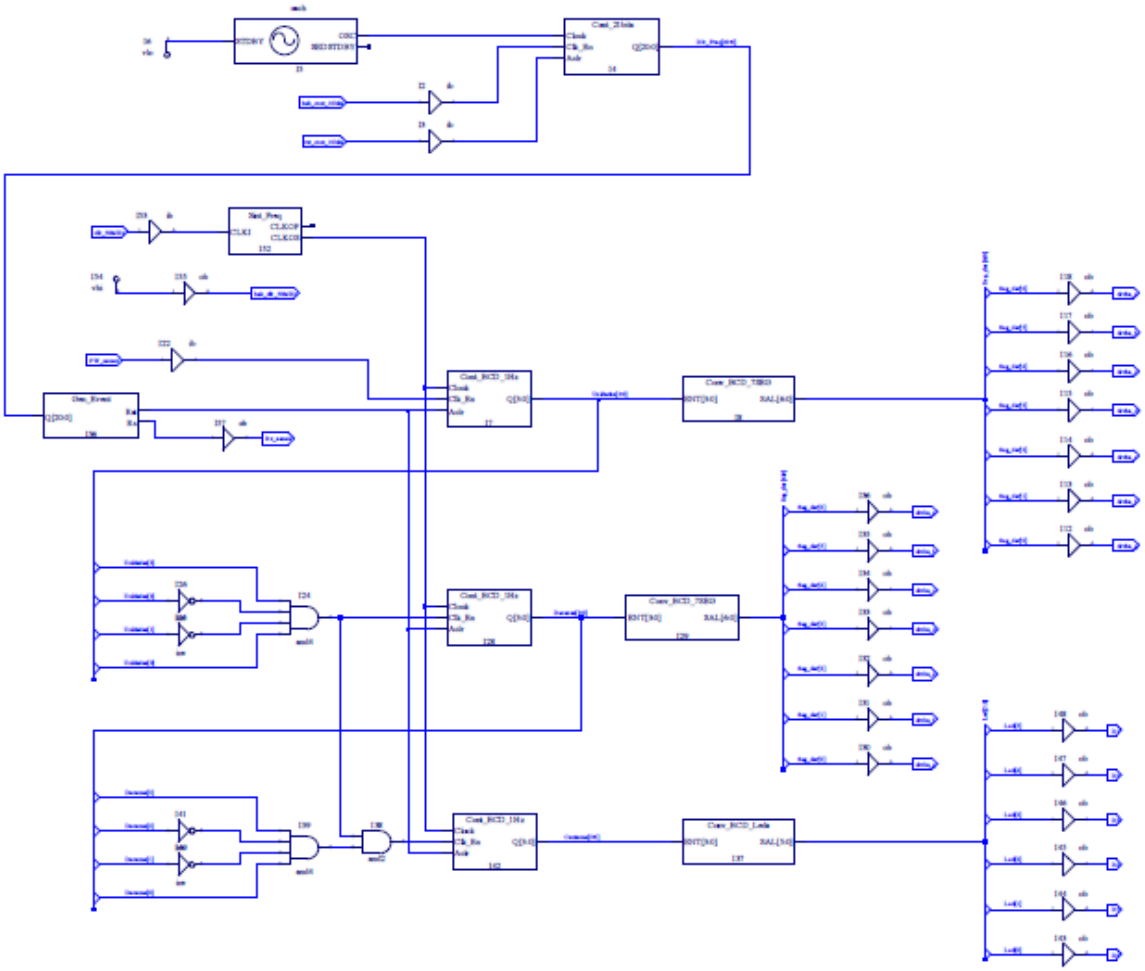


Figura 36. Esquema para la medida de distancias

El diseño actual consiste en un contador BCD de 3 dígitos que se habilita cuando el pulso PW del sensor está activo y cuenta durante este periodo. El valor de la cuenta es la distancia medida expresada en centímetros, ya que se ha construido el reloj fino con una frecuencia que permita que el factor de proporcionalidad sea la unidad. El conjunto se inicializa cada segundo.

Con esta configuración ya es posible realizar medidas de distancias dentro del rango permitido.

Mejora de la visualización mediante registros:

Para evitar que el valor mostrado por pantalla parpadee constantemente al realizar una medida, se introducen registros de tipo D, que se denominan fd1s3ax en la biblioteca de símbolos de Lattice Diamond.

Este tipo de registro trabaja solo con un bit, por lo que se necesitarían cuatro registros para cada contador BCD (unidades, decenas y centenas). Para simplificar el esquema general se ha realizado el esquema del almacenamiento de cada contador en un proyecto aparte y luego se ha implementado como un bloque.

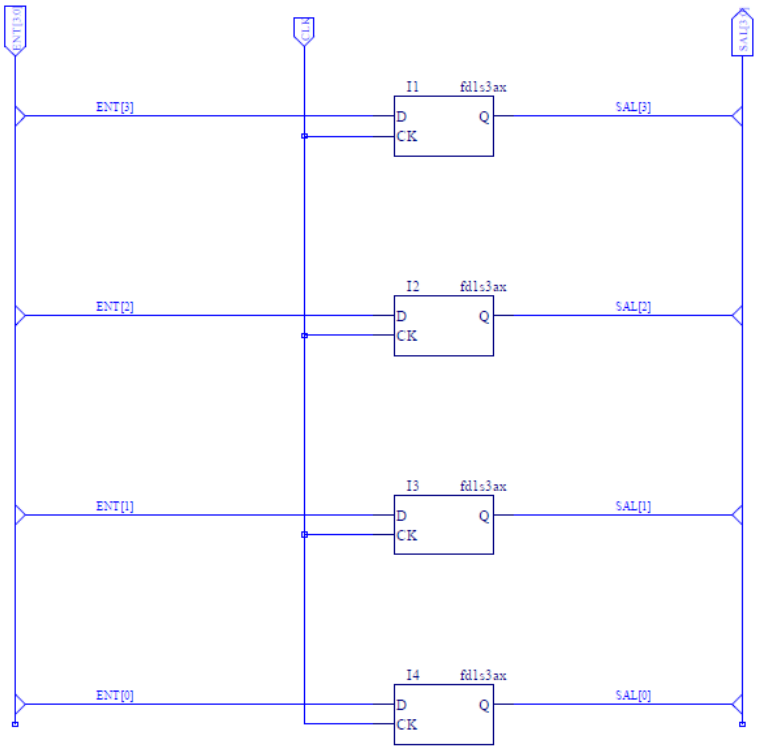


Figura 37. Esquema de almacenamiento en registro de 1 dígito

En este esquema se almacenan los 4 bits que conforman 1 dígito y se utiliza un reloj para establecer la frecuencia de almacenamiento.

En el esquema general se introducen como bloques que tiene 2 entradas y una salida, una de las entradas es el dígito a almacenar y la otra entrada es el reloj, que en este caso será el reloj de 1 Hz. Esto significa que se almacenará cada segundo el valor que se mida con el sensor, frecuencia suficiente para mantener estable la salida por los displays.

El esquema final para la lectura del sensor de ultrasonidos con la utilización de los registros para la mejora en la visualización de la medida es el siguiente:

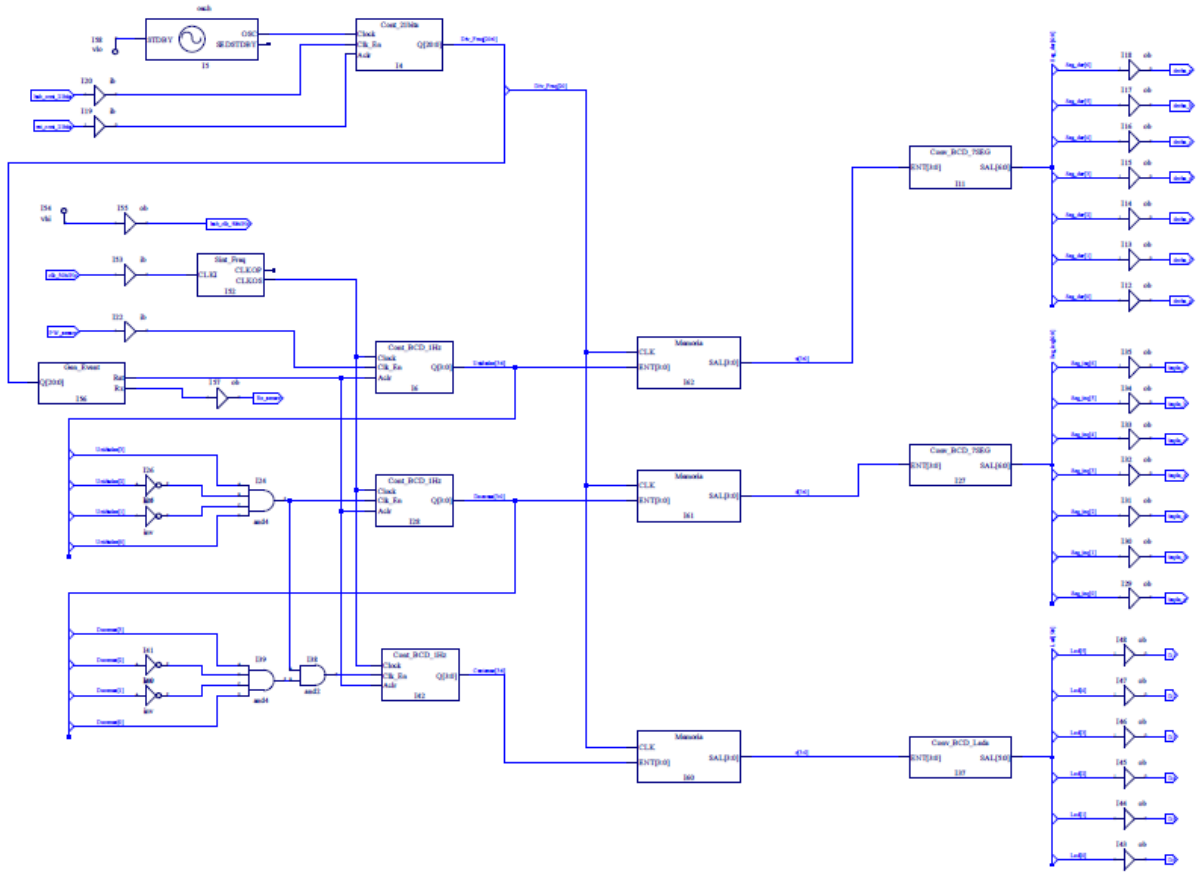


Figura 38. Esquema para la medida de distancias con memorias

Una vez terminado el diseño esquemático, se realiza la generación del fichero JEDEC que posteriormente se cargará en la FPGA.

Cuando se genera por primera vez el fichero JEDEC, se realiza la asignación automática de los pines utilizados en el esquema. Es necesario modificar esta asignación manualmente para hacer coincidir los pines con los elementos de la placa.

Para modificar la asignación de los pines se utiliza la herramienta Spreadsheet View, en esta tabla se cambian las columnas PIN y IO_TYPE. En esta última columna se elige LVCMOS33.

La asignación de pines queda de la siguiente forma:

	Name	Group By	Pin	BANK	BANK_VCC	VREF	IO_TYPE
1	All Ports	N/A	N/A	N/A	N/A	N/A	
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	Clock	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1.1	clk_50MHz	N/A	27(27)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.2	PW_sensor	N/A	1(1)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.3	hab_cont_21bits	N/A	77(77)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.4	rst_cont_21bits	N/A	76(76)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	D1	N/A	97(97)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.2	D2	N/A	98(98)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.3	D3	N/A	99(99)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.4	D4	N/A	100(100)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.5	D5	N/A	104(104)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.6	D6	N/A	105(105)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.7	Rx_sensor	N/A	3(3)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.8	drcha_a	N/A	21(21)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.9	drcha_b	N/A	22(22)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.10	drcha_c	N/A	11(11)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.11	drcha_d	N/A	10(10)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.12	drcha_e	N/A	13(13)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.13	drcha_f	N/A	12(12)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.14	drcha_g	N/A	14(14)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.15	hab_clk_50MHz	N/A	32(32)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.16	izqda_a	N/A	23(23)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)

1.2.17	izqda_b	N/A	25(25)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.18	izqda_c	N/A	33(33)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.19	izqda_d	N/A	34(34)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.20	izqda_e	N/A	35(35)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.21	izqda_f	N/A	26(26)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.22	izqda_g	N/A	24(24)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)

Figura 39. Spreadsheet View del sensor de ultrasonidos

Finalmente, se procede a realizar la carga del programa a la FPGA conectando el dispositivo mediante el cable USB al ordenador donde se ha realizado la configuración y donde se ha generado el fichero JEDEC.

Recursos empleados:

Cuando se compila el diseño y se genera el JEDEC, el software genera una serie de informes que indican los recursos empleados.

```
Design Summary
   Number of registers:     71 out of  1604 (4%)
      PFU registers:           71 out of  1280 (6%)
      PIO registers:            0 out of   324 (0%)
   Number of SLICEs:        91 out of   640 (14%)
      SLICEs as Logic/ROM:     91 out of   640 (14%)
      SLICEs as RAM:            0 out of   480 (0%)
      SLICEs as Carry:         21 out of   640 (3%)
   Number of LUT4s:        161 out of  1280 (13%)
      Number used as logic LUTs:        119
      Number used as distributed RAM:     0
      Number used as ripple logic:       42
      Number used as shift registers:     0
   Number of PIO sites used: 26 + 4(JTAG) out of 108 (28%)
   Number of block RAMs:  0 out of 7 (0%)
   Number of GSRs:  1 out of 1 (100%)
   EFB used :       No
   JTAG used :      No
   Readback used :  No
   Oscillator used :  Yes
   Startup used :   No
   POR :            On
   Bandgap :        On
   Number of Power Controller:  0 out of 1 (0%)
   Number of Dynamic Bank Controller (BCINRD):  0 out of 4 (0%)
   Number of Dynamic Bank Controller (BCLVDSO):  0 out of 1 (0%)
   Number of DCCA:  0 out of 8 (0%)
   Number of DCMA:  0 out of 2 (0%)
   Number of PLLs:  1 out of 1 (100%)
   Number of DQSDLLs:  0 out of 2 (0%)
   Number of CLKDIVC:  0 out of 4 (0%)
   Number of ECLKSYNCA:  0 out of 4 (0%)
   Number of ECLKBRIDGECS:  0 out of 2 (0%)
```

```
   Number of warnings:  1
   Number of errors:    0
```

Device utilization summary:

```
   PIO (prelim)    26+4(JTAG)/108      28% used
                   26+4(JTAG)/108      28% bonded

   SLICE              91/640           14% used

   GSR                 1/1            100% used
   OSC                 1/1            100% used
   PLL                 1/1            100% used


Number of Signals: 247
Number of Connections: 586
```

Se observa que el diseño ha consumido el 13% de la capacidad de lógica de la FPGA, es decir, 161 de 1280 LUTs.

3.2. Lectura del convertidor de luz-frecuencia

Se pretende realizar una lectura de la salida proporcionada por el convertidor de luz-frecuencia y mostrar el resultado a través de los leds de la placa y los displays disponibles. La salida proporcionada por el sensor es una onda con una frecuencia que varía en función de la luz incidente, por lo tanto, el valor que se muestre por pantalla será indicativo del nivel de iluminación existente.

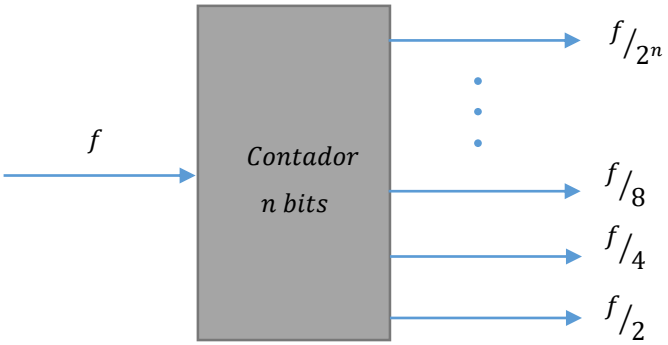
La información se actualizará cada segundo y se mostrará mediante tres dígitos (unidades, decenas y centenas).

Para la lectura del convertidor de luz-frecuencia se ha utilizado la siguiente organización de carpetas:

- Lectura_Sensor_Luz (Creado como Proyecto General)
  - └ Conv_BCD_7SEG (Creado como Proyecto)
  - └ Conv_BCD_Leds (Creado como Proyecto)
  - └ Memoria (Creado como Proyecto)

La señal de reloj de 1 Hz se utilizará aquí para definir la entrada de habilitación y reset del contador BCD de tres dígitos. Estas señales serán complementarias, es decir, en cada ciclo de medida, la mitad del tiempo se manda una señal que habilita el contador y la otra mitad del tiempo se realiza un reset para poder realizar la siguiente medida. Con esto se consigue que el contador se habilite y deshabilite siguiendo una frecuencia fija (1 Hz) y durante estos pulsos de frecuencia fija el contador BCD recibirá la frecuencia detectada por el sensor como una señal de reloj que indicará el nivel de iluminación.

Para evitar los desbordamientos se escalará la frecuencia que proporciona el sensor antes de llevarla al contador. Para ello se utilizará un contador de 10 bits que servirá de divisor de frecuencia para dividir la frecuencia entre 1000.



$$\frac{f}{2^n} \cong \frac{f}{1000} \rightarrow \boldsymbol{n = 10\ bits}$$

Con la ayuda de la herramienta IPexpress se genera el contador de 10 bits, del cual se utilizará la salida correspondiente al bit número 9.

Para estabilizar la salida por los displays de la placa se utilizan los registros de tipo D con señal de reloj de 1 Hz.

El esquema final para la lectura del convertidor de luz-frecuencia es el siguiente:

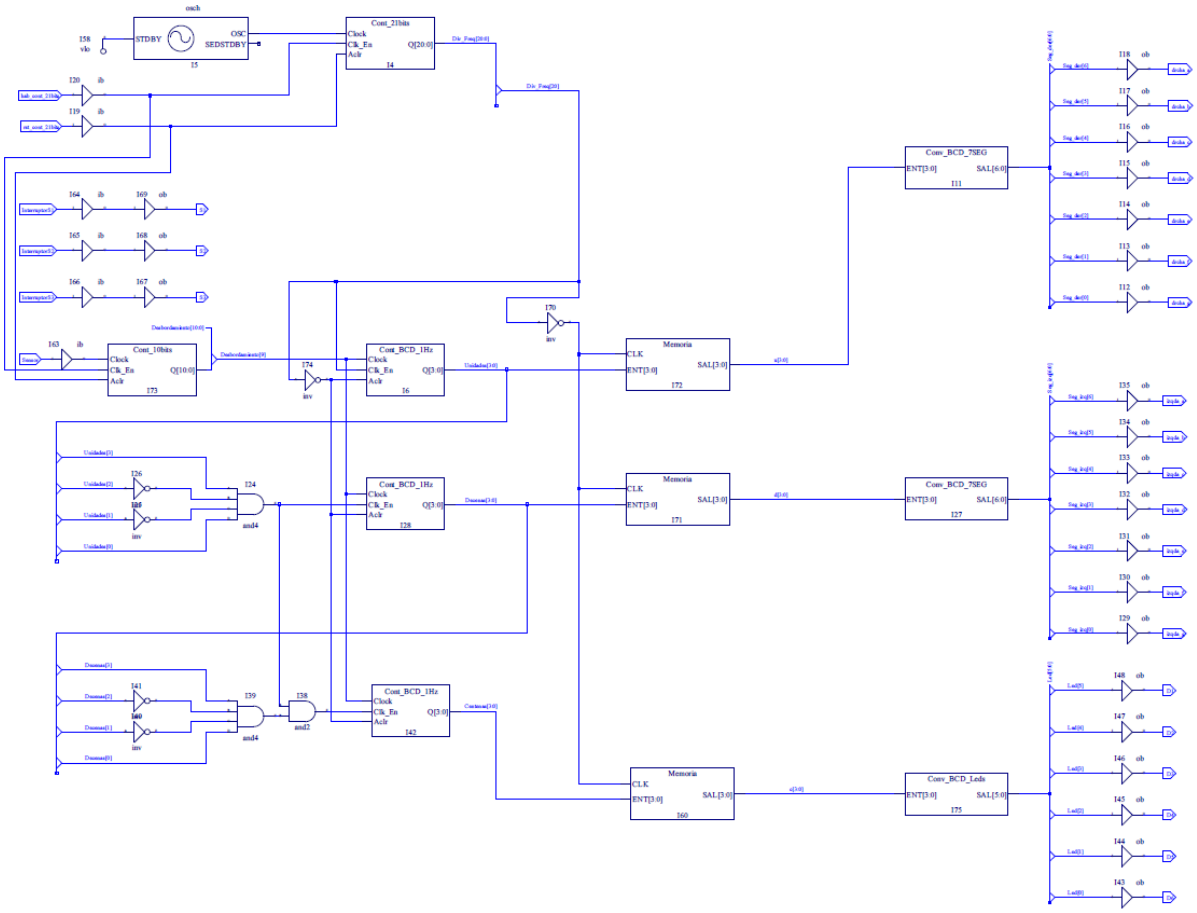


Figura 40. Esquema para la lectura del convertidor de luz-frecuencia

Para modificar la asignación de los pines se utiliza la herramienta Spreadsheet View, en esta tabla se cambian las columnas PIN y IO_TYPE. En esta última columna se elige LVCMOS33.

La asignación de pines queda de la siguiente forma:

	Name	Group By	Pin	BANK	BANK_VCC	VREF	IO_TYPE
1	All Ports	N/A	N/A	N/A	N/A	N/A	
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	Clock	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1.1	Sensor	N/A	1(1)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.2	InterruptorS1	N/A	86(86)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.3	InterruptorS2	N/A	85(85)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.4	InterruptorS3	N/A	84(84)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.5	hab_cont_21bits	N/A	77(77)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.1.6	rst_cont_21bits	N/A	76(76)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	D1	N/A	97(97)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.2	D2	N/A	98(98)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.3	D3	N/A	99(99)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.4	D4	N/A	100(100)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.5	D5	N/A	104(104)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.6	D6	N/A	105(105)	1(1)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.7	S1	N/A	4(4)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.8	S2	N/A	3(3)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.9	S3	N/A	2(2)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.10	drcha_a	N/A	21(21)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.11	drcha_b	N/A	22(22)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.12	drcha_c	N/A	11(11)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.13	drcha_d	N/A	10(10)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.14	drcha_e	N/A	13(13)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)

1.2.15	drcha_f	N/A	12(12)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.16	drcha_g	N/A	14(14)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.17	izqda_a	N/A	23(23)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.18	izqda_b	N/A	25(25)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.19	izqda_c	N/A	33(33)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.20	izqda_d	N/A	34(34)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.21	izqda_e	N/A	35(35)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.22	izqda_f	N/A	26(26)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)
1.2.23	izqda_g	N/A	24(24)	3(3)	Auto	N/A	LVCMOS33(LVCMOS33)

Figura 41. Spreadsheet View del convertidor de luz-frecuencia

Finalmente, se procede a realizar la carga del programa a la FPGA conectando el dispositivo mediante el cable USB al ordenador donde se ha realizado la configuración y donde se ha generado el fichero JEDEC.

Recursos empleados:

Cuando se compila el diseño y se genera el JEDEC, el software genera una serie de informes que indican los recursos empleados.

```
Design Summary
   Number of registers:     81 out of  1604 (5%)
      PFU registers:           81 out of  1280 (6%)
      PIO registers:            0 out of   324 (0%)
   Number of SLICEs:        93 out of   640 (15%)
      SLICEs as Logic/ROM:     93 out of   640 (15%)
      SLICEs as RAM:            0 out of   480 (0%)
      SLICEs as Carry:         27 out of   640 (4%)
   Number of LUT4s:        165 out of  1280 (13%)
      Number used as logic LUTs:        111
      Number used as distributed RAM:     0
      Number used as ripple logic:       54
      Number used as shift registers:     0
   Number of PIO sites used: 29 + 4(JTAG) out of 108 (31%)
   Number of block RAMs:  0 out of 7 (0%)
   Number of GSRs:  1 out of 1 (100%)
   EFB used :       No
   JTAG used :      No
   Readback used :  No
   Oscillator used :  Yes
   Startup used :   No
   POR :            On
   Bandgap :        On
   Number of Power Controller:  0 out of 1 (0%)
   Number of Dynamic Bank Controller (BCINRD):  0 out of 4 (0%)
   Number of Dynamic Bank Controller (BCLVDSO):  0 out of 1 (0%)
   Number of DCCA:  0 out of 8 (0%)
   Number of DCMA:  0 out of 2 (0%)
   Number of PLLs:  0 out of 1 (0%)
```

```
   Number of DQSDLLs:  0 out of 2 (0%)
   Number of CLKDIVC:  0 out of 4 (0%)
   Number of ECLKSYNCA:  0 out of 4 (0%)
   Number of ECLKBRIDGECS:  0 out of 2 (0%)

   Number of warnings:  1
   Number of errors:    0
```

Device utilization summary:

```
   PIO (prelim)    29+4(JTAG)/108      31% used
                   29+4(JTAG)/108      31% bonded

   SLICE              93/640           14% used

   GSR                 1/1            100% used
   OSC                 1/1            100% used


Number of Signals: 264
Number of Connections: 592
```

Se observa que el diseño ha consumido el 13% de la capacidad de lógica de la FPGA, es decir, 165 de 1280 LUTs.

4. RESULTADOS

En este apartado se presentan los resultados obtenidos de la realización de las pruebas de medición y el análisis de algunas señales.

4.1. Analizador lógico

Para poder observar los valores de las señales de los distintos sensores se utilizará un analizador lógico y el software Saleae Logic 1.2.18.

Figura 42. Analizador lógico

Para registrar los datos de las señales que se quiera observar se conectan mediante unas pinzas y unos cables en los pines correspondientes. El analizador que se utiliza en este caso es de 24 MHz y tiene una capacidad de hasta 8 canales.

El software permite configurar los canales, la velocidad de muestreo y la duración de la toma de muestras. Una vez recogida la muestra se puede realizar mediciones de las señales como pueden ser anchos de pulso, frecuencias, etc.

Sensor de ultrasonidos:

Para el sensor de ultrasonidos se utilizarán 3 canales para las señales de GND, RX y PW. De este modo se podrá comprobar el correcto funcionamiento en la toma de medidas.

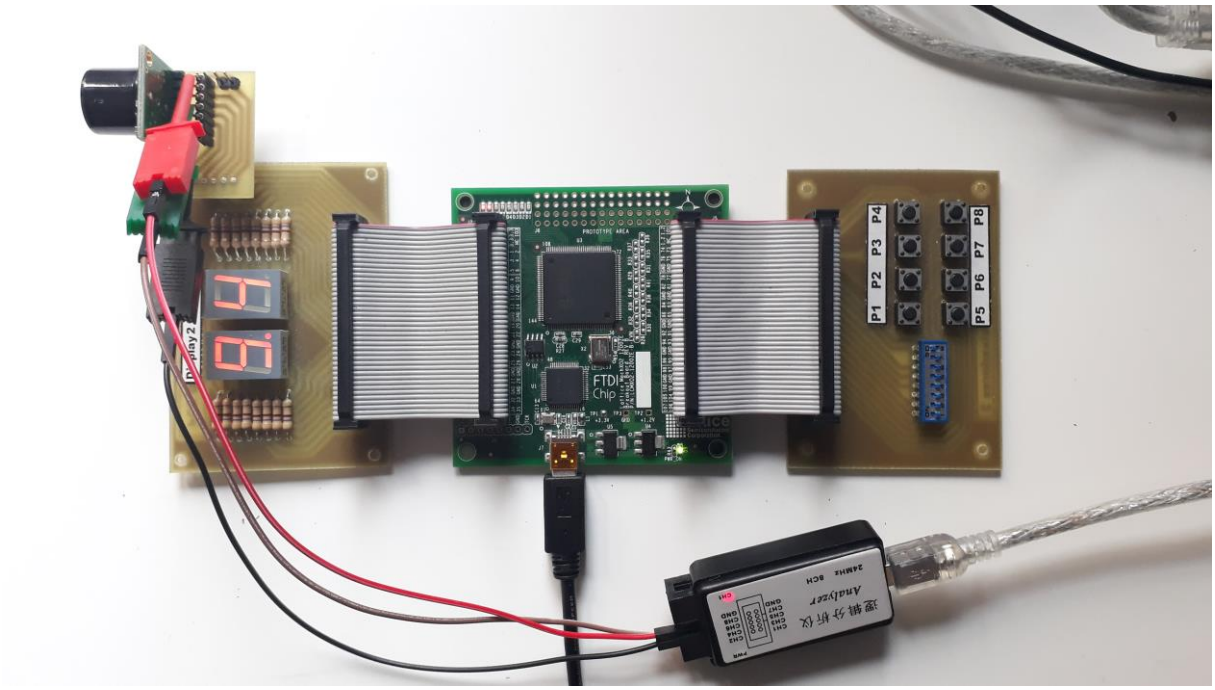


Figura 43. Conexión del analizador lógico en el sensor de ultrasonidos

En el primer canal se conecta la señal de GND que se tomará como referencia, este valor siempre se encontrará a 0.

Se tomarán dos muestras con dos medidas diferentes para observar las diferencias, ambas muestras son de 2 segundos de duración.

En la primera muestra se observan 2 pulsos en la señal RX con una frecuencia de 1,059 Hz (aprox. 1 Hz), estos pulsos son los que se generaron con el generador de eventos.

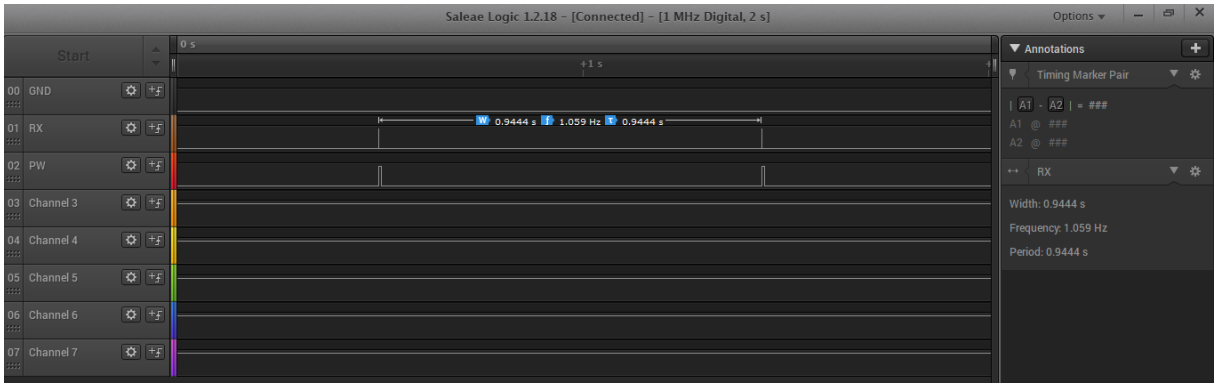


Figura 44. Datos obtenidos de la primera muestra del sensor de ultrasonidos

Aplicando un zoom en uno de los pulsos, se observa con más detalle las duraciones de las señales RX y PW. La señal RX tiene una duración de 28 µs (se necesitaba un mínimo de 20 µs para poder realizar la medida), luego tras 0,483 ms se realiza la medida con la señal PW.

Figura 45. Datos obtenidos de la primera muestra del sensor de ultrasonidos

Finalmente, se observa que la duración de la señal PW para esta muestra en concreto es de 5,3 ms.

Figura 46. Datos obtenidos de la primera muestra del sensor de ultrasonidos

En la segunda muestra se observan también 2 pulsos en la señal RX con una frecuencia de 1,059 Hz (aprox. 1 Hz), igual que en la primera muestra.

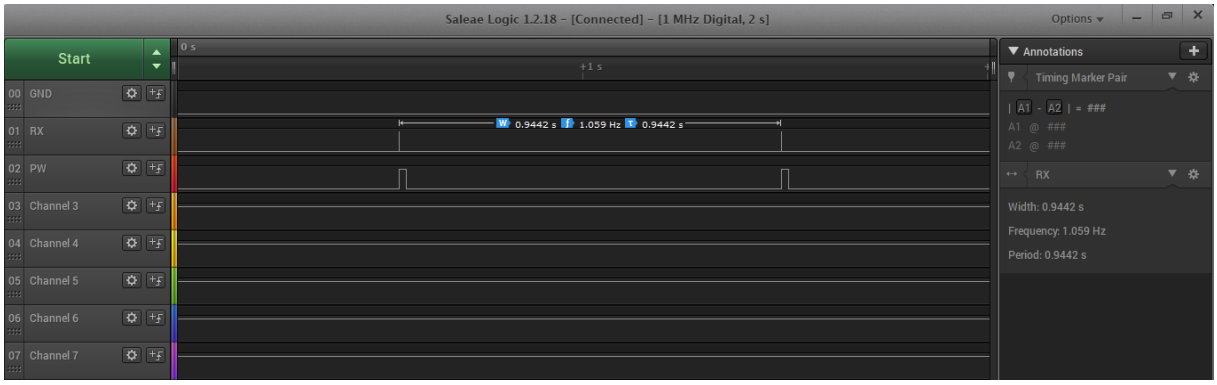


Figura 47. Datos obtenidos de la segunda muestra del sensor de ultrasonidos

Aplicando un zoom en uno de los pulsos, se observa con más detalle las duraciones de las señales RX y PW. La señal RX tiene una duración de 29 µs (se necesitaba un mínimo de 20 µs para poder realizar la medida), luego tras 0,458 ms se realiza la medida con la señal PW.

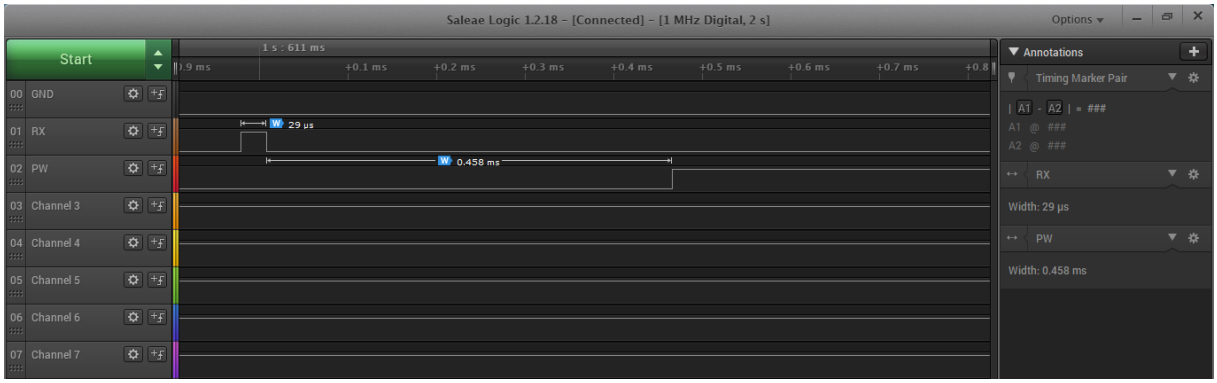


Figura 48. Datos obtenidos de la segunda muestra del sensor de ultrasonidos

Finalmente, se observa que la duración de la señal PW para esta muestra en concreto es de 17 ms.

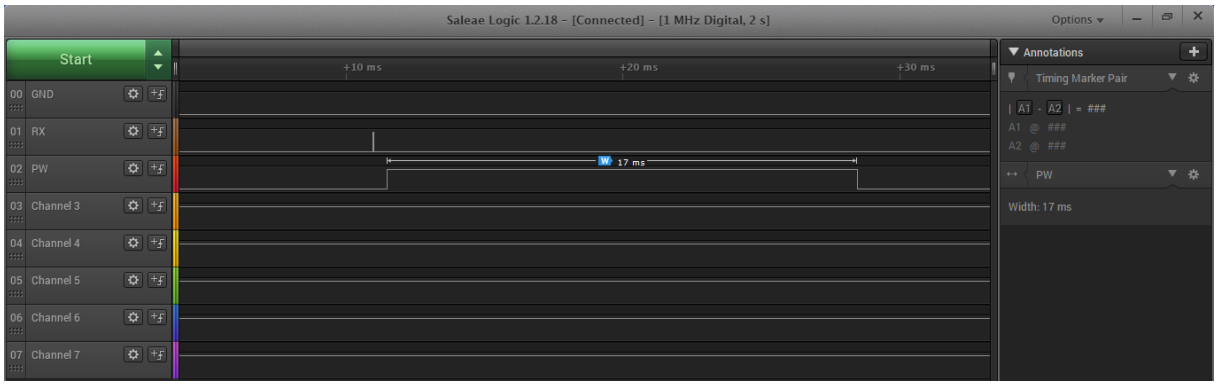


Figura 49. Datos obtenidos de la segunda muestra del sensor de ultrasonidos

Se llega a la conclusión de que los tiempos de ejecución de la medida son iguales siempre, lo único que varía es la amplitud de la señal PW que es proporcional a la distancia medida.

Convertidor de luz-frecuencia:

Para el convertidor de luz-frecuencia se utilizarán 4 canales para las señales de GND, S0, S1 y OUT. De este modo se podrá comprobar el correcto funcionamiento en la toma de medidas.

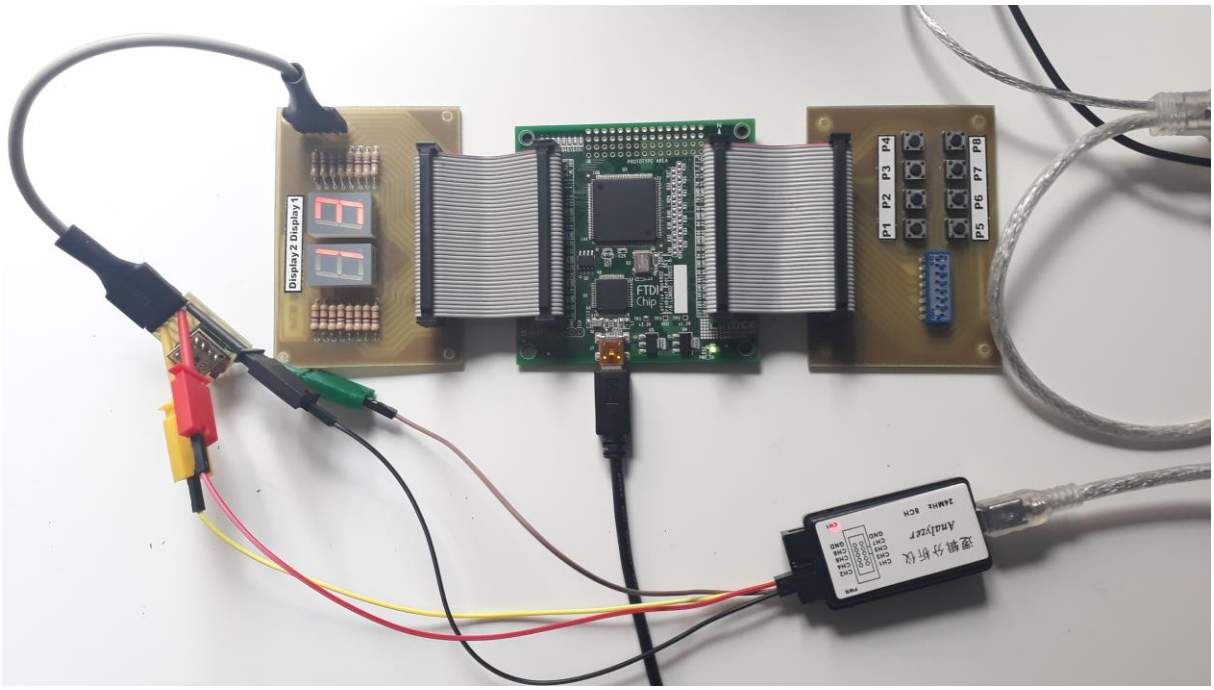


Figura 50. Conexión del analizador lógico en el convertidor de luz-frecuencia

En el primer canal se conecta la señal de GND que se tomará como referencia, este valor siempre se encontrará a 0.

Se tomarán dos muestras con dos medidas diferentes para observar las diferencias, ambas muestras son de 10 ms de duración.

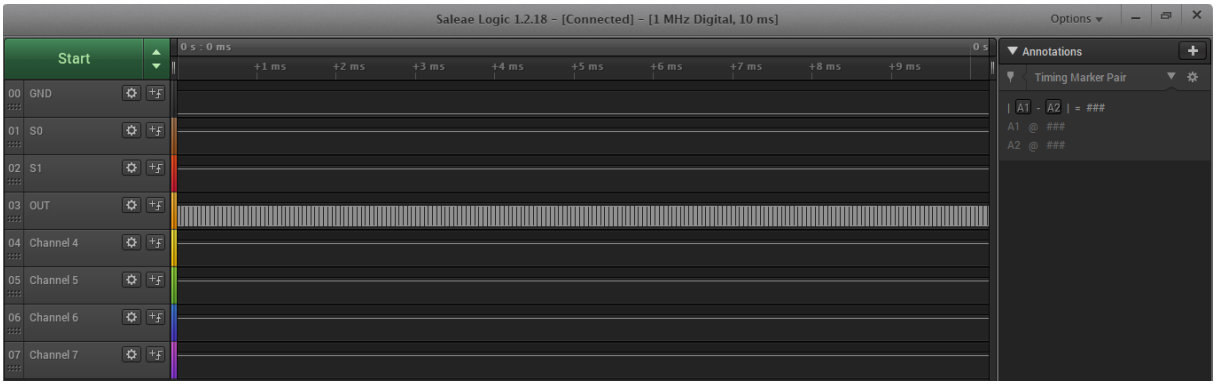


Figura 51. Datos obtenidos del convertidor de luz-frecuencia al 100%

Como se puede ver en la figura anterior, las señales S0 y S1 están a nivel alto, valen 1, esto significa que se toma el 100% de la frecuencia de medida sin realizar ningún escalado. Con esta configuración no se puede observar con claridad los datos, ya que la frecuencia es muy alta.

Se decide entonces cambiar la configuración de la medida para que se escale la frecuencia a un 2% de la frecuencia medida. Esto se logra poniendo el valor de S0 a 0 (se quita el jumper que tiene el convertidor de luz-frecuencia) y el de S1 a 1.

En la primera muestra se observa que la señal de OUT tiene una duración de 1,21 ms y una frecuencia de 826,4 Hz.

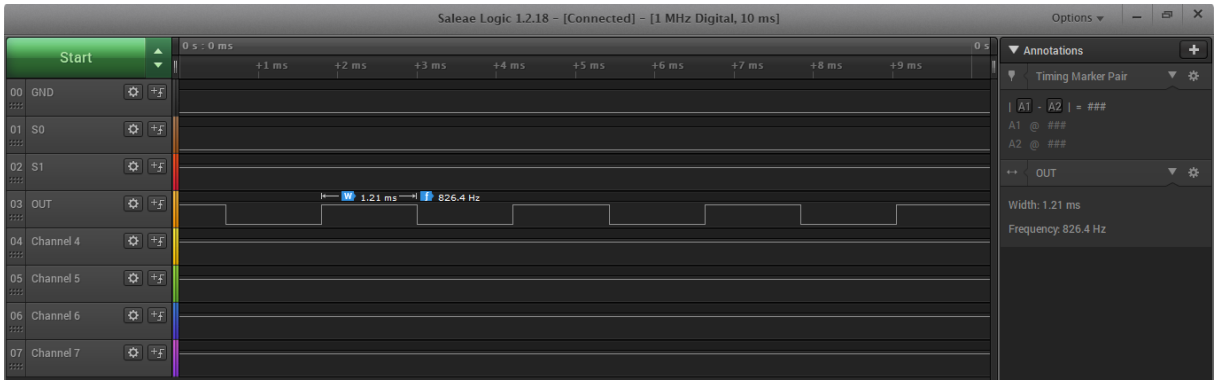


Figura 52. Datos obtenidos de la primera muestra del convertidor de luz-frecuencia

En la segunda muestra se observa que la señal de OUT tiene una duración de 0,156 ms y una frecuencia de 6,41 kHz.

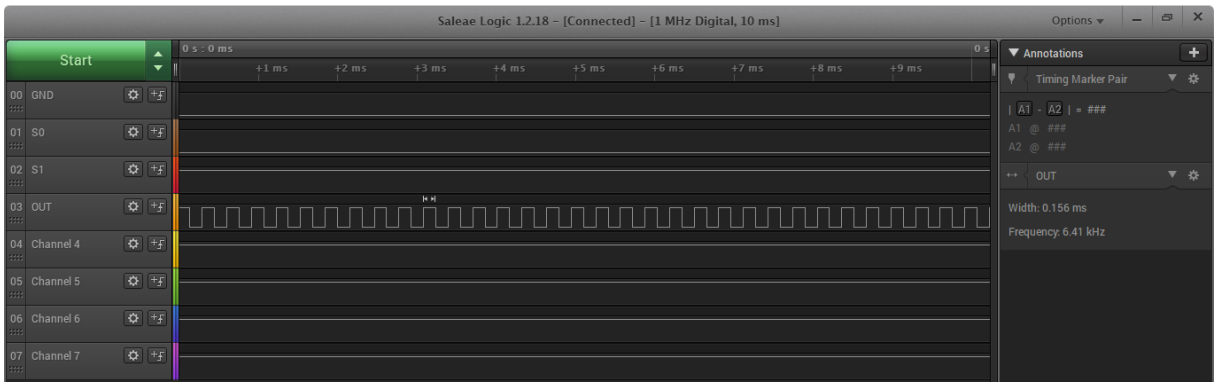


Figura 53. Datos obtenidos de la segunda muestra del convertidor de luz-frecuencia

Se llega a la conclusión de que la segunda muestra es correspondiente a una medida con mayor luz incidente porque la señal de OUT varía su frecuencia en función de la luz incidente.

4.2. Prueba de mediciones

En esta parte se adjuntan fotografías que demuestran el correcto funcionamiento de ambos sistemas, mostrando por los displays y leds de la placa las medidas de prueba que se han realizado.

Sensor de ultrasonidos:

Para comprobar la correcta salida por pantalla del resultado de la medición se ha realizado una serie de medidas colocando algún objeto a una distancia determinada y con la referencia de una cinta métrica.

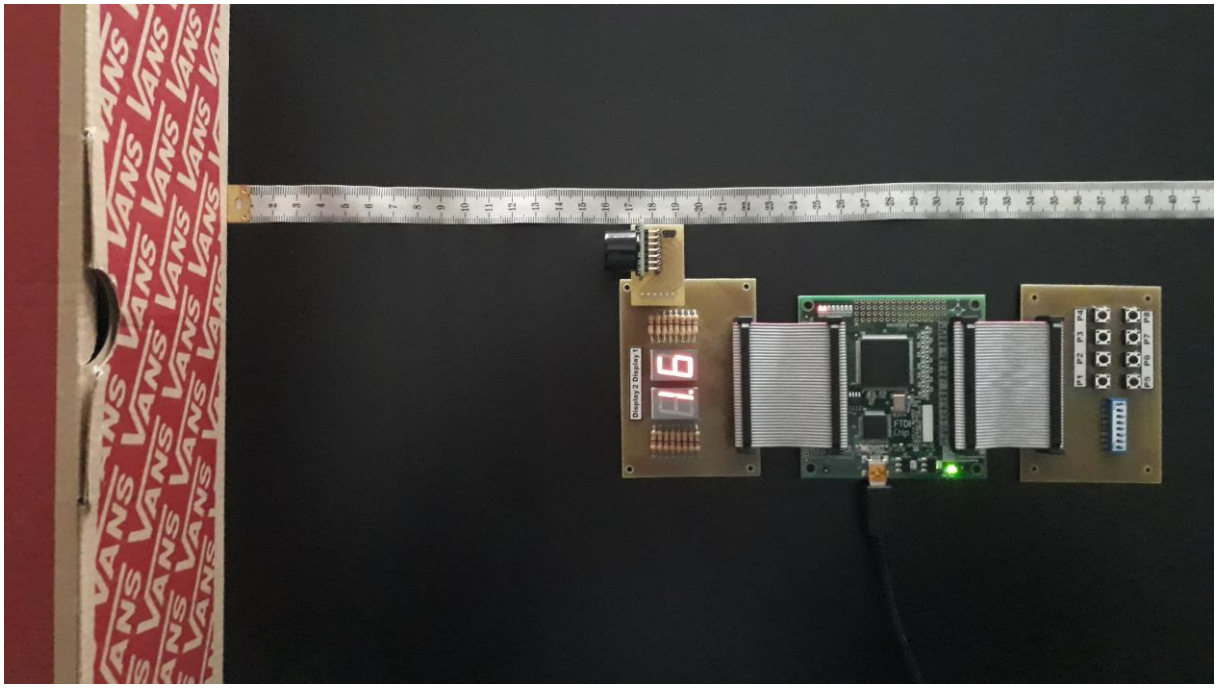


Figura 54. Medida de objeto a 16 cm de distancia

Figura 55. Medida de objeto a 52 cm de distancia

En estas dos primeras medidas se observa que solo se utilizan los 2 displays para mostrar la medida porque son valores de solo 2 dígitos (unidades y decenas).

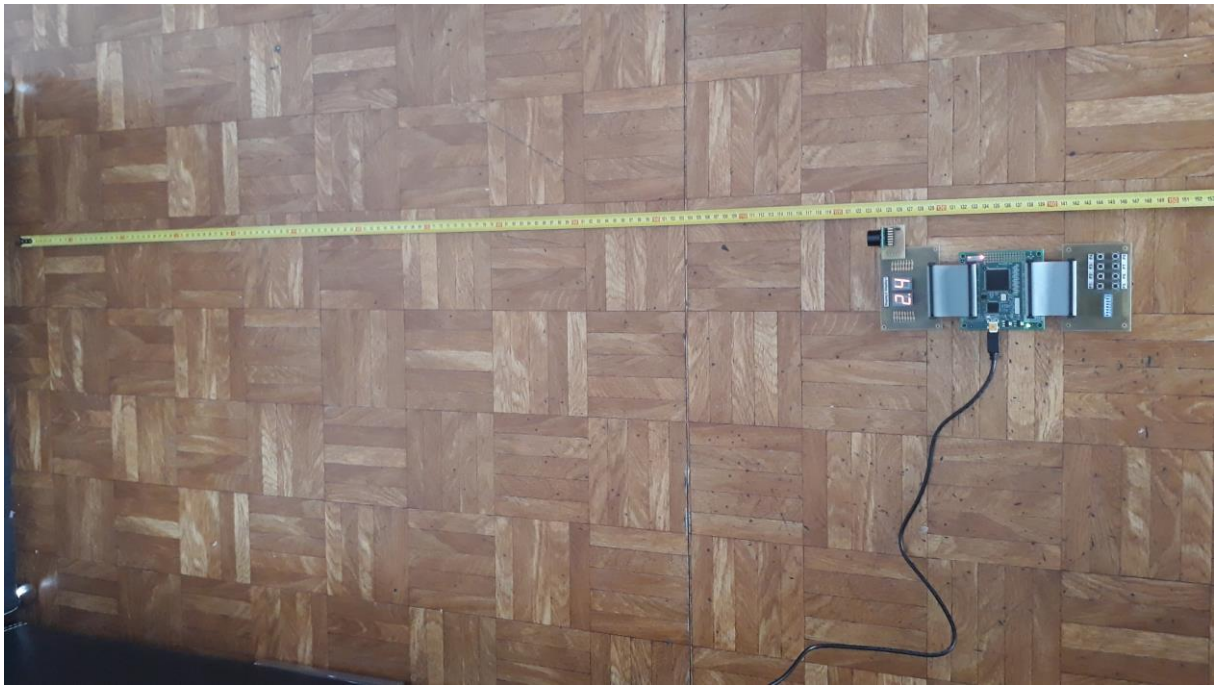


Figura 56. Medida de objeto a 124 cm de distancia

En el caso de la tercera medida se utiliza además el primer led que indica la primera centena.

Convertidor de luz-frecuencia:

En el caso del convertidor de luz-frecuencia, para comprobar la correcta salida por pantalla del resultado de la medición se ha realizado una serie de medidas con diferentes iluminaciones. Para realizar estas medidas se ha configurado el sensor para que no realice ningún escalado, es decir con S0 y S1 igual a 1.

En el primer caso se ha cubierto totalmente el sensor y la medida que se obtiene en este caso es un valor de 1.

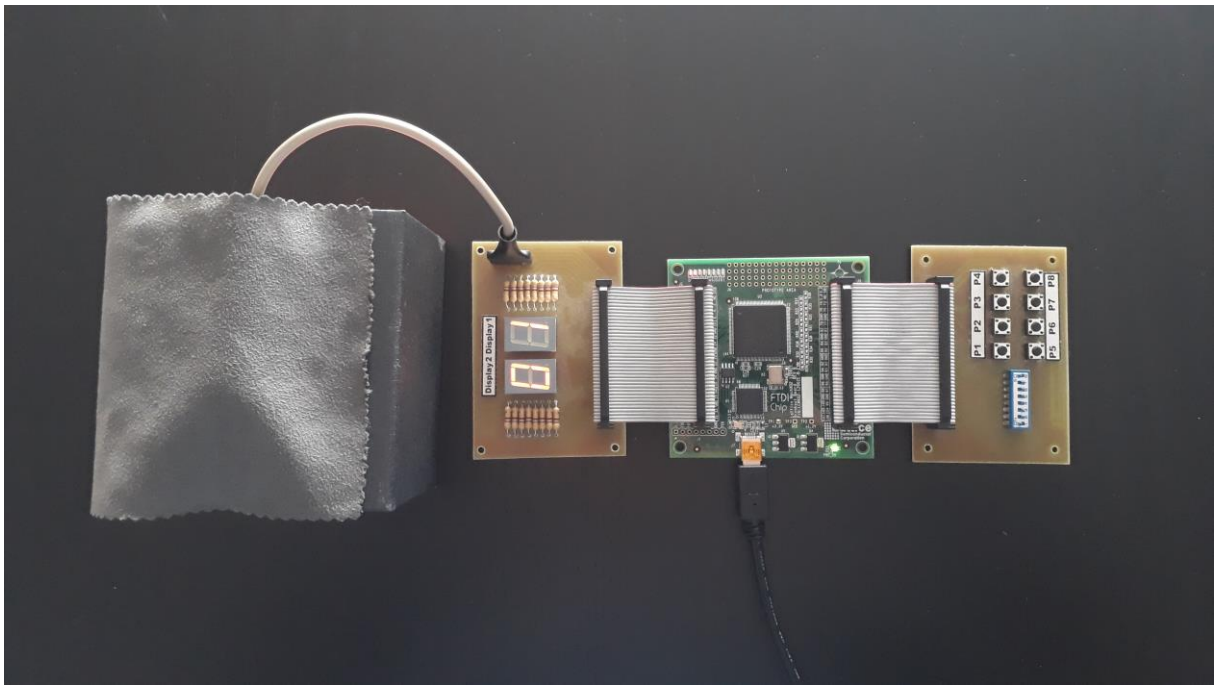


Figura 57. Primera medida del convertidor de luz-frecuencia

Para la segunda medida se ha descubierto el sensor y se ha realizado la medida con la luz ambiente. El resultado obtenido es el valor 29.

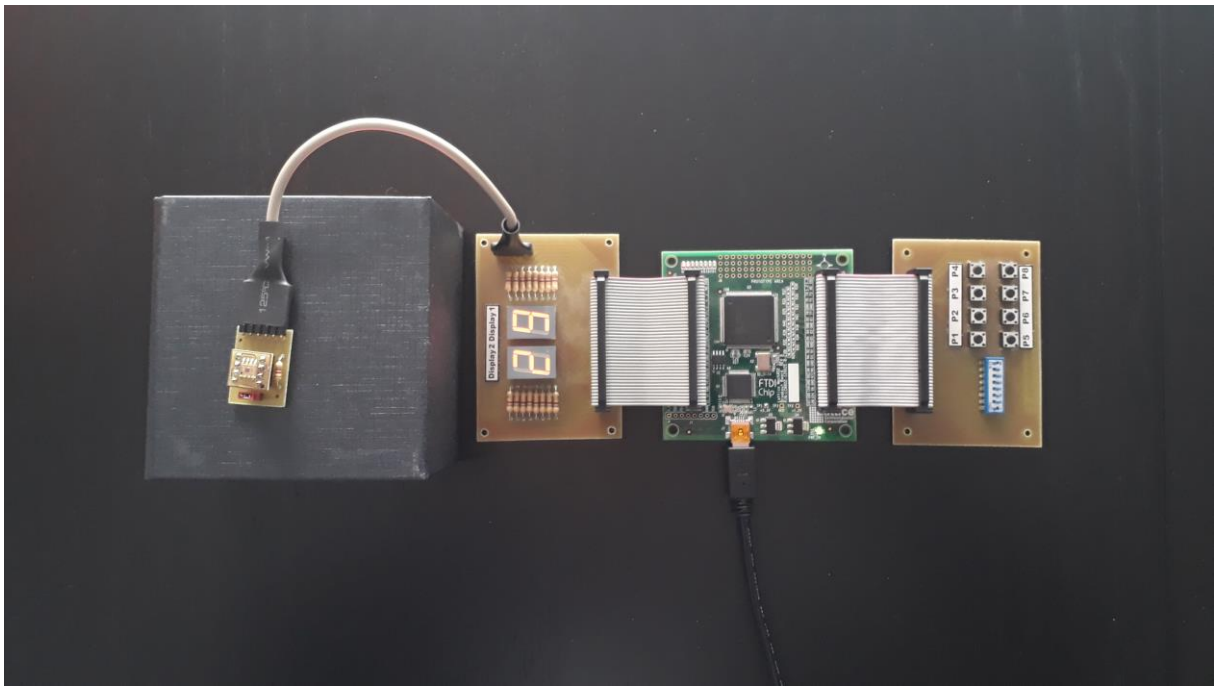


Figura 58. Segunda medida del convertidor de luz-frecuencia

Para las siguientes medidas se ha hecho uso de una linterna a diferentes distancias del sensor. Los resultados obtenidos en este caso son los valores 80, 144 y 360 respectivamente.

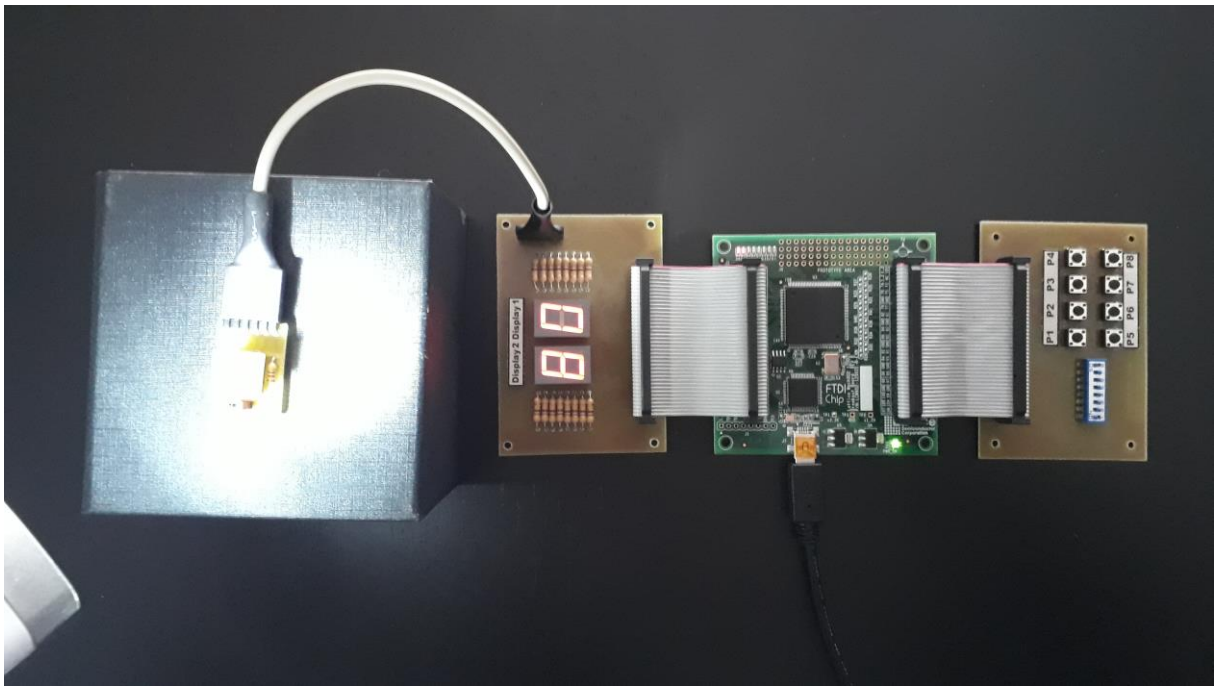


Figura 59. Tercera medida del convertidor de luz-frecuencia

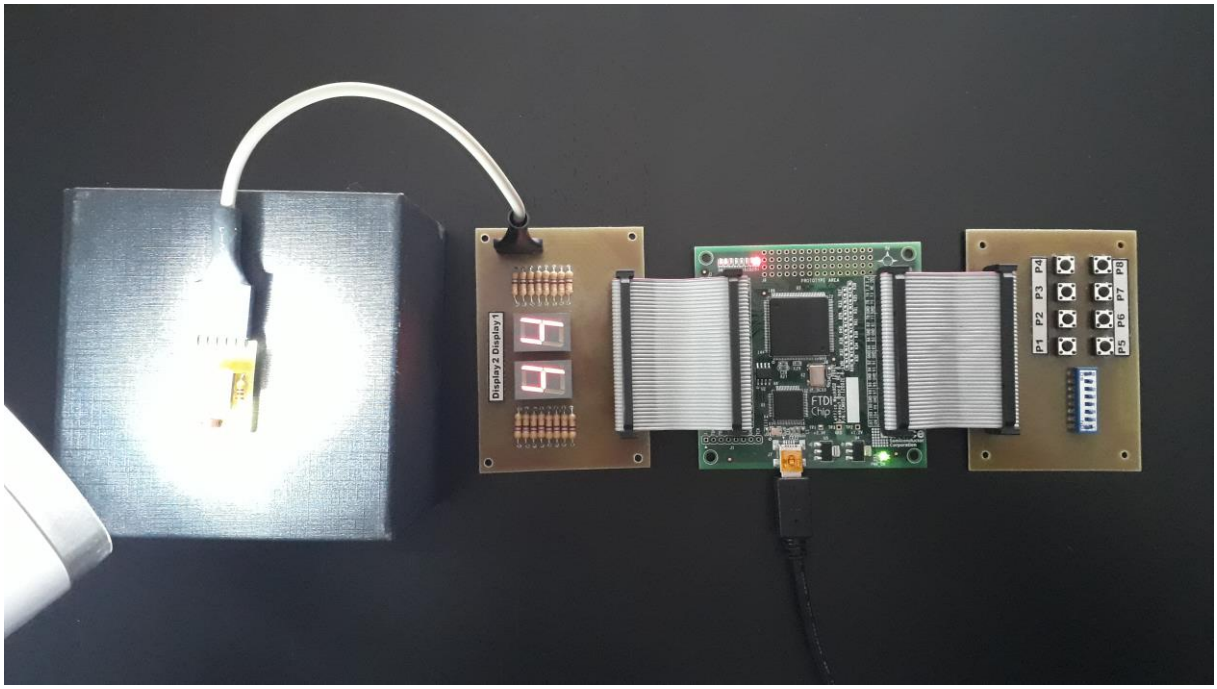


Figura 60. Cuarta medida del convertidor de luz-frecuencia

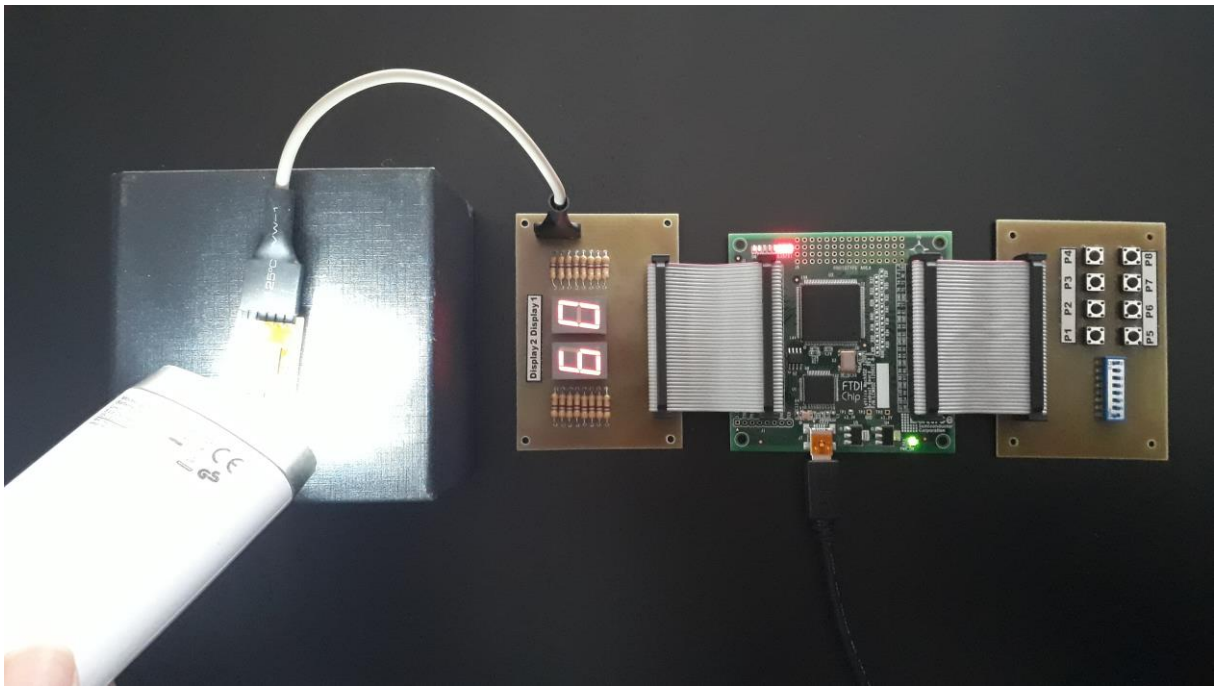


Figura 61. Quinta medida del convertidor de luz-frecuencia

Se puede observar como el valor detectado varía en función de la luz incidente en el sensor. La última medida es la máxima que puede detectar el sensor y corresponde con el valor 360.

5. ESTUDIO ECONÓMICO

Una vez concluido el proyecto, se ha realizado un cálculo de los recursos económicos necesarios para la realización del mismo.

Por una parte, se encuentran los costes de inversión, estos hacen referencia a los precios de adquisición de los materiales utilizados. En la siguiente tabla se resumen estos costes.

Componente	Coste Unitario	Cantidad	Coste Total
MachXO2-1200ZE	30 €	1	30 €
LV-MaxSonar-EZ3	32 €	1	32 €
TCS3210	4 €	1	4 €
Analizador lógico	10 €	1	10 €
			76 €

Figura 62. Costes de material utilizado

El uso del software necesario para la configuración de la FPGA, simulación de circuitos y análisis de señales no supone un coste adicional. Esto se debe a que todos los programas utilizados durante el proyecto son de licencia gratuita.

De esta forma, el coste total de inversión en material para el proyecto es de 76 €.

Por otra parte, estaría el coste de oportunidad, relacionado con el trabajo realizado por parte del ingeniero. Estimando una dedicación de 240 horas a la realización de la configuración del sistema y suponiendo un coste de 10 €/hora, el coste de oportunidad del proyecto ascendería a 2400 €.

En conclusión, si se suman los dos costes anteriores se obtendría que el coste total de la realización del presente proyecto es de 2476 €.

6. CONCLUSIONES

En este apartado se exponen las conclusiones alcanzadas después de la realización y la ejecución del proyecto.

Se ha comprobado la viabilidad de implantar, en un único dispositivo tipo FPGA, toda la lógica de control y monitorización de sensores externos. Esta implantación utiliza una pequeña parte de la lógica disponible en el dispositivo empleado, por lo que queda disponible, en dicho dispositivo, una gran cantidad de lógica.

Para llegar a este resultado, se han debido cumplir una serie de objetivos parciales:

- ✓ Se ha seleccionado el hardware de partida (FPGA) que se ha considerado más oportuno, tras analizar las alternativas que existen en el mercado.
- ✓ Se ha dotado al hardware de partida de un interfaz a base de interruptores, pulsadores y displays de 7 segmentos.
- ✓ Se ha integrado en el sistema un sensor de ultrasonidos.
- ✓ Se ha integrado en el sistema un convertidor de luz-frecuencia.
- ✓ Se han utilizado herramientas de diseño, simulación y síntesis modernas, obteniendo un diseño estructurado y jerárquico.
- ✓ Se han empleado técnicas de diseño mixtas, mezclando el uso de lenguajes de descripción de hardware y el diseño mediante esquemas. La descripción global se ha realizado a través de esquemas y algunos bloques funcionales se han descrito en lenguaje VHDL.
- ✓ Se han utilizado herramientas automáticas para la generación de descripciones VHDL.
- ✓ Se han utilizado herramientas de simulación de circuitos.
- ✓ Se han utilizado herramientas de análisis de señales (analizador lógico).

Líneas futuras:

Como líneas futuras de este proyecto, se podría estudiar el desarrollo de algunas posibles aplicaciones tales como:

- Utilizando la posibilidad que tiene el convertidor de luz-frecuencia de detectar la luz incidente en forma de luz roja, verde o azul, se podría diseñar un sistema para extraer las componentes RGB, indicando el porcentaje de cada uno de estos colores.
- Incorporar al sistema un microprocesador, en la lógica de propósito general que queda disponible.
- Añadir otro tipo de sensores con una configuración semejante a la utilizada en este proyecto para poder obtener un sistema de control completo.

7. BIBLIOGRAFÍA

- Lattice Semiconductor. *MachXO2 Family Data Sheet.* (2017)
- Lattice Semiconductor. *MachXO2-1200ZE Breakout Board Evaluation Kit User's Guide* (2012)
- MaxBotix. *LV-MaxSonar-EZ Series Data Sheet.*
- Texas Advanced Optoelectronic Solutions. *TCS3210 Light-to-Frequency Converter Data Sheet.* (2009)
- García Gallego, G. *Monitorización y control de la intensidad luminosa mediante FPGA.* Universidad de Valladolid. (2014)
- González Antón, J. *Medición de distancia y velocidad empleando un sensor de ultrasonidos.* Universidad de Valladolid. (2015)
- https://www.xilinx.com/products/silicon-devices/fpga.html [Consulta: marzo de 2019]
- https://www.intel.com/content/www/us/en/products/programmable/fpga.html [Consulta: marzo de 2019]
- http://www.latticesemi.com/en/Products [Consulta: marzo de 2019]

8. ANEXOS

LV-MaxSonar®-EZ™ Series

High Performance Sonar Range Finder

MB1000, MB1010, MB1020, MB1030, MB1040[2]

With 2.5V - 5.5V power the LV-MaxSonar-EZ provides very short to long-range detection and ranging in a very small package. The LV-MaxSonar-EZ detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects from 0-inches to 6-inches typically range as 6-inches[1]. The interface output formats included are pulse width output, analog voltage output, and RS232 serial output. Factory calibration and testing is completed with a flat object. ***[1]See Close Range Operation***

Features

- Continuously variable gain for control and side lobe suppression
- Object detection to zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- Interfaces are active simultaneously
- Serial, 0 to Vcc, 9600 Baud, 81N
- Analog, (Vcc/512) / inch
- Pulse width, (147uS/inch)
- Learns ringdown pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor operates at 42KHz
- High output square wave sensor drive (double Vcc)
- Actual operating temperature range from –40°C to +65°C, Recommended operating temperature range from 0°C to +60°C [1]

Benefits

- Very low cost ultrasonic rangefinder
- Reliable and stable range data
- Quality beam characteristics
- Mounting holes provided on the circuit board
- Very low power ranger, excellent for multiple sensor or battery-based systems
- Fast measurement cycles
- Sensor reports the range reading directly and frees up user processor
- Choose one of three sensor outputs
- Triggered externally or internally

Applications and Uses

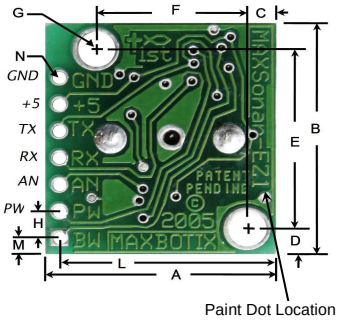
- UAV blimps, micro planes and some helicopters
- Bin level measurement
- Proximity zone detection
- People detection
- Robot ranging sensor
- Autonomous navigation
- Multi-sensor arrays
- Distance measuring
- Long range object detection
- Wide beam sensitivity

Notes:

[1]Please reference page 4 for minimum operating voltage verses temperature information.

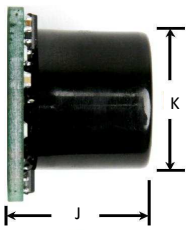
[2] Please reference page 12 for part number key.

LV-MaxSonar-EZ Mechanical Dimensions



A	0.785"	19.9 mm	H	0.100"	2.54 mm
B	0.870"	22.1 mm	J	0.610"	15.5 mm
C	0.100"	2.54 mm	K	0.645"	16.4 mm
D	0.100"	2.54 mm	L	0.735"	18.7 mm
E	0.670"	17.0 mm	M	0.065"	1.7 mm
F	0.510"	12.6 mm	N	0.038" dia.	1.0 mm dia.
G	0.124" dia.	3.1 mm dia.	weight, 4.3 grams		

Part Number	MB1000	MB1010	MB1020	MB1030	MB1040
Paint Dot Color	Black	Brown	Red	Orange	Yellow



Close Range Operation

Applications requiring 100% reading-to-reading reliability should not use MaxSonar sensors at a distance closer than 6 inches. Although most users find MaxSonar sensors to work reliably from 0 to 6 inches for detecting objects in many applications, MaxBotix® Inc. does not guarantee operational reliability for objects closer than the minimum reported distance. Because of ultrasonic physics, these sensors are unable to achieve 100% reliability at close distances.

Warning: Personal Safety Applications

We do not recommend or endorse this product be used as a component in any personal safety applications. This product is not designed, intended or authorized for such use. These sensors and controls do not include the self-checking redundant circuitry needed for such use. Such unauthorized use may create a failure of the MaxBotix® Inc. product which may result in personal injury or death. MaxBotix® Inc. will not be held liable for unauthorized use of this component.

About Ultrasonic Sensors

Our ultrasonic sensors are in air, non-contact object detection and ranging sensors that detect objects within an area. These sensors are not affected by the color or other visual characteristics of the detected object. Ultrasonic sensors use high frequency sound to detect and localize objects in a variety of environments. Ultrasonic sensors measure the time of flight for sound that has been transmitted to and reflected back from nearby objects. Based upon the time of flight, the sensor then outputs a range reading.

Pin Out Description

Pin 1-BW-*Leave open or hold low for serial output on the TX output. When BW pin is held high the TX output sends a pulse (instead of serial data), suitable for low noise chaining.

Pin 2-PW- This pin outputs a pulse width representation of range. The distance can be calculated using the scale factor of 147uS per inch.

Pin 3-AN- Outputs analog voltage with a scaling factor of (Vcc/512) per inch. A supply of 5V yields ~9.8mV/in. and 3.3V yields ~6.4mV/in. The output is buffered and corresponds to the most recent range data.

Pin 4-RX– This pin is internally pulled high. The LV-MaxSonar-EZ will continually measure range and output if RX data is left unconnected or held high. If held low the sensor will stop ranging. Bring high for 20uS or more to command a range reading.

Pin 5-TX- When the *BW is open or held low, the TX output delivers asynchronous serial with an RS232 format, except voltages are 0-Vcc. The output is an ASCII capital "R", followed by three ASCII character digits representing the range in inches up to a maximum of 255, followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltage of 0-Vcc is outside the RS232 standard, most RS232 devices have sufficient margin to read 0-Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232. When BW pin is held high the TX output sends a single pulse, suitable for low noise chaining. (no serial data)

Pin 6-+5V- Vcc – Operates on 2.5V - 5.5V. Recommended current capability of 3mA for 5V, and 2mA for 3V. Please reference page 4 for minimum operating voltage verses temperature information.

Pin 7-GND- Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

Range "0" Location

Range Zero

The range is measured from the front of the transducer.

The LV-MaxSonar-EZ reports the range to distant targets starting from the front of the sensor as shown in the diagram below.

In general, the LV-MaxSonar-EZ will report the range to the leading edge of the closest detectable object. Target detection has been characterized in the sensor beam patterns.

Sensor Minimum Distance

The sensor minimum reported distance is 6-inches (15.2 cm). However, the LV-MaxSonar-EZ will range and report targets to the front sensor face. Large targets closer than 6-inches will typically range as 6-inches.

Sensor Operation from 6-inches to 20-inches

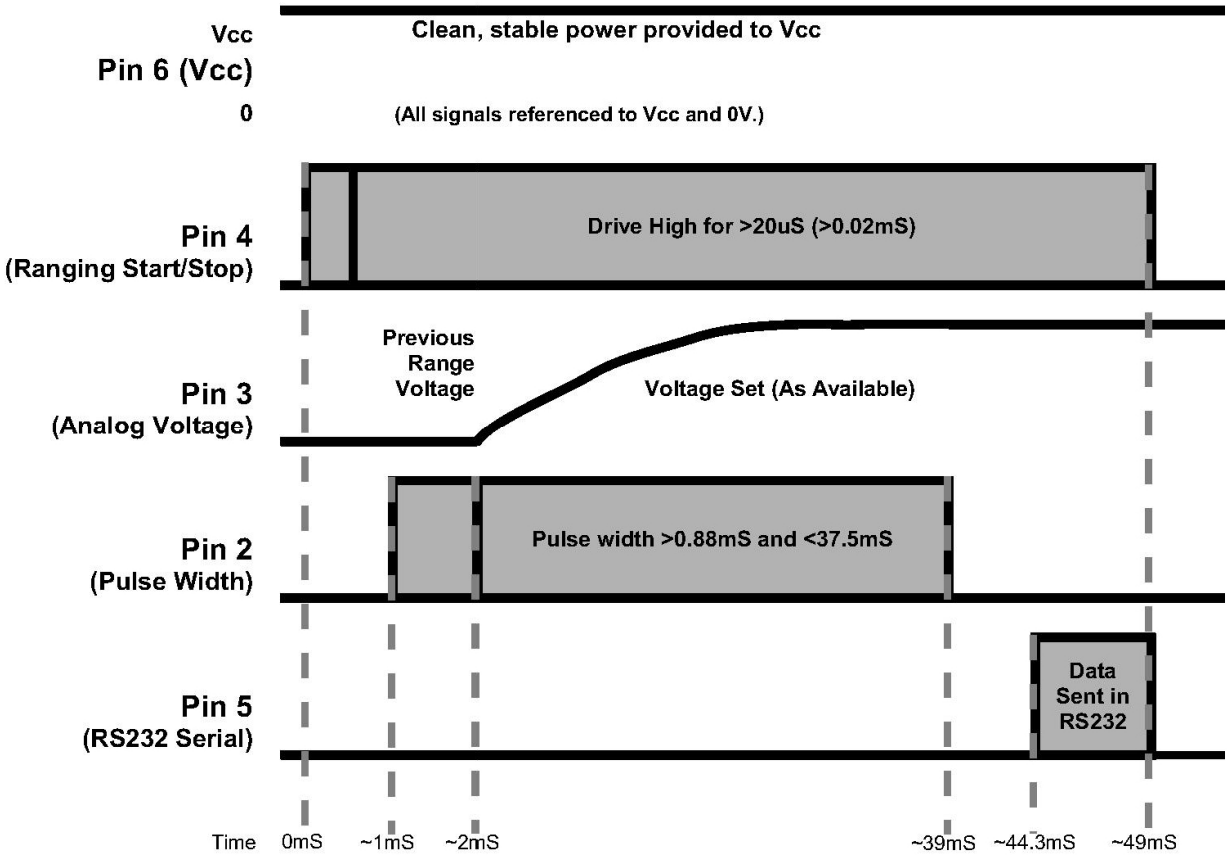
Because of acoustic phase effects in the near field, objects between 6-inches and 20-inches may experience acoustic phase

cancellation of the returning waveform resulting in inaccuracies of up to 2-inches. These effects become less prevalent as the target distance increases, and has not been observed past 20-inches.

General Power-Up Instruction

Each time the LV-MaxSonar-EZ is powered up, it will calibrate during its first read cycle. The sensor uses this stored information to range a close object. It is important that objects not be close to the sensor during this calibration cycle. The best sensitivity is obtained when the detection area is clear for fourteen inches, but good results are common when clear for at least seven inches. If an object is too close during the calibration cycle, the sensor may ignore objects at that distance.

The LV-MaxSonar-EZ does not use the calibration data to temperature compensate for range, but instead to compensate for the sensor ringdown pattern. If the temperature, humidity, or applied voltage changes during operation, the sensor may require recalibration to reacquire the ringdown pattern. Unless recalibrated, if the temperature increases, the sensor is more likely to have false close readings. If the temperature decreases, the sensor is more likely to have reduced up close sensitivity. To recalibrate the LV-MaxSonar-EZ, cycle power, then command a read cycle.



Timing Diagram

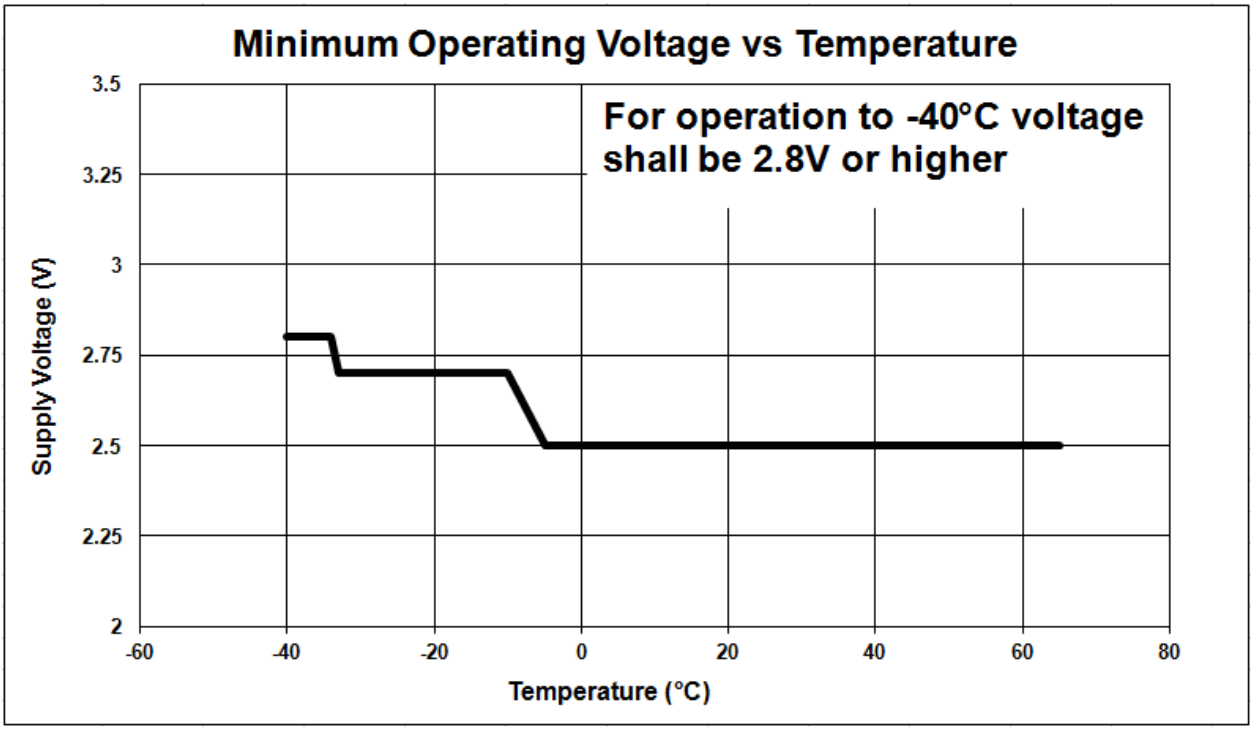
Timing Description

250mS after power-up, the LV-MaxSonar-EZ is ready to accept the RX command. If the RX pin is left open or held high, the sensor will first run a calibration cycle (49mS), and then it will take a range reading (49mS). After the power up delay, the first reading will take an additional ~100mS. Subsequent readings will take 49mS. The LV-MaxSonar-EZ checks the RX pin at the end of every cycle. Range data can be acquired once every 49mS.

Each 49mS period starts by the RX being high or open, after which the LV-MaxSonar-EZ sends the transmit burst, after which the pulse width pin (PW) is set high. When a target is detected the PW pin is pulled low. The PW pin is high for up to 37.5mS if no target is detected. The remainder of the 49mS time (less 4.7mS) is spent adjusting the analog voltage to the correct level. When a long distance is measured immediately after a short distance reading, the analog voltage may not reach the exact level within one read cycle. During the last 4.7mS, the serial data is sent.

Voltage vs Temperature

The graph below shows minimum operating voltage of the sensor verses temperature.

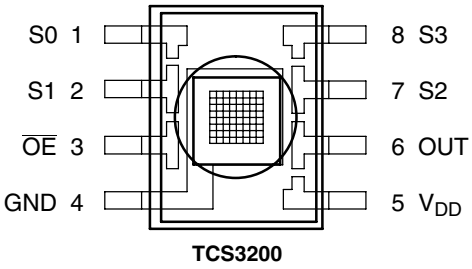


TCS3200, TCS3210 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

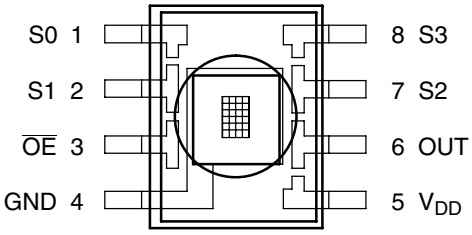
TAOS099 – JULY 2009

- **High-Resolution Conversion of Light Intensity to Frequency**
- **Programmable Color and Full-Scale Output Frequency**
- **Communicates Directly With a Microcontroller**
- **Single-Supply Operation (2.7 V to 5.5 V)**
- **Power Down Feature**
- **Nonlinearity Error Typically 0.2% at 50 kHz**
- **Stable 200 ppm/°C Temperature Coefficient**
- **Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package**

PACKAGE D
8-LEAD SOIC
(TOP VIEW)



TCS3200



TCS3210

Description

The TCS3200 and TCS3210 programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance).

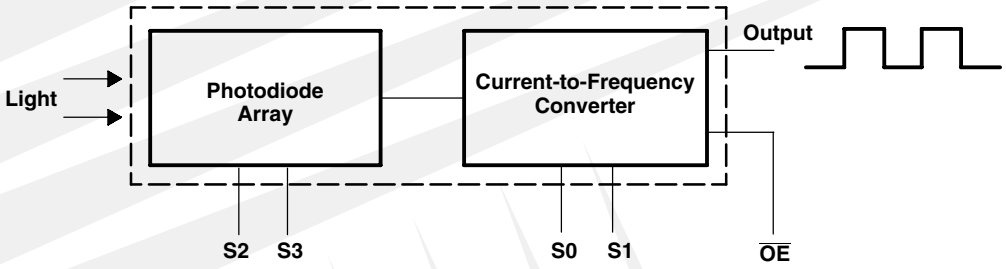
The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable ($\overline{OE}$) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

In the TCS3200, the light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters.

In the TCS3210, the light-to-frequency converter reads a 4 x 6 array of photodiodes. Six photodiodes have blue filters, 6 photodiodes have green filters, 6 photodiodes have red filters, and 6 photodiodes are clear with no filters.

The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of the same color are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 µm x 110 µm in size and are on 134-µm centers.

Functional Block Diagram



Texas Advanced Optoelectronic Solutions Inc.
1001 Klein Road • Suite 300 • Plano, TX 75074 • (972) 673-0759
www.taosinc.com

Terminal Functions

TERMINAL NAME	TERMINAL NO.	I/O	DESCRIPTION
GND	4		Power supply ground. All voltages are referenced to GND.
$\overline{OE}$	3	I	Enable for f_o (active low).
OUT	6	O	Output frequency (f_o).
S0, S1	1, 2	I	Output frequency scaling selection inputs.
S2, S3	7, 8	I	Photodiode type selection inputs.
V_{DD}	5		Supply voltage

Table 1. Selectable Options

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Available Options

DEVICE	T_A	PACKAGE – LEADS	PACKAGE DESIGNATOR	ORDERING NUMBER
TCS3200	–40°C to 85°C	SOIC–8	D	TCS3200D
TCS3210	–40°C to 85°C	SOIC–8	D	TCS3210D

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{DD} (see Note 1) . . . . . 6 V
Input voltage range, all inputs, V_I . . . . . –0.3 V to V_{DD} + 0.3 V
Operating free-air temperature range, T_A (see Note 2) . . . . . –40°C to 85°C
Storage temperature range (see Note 2) . . . . . –40°C to 85°C
Solder conditions in accordance with JEDEC J–STD–020A, maximum temperature (see Note 3) . . . 260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to GND.
2. Long-term storage or operation above 70°C could cause package yellowing that will lower the sensitivity to wavelengths < 500nm.
3. The device may be hand soldered provided that heat is applied only to the solder pad and no contact is made between the tip of the solder iron and the device lead. The maximum time heat should be applied to the device is 5 seconds.

Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
Supply voltage, V_{DD}		2.7	5	5.5	V
High-level input voltage, V_{IH}	V_{DD} = 2.7 V to 5.5 V	2		V_{DD}	V
Low-level input voltage, V_{IL}	V_{DD} = 2.7 V to 5.5 V	0		0.8	V
Operating free-air temperature range, T_A		–40		70	°C

Electrical Characteristics at T_A = 25°C, V_{DD} = 5 V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = –2 mA	4	4.5		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA		0.25	0.40	V
I_{IH}	High-level input current				5	µA
I_{IL}	Low-level input current				5	µA
I_{DD}	Supply current	Power-on mode		1.4	2	mA
		Power-down mode			0.1	µA
	Full-scale frequency (See Note 4)	S0 = H, S1 = H	500	600		kHz
		S0 = H, S1 = L	100	120		kHz
		S0 = L, S1 = H	10	12		kHz
	Temperature coefficient of responsivity	λ ≤ 700 nm, –25°C ≤ T_A ≤ 70°C		±200		ppm/°C
k_{SVS}	Supply voltage sensitivity	V_{DD} = 5 V ±10%		±0.5		%/V

NOTE 4: Full-scale frequency is the maximum operating frequency of the device without saturation.

Operating Characteristics at V_{DD} = 5 V, T_A = 25°C, S0 = H, S1 = H (unless otherwise noted) (See Notes 5, 6, 7, and 8). Values for TCS3200 (TCS3210) are below.

PARAMETER		TEST CONDITIONS	CLEAR PHOTODIODE S2 = H, S3 = L			BLUE PHOTODIODE S2 = L, S3 = H			GREEN PHOTODIODE S2 = H, S3 = H			RED PHOTODIODE S2 = L, S3 = L			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_O	Output frequency (Note 9)	E_e = 47.2 µW/cm², λ_p = 470 nm	12.5	15.6	18.7	61%		84%	22%		43%	0%		6%	kHz
			(4.7)	(5.85)	(7)										
		E_e = 40.4 µW/cm², λ_p = 524 nm	12.5	15.6	18.7	8%		28%	57%		80%	9%		27%	
			(4.7)	(5.85)	(7)										
		E_e = 34.6 µW/cm², λ_p = 640 nm	13.1	16.4	19.7	5%		21%	0%		12%	84%		105%	
			(4.9)	(6.15)	(7.4)										
R_e	Irradiance responsivity (Note 10)	λ_p = 470 nm		331		61%		84%	22%		43%	0%		6%	Hz/ (µW/ cm²)
				(124)											
		λ_p = 524 nm		386		8%		28%	57%		80%	9%		27%	
				(145)											
		λ_p = 640 nm		474		5%		21%	0%		12%	84%		105%	
				(178)											
	Saturation irradiance (Note 11)	λ_p = 470 nm		1813			––			––			––		µW/ cm²
				(4839)											
		λ_p = 524 nm		1554			––			––			––		
				(4138)											
		λ_p = 640 nm		1266			––			––			––		
				(3371)											
f_D	Dark frequency	E_e = 0		2	10		2	10		2	10		2	10	Hz
	Nonlinearity (Note 12)	f_O = 0 to 5 kHz		±0.1			±0.1			±0.1			±0.1		% F.S.
		f_O = 0 to 50 kHz		±0.2			±0.2			±0.2			±0.2		
		f_O = 0 to 500 kHz		±0.5			±0.5			±0.5			±0.5		
	Recovery from power down			100			100			100			100		µs
	Response time to output enable ($\overline{OE}$)			100			100			100			100		ns

NOTES: 5. Optical measurements are made using small-angle incident radiation from a light-emitting diode (LED) optical source.
6. The 470 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength λ_p = 470 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 35 nm, and luminous efficacy = 75 lm/W.
7. The 524 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength λ_p = 524 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 47 nm, and luminous efficacy = 520 lm/W.
8. The 640 nm input irradiance is supplied by a AlInGaP light-emitting diode with the following characteristics: peak wavelength λ_p = 640 nm, spectral halfwidth $\Delta\lambda_{1/2}$ = 17 nm, and luminous efficacy = 155 lm/W.
9. Output frequency Blue, Green, Red percentage represents the ratio of the respective color to the Clear channel absolute value.
10. Irradiance responsivity R_e is characterized over the range from zero to 5 kHz.
11. Saturation irradiance = (full-scale frequency)/(irradiance responsivity) for the Clear reference channel.
12. Nonlinearity is defined as the deviation of f_O from a straight line between zero and full scale, expressed as a percent of full scale.

TYPICAL CHARACTERISTICS

PHOTODIODE SPECTRAL RESPONSIVITY

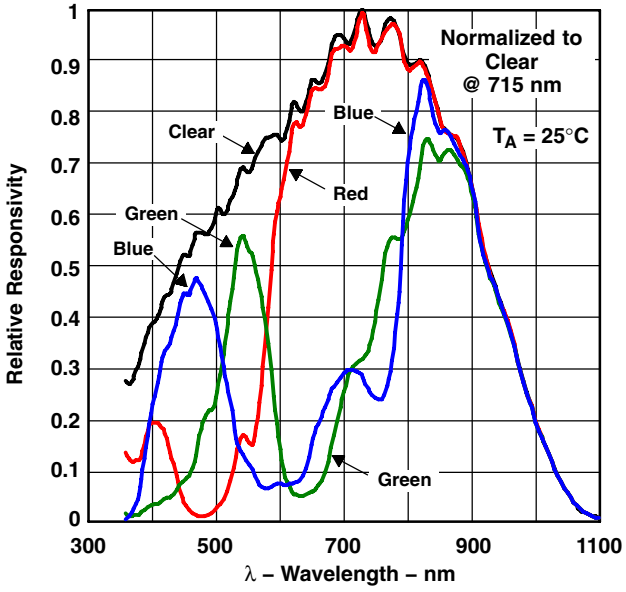


Figure 1

NORMALIZED OUTPUT FREQUENCY
vs.
ANGULAR DISPLACEMENT

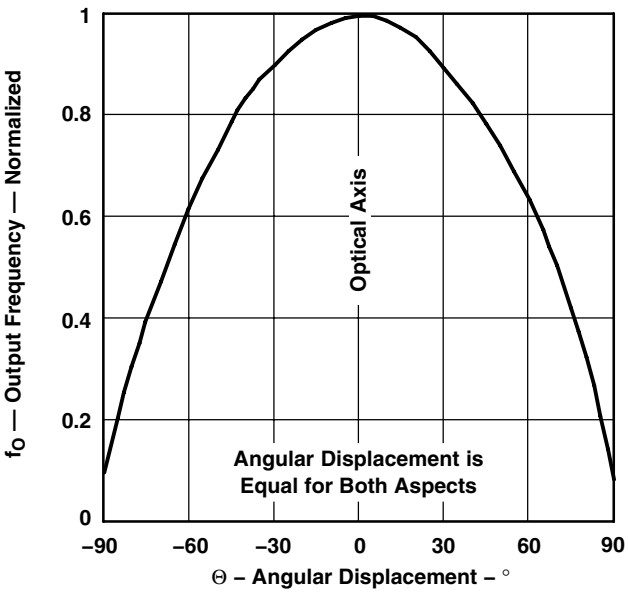


Figure 2

I_{DD} vs.
V_{DD} vs.
TEMPERATURE

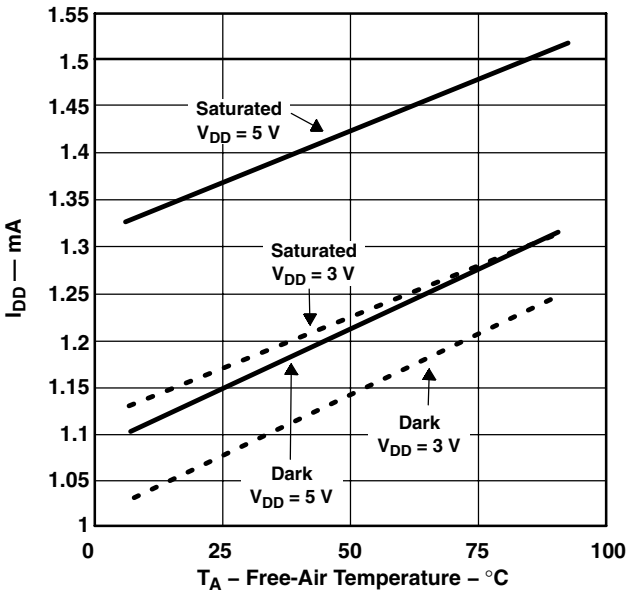


Figure 3

NORMALIZED OUTPUT
vs.
V_{DD}

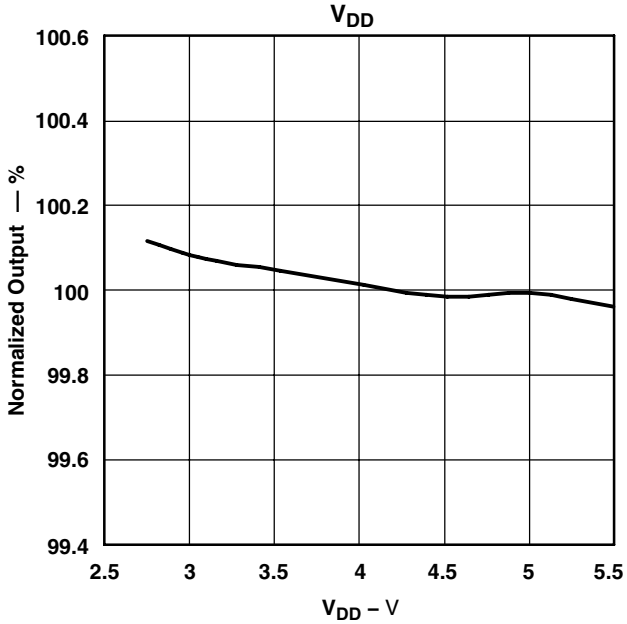


Figure 4

TYPICAL CHARACTERISTICS

PHOTODIODE RESPONSIVITY TEMPERATURE COEFFICIENT vs. WAVELENGTH OF INCIDENT LIGHT

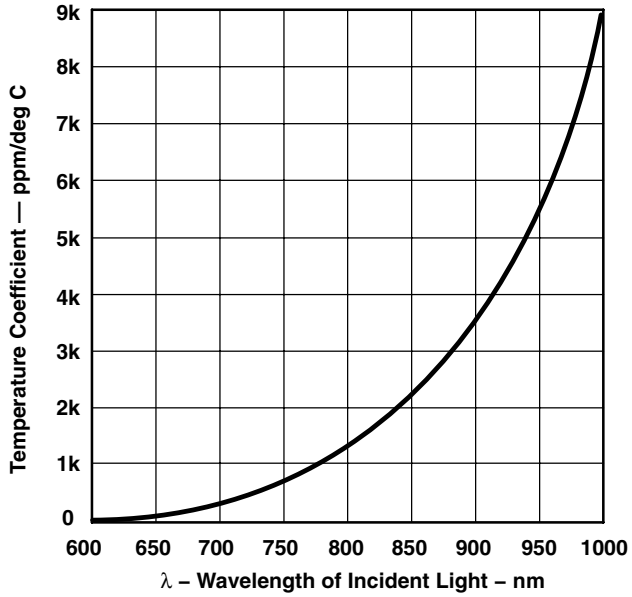


Figure 5

APPLICATION INFORMATION

Power supply considerations

Power-supply lines must be decoupled by a 0.01-μF to 0.1-μF capacitor with short leads mounted close to the device package.

Input interface

A low-impedance electrical connection between the device $\overline{OE}$ pin and the device GND pin is required for improved noise immunity. All input pins must be either driven by a logic signal or connected to VDD or GND — they should not be left unconnected (floating).

Output interface

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances. If lines greater than 12 inches are used on the output, a buffer or line driver is recommended.

A high state on Output Enable (OE) places the output in a high-impedance state for multiple-unit sharing of a microcontroller input line.

Power down

Powering down the sensor using S0/S1 (L/L) will cause the output to be held in a high-impedance state. This is similar to the behavior of the output enable pin, however powering down the sensor saves significantly more power than disabling the sensor with the output enable pin.

Photodiode type (color) selection

The type of photodiode (blue, green, red, or clear) used by the device is controlled by two logic inputs, S2 and S3 (see Table 1).

Output frequency scaling

Output-frequency scaling is controlled by two logic inputs, S0 and S1. The internal light-to-frequency converter generates a fixed-pulsewidth pulse train. Scaling is accomplished by internally connecting the pulse-train output of the converter to a series of frequency dividers. Divided outputs are 50%-duty cycle square waves with relative frequency values of 100%, 20%, and 2%. Because division of the output frequency is accomplished by counting pulses of the principal internal frequency, the final-output period represents an average of the multiple periods of the principle frequency.

The output-scaling counter registers are cleared upon the next pulse of the principal frequency after any transition of the S0, S1, S2, S3, and $\overline{OE}$ lines. The output goes high upon the next subsequent pulse of the principal frequency, beginning a new valid period. This minimizes the time delay between a change on the input lines and the resulting new output period. The response time to an input programming change or to an irradiance step change is one period of new frequency plus 1 μs. The scaled output changes both the full-scale frequency and the dark frequency by the selected scale factor.

The frequency-scaling function allows the output range to be optimized for a variety of measurement techniques. The scaled-down outputs may be used where only a slower frequency counter is available, such as low-cost microcontroller, or where period measurement techniques are used.

APPLICATION INFORMATION

Measuring the frequency

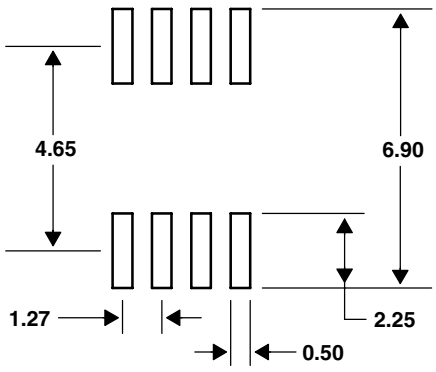
The choice of interface and measurement technique depends on the desired resolution and data acquisition rate. For maximum data-acquisition rate, period-measurement techniques are used.

Output data can be collected at a rate of twice the output frequency or one data point every microsecond for full-scale output. Period measurement requires the use of a fast reference clock with available resolution directly related to reference clock rate. Output scaling can be used to increase the resolution for a given clock rate or to maximize resolution as the light input changes. Period measurement is used to measure rapidly varying light levels or to make a very fast measurement of a constant light source.

Maximum resolution and accuracy may be obtained using frequency-measurement, pulse-accumulation, or integration techniques. Frequency measurements provide the added benefit of averaging out random- or high-frequency variations (jitter) resulting from noise in the light signal. Resolution is limited mainly by available counter registers and allowable measurement time. Frequency measurement is well suited for slowly varying or constant light levels and for reading average light levels over short periods of time. Integration (the accumulation of pulses over a very long period of time) can be used to measure exposure, the amount of light present in an area over a given time period.

PCB Pad Layout

Suggested PCB pad layout guidelines for the D package are shown in Figure 6.



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Figure 6. Suggested D Package PCB Layout

MachXO2-1200ZE Breakout Board Evaluation Kit

User's Guide

January 2012
Revision: EB68_01.1

Introduction

Thank you for choosing the Lattice Semiconductor MachXO2™-1200ZE Breakout Board Evaluation Kit!

This user's guide describes how to start using the MachXO2-1200ZE Breakout Board, an easy-to-use platform for evaluating and designing with the MachXO2-1200ZE PLD. Along with the board and accessories, this kit includes a pre-loaded demonstration design. You may also reprogram the on-board MachXO2-1200ZE device to review your own custom designs.

Note: Static electricity can severely shorten the lifespan of electronic components. See the Storage and Handling section of this document for handling and storage tips.

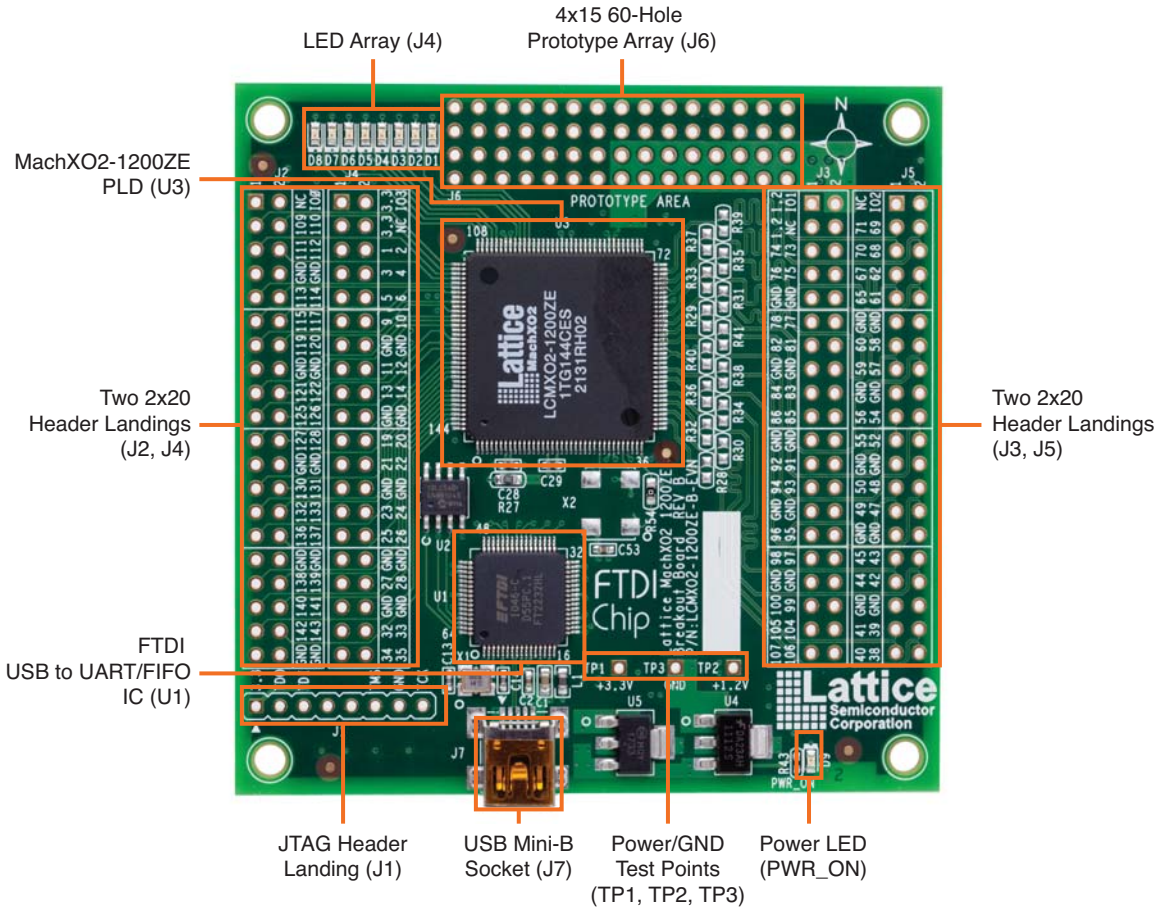
Features

The MachXO2-1200ZE Breakout Board Evaluation Kit includes:

- **MachXO2-1200ZE Breakout Board** – The board is a 3" x 3" form factor that features the following on-board components and circuits:
  - MachXO2-1200ZE PLD (LCMXO2-1200ZE-1TG144C)
  - USB mini-B connector for power and programming
  - Eight LEDs
  - 60-hole prototype area
  - Four 2x20 expansion header landings for general I/O, JTAG, and external power
  - 1x8 expansion header landing for JTAG
  - 3.3V and 1.2V supply rails
- **Pre-loaded Demo** – The kit includes a pre-loaded counter design that highlights use of the embedded MachXO2-1200ZE oscillator and programmable I/Os configured for LED drive.
- **USB Connector Cable** – The board is powered from the USB mini-B socket when connected to a host PC. The USB channel also provides a programming interface to the LCMXO2-1200ZE JTAG port.
- **Lattice Breakout Board Evaluation Kits Web Page** – Visit www.latticesemi.com/breakoutboards for the latest documentation (including this guide) and drivers for the kit.

The content of this user's guide includes demo operation, programming instructions, top-level functional descriptions of the Breakout Board, descriptions of the on-board connectors, and a complete set of schematics.

Figure 1. MachXO2-1200ZE Breakout Board, Top Side



Storage and Handling

Static electricity can shorten the lifespan of electronic components. Please observe these tips to prevent damage that could occur from electro-static discharge:

- Use anti-static precautions such as operating on an anti-static mat and wearing an anti-static wrist-band.
- Store the evaluation board in the packaging provided.
- Touch a metal USB housing to equalize voltage potential between you and the board.

Software Requirements

You should install the following software before you begin developing new designs for the Breakout board:

- Lattice Diamond® design software
- FTDI Chip USB hardware drivers (installed as an option within the Diamond installation program)

MachXO2-1200ZE Device

This board features the MachXO2-1200ZE PLD which offers embedded Flash technology for instant-on, non-volatile operation in a single chip. Numerous system functions are included, such as a PLL and 64Kbits of embedded RAM plus hardened implementations of I^2C, SPI, timer/counter, and user Flash memory. Flexible, high performance I/Os support numerous single-ended and differential standards including LVDS, and also source synchro-

nous interfaces to DDR/DDR2/LPDDR DRAM memory. The 144-pin TQFP package provides 108 user I/Os in a 20mm x 20mm form factor. A complete description of this device can be found in the MachXO2 Family Data Sheet.

Demonstration Design

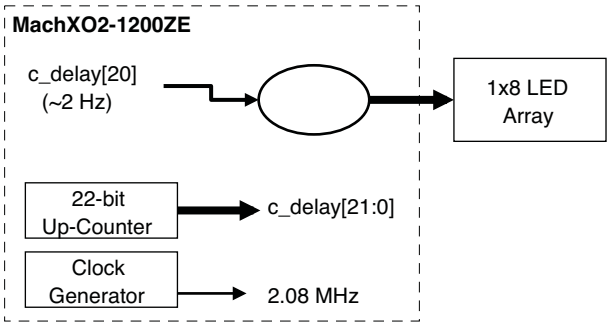
Lattice provides a simple, pre-programmed demo to illustrate basic operation of the MachXO2-1200ZE device. The design integrates an up-counter with the on-chip oscillator.

Note: You may obtain your Breakout Board after it has been reprogrammed. To restore the factory default demo and program it with other Lattice-supplied examples see the Download Demo Designs section of this document.

Run the Demonstration Design

Upon power-up, the preprogrammed demonstration design automatically loads and drives the LED array in an alternating pattern. The program shows a clock generator based on the MachXO2-1200ZE on-chip oscillator. The counter module is clocked at the oscillator default frequency of 2.08MHz to illustrate how low speed timer functions can be implemented with a PLD. The 22-bit up-counter further divides the clock to advance the LED display approximately every 500ms. The resulting light pattern will appear as an alternating pair of lit LEDs per row.

Figure 2. Demonstration Design Block Diagram



WARNING: Do not connect the Breakout Board to your PC before you follow the driver installation procedure of this section.

Communication with the Breakout Board with a PC via the USB connection cable requires installation of the FTDI chip USB hardware drivers. Loading these drivers enables the computer to recognize and program the Breakout Board. Drivers can be loaded as part of the installation of Lattice Diamond design software or Diamond Programmer, or as a stand-alone package.

To load the FTDI Chip USB hardware drivers as part of the Lattice Diamond installation:

1. Select **Programmer Drivers** in the Product Options of Lattice Diamond Setup.
2. Select **FTDI Windows USB Driver** or **All Drivers** in the LSC Drivers Install/Uninstall dialog box.
3. Click **Finish** to install the USB driver.
4. After the driver installation is complete, connect the USB cable from a USB port on your PC to the board's USB mini-B socket (J2). After the connection is made, a green Power LED (D9) will light indicating the board is powered on.
5. The demonstration design will automatically load and drive the LED array in an alternating pattern.

To load the FTDI chip USB hardware drivers via the stand-alone package:

1. Browse to www.latticesemi.com/breakoutboards and download the FTDI Chip USB Hardware Drivers package.
2. Extract the FTDI chip USB Hardware driver package to your PC hard drive.
3. Connect the USB cable from a USB port on your PC to the board's USB mini-B socket (J7). After the connection is made, a green Power LED (D9) will light indicating the board is powered on.
4. If you are prompted, "Windows may connect to Windows Update" select **No, not this time** from available options and click **Next** to proceed with the installation. Choose the **Install from specific location (Advanced)** option and click **Next**.
5. Search for the best driver in these locations and click the **Browse** button to browse to the Windows driver folder created in the Download Windows USB Hardware Drivers section. Select the **CDM 2.04.06 WHQL Certified** folder and click **OK**.
6. Click **Next**. A screen will display as Windows copies the required driver files. Windows will display a message indicating that the installation was successful.
7. Click **Finish** to install the USB driver.
8. The demonstration design will automatically load and drive the LED array in an alternating pattern.

See the Troubleshooting section of this guide if the board does not function as expected.

Download Demo Designs

The counter demo is preprogrammed into the Breakout Board, however over time it is likely your board will be modified. Lattice distributes source and programming files for demonstration designs compatible with the Breakout Board.

To download demo designs:

1. Browse to the Lattice Breakout Board Evaluation Kits web page (www.latticesemi.com/breakoutboards) of the Lattice web site. Select **MachXO2 Breakout Board Demo Source** and save the file.
2. Extract the contents of **MachXO2_1200ZE_BB_Eval_Kit_v01.0.zip** to an accessible location on your hard drive.

The demo design directory **Demo_LED** is unpacked with all design files needed for the demo, including the JEDEC programming data file.

Continue to Programming a Demo Design with Lattice Diamond Design Software.

Programming a Demo Design with the Lattice Diamond Programmer

The demonstration design is pre-programmed into the MachXO2-1200ZE Breakout Board by Lattice. If you have changed the design but now want to restore the Breakout Board to factory settings, use the procedure described below.

To program the MachXO2-1200ZE device:

1. Install, license and run Lattice Diamond software. See www.latticesemi.com/latticediamond for download and licensing information.
2. Connect the USB cable to the host PC and the MachXO2-1200ZE Breakout Board.
3. From Diamond, open the **Demo_LED_OSC.ldf** project file.

4. Click the **Programmer** icon.
5. Click **Detect Cable**. The Programmer will detect the cable (Cable: USB2, Port: FTUSB-0).
6. Click the **Program** icon. When complete, **PASS** is displayed in the Status column.

MachXO2-1200ZE Breakout Board

This section describes the features of the MachXO2-1200ZE Breakout Board in detail.

Overview

The Breakout Board is a complete development platform for the MachXO2-1200ZE PLD. The board includes a prototyping area, a USB program/power port, an LED array, and header landings with electrical connections to most of the PLD's programmable I/O, power, and JTAG pins. The board is powered by the PC's USB port or optionally with external power. You may create or modify the program files and reprogram the board using Lattice Diamond software.

Figure 3. MachXO2-1200ZE Breakout Board Block Diagram

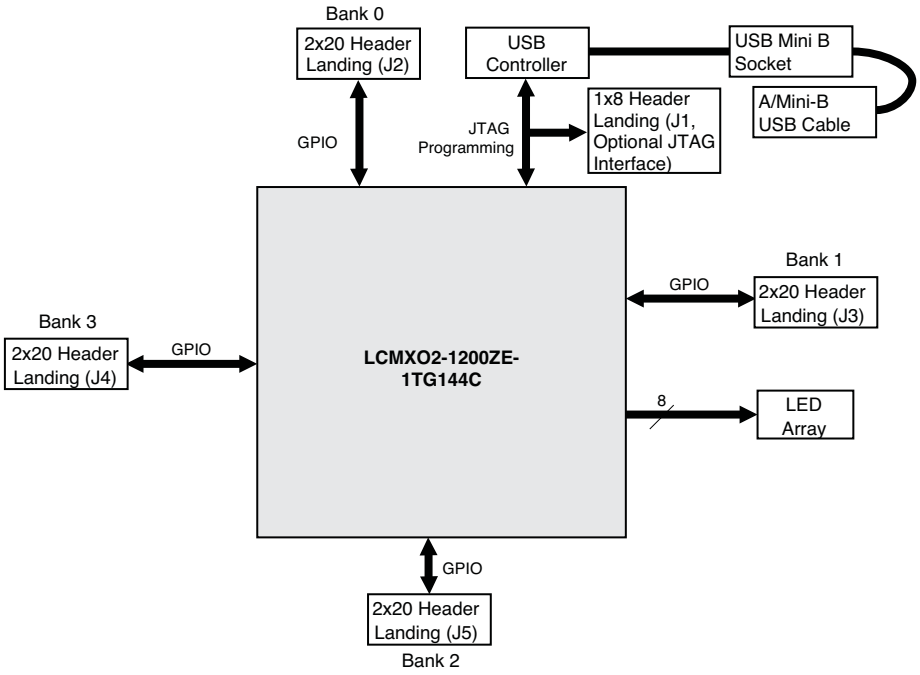


Table 1 describes the components on the board and the interfaces it supports.

Table 1. Breakout Board Components and Interfaces

Component/Interface	Type	Schematic Reference	Description
Circuits			
USB Controller	Circuit	U2: FT2232H	USB-to-JTAG interface and dual USB UART/FIFO IC
USB Mini-B Socket	I/O	J7:USB_MINI_B	Programming and debug interface
Components			
LCMXO2-1200ZE	PLD	U3: LCMXO2-1200ZE-1TG144C	1200-LUT device packaged in a 20 x 20mm, 144-pin TQFP
Interfaces			
LED Array	Output	D8-D1	Red LEDs
Four 2x20 Header Landings	I/O	J2: header_2x20 J3: header_2x20 J4: header_2x20 J5: header_2x20	User-definable I/O
1x8 Header Landing	I/O	J1: header_1x8	Optional JTAG interface
4x15 60-Hole Prototype Area			Prototype area 100mil centered holes.
Test points	Power	TP1: +3.3V TP2: +1.2V TP3: GND	Power and ground reference points

Subsystems

This section describes the principle sub systems for the Breakout Board in alphabetical order.

Clock Sources

All clocks for the counter demonstration designs originate from the MachXO2-1200ZE PLD on-chip oscillator. You may use an expansion header landing to drive a PLD input with an external clock source.

Expansion Header Landings

The expansion header landings provide access to user GPIOs, primary inputs, clocks, and VCCO pins of the MachXO2-1200ZE. The remaining pins serve as power supplies for external connections. Each landing is configured as one 2x20 100 mil.

Table 2. Expansion Connector Reference

Item	Description
Reference Designators	J2, J3, J4, J5
Part Number	header_2x20

Table 3. Expansion Header Pin Information (J2)

Pin Number	Function	MachXO2-1200ZE Pin
1	NC	N/A
2	VCCIO0	118, 123, 135
3	PT17D/DONE	109
4	PT17C/INITn	110
5	PT17B	111
6	PT17A	112
7	GND	N/A
8	GND	N/A
9	PT16D	113
10	PT16C	114
11	PT16B	115
12	PT16A	117
13	PT15D/PROGn	119
14	PT15C/JTAGen	120
15	GND	N/A
16	GND	N/A
17	PT15B	121
18	PT15A	122
19	PT12D/SDA/PCLKC0_0	125
20	PT12C/SCL/PCLKT0_0	126
21	PT12B/PCLKC0_1	127
22	PT12A/PCLKT0_1	128
23	GND	N/A
24	GND	N/A
25	PT11D/TMS	130
26	PT11C/TCK/TEST_CLK	131
27	PT11B	132
28	PT11A	133
29	PT10D/TDI	136
30	PT10C/TDO	137
31	GND	N/A
32	GND	N/A
33	PT10B	138
34	PT10A	139
35	PT9D	140
36	PT9C	141
37	PT9B	142
38	PT9A	143
39	GND	N/A
40	GND	N/A

Table 4. Expansion Header Pin Information (J3)

Pin Number	Function	MachXO2-1200ZE Pin
1	VCC_1.2V	36, 72, 108, 144
2	VCCIO1	79, 88, 102
3	VCC_1.2V	36, 72, 108, 144
4	NC	N/A
5	PR10C	74
6	PR10D	73
7	PR10A	76
8	PR10B	75
9	GND	N/A
10	GND	N/A
11	PR9C	78
12	PR9D	77
13	PR9A	82
14	PR9B	81
15	GND	N/A
16	GND	N/A
17	PR8C	84
18	PR8D	83
19	PR8A	86
20	PR8B	85
21	GND	N/A
22	GND	N/A
23	PCLKT1_0/PR5C	92
24	PCLKC1_0/PR5D	91
25	PR5A	94
26	PR5B	93
27	GND	N/A
28	GND	N/A
29	PR4C	96
30	PR4D	95
31	PR4A	98
32	PR4B	97
33	GND	N/A
34	GND	N/A
35	PR3A	100
36	PR3B	99
37	PR2C	105
38	PR2D	104
39	PR2A	107
40	PR2B	106

Table 5. Expansion Header Pin Information (J4)

Pin Number	Function	MachXO2-1200ZE Pin
1	VCC_3.3V	N/A
2	VCCIO3	7, 16, 30
3	VCC_3.3V	N/A
4	NC	N/A
5	PL2A/L_GPLLT_FB	1
6	PL2B/L_GPPLC_FB	2
7	PL2C/L_GPLLT_IN	3
8	PL2D/L_GPLLC_IN	4
9	PL3A/PCLKT3_2	5
10	PL3B/PCLKC3_2	6
11	PL3C	9
12	PL3D	10
13	GND	N/A
14	GND	N/A
15	PL4A	11
16	PL4B	12
17	PL4C	13
18	PL4D	14
19	GND	N/A
20	GND	N/A
21	PL5A/PCLKT3_1	19
22	PL5B/PCLKC3_1	20
23	PL5C	21
24	PL5D	22
25	GND	N/A
26	GND	N/A
27	PL8A	23
28	PL8B	24
29	PL8C	25
30	PL8D	26
31	GND	N/A
32	GND	N/A
33	PL9A/PCLKT3_0	27
34	PL9B/PCLKC3_0	28
35	GND	N/A
36	GND	N/A
37	PL10A	32
38	PL10B	33
39	PL10C	34
40	PL10D	35

Table 6. Expansion Header Pin Information (J5)

Pin Number	Function	MachXO2-1200ZE Pin
1	NC	N/A
2	VCCIO2	37, 51, 66
3	SI/SISPI/PB20D	71
4	PB20B	69
5	SN/PB20C	70
6	PB20A	68
7	PB18D	67
8	PB18B	62
9	PB18C	65
10	PB18A	61
11	GND	N/A
12	GND	N/A
13	PB15D	60
14	PB15B	58
15	PB15C	59
16	PB15A	57
17	GND	N/A
18	GND	N/A
19	PCLKC2_1/PB11B	56
20	PB11D	54
21	PCLKT2_1/PB11A	55
22	PB11C	52
23	GND	N/A
24	GND	N/A
25	PCLKC2_0/PB9B	50
26	PB9D	48
27	PCLKT2_0/PB9A	49
28	PB9C	47
29	GND	N/A
30	GND	N/A
31	S0/SPISO/PB6D	45
32	PB6B	43
33	MCLK/CLK/PB6C	44
34	PB6A	42
35	GND	N/A
36	GND	N/A
37	PB4D	41
38	PB4B	39
39	CSSPIN/PB4C	40
40	PB4A	38

Figure 4. J2/J4 Header Landing Callout

J2	
1	2
NC	IO0
109	110
111	112
GND	GND
113	114
115	117
119	120
GND	GND
121	122
125	126
127	128
GND	GND
130	131
132	133
136	137
GND	GND
138	139
140	141
142	143
GND	GND

J4	
1	2
3.3	IO3
3.3	NC
1	2
3	4
5	6
9	10
GND	GND
11	12
13	14
GND	GND
19	20
21	22
GND	GND
23	24
25	26
GND	GND
27	28
GND	GND
32	33
34	35

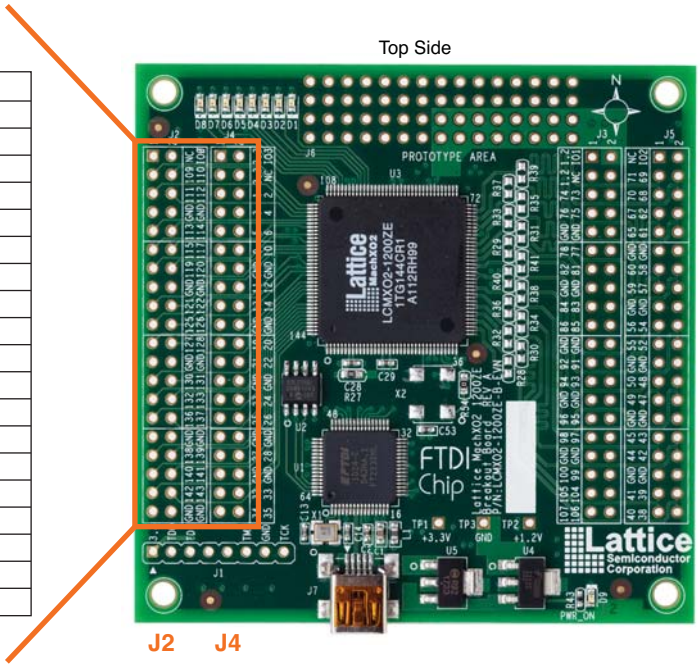
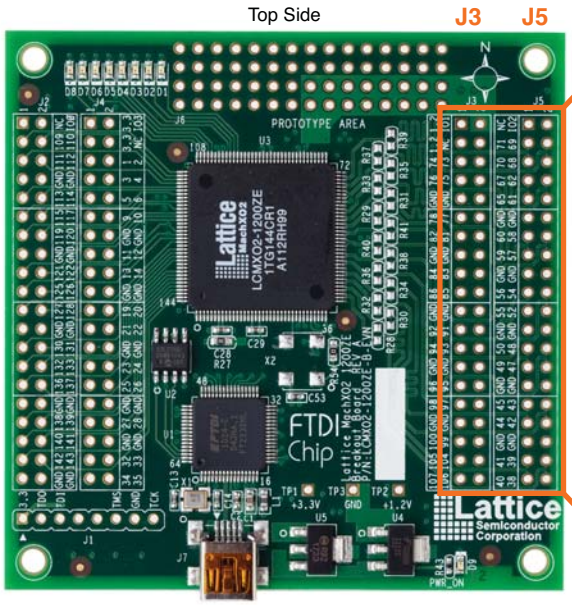


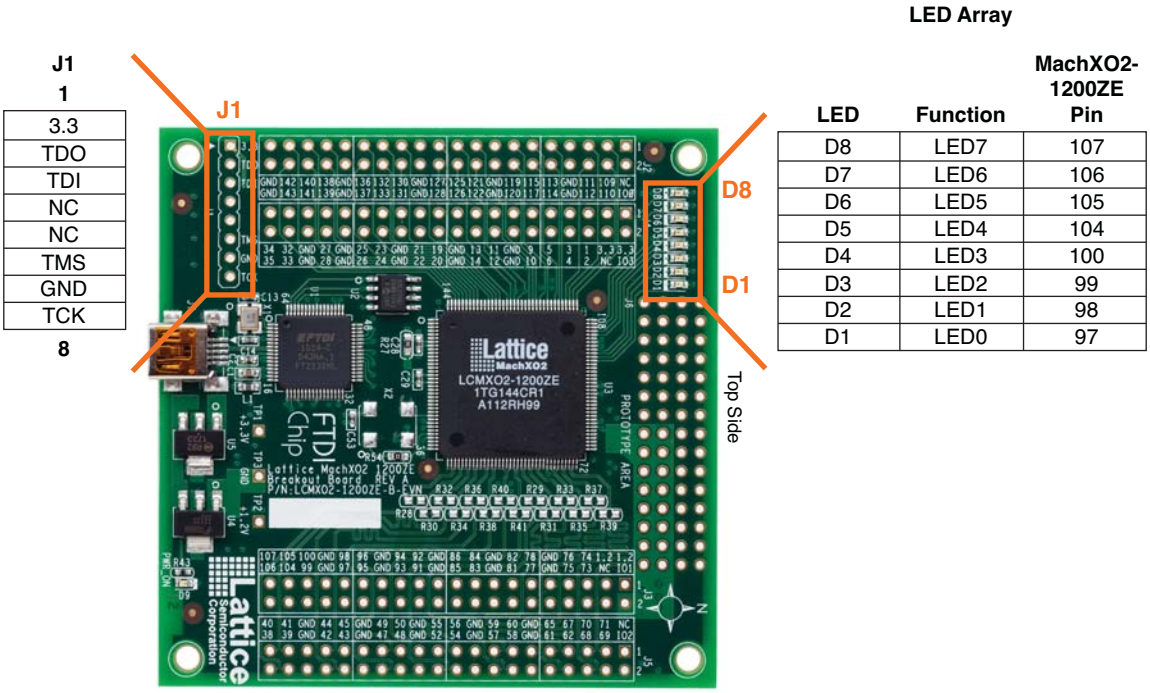
Figure 5. J3/J5 Header Landing Callout



J3	
1	2
1.2	IO1
1.2	NC
74	73
76	75
GND	GND
78	77
82	81
GND	GND
84	83
86	85
GND	GND
92	91
94	93
GND	GND
96	95
98	97
GND	GND
100	99
105	104
107	106

J5	
1	2
NC	IO2
71	69
70	68
67	62
65	61
GND	GND
60	58
59	57
GND	GND
56	54
55	52
GND	GND
50	48
49	47
GND	GND
45	43
44	42
GND	GND
41	39
40	38

Figure 6. J1 Header Landing and LED Array Callout



MachXO2-1200ZE PLD

The MachXO2-1200ZE-1TG144C is a 144-pin TQFP package PLD device which provides 108 I/Os in a 20 x 20mm package.

Table 7. MachXO PLD Interface Reference

Item	Description
Reference Designators	U3
Part Number	LCMXO2-1200ZE-1TG144C
Manufacturer	Lattice Semiconductor
Web Site	www.latticesemi.com

JTAG Interface Circuits

For power and programming an FTDI USB UART/FIFO IC converter provides a communication interface between a PC host and the JTAG programming chain of the Breakout Board. The USB 5V supply is also used as a source for the 3.3V supply rail. A USB mini-B socket is provided for the USB connector cable.

Table 8. JTAG Interface Reference

Item	Description
Reference Designators	U1
Part Number	FT2232HL
Manufacturer	Future Technology Devices International (FTDI)
Web Site	www.ftdichip.com

Table 9. JTAG Programming Pin Information

Description	MachXO2-1200ZE Pin
Test Data Output	137:TDO
Test Data Input	136:TDI
Test Mode Select	130:TMS
Test Clock	131:TCK

LEDs

A green LED (D9) is used to indicate USB 5V power. Eight red LEDs are driven by I/O pins of the MachXO2-1200ZE device.

Table 10. Power and User LEDs Reference

Item	Description
Reference Designators	D1, D2, D3, D4, D5, D6, D7, D8, D9
Part Number	LTST-C190KRKT (D1-D8) LTST-C190KGKT (D9)
Manufacturer	Lite-On It Corporation
Web Site	www.liteonit.com

Power Supply

3.3V and 1.2V power supply rails are converted from the USB 5V interface when the board is connected to a host PC.

Test Points

In order to check the various voltage levels used, test points are provided:

- TP1: +3.3V
- TP2: +1.2V
- TP3: GND

USB Programming and Debug Interface

The USB mini-B socket of the Breakout Board serves as the programming and debug interface.

JTAG Programming: For JTAG programming, a preprogrammed USB PHY peripheral controller is provided on the Breakout Board to serve as the programming interface to the MachXO2-1200ZE PLD.

Programming requires the Lattice Diamond or ispVM System software.

Table 11. USB Interface Reference

Item	Description
Reference Designators	U1
Part Number	FT2232HL
Manufacturer	Future Technology Devices International (FTDI)
Web Site	www.ftdichip.com

Board Modifications

This section describes modifications to the board to change or add functionality.

Bypassing the USB Programming Interface

The USB programming interface circuit (USB Programming and Debug Interface section) may be optionally bypassed by removing the 0 ohm resistors: R5, R6, R7, R8 (See Appendix A. Schematics, Sheet 2 of 5). Header landing J1 provides JTAG signal access for jumper wires or a 1x8 pin header.

Applying External Power

The Breakout Board is powered by the circuit of Schematic Sheet 5 of 5 based on the 5V USB power source. You may disconnect this power source by removing the 0 ohm resistors: R42 (VCC_1.2V) and R44 (VCC_3.3V). Power connections are available from the expansion header landings, J3 (+1.2V, pins 1 and 3, schematic sheet 3 of 5) and J4 (+3.3V, pins 1 and 3, schematic sheet 4 of 5).

Measuring Bank and Core Power

In addition to the expansion headers, test points (TP1, TP2) provide access to power supplies of the MachXO2-1200ZE PLD. Inline 1 ohm resistors: R24 (VCCIO0, +3.3V, Bank 0), R25 (VCCIO1, +3.3V, Bank 1), R26 (VCCIO2, +3.3V, Bank 2), R27 (VCCIO3, +3.3V, Bank 3), R56 (VCC core, +1.2V) can be used to measure current for the power supplies.

Mechanical Specifications

Dimensions: 3 in. [L] x 3 in. [W] x 1/2 in. [H]

Environmental Requirements

The evaluation board must be stored between -40° C and 100° C. The recommended operating temperature is between 0° C and 90° C.

The board can be damaged without proper anti-static handling.

Glossary

PLD: Programmable Logic Device

DIP: Dual in-line package

LED: Light Emitting Diode.

LUT: Look Up Table

PCB: Printed Circuit Board

RoHS: Restriction of Hazardous Substances Directive

USB: Universal Serial Bus

WDT: Watchdog Timer

Troubleshooting

Use the tips in this section to diagnose problems with the Breakout Board.

LEDs Do Not Flash

If power is applied but the board does not flash according to the preprogrammed counter demonstration then it is likely the board has been reprogrammed with a new design. Follow the directions in the Demonstration Design section to restore the factory default.

USB Cable Not Detected

If Lattice Diamond Programmer or ispVM System does not recognize the USB cable after installing the Lattice USB port drivers and rebooting, the incorrect USB driver may have been installed. This usually occurs if you attach the board to your PC prior to installing the Lattice-supplied USB driver.

To access the *Troubleshooting the USB Driver Installation Guide*:

For Diamond software and standalone Diamond Programmer:

1. Start Diamond or Diamond Programmer and choose **Help**.
2. Search for **USB driver** or **Troubleshooting**, then select the **Troubleshooting the USB Driver** topic.
3. Follow the directions to install the Lattice USB driver.

For ispVM:

1. Start ispVM System and choose **Options > Cable and I/O Port Setup**.
   The Cable and I/O Port Setup Dialog appears.
2. Click the **Troubleshooting the USB Driver Installation Guide** link.
   The *Troubleshooting the USB Driver Installation Guide* document appears in your system's PDF file reader.
3. Follow the directions to install the Lattice USB driver.

Determine the Source of a Pre-Programmed Device

If the Breakout Board has been reprogrammed, the original demo design can be restored. To restore the board to the factory default, see the Download Demo Designs section for details on downloading and reprogramming the device.

Ordering Information

Description	Ordering Part Number	China RoHS Environment-Friendly Use Period (EFUP)
MachXO2-1200ZE Breakout Board Evaluation Kit	LCMXO2-1200ZE-B-EVN	

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

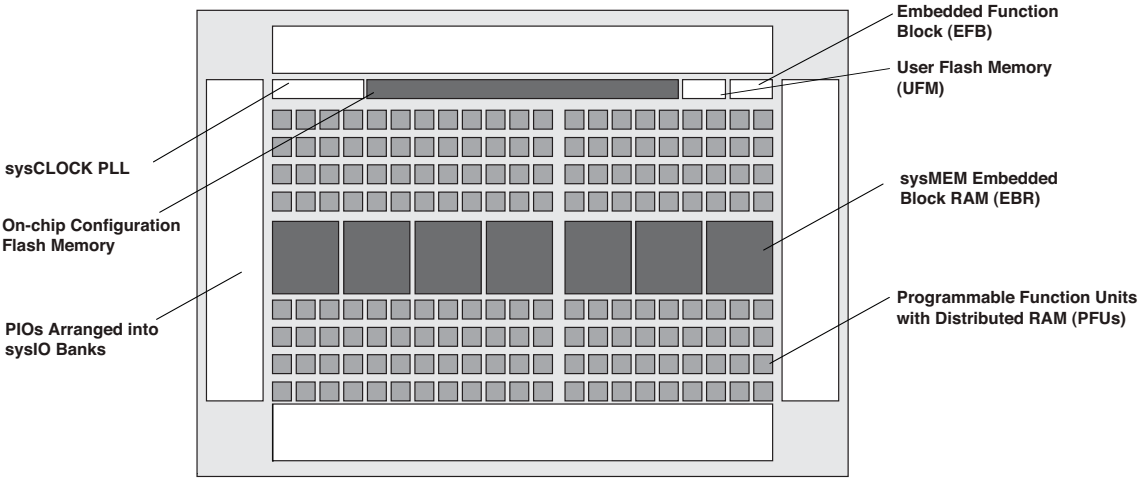
Revision History

Date	Version	Change Summary
December 2011	01.0	Initial release.
January 2012	01.1	Figure "MachXO2-1200ZE Breakout Board, Top Side" updated with revision B board photo.

Architecture Overview

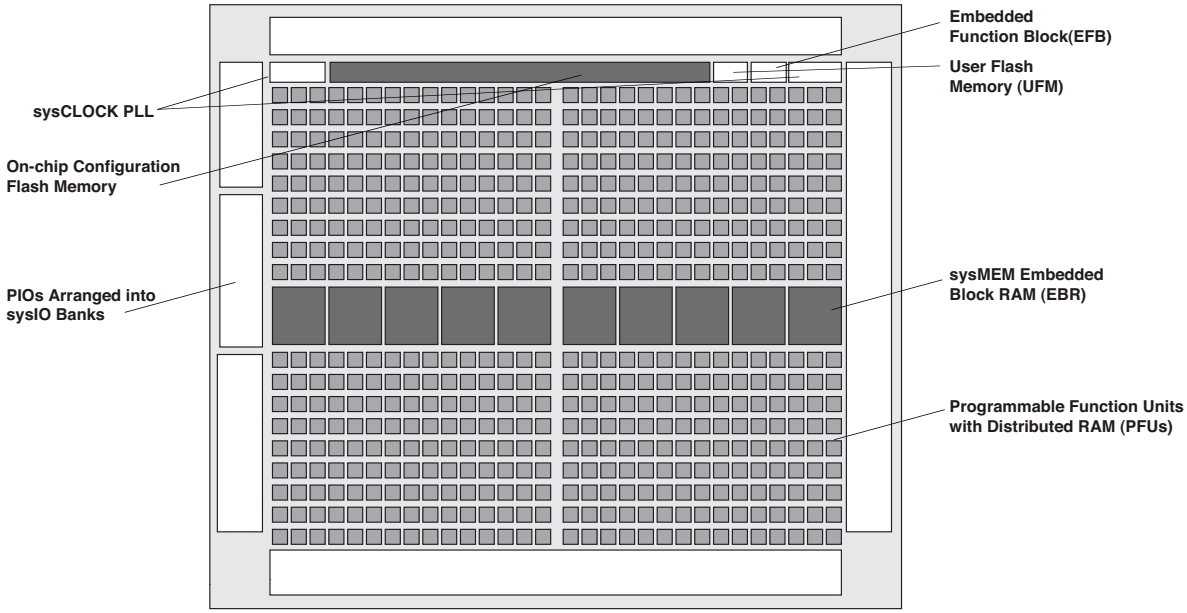
The MachXO2 family architecture contains an array of logic blocks surrounded by Programmable I/O (PIO). The larger logic density devices in this family have sysCLOCK™ PLLs and blocks of sysMEM Embedded Block RAM (EBRs). Figure 2-1 and Figure 2-2 show the block diagrams of the various family members.

Figure 2-1. Top View of the MachXO2-1200 Device



Note: MachXO2-256, and MachXO2-640/U are similar to MachXO2-1200. MachXO2-256 has a lower LUT count and no PLL or EBR blocks. MachXO2-640 has no PLL, a lower LUT count and two EBR blocks. MachXO2-640U has a lower LUT count, one PLL and seven EBR blocks.

Figure 2-2. Top View of the MachXO2-4000 Device



Note: MachXO2-1200U, MachXO2-2000/U and MachXO2-7000 are similar to MachXO2-4000. MachXO2-1200U and MachXO2-2000 have a lower LUT count, one PLL, and eight EBR blocks. MachXO2-2000U has a lower LUT count, two PLLs, and 10 EBR blocks. MachXO2-7000 has a higher LUT count, two PLLs, and 26 EBR blocks.

The logic blocks, Programmable Functional Unit (PFU) and sysMEM EBR blocks, are arranged in a two-dimensional grid with rows and columns. Each row has either the logic blocks or the EBR blocks. The PIO cells are located at the periphery of the device, arranged in banks. The PFU contains the building blocks for logic, arithmetic, RAM, ROM, and register functions. The PIOs utilize a flexible I/O buffer referred to as a sysIO buffer that supports operation with a variety of interface standards. The blocks are connected with many vertical and horizontal routing channel resources. The place and route software tool automatically allocates these routing resources.

In the MachXO2 family, the number of sysIO banks varies by device. There are different types of I/O buffers on the different banks. Refer to the details in later sections of this document. The sysMEM EBRs are large, dedicated fast memory blocks; these blocks are found in MachXO2-640/U and larger devices. These blocks can be configured as RAM, ROM or FIFO. FIFO support includes dedicated FIFO pointer and flag “hard” control logic to minimize LUT usage.

The MachXO2 registers in PFU and sysI/O can be configured to be SET or RESET. After power up and device is configured, the device enters into user mode with these registers SET/RESET according to the configuration setting, allowing device entering to a known state for predictable system function.

The MachXO2 architecture also provides up to two sysCLOCK Phase Locked Loop (PLL) blocks on MachXO2-640U, MachXO2-1200/U and larger devices. These blocks are located at the ends of the on-chip Flash block. The PLLs have multiply, divide, and phase shifting capabilities that are used to manage the frequency and phase relationships of the clocks.

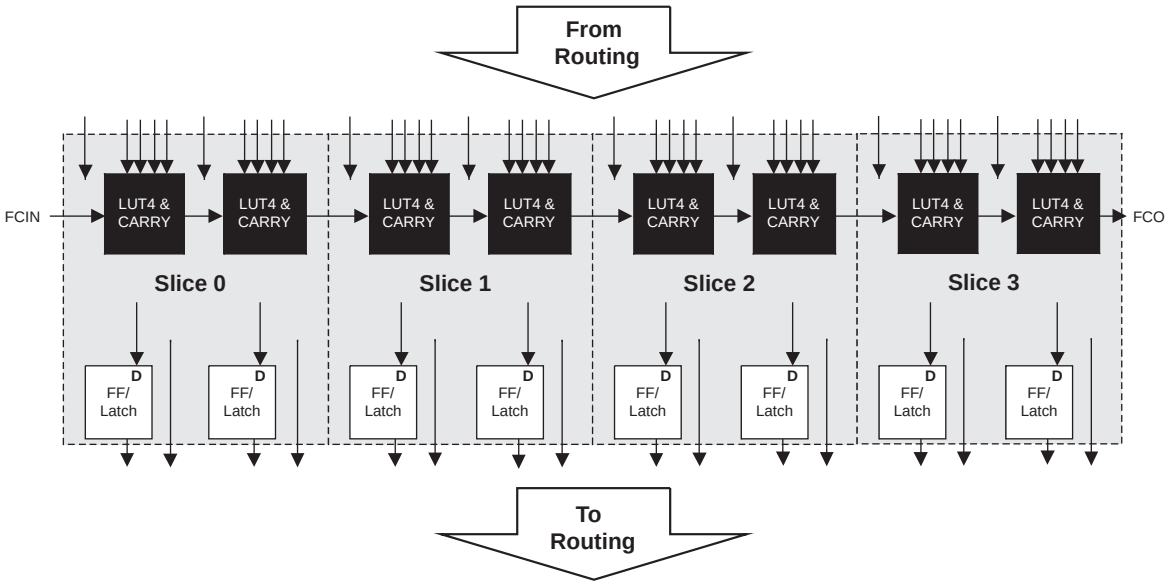
MachXO2 devices provide commonly used hardened functions such as SPI controller, I^2C controller and timer/counter. MachXO2-640/U and higher density devices also provide User Flash Memory (UFM). These hardened functions and the UFM interface to the core logic and routing through a WISHBONE interface. The UFM can also be accessed through the SPI, I^2C and JTAG ports.

Every device in the family has a JTAG port that supports programming and configuration of the device as well as access to the user logic. The MachXO2 devices are available for operation from 3.3 V, 2.5 V and 1.2 V power supplies, providing easy integration into the overall system.

PFU Blocks

The core of the MachXO2 device consists of PFU blocks, which can be programmed to perform logic, arithmetic, distributed RAM and distributed ROM functions. Each PFU block consists of four interconnected slices numbered 0 to 3 as shown in Figure 2-3. Each slice contains two LUTs and two registers. There are 53 inputs and 25 outputs associated with each PFU block.

Figure 2-3. PFU Block Diagram



Slices

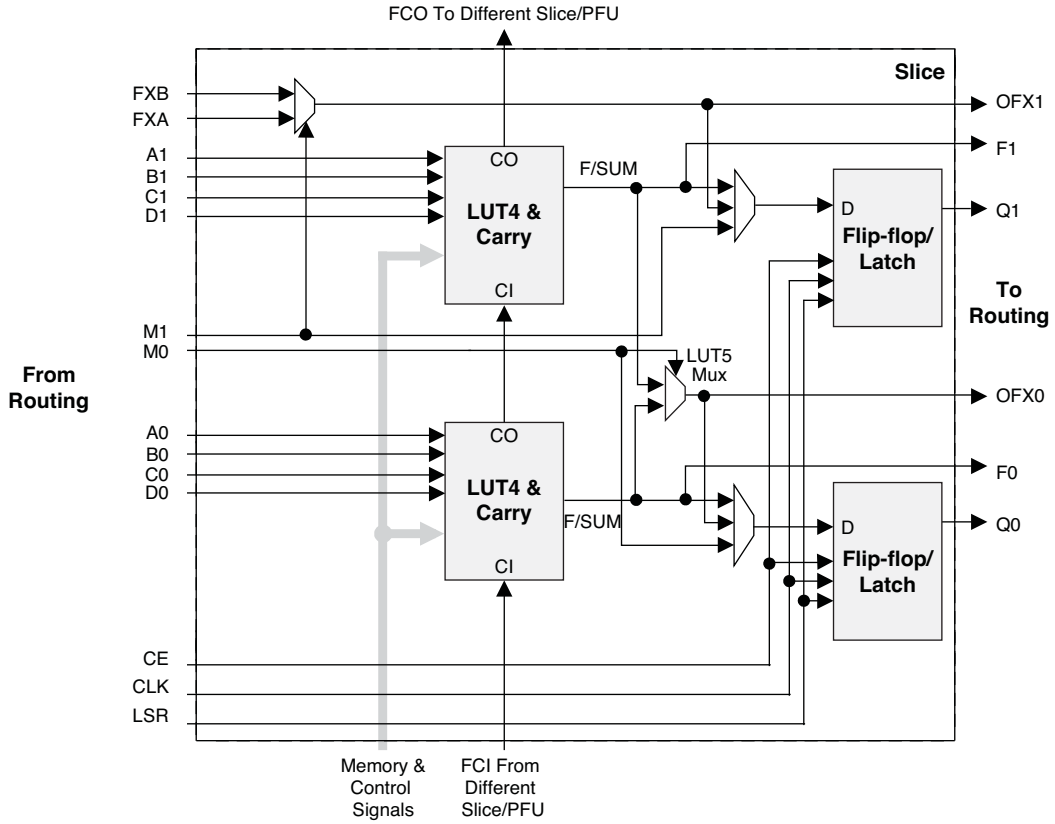
Slices 0-3 contain two LUT4s feeding two registers. Slices 0-2 can be configured as distributed memory. Table 2-1 shows the capability of the slices in PFU blocks along with the operation modes they enable. In addition, each PFU contains logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. The control logic performs set/reset functions (programmable as synchronous/ asynchronous), clock select, chip-select and wider RAM/ROM functions.

Table 2-1. Resources and Modes Available per Slice

Slice	PFU Block	
	Resources	Modes
Slice 0	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 1	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 2	2 LUT4s and 2 Registers	Logic, Ripple, RAM, ROM
Slice 3	2 LUT4s and 2 Registers	Logic, Ripple, ROM

Figure 2-4 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge triggered or level sensitive clocks. All slices have 15 inputs from routing and one from the carry-chain (from the adjacent slice or PFU). There are seven outputs: six for routing and one to carry-chain (to the adjacent PFU). Table 2-2 lists the signals associated with Slices 0-3.

Figure 2-4. Slice Diagram



For Slices 0 and 1, memory control signals are generated from Slice 2 as follows:

- WCK is CLK
- WRE is from LSR
- DI[3:2] for Slice 1 and DI[1:0] for Slice 0 data from Slice 2
- WAD [A:D] is a 4-bit address from slice 2 LUT input

Table 2-2. Slice Signal Descriptions

Function	Type	Signal Names	Description
Input	Data signal	A0, B0, C0, D0	Inputs to LUT4
Input	Data signal	A1, B1, C1, D1	Inputs to LUT4
Input	Multi-purpose	M0/M1	Multi-purpose input
Input	Control signal	CE	Clock enable
Input	Control signal	LSR	Local set/reset
Input	Control signal	CLK	System clock
Input	Inter-PFU signal	FCIN	Fast carry in[1]
Output	Data signals	F0, F1	LUT4 output register bypass signals
Output	Data signals	Q0, Q1	Register outputs
Output	Data signals	OFX0	Output of a LUT5 MUX
Output	Data signals	OFX1	Output of a LUT6, LUT7, LUT8[2] MUX depending on the slice
Output	Inter-PFU signal	FCO	Fast carry out[1]

1. See Figure 2-3 for connection details.
2. Requires two PFUs.

Modes of Operation

Each slice has up to four potential modes of operation: Logic, Ripple, RAM and ROM.

Logic Mode
In this mode, the LUTs in each slice are configured as 4-input combinatorial lookup tables. A LUT4 can have 16 possible input combinations. Any four input logic functions can be generated by programming this lookup table. Since there are two LUT4s per slice, a LUT5 can be constructed within one slice. Larger look-up tables such as LUT6, LUT7 and LUT8 can be constructed by concatenating other slices. Note LUT8 requires more than four slices.

Ripple Mode
Ripple mode supports the efficient implementation of small arithmetic functions. In Ripple mode, the following functions can be implemented by each slice:

- Addition 2-bit
- Subtraction 2-bit
- Add/subtract 2-bit using dynamic control
- Up counter 2-bit
- Down counter 2-bit
- Up/down counter with asynchronous clear
- Up/down counter with preload (sync)
- Ripple mode multiplier building block
- Multiplier support
- Comparator functions of A and B inputs
  — A greater-than-or-equal-to B
  — A not-equal-to B
  — A less-than-or-equal-to B

Ripple mode includes an optional configuration that performs arithmetic using fast carry chain methods. In this configuration (also referred to as CCU2 mode) two additional signals, Carry Generate and Carry Propagate, are generated on a per-slice basis to allow fast arithmetic functions to be constructed by concatenating slices.

RAM Mode

In this mode, a 16x4-bit distributed single port RAM (SPR) can be constructed by using each LUT block in Slice 0 and Slice 1 as a 16x1-bit memory. Slice 2 is used to provide memory address and control signals.

MachXO2 devices support distributed memory initialization.

The Lattice design tools support the creation of a variety of different size memories. Where appropriate, the software will construct these using distributed memory primitives that represent the capabilities of the PFU. Table 2-3 shows the number of slices required to implement different distributed RAM primitives. For more information about using RAM in MachXO2 devices, please see TN1201, Memory Usage Guide for MachXO2 Devices.

Table 2-3. Number of Slices Required For Implementing Distributed RAM

	SPR 16x4	PDPR 16x4
Number of slices	3	3

Note: SPR = Single Port RAM, PDPR = Pseudo Dual Port RAM

ROM Mode

ROM mode uses the LUT logic; hence, slices 0-3 can be used in ROM mode. Preloading is accomplished through the programming interface during PFU configuration.

For more information on the RAM and ROM modes, please refer to TN1201, Memory Usage Guide for MachXO2 Devices.

Routing

There are many resources provided in the MachXO2 devices to route signals individually or as buses with related control signals. The routing resources consist of switching circuitry, buffers and metal interconnect (routing) segments.

The inter-PFU connections are made with three different types of routing resources: x1 (spans two PFUs), x2 (spans three PFUs) and x6 (spans seven PFUs). The x1, x2, and x6 connections provide fast and efficient connections in the horizontal and vertical directions.

The design tools take the output of the synthesis tool and places and routes the design. Generally, the place and route tool is completely automatic, although an interactive routing editor is available to optimize the design.

Clock/Control Distribution Network

Each MachXO2 device has eight clock inputs (PCLK [T, C] [Banknum]_[2..0]) – three pins on the left side, two pins each on the bottom and top sides and one pin on the right side. These clock inputs drive the clock nets. These eight inputs can be differential or single-ended and may be used as general purpose I/O if they are not used to drive the clock nets. When using a single ended clock input, only the PCLKT input can drive the clock tree directly.

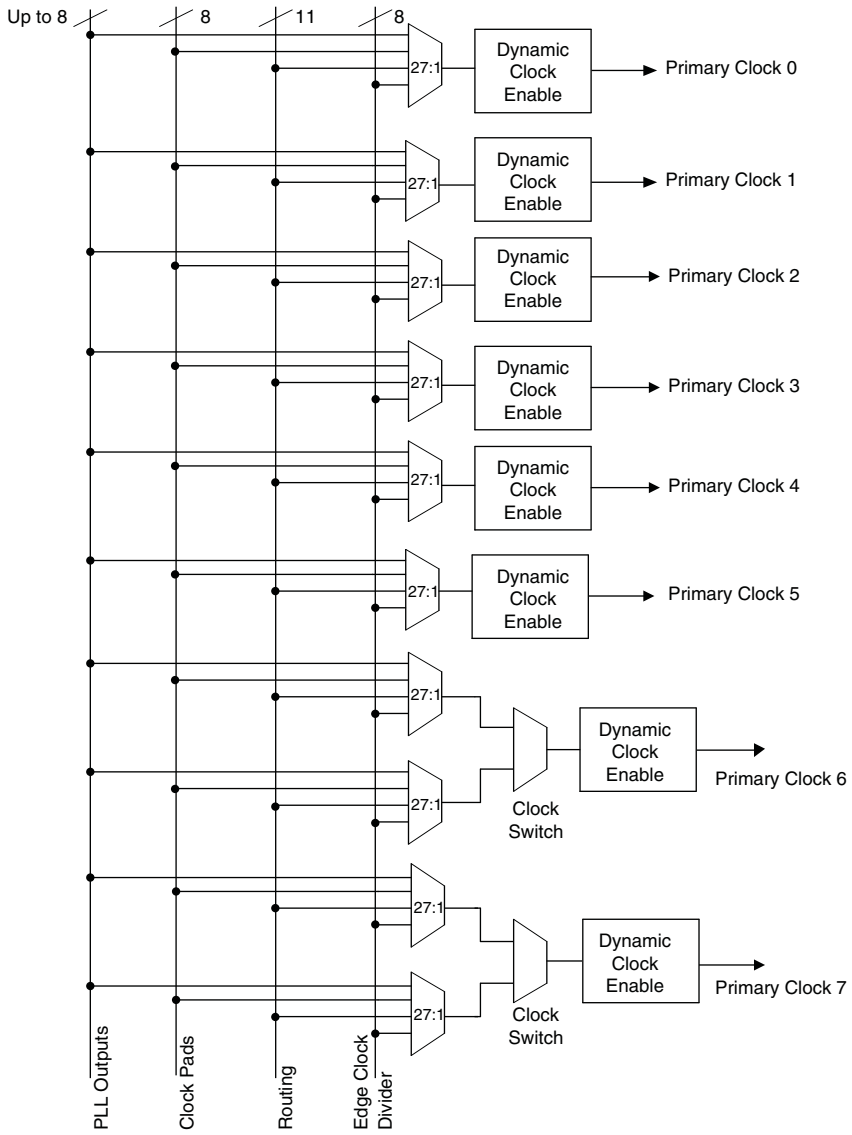
The MachXO2 architecture has three types of clocking resources: edge clocks, primary clocks and secondary high fanout nets. MachXO2-640U, MachXO2-1200/U and higher density devices have two edge clocks each on the top and bottom edges. Lower density devices have no edge clocks. Edge clocks are used to clock I/O registers and have low injection time and skew. Edge clock inputs are from PLL outputs, primary clock pads, edge clock bridge outputs and CIB sources.

The eight primary clock lines in the primary clock network drive throughout the entire device and can provide clocks for all resources within the device including PFUs, EBRs and PICs. In addition to the primary clock signals, MachXO2 devices also have eight secondary high fanout signals which can be used for global control signals, such as clock enables, synchronous or asynchronous clears, presets, output enables, etc. Internal logic can drive the global clock network for internally-generated global clocks and control signals.

The maximum frequency for the primary clock network is shown in the MachXO2 External Switching Characteristics table.

The primary clock signals for the MachXO2-256 and MachXO2-640 are generated from eight 17:1 muxes The available clock sources include eight I/O sources and 9 routing inputs. Primary clock signals for the MachXO2-640U, MachXO2-1200/U and larger devices are generated from eight 27:1 muxes The available clock sources include eight I/O sources, 11 routing inputs, eight clock divider inputs and up to eight sysCLOCK PLL outputs.

Figure 2-5. Primary Clocks for MachXO2 Devices

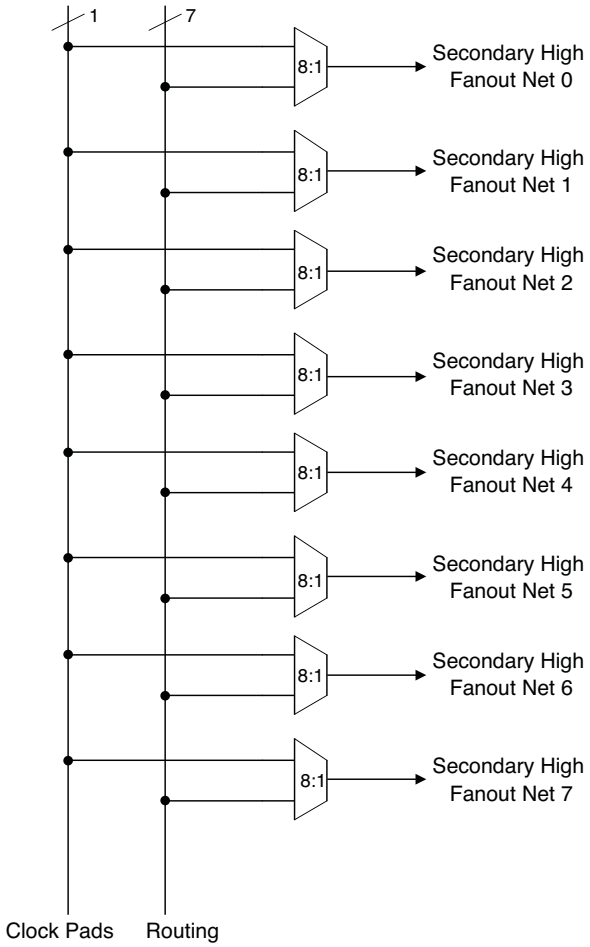


Primary clocks for MachXO2-640U, MachXO2-1200/U and larger devices.

Note: MachXO2-640 and smaller devices do not have inputs from the Edge Clock Divider or PLL and fewer routing inputs. These devices have 17:1 muxes instead of 27:1 muxes.

Eight secondary high fanout nets are generated from eight 8:1 muxes as shown in Figure 2-6. One of the eight inputs to the secondary high fanout net input mux comes from dual function clock pins and the remaining seven come from internal routing. The maximum frequency for the secondary clock network is shown in MachXO2 External Switching Characteristics table.

Figure 2-6. Secondary High Fanout Nets for MachXO2 Devices



sysCLOCK Phase Locked Loops (PLLs)

The sysCLOCK PLLs provide the ability to synthesize clock frequencies. The MachXO2-640U, MachXO2-1200/U and larger devices have one or more sysCLOCK PLL. CLKI is the reference frequency input to the PLL and its source can come from an external I/O pin or from internal routing. CLKFB is the feedback signal to the PLL which can come from internal routing or an external I/O pin. The feedback divider is used to multiply the reference frequency and thus synthesize a higher frequency clock output.

The MachXO2 sysCLOCK PLLs support high resolution (16-bit) fractional-N synthesis. Fractional-N frequency synthesis allows the user to generate an output clock which is a non-integer multiple of the input frequency. For more information about using the PLL with Fractional-N synthesis, please see TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide.

Each output has its own output divider, thus allowing the PLL to generate different frequencies for each output. The output dividers can have a value from 1 to 128. The output dividers may also be cascaded together to generate low frequency clocks. The CLKOP, CLKOS, CLKOS2, and CLKOS3 outputs can all be used to drive the MachXO2 clock distribution network directly or general purpose routing resources can be used.

The LOCK signal is asserted when the PLL determines it has achieved lock and de-asserted if a loss of lock is detected. A block diagram of the PLL is shown in Figure 2-7.

The setup and hold times of the device can be improved by programming a phase shift into the CLKOS, CLKOS2, and CLKOS3 output clocks which will advance or delay the output clock with reference to the CLKOP output clock.

This phase shift can be either programmed during configuration or can be adjusted dynamically. In dynamic mode, the PLL may lose lock after a phase adjustment on the output used as the feedback source and not relock until the t_{LOCK} parameter has been satisfied.

The MachXO2 also has a feature that allows the user to select between two different reference clock sources dynamically. This feature is implemented using the PLLREFCS primitive. The timing parameters for the PLL are shown in the sysCLOCK PLL Timing table.

The MachXO2 PLL contains a WISHBONE port feature that allows the PLL settings, including divider values, to be dynamically changed from the user logic. When using this feature the EFB block must also be instantiated in the design to allow access to the WISHBONE ports. Similar to the dynamic phase adjustment, when PLL settings are updated through the WISHBONE port the PLL may lose lock and not relock until the t_{LOCK} parameter has been satisfied. The timing parameters for the PLL are shown in the sysCLOCK PLL Timing table.

For more details on the PLL and the WISHBONE interface, see TN1199, MachXO2 sysCLOCK PLL Design and Usage Guide.

Figure 2-7. PLL Diagram

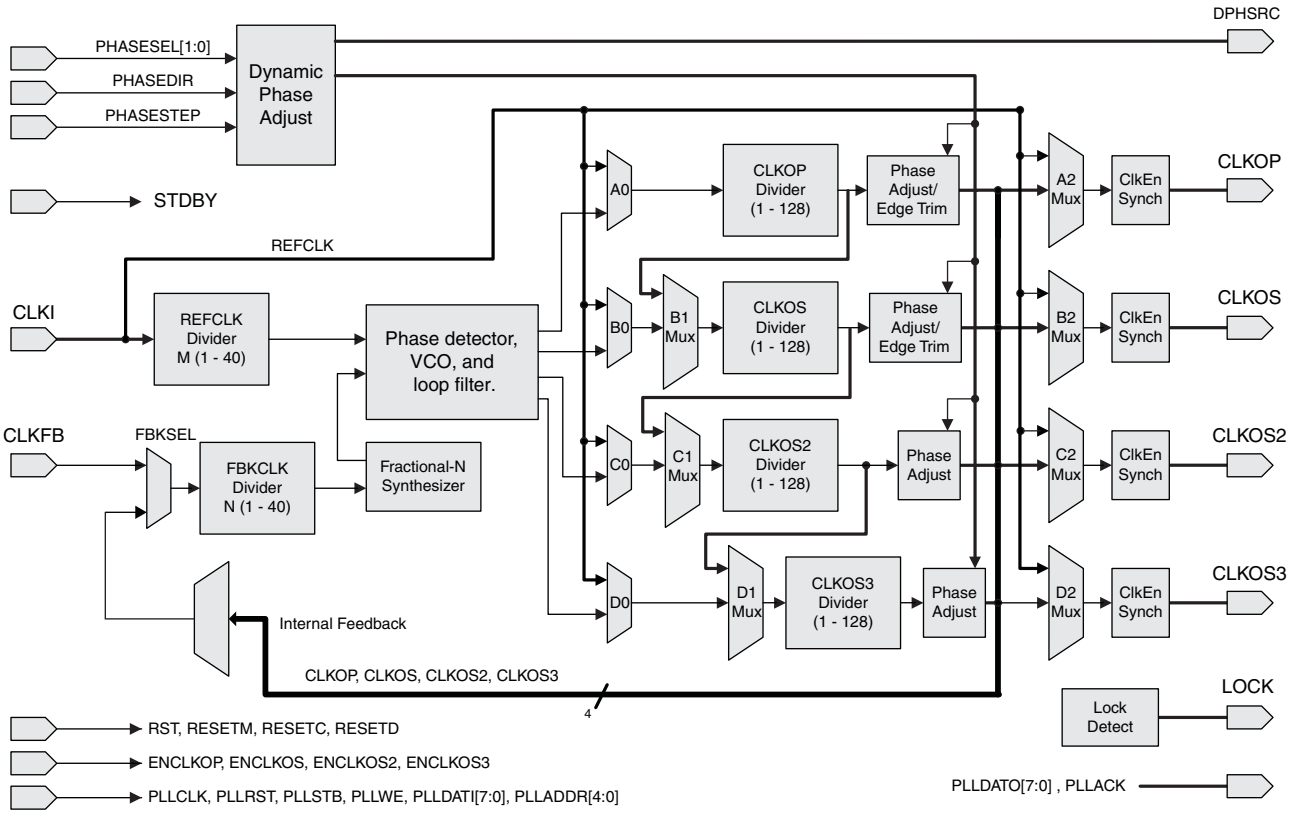


Table 2-4 provides signal descriptions of the PLL block.

Table 2-4. PLL Signal Descriptions

Port Name	I/O	Description
CLKI	I	Input clock to PLL
CLKFB	I	Feedback clock
PHASESEL[1:0]	I	Select which output is affected by Dynamic Phase adjustment ports
PHASEDIR	I	Dynamic Phase adjustment direction
PHASESTEP	I	Dynamic Phase step – toggle shifts VCO phase adjust by one step.

Table 2-4. PLL Signal Descriptions (Continued)

Port Name	I/O	Description
CLKOP	O	Primary PLL output clock (with phase shift adjustment)
CLKOS	O	Secondary PLL output clock (with phase shift adjust)
CLKOS2	O	Secondary PLL output clock2 (with phase shift adjust)
CLKOS3	O	Secondary PLL output clock3 (with phase shift adjust)
LOCK	O	PLL LOCK, asynchronous signal. Active high indicates PLL is locked to input and feedback signals.
DPHSRC	O	Dynamic Phase source – ports or WISHBONE is active
STDBY	I	Standby signal to power down the PLL
RST	I	PLL reset without resetting the M-divider. Active high reset.
RESETM	I	PLL reset - includes resetting the M-divider. Active high reset.
RESETC	I	Reset for CLKOS2 output divider only. Active high reset.
RESETD	I	Reset for CLKOS3 output divider only. Active high reset.
ENCLKOP	I	Enable PLL output CLKOP
ENCLKOS	I	Enable PLL output CLKOS when port is active
ENCLKOS2	I	Enable PLL output CLKOS2 when port is active
ENCLKOS3	I	Enable PLL output CLKOS3 when port is active
PLLCLK	I	PLL data bus clock input signal
PLLRST	I	PLL data bus reset. This resets only the data bus not any register values.
PLLSTB	I	PLL data bus strobe signal
PLLWE	I	PLL data bus write enable signal
PLLADDR [4:0]	I	PLL data bus address
PLLDATI [7:0]	I	PLL data bus data input
PLLDATO [7:0]	O	PLL data bus data output
PLLACK	O	PLL data bus acknowledge signal

sysMEM Embedded Block RAM Memory

The MachXO2-640/U and larger devices contain sysMEM Embedded Block RAMs (EBRs). The EBR consists of a 9-kbit RAM, with dedicated input and output registers. This memory can be used for a wide variety of purposes including data buffering, PROM for the soft processor and FIFO.

sysMEM Memory Block

The sysMEM block can implement single port, dual port, pseudo dual port, or FIFO memories. Each block can be used in a variety of depths and widths as shown in Table 2-5.

Table 2-5. sysMEM Block Configurations

Memory Mode	Configurations
Single Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
True Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
Pseudo Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18
FIFO	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18

Bus Size Matching

All of the multi-port memory modes support different widths on each of the ports. The RAM bits are mapped LSB word 0 to MSB word 0, LSB word 1 to MSB word 1, and so on. Although the word size and number of words for each port varies, this mapping scheme applies to each port.

RAM Initialization and ROM Operation

If desired, the contents of the RAM can be pre-loaded during device configuration. EBR initialization data can be loaded from the UFM. To maximize the number of UFM bits, initialize the EBRs used in your design to an all-zero pattern. Initializing to an all-zero pattern does not use up UFM bits. MachXO2 devices have been designed such that multiple EBRs share the same initialization memory space if they are initialized to the same pattern.

By preloading the RAM block during the chip configuration cycle and disabling the write controls, the sysMEM block can also be utilized as a ROM.

Memory Cascading

Larger and deeper blocks of RAM can be created using EBR sysMEM Blocks. Typically, the Lattice design tools cascade memory transparently, based on specific design inputs.

Single, Dual, Pseudo-Dual Port and FIFO Modes

Figure 2-8 shows the five basic memory configurations and their input/output names. In all the sysMEM RAM modes, the input data and addresses for the ports are registered at the input of the memory array. The output data of the memory is optionally registered at the memory array output.

Figure 2-8. sysMEM Memory Primitives

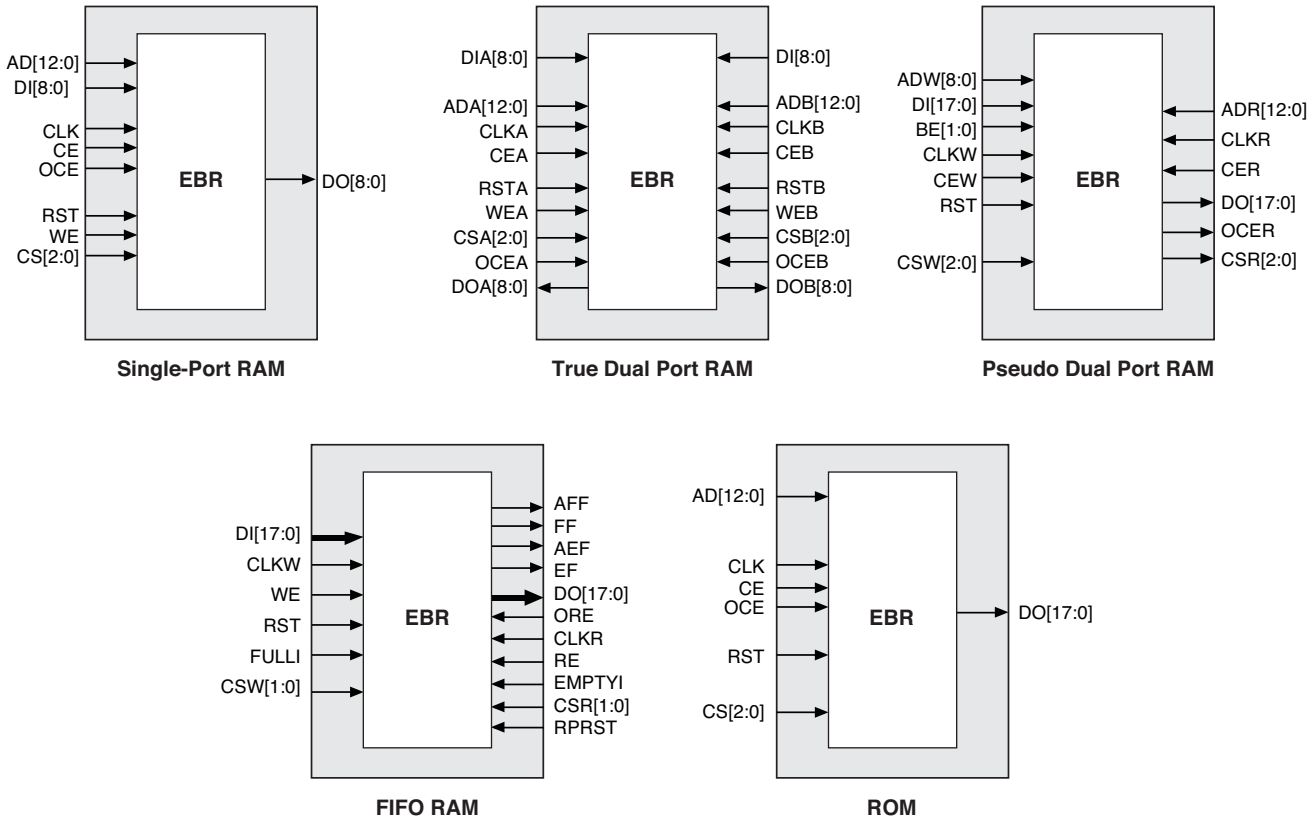


Table 2-6. EBR Signal Descriptions

Port Name	Description	Active State
CLK	Clock	Rising Clock Edge
CE	Clock Enable	Active High
OCE[1]	Output Clock Enable	Active High
RST	Reset	Active High
BE[1]	Byte Enable	Active High
WE	Write Enable	Active High
AD	Address Bus	—
DI	Data In	—
DO	Data Out	—
CS	Chip Select	Active High
AFF	FIFO RAM Almost Full Flag	—
FF	FIFO RAM Full Flag	—
AEF	FIFO RAM Almost Empty Flag	—
EF	FIFO RAM Empty Flag	—
RPRST	FIFO RAM Read Pointer Reset	—

1. Optional signals.
2. For dual port EBR primitives a trailing ‘A’ or ‘B’ in the signal name specifies the EBR port A or port B respectively.
3. For FIFO RAM mode primitive, a trailing ‘R’ or ‘W’ in the signal name specifies the FIFO read port or write port respectively.
4. For FIFO RAM mode primitive FULLI has the same function as CSW(2) and EMPTYI has the same function as CSR(2).
5. In FIFO mode, CLKW is the write port clock, CSW is the write port chip select, CLKR is the read port clock, CSR is the read port chip select, ORE is the output read enable.

The EBR memory supports three forms of write behavior for single or dual port operation:

1. **Normal** – Data on the output appears only during the read cycle. During a write cycle, the data (at the current address) does not appear on the output. This mode is supported for all data widths.

2. **Write Through** – A copy of the input data appears at the output of the same port. This mode is supported for all data widths.

3. **Read-Before-Write** – When new data is being written, the old contents of the address appears at the output.

FIFO Configuration

The FIFO has a write port with data-in, CEW, WE and CLKW signals. There is a separate read port with data-out, RCE, RE and CLKR signals. The FIFO internally generates Almost Full, Full, Almost Empty and Empty Flags. The Full and Almost Full flags are registered with CLKW. The Empty and Almost Empty flags are registered with CLKR. Table 2-7 shows the range of programming values for these flags.

Table 2-7. Programmable FIFO Flag Ranges

Flag Name	Programming Range
Full (FF)	1 to max (up to 2^N-1)
Almost Full (AF)	1 to Full-1
Almost Empty (AE)	1 to Full-1
Empty (EF)	0

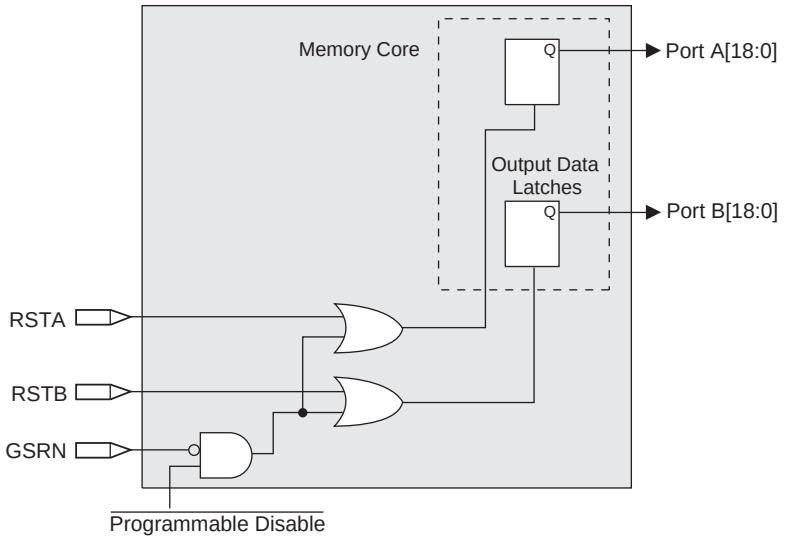
N = Address bit width.

The FIFO state machine supports two types of reset signals: RST and RPRST. The RST signal is a global reset that clears the contents of the FIFO by resetting the read/write pointer and puts the FIFO flags in their initial reset state. The RPRST signal is used to reset the read pointer. The purpose of this reset is to retransmit the data that is in the FIFO. In these applications it is important to keep careful track of when a packet is written into or read from the FIFO.

Memory Core Reset

The memory core contains data output latches for ports A and B. These are simple latches that can be reset synchronously or asynchronously. RSTA and RSTB are local signals, which reset the output latches associated with port A and port B respectively. The Global Reset (GSRN) signal resets both ports. The output data latches and associated resets for both ports are as shown in Figure 2-9.

Figure 2-9. Memory Core Reset

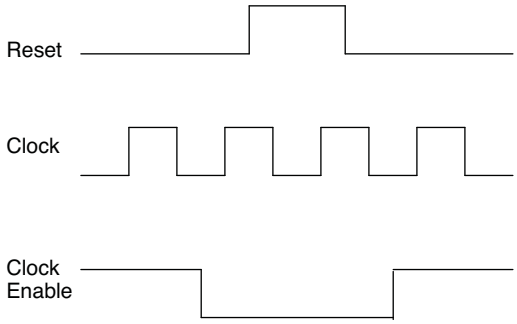


For further information on the sysMEM EBR block, please refer to TN1201, Memory Usage Guide for MachXO2 Devices.

EBR Asynchronous Reset

EBR asynchronous reset or GSR (if used) can only be applied if all clock enables are low for a clock cycle before the reset is applied and released a clock cycle after the reset is released, as shown in Figure 2-10. The GSR input to the EBR is always asynchronous.

Figure 2-10. EBR Asynchronous Reset (Including GSR) Timing Diagram



If all clock enables remain enabled, the EBR asynchronous reset or GSR may only be applied and released after the EBR read and write clock inputs are in a steady state condition for a minimum of $1/f_{MAX}$ (EBR clock). The reset release must adhere to the EBR synchronous reset setup time before the next active read or write clock edge.

If an EBR is pre-loaded during configuration, the GSR input must be disabled or the release of the GSR during device wake up must occur before the release of the device I/Os becoming active.

These instructions apply to all EBR RAM, ROM and FIFO implementations. For the EBR FIFO mode, the GSR signal is always enabled and the WE and RE signals act like the clock enable signals in Figure 2-10. The reset timing rules apply to the RPReset input versus the RE input and the RST input versus the WE and RE inputs. Both RST and RPReset are always asynchronous EBR inputs. For more details refer to TN1201, Memory Usage Guide for MachXO2 Devices.

Note that there are no reset restrictions if the EBR synchronous reset is used and the EBR GSR input is disabled.

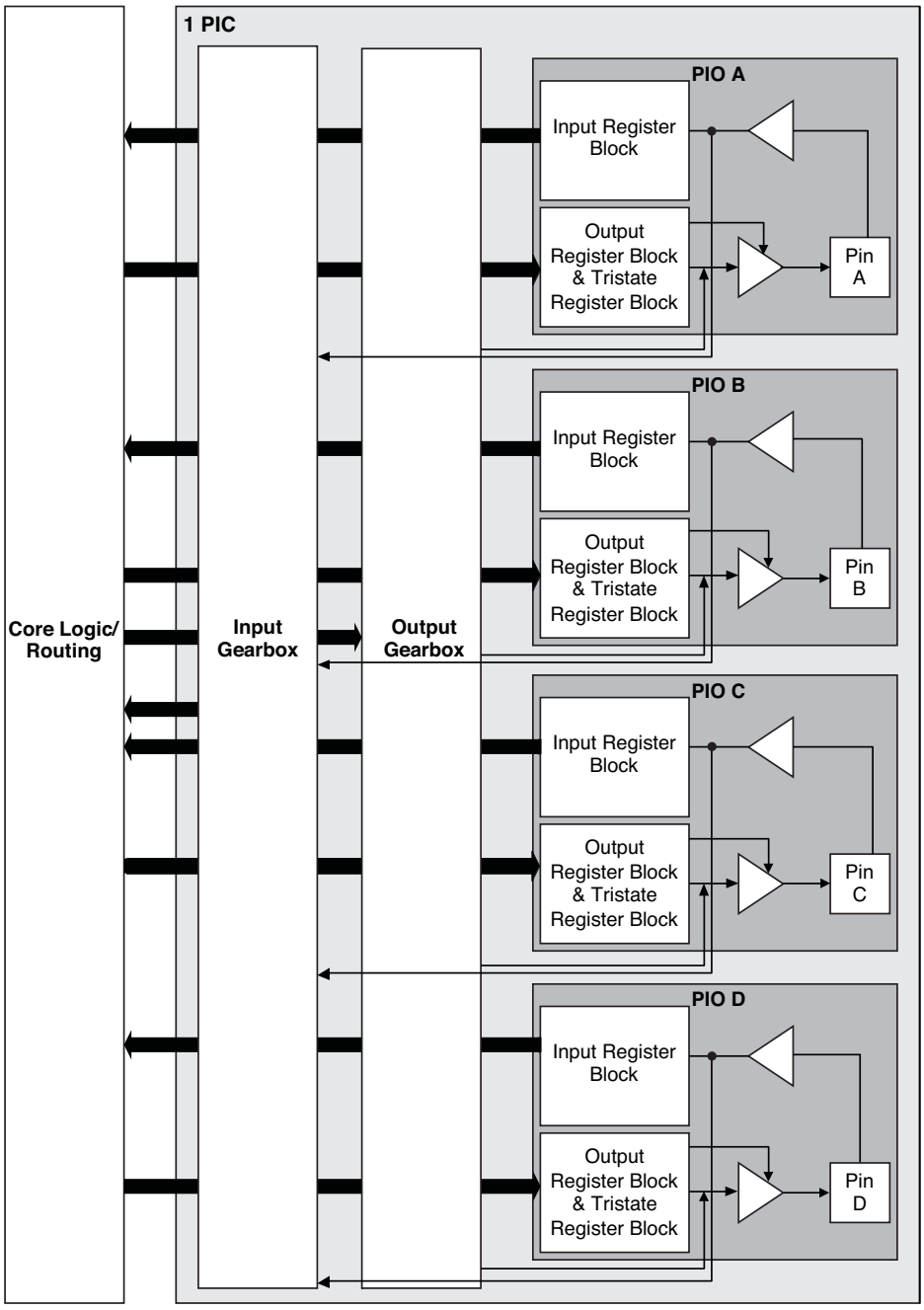
Programmable I/O Cells (PIC)

The programmable logic associated with an I/O is called a PIO. The individual PIO are connected to their respective sysIO buffers and pads. On the MachXO2 devices, the PIO cells are assembled into groups of four PIO cells called a Programmable I/O Cell or PIC. The PICs are placed on all four sides of the device.

On all the MachXO2 devices, two adjacent PIOs can be combined to provide a complementary output driver pair.

The MachXO2-640U, MachXO2-1200/U and higher density devices contain enhanced I/O capability. All PIO pairs on these larger devices can implement differential receivers. Half of the PIO pairs on the top edge of these devices can be configured as true LVDS transmit pairs. The PIO pairs on the bottom edge of these higher density devices have on-chip differential termination and also provide PCI support.

Figure 2-11. Group of Four Programmable I/O Cells



Notes:
1. Input gearbox is available only in PIC on the bottom edge of MachXO2-640U, MachXO2-1200/U and larger devices.
2. Output gearbox is available only in PIC on the top edge of MachXO2-640U, MachXO2-1200/U and larger devices.

PIO

The PIO contains three blocks: an input register block, output register block and tri-state register block. These blocks contain registers for operating in a variety of modes along with the necessary clock and selection logic.

Table 2-8. PIO Signal List

Pin Name	I/O Type	Description
CE	Input	Clock Enable
D	Input	Pin input from sysIO buffer.
INDD	Output	Register bypassed input.
INCK	Output	Clock input
Q0	Output	DDR positive edge input
Q1	Output	Registered input/DDR negative edge input
D0	Input	Output signal from the core (SDR and DDR)
D1	Input	Output signal from the core (DDR)
TD	Input	Tri-state signal from the core
Q	Output	Data output signals to sysIO Buffer
TQ	Output	Tri-state output signals to sysIO Buffer
DQSR90[1]	Input	DQS shift 90-degree read clock
DQSW90[1]	Input	DQS shift 90-degree write clock
DDRCLKPOL[1]	Input	DDR input register polarity control signal from DQS
SCLK	Input	System clock for input and output/tri-state blocks.
RST	Input	Local set reset signal

1. Available in PIO on right edge only.

Input Register Block

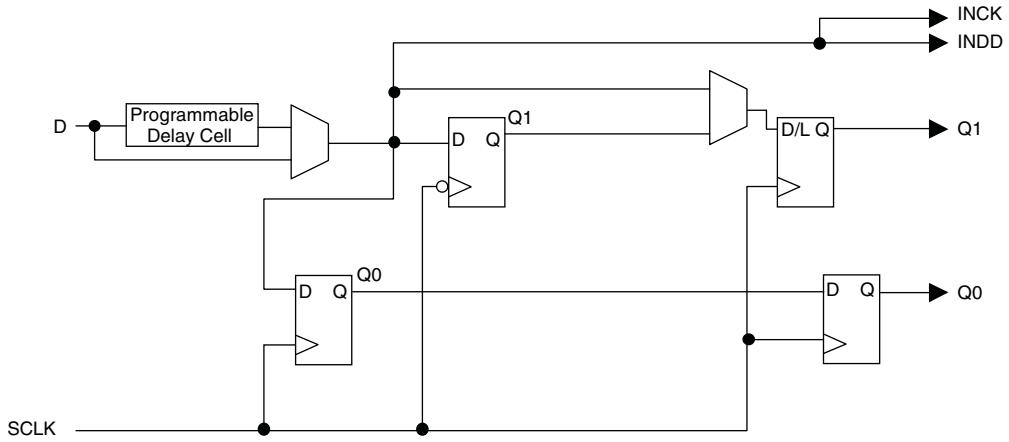
The input register blocks for the PIOs on all edges contain delay elements and registers that can be used to condition high-speed interface signals before they are passed to the device core. In addition to this functionality, the input register blocks for the PIOs on the right edge include built-in logic to interface to DDR memory.

Figure 2-12 shows the input register block for the PIOs located on the left, top and bottom edges. Figure 2-13 shows the input register block for the PIOs on the right edge.

Left, Top, Bottom Edges

Input signals are fed from the sysIO buffer to the input register block (as signal D). If desired, the input signal can bypass the register and delay elements and be used directly as a combinatorial signal (INDD), and a clock (INCK). If an input delay is desired, users can select a fixed delay. I/Os on the bottom edge also have a dynamic delay, DEL[4:0]. The delay, if selected, reduces input register hold time requirements when using a global clock. The input block allows two modes of operation. In single data rate (SDR) the data is registered with the system clock (SCLK) by one of the registers in the single data rate sync register block. In Generic DDR mode, two registers are used to sample the data on the positive and negative edges of the system clock (SCLK) signal, creating two data streams.

Figure 2-12. MachXO2 Input Register Block Diagram (PIO on Left, Top and Bottom Edges)



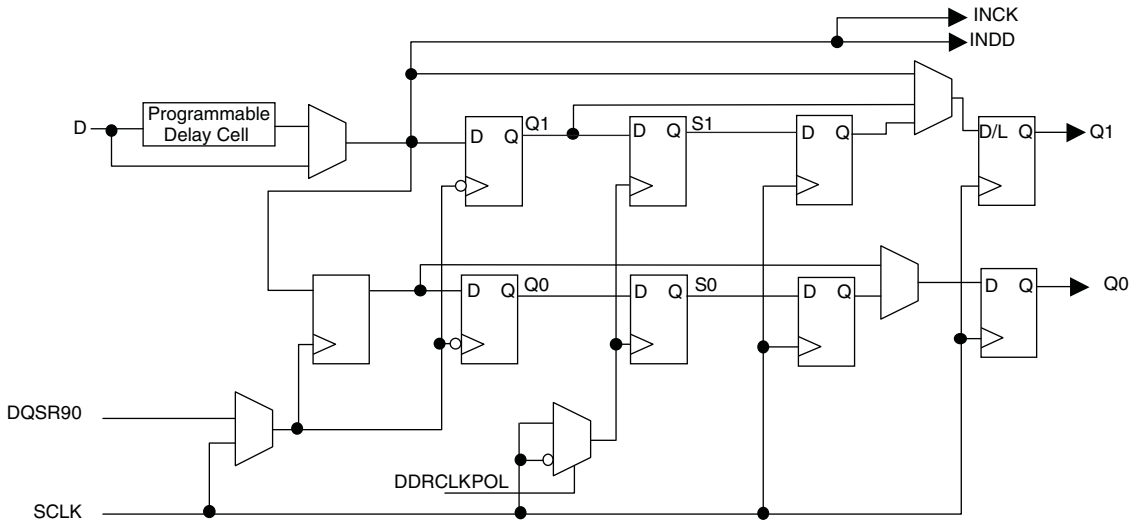
Right Edge

The input register block on the right edge is a superset of the same block on the top, bottom, and left edges. In addition to the modes described above, the input register block on the right edge also supports DDR memory mode.

In DDR memory mode, two registers are used to sample the data on the positive and negative edges of the modified DQS (DQSR90) in the DDR Memory mode creating two data streams. Before entering the core, these two data streams are synchronized to the system clock to generate two data streams.

The signal DDRCLKPOL controls the polarity of the clock used in the synchronization registers. It ensures adequate timing when data is transferred to the system clock domain from the DQS domain. The DQSR90 and DDRCLKPOL signals are generated in the DQS read-write block.

Figure 2-13. MachXO2 Input Register Block Diagram (PIO on Right Edge)



Output Register Block

The output register block registers signals from the core of the device before they are passed to the sysIO buffers.

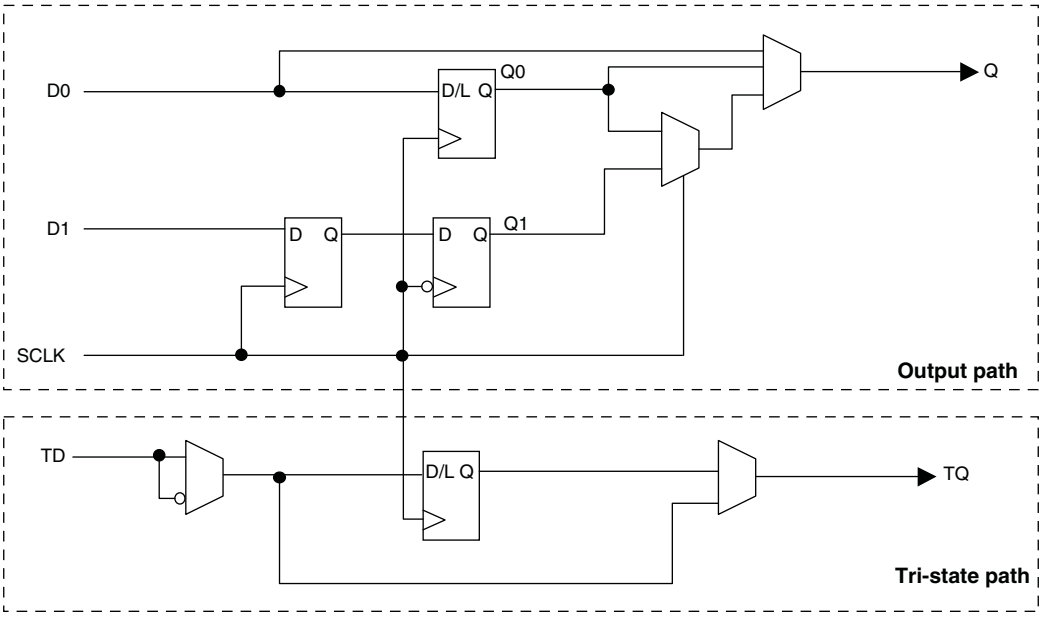
Left, Top, Bottom Edges

In SDR mode, D0 feeds one of the flip-flops that then feeds the output. The flip-flop can be configured as a D-type register or latch.

In DDR generic mode, D0 and D1 inputs are fed into registers on the positive edge of the clock. At the next falling edge the registered D1 input is registered into the register Q1. A multiplexer running off the same clock is used to switch the mux between the outputs of registers Q0 and Q1 that will then feed the output.

Figure 2-14 shows the output register block on the left, top and bottom edges.

Figure 2-14. MachXO2 Output Register Block Diagram (PIO on the Left, Top and Bottom Edges)



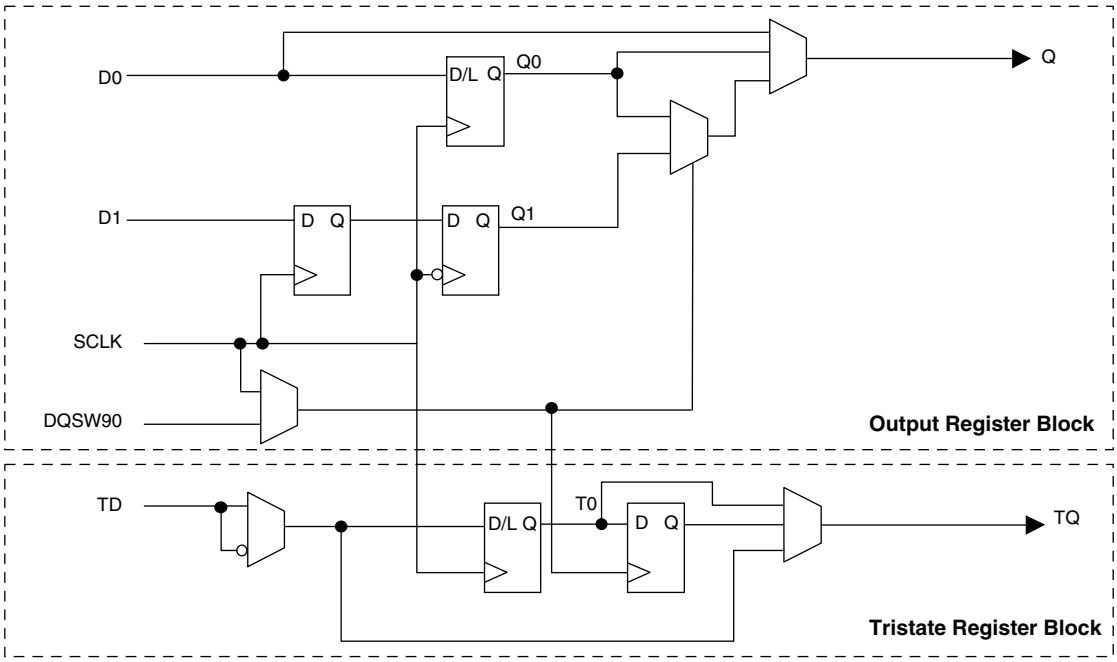
Right Edge

The output register block on the right edge is a superset of the output register on left, top and bottom edges of the device. In addition to supporting SDR and Generic DDR modes, the output register blocks for PIOs on the right edge include additional logic to support DDR-memory interfaces. Operation of this block is similar to that of the output register block on other edges.

In DDR memory mode, D0 and D1 inputs are fed into registers on the positive edge of the clock. At the next falling edge the registered D1 input is registered into the register Q1. A multiplexer running off the DQSW90 signal is used to switch the mux between the outputs of registers Q0 and Q1 that will then feed the output.

Figure 2-15 shows the output register block on the right edge.

Figure 2-15. MachXO2 Output Register Block Diagram (PIO on the Right Edges)



Tri-state Register Block

The tri-state register block registers tri-state control signals from the core of the device before they are passed to the sysIO buffers. The block contains a register for SDR operation. In SDR, TD input feeds one of the flip-flops that then feeds the output.

The tri-state register blocks on the right edge contain an additional register for DDR memory operation. In DDR memory mode, the register TS input is fed into another register that is clocked using the DQSW90 signal. The output of this register is used as a tri-state control.

Input Gearbox

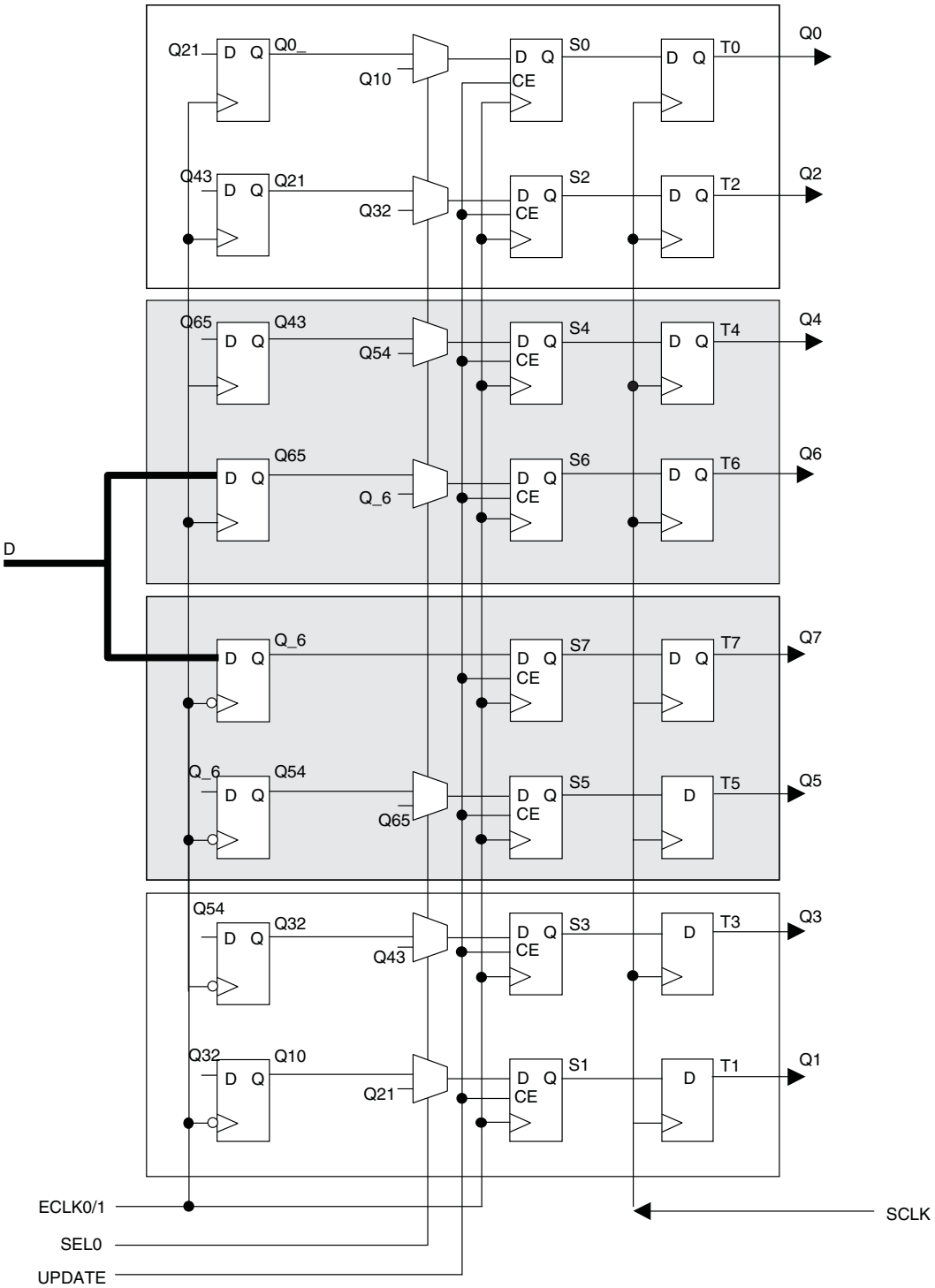
Each PIC on the bottom edge has a built-in 1:8 input gearbox. Each of these input gearboxes may be programmed as a 1:7 de-serializer or as one IDDRX4 (1:8) gearbox or as two IDDRX2 (1:4) gearboxes. Table 2-9 shows the gearbox signals.

Table 2-9. Input Gearbox Signal List

Name	I/O Type	Description
D	Input	High-speed data input after programmable delay in PIO A input register block
ALIGNWD	Input	Data alignment signal from device core
SCLK	Input	Slow-speed system clock
ECLK[1:0]	Input	High-speed edge clock
RST	Input	Reset
Q[7:0]	Output	Low-speed data to device core: Video RX(1:7): Q[6:0] GDDRX4(1:8): Q[7:0] GDDRX2(1:4)(IOL-A): Q4, Q5, Q6, Q7 GDDRX2(1:4)(IOL-C): Q0, Q1, Q2, Q3

These gearboxes have three stage pipeline registers. The first stage registers sample the high-speed input data by the high-speed edge clock on its rising and falling edges. The second stage registers perform data alignment based on the control signals UPDATE and SEL0 from the control block. The third stage pipeline registers pass the data to the device core synchronized to the low-speed system clock. Figure 2-16 shows a block diagram of the input gearbox.

Figure 2-16. Input Gearbox



More information on the input gearbox is available in TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

Output Gearbox

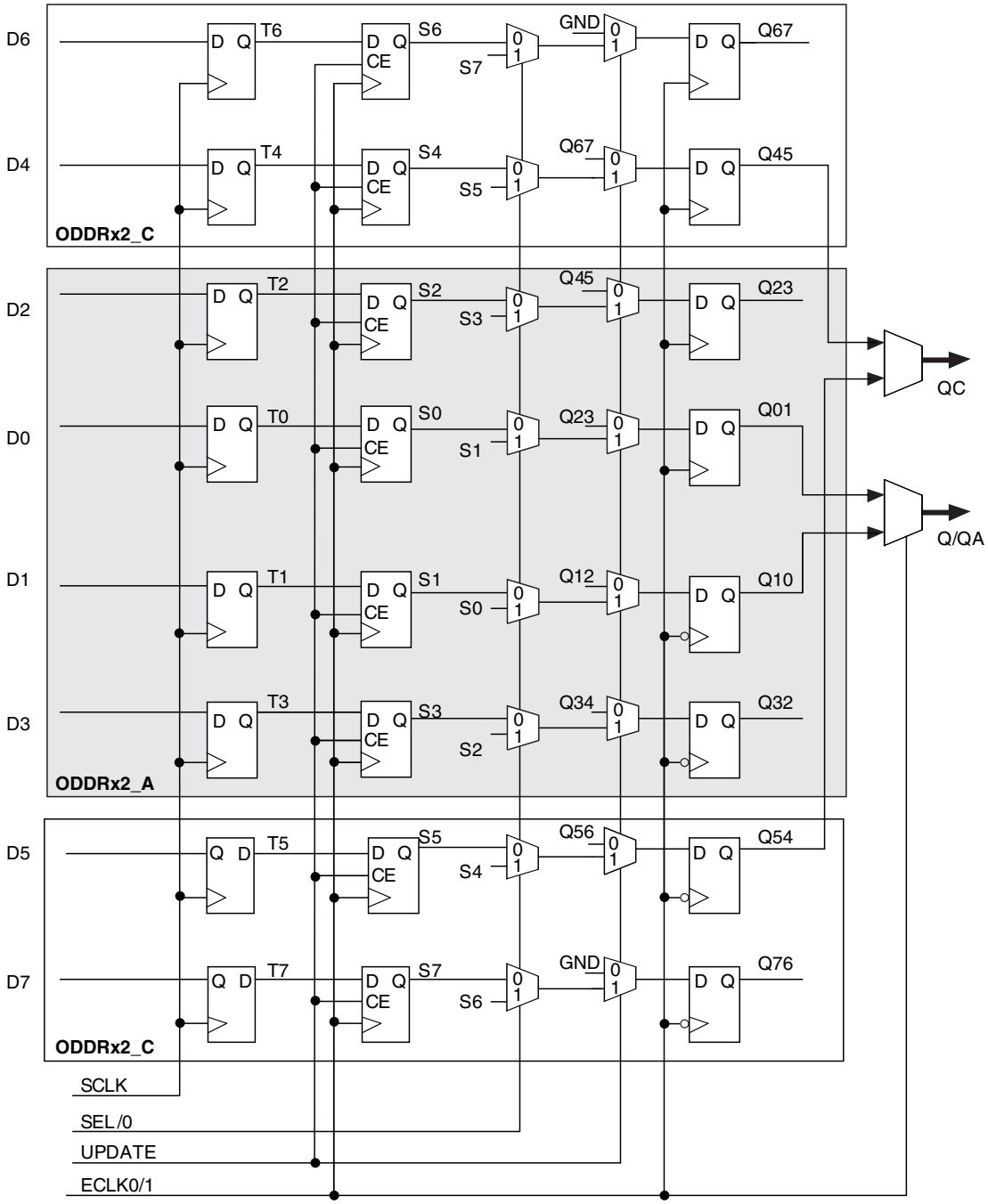
Each PIC on the top edge has a built-in 8:1 output gearbox. Each of these output gearboxes may be programmed as a 7:1 serializer or as one ODDRX4 (8:1) gearbox or as two ODDRX2 (4:1) gearboxes. Table 2-10 shows the gearbox signals.

Table 2-10. Output Gearbox Signal List

Name	I/O Type	Description
Q	Output	High-speed data output
D[7:0]	Input	Low-speed data from device core
Video TX(7:1): D[6:0]		
GDDRX4(8:1): D[7:0]		
GDDRX2(4:1)(IOL-A): D[3:0]		
GDDRX2(4:1)(IOL-C): D[7:4]		
SCLK	Input	Slow-speed system clock
ECLK [1:0]	Input	High-speed edge clock
RST	Input	Reset

The gearboxes have three stage pipeline registers. The first stage registers sample the low-speed input data on the low-speed system clock. The second stage registers transfer data from the low-speed clock registers to the high-speed clock registers. The third stage pipeline registers controlled by high-speed edge clock shift and mux the high-speed data out to the sysIO buffer. Figure 2-17 shows the output gearbox block diagram.

Figure 2-17. Output Gearbox



More information on the output gearbox is available in TN1203, Implementing High-Speed Interfaces with MachXO2 Devices.

DDR Memory Support

Certain PICs on the right edge of MachXO2-640U, MachXO2-1200/U and larger devices, have additional circuitry to allow the implementation of DDR memory interfaces. There are two groups of 14 or 12 PIOs each on the right edge with additional circuitry to implement DDR memory interfaces. This capability allows the implementation of up to 16-bit wide memory interfaces. One PIO from each group contains a control element, the DQS Read/Write Block, to facilitate the generation of clock and control signals (DQSR90, DQSW90, DDRCLKPOL and DATAVALID). These clock and control signals are distributed to the other PIO in the group through dedicated low skew routing.

DQS Read Write Block

Source synchronous interfaces generally require the input clock to be adjusted in order to correctly capture data at the input register. For most interfaces a PLL is used for this adjustment. However, in DDR memories the clock (referred to as DQS) is not free-running so this approach cannot be used. The DQS Read Write block provides the required clock alignment for DDR memory interfaces. DQSR90 and DQSW90 signals are generated by the DQS Read Write block from the DQS input.

In a typical DDR memory interface design, the phase relationship between the incoming delayed DQS strobe and the internal system clock (during the read cycle) is unknown. The MachXO2 family contains dedicated circuits to transfer data between these domains. To prevent set-up and hold violations, at the domain transfer between DQS (delayed) and the system clock, a clock polarity selector is used. This circuit changes the edge on which the data is registered in the synchronizing registers in the input register block. This requires evaluation at the start of each read cycle for the correct clock polarity. Prior to the read operation in DDR memories, DQS is in tri-state (pulled by termination). The DDR memory device drives DQS low at the start of the preamble state. A dedicated circuit in the DQS Read Write block detects the first DQS rising edge after the preamble state and generates the DDRCLKPOL signal. This signal is used to control the polarity of the clock to the synchronizing registers.

The temperature, voltage and process variations of the DQS delay block are compensated by a set of calibration signals (6-bit bus) from a DLL on the right edge of the device. The DLL loop is compensated for temperature, voltage and process variations by the system clock and feedback loop.

sysIO Buffer

Each I/O is associated with a flexible buffer referred to as a sysIO buffer. These buffers are arranged around the periphery of the device in groups referred to as banks. The sysIO buffers allow users to implement a wide variety of standards that are found in today's systems including LVCMOS, TTL, PCI, SSTL, HSTL, LVDS, BLVDS, MLVDS and LVPECL.

Each bank is capable of supporting multiple I/O standards. In the MachXO2 devices, single-ended output buffers, ratioed input buffers (LVTTL, LVCMOS and PCI), differential (LVDS) and referenced input buffers (SSTL and HSTL) are powered using I/O supply voltage (V_{CCIO}). Each sysIO bank has its own V_{CCIO}. In addition, each bank has a voltage reference, V_{REF}, which allows the use of referenced input buffers independent of the bank V_{CCIO}.

MachXO2-256 and MachXO2-640 devices contain single-ended ratioed input buffers and single-ended output buffers with complementary outputs on all the I/O banks. Note that the single-ended input buffers on these devices do not contain PCI clamps. In addition to the single-ended I/O buffers these two devices also have differential and referenced input buffers on all I/Os. The I/Os are arranged in pairs, the two pads in the pair are described as "T" and "C", where the true pad is associated with the positive side of the differential input buffer and the comp (complementary) pad is associated with the negative side of the differential input buffer.

MachXO2-640U, MachXO2-1200/U, MachXO2-2000/U, MachXO2-4000 and MachXO2-7000 devices contain three types of sysIO buffer pairs.

1. **Left and Right sysIO Buffer Pairs**
   The sysIO buffer pairs in the left and right banks of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the left and right of the devices also have differential and referenced input buffers.

2. **Bottom sysIO Buffer Pairs**
   The sysIO buffer pairs in the bottom bank of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the bottom also have differential and referenced input buffers. Only the I/Os on the bottom banks have programmable PCI clamps and differential input termination. The PCI clamp is enabled after V_{CC} and V_{CCIO} are at valid operating levels and the device has been configured.

3. **Top sysIO Buffer Pairs**
   The sysIO buffer pairs in the top bank of the device consist of two single-ended output drivers and two single-ended input buffers (for ratioed inputs such as LVCMOS and LVTTL). The I/O pairs on the top also have differential and referenced I/O buffers. Half of the sysIO buffer pairs on the top edge have true differential outputs. The sysIO buffer pair comprising of the A and B PIOs in every PIC on the top edge have a differential output driver. The referenced input buffer can also be configured as a differential input buffer.

Typical I/O Behavior During Power-up

The internal power-on-reset (POR) signal is deactivated when V_{CC} and V_{CCIO0} have reached V_{PORUP} level defined in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet. After the POR signal is deactivated, the FPGA core logic becomes active. It is the user's responsibility to ensure that all V_{CCIO} banks are active with valid input logic levels to properly control the output logic states of all the I/O banks that are critical to the application. The default configuration of the I/O pins in a blank device is tri-state with a weak pull-down to GND (some pins such as PROGRAMN and the JTAG pins have weak pull-up to V_{CCIO} as the default functionality). The I/O pins will maintain the blank configuration until V_{CC} and V_{CCIO} (for I/O banks containing configuration I/Os) have reached V_{PORUP} levels at which time the I/Os will take on the user-configured settings only after a proper download/configuration.

Supported Standards

The MachXO2 sysIO buffer supports both single-ended and differential standards. Single-ended standards can be further subdivided into LVCMOS, LVTTL, and PCI. The buffer supports the LVTTL, PCI, LVCMOS 1.2, 1.5, 1.8, 2.5, and 3.3 V standards. In the LVCMOS and LVTTL modes, the buffer has individually configurable options for drive strength, bus maintenance (weak pull-up, weak pull-down, bus-keeper latch or none) and open drain. BLVDS, MLVDS and LVPECL output emulation is supported on all devices. The MachXO2-640U, MachXO2-1200/U and higher devices support on-chip LVDS output buffers on approximately 50% of the I/Os on the top bank. Differential receivers for LVDS, BLVDS, MLVDS and LVPECL are supported on all banks of MachXO2 devices. PCI support is provided in the bottom bank of theMachXO2-640U, MachXO2-1200/U and higher density devices. Table 2-11 summarizes the I/O characteristics of the MachXO2 PLDs.

Tables 2-11 and 2-12 show the I/O standards (together with their supply and reference voltages) supported by the MachXO2 devices. For further information on utilizing the sysIO buffer to support a variety of standards please see TN1202, MachXO2 sysIO Usage Guide.

Table 2-11. I/O Support Device by Device

	MachXO2-256, MachXO2-640	MachXO2-640U, MachXO2-1200	MachXO2-1200U MachXO2-2000/U, MachXO2-4000, MachXO2-7000
Number of I/O Banks	4	4	6
Type of Input Buffers	Single-ended (all I/O banks) Differential Receivers (all I/O banks)	Single-ended (all I/O banks) Differential Receivers (all I/O banks) Differential input termination (bottom side)	Single-ended (all I/O banks) Differential Receivers (all I/O banks) Differential input termination (bottom side)
Types of Output Buffers	Single-ended buffers with complementary outputs (all I/O banks)	Single-ended buffers with complementary outputs (all I/O banks) Differential buffers with true LVDS outputs (50% on top side)	Single-ended buffers with complementary outputs (all I/O banks) Differential buffers with true LVDS outputs (50% on top side)
Differential Output Emulation Capability	All I/O banks	All I/O banks	All I/O banks
PCI Clamp Support	No	Clamp on bottom side only	Clamp on bottom side only

Table 2-12. Supported Input Standards

	VCCIO (Typ.)				
Input Standard	**3.3 V**	**2.5 V**	**1.8 V**	**1.5**	**1.2 V**
Single-Ended Interfaces					
LVTTL	✓	✓[2]	✓[2]	✓[2]	
LVCMOS33	✓	✓[2]	✓[2]	✓[2]	
LVCMOS25	✓[2]	✓	✓[2]	✓[2]	
LVCMOS18	✓[2]	✓[2]	✓	✓[2]	
LVCMOS15	✓[2]	✓[2]	✓[2]	✓	✓[2]
LVCMOS12	✓[2]	✓[2]	✓[2]	✓[2]	✓
PCI[1]	✓				
SSTL18 (Class I, Class II)	✓	✓	✓		
SSTL25 (Class I, Class II)	✓	✓			
HSTL18 (Class I, Class II)	✓	✓	✓		
Differential Interfaces					
LVDS	✓	✓			
BLVDS, MVDS, LVPECL, RSDS	✓	✓			
MIPI[3]	✓	✓			
Differential SSTL18 Class I, II	✓	✓	✓		
Differential SSTL25 Class I, II	✓	✓			
Differential HSTL18 Class I, II	✓	✓	✓		

1. Bottom banks of MachXO2-640U, MachXO2-1200/U and higher density devices only.
2. Reduced functionality. Refer to TN1202, MachXO2 sysIO Usage Guide for more detail.
3. These interfaces can be emulated with external resistors in all devices.

Table 2-13. Supported Output Standards

Output Standard	V_{CCIO} (Typ.)
Single-Ended Interfaces	
LVTTL	3.3
LVCMOS33	3.3
LVCMOS25	2.5
LVCMOS18	1.8
LVCMOS15	1.5
LVCMOS12	1.2
LVCMOS33, Open Drain	—
LVCMOS25, Open Drain	—
LVCMOS18, Open Drain	—
LVCMOS15, Open Drain	—
LVCMOS12, Open Drain	—
PCI33	3.3
SSTL25 (Class I)	2.5
SSTL18 (Class I)	1.8
HSTL18(Class I)	1.8
Differential Interfaces	
LVDS[1, 2]	2.5, 3.3
BLVDS, MLVDS, RSDS [2]	2.5
LVPECL[2]	3.3
MIPI[2]	2.5
Differential SSTL18	1.8
Differential SSTL25	2.5
Differential HSTL18	1.8

1. MachXO2-640U, MachXO2-1200/U and larger devices have dedicated LVDS buffers.
2. These interfaces can be emulated with external resistors in all devices.

sysIO Buffer Banks

The numbers of banks vary between the devices of this family. MachXO2-1200U, MachXO2-2000/U and higher density devices have six I/O banks (one bank on the top, right and bottom side and three banks on the left side). The MachXO2-1200 and lower density devices have four banks (one bank per side). Figures 2-18 and 2-19 show the sysIO banks and their associated supplies for all devices.

Figure 2-18. MachXO2-1200U, MachXO2-2000/U, MachXO2-4000 and MachXO2-7000 Banks

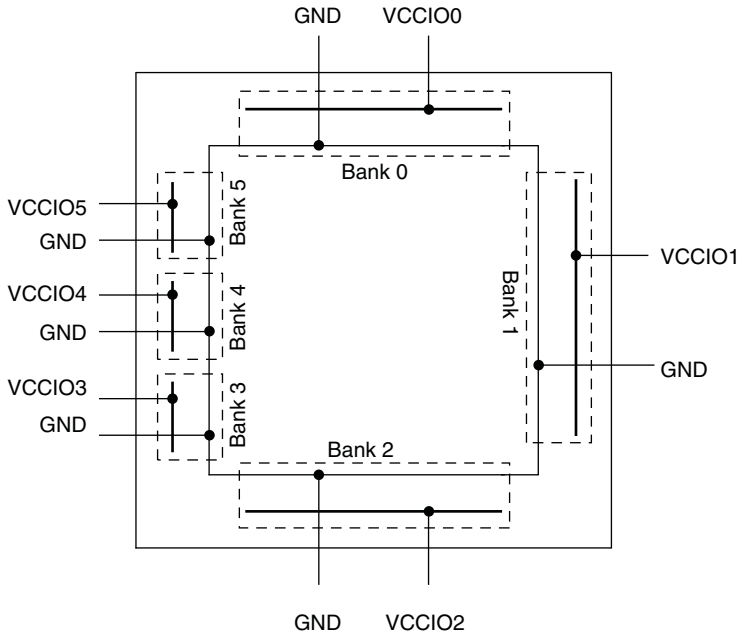
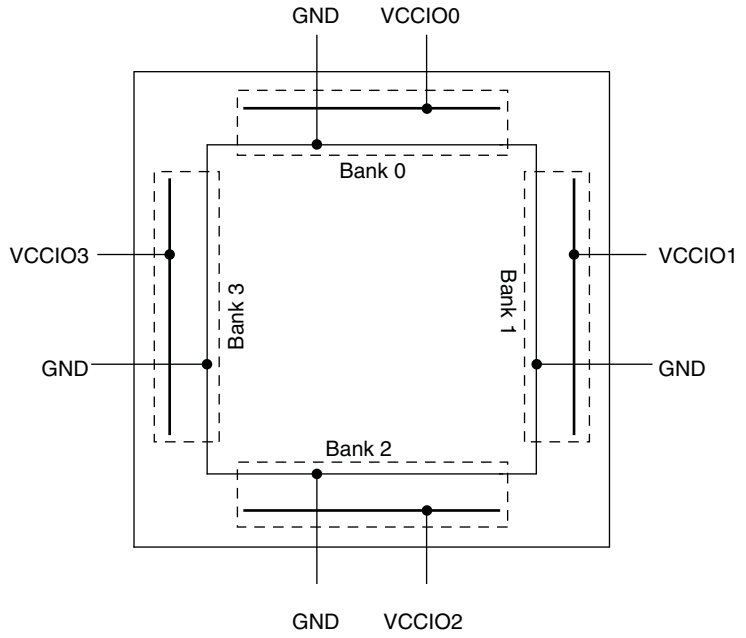


Figure 2-19. MachXO2-256, MachXO2-640/U and MachXO2-1200 Banks



Hot Socketing

The MachXO2 devices have been carefully designed to ensure predictable behavior during power-up and power-down. Leakage into I/O pins is controlled to within specified limits. This allows for easy integration with the rest of the system. These capabilities make the MachXO2 ideal for many multiple power supply and hot-swap applications.

On-chip Oscillator

Every MachXO2 device has an internal CMOS oscillator. The oscillator output can be routed as a clock to the clock tree or as a reference clock to the sysCLOCK PLL using general routing resources. The oscillator frequency can be divided by internal logic. There is a dedicated programming bit and a user input to enable/disable the oscillator. The oscillator frequency ranges from 2.08 MHz to 133 MHz. The software default value of the Master Clock (MCLK) is nominally 2.08 MHz. When a different MCLK is selected during the design process, the following sequence takes place:

1. Device powers up with a nominal MCLK frequency of 2.08 MHz.
2. During configuration, users select a different master clock frequency.
3. The MCLK frequency changes to the selected frequency once the clock configuration bits are received.
4. If the user does not select a master clock frequency, then the configuration bitstream defaults to the MCLK frequency of 2.08 MHz.

Table 2-14 lists all the available MCLK frequencies.

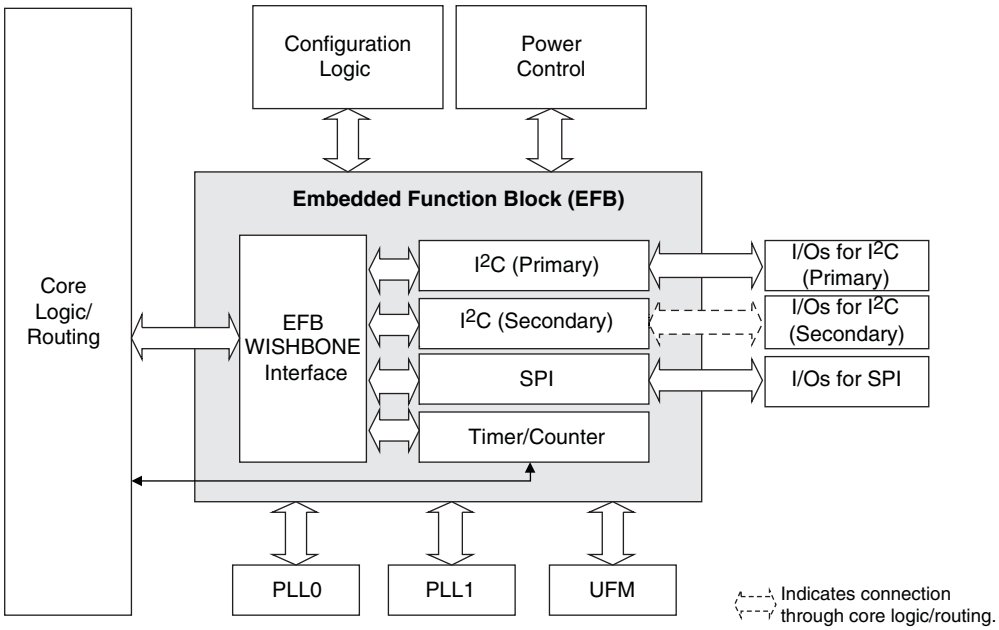
Table 2-14. Available MCLK Frequencies

MCLK (MHz, Nominal)	MCLK (MHz, Nominal)	MCLK (MHz, Nominal)
2.08 (default)	9.17	33.25
2.46	10.23	38
3.17	13.3	44.33
4.29	14.78	53.2
5.54	20.46	66.5
7	26.6	88.67
8.31	29.56	133

Embedded Hardened IP Functions and User Flash Memory

All MachXO2 devices provide embedded hardened functions such as SPI, I^2C and Timer/Counter. MachXO2-640/U and higher density devices also provide User Flash Memory (UFM). These embedded blocks interface through the WISHBONE interface with routing as shown in Figure 2-20.

Figure 2-20. Embedded Function Block Interface



Hardened I²C IP Core

Every MachXO2 device contains two I²C IP cores. These are the primary and secondary I²C IP cores. Either of the two cores can be configured either as an I²C master or as an I²C slave. The only difference between the two IP cores is that the primary core has pre-assigned I/O pins whereas users can assign I/O pins for the secondary core.

When the IP core is configured as a master it will be able to control other devices on the I²C bus through the interface. When the core is configured as the slave, the device will be able to provide I/O expansion to an I²C Master. The I²C cores support the following functionality:

- Master and Slave operation
- 7-bit and 10-bit addressing
- Multi-master arbitration support
- Up to 400 kHz data transfer speed
- General call support
- Interface to custom logic through 8-bit WISHBONE interface

Figure 2-21. I²C Core Block Diagram

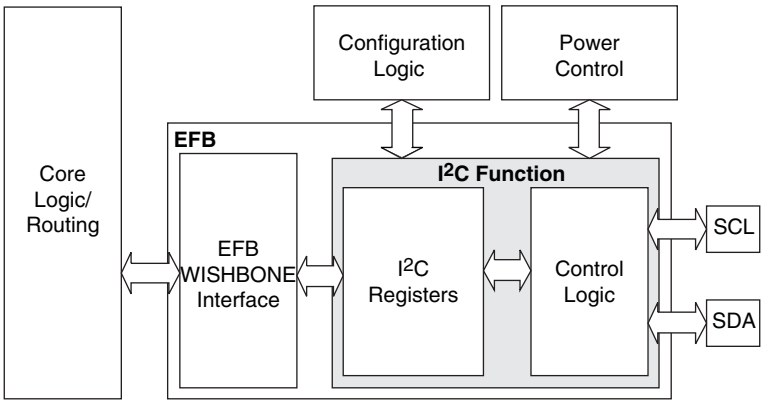


Table 2-15 describes the signals interfacing with the I²C cores.

Table 2-15. I²C Core Signal Description

Signal Name	I/O	Description
i2c_scl	Bi-directional	Bi-directional clock line of the I²C core. The signal is an output if the I²C core is in master mode. The signal is an input if the I²C core is in slave mode. MUST be routed directly to the pre-assigned I/O of the chip. Refer to the Pinout Information section of this document for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c_sda	Bi-directional	Bi-directional data line of the I²C core. The signal is an output when data is transmitted from the I²C core. The signal is an input when data is received into the I²C core. MUST be routed directly to the pre-assigned I/O of the chip. Refer to the Pinout Information section of this document for detailed pad and pin locations of I²C ports in each MachXO2 device.
i2c_irqo	Output	Interrupt request output signal of the I²C core. The intended usage of this signal is for it to be connected to the WISHBONE master controller (i.e. a microcontroller or state machine) and request an interrupt when a specific condition is met. These conditions are described with the I²C register definitions.
cfg_wake	Output	Wake-up signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, I²C Tab.
cfg_stdby	Output	Stand-by signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, I²C Tab.

Hardened SPI IP Core

Every MachXO2 device has a hard SPI IP core that can be configured as a SPI master or slave. When the IP core is configured as a master it will be able to control other SPI enabled chips connected to the SPI bus. When the core is configured as the slave, the device will be able to interface to an external SPI master. The SPI IP core on MachXO2 devices supports the following functions:

- Configurable Master and Slave modes
- Full-Duplex data transfer
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- LSB First or MSB First Data Transfer
- Interface to custom logic through 8-bit WISHBONE interface

There are some limitations on the use of the hardened user SPI. These are defined in the following technical notes:

- TN1087, Minimizing System Interruption During Configuration Using TransFR Technology (Appendix B)
- TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices

Figure 2-22. SPI Core Block Diagram

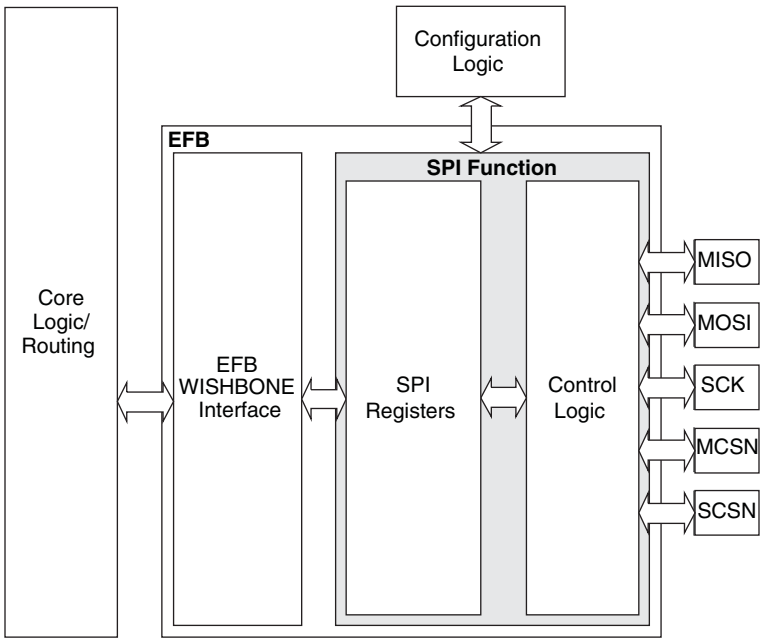


Table 2-16 describes the signals interfacing with the SPI cores.

Table 2-16. SPI Core Signal Description

Signal Name	I/O	Master/Slave	Description
spi_csn[0]	O	Master	SPI master chip-select output
spi_csn[1..7]	O	Master	Additional SPI chip-select outputs (total up to eight slaves)
spi_scsn	I	Slave	SPI slave chip-select input
spi_irq	O	Master/Slave	Interrupt request
spi_clk	I/O	Master/Slave	SPI clock. Output in master mode. Input in slave mode.
spi_miso	I/O	Master/Slave	SPI data. Input in master mode. Output in slave mode.
spi_mosi	I/O	Master/Slave	SPI data. Output in master mode. Input in slave mode.
ufm_sn	I	Slave	Configuration Slave Chip Select (active low), dedicated for selecting the User Flash Memory (UFM).
cfg_stdby	O	Master/Slave	Stand-by signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.
cfg_wake	O	Master/Slave	Wake-up signal – To be connected only to the power module of the MachXO2 device. The signal is enabled only if the “Wakeup Enable” feature has been set within the EFB GUI, SPI Tab.

Hardened Timer/Counter

MachXO2 devices provide a hard Timer/Counter IP core. This Timer/Counter is a general purpose, bi-directional, 16-bit timer/counter module with independent output compare units and PWM support. The Timer/Counter supports the following functions:

- Supports the following modes of operation:
  — Watchdog timer
  — Clear timer on compare match
  — Fast PWM
  — Phase and Frequency Correct PWM
- Programmable clock input source
- Programmable input clock prescaler
- One static interrupt output to routing
- One wake-up interrupt to on-chip standby mode controller.
- Three independent interrupt sources: overflow, output compare match, and input capture
- Auto reload
- Time-stamping support on the input capture unit
- Waveform generation on the output
- Glitch-free PWM waveform generation with variable PWM period
- Internal WISHBONE bus access to the control and status registers
- Stand-alone mode with preloaded control registers and direct reset input

Figure 2-23. Timer/Counter Block Diagram

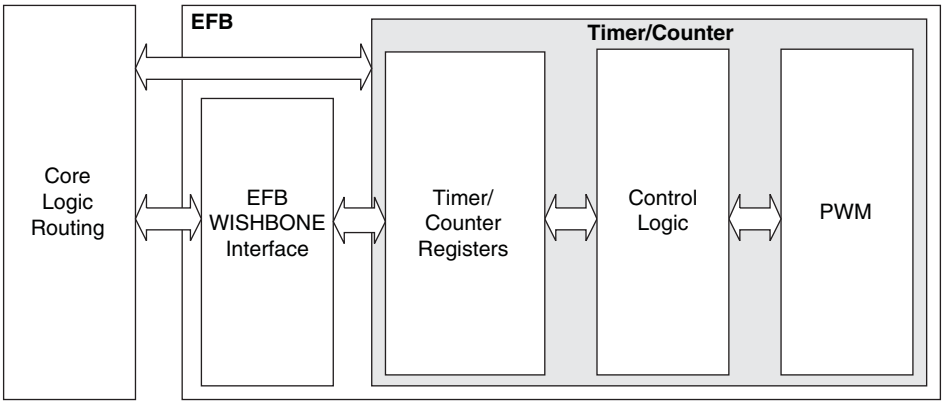


Table 2-17. Timer/Counter Signal Description

Port	I/O	Description
tc_clki	I	Timer/Counter input clock signal
tc_rstn	I	Register tc_rstn_ena is preloaded by configuration to always keep this pin enabled
tc_ic	I	Input capture trigger event, applicable for non-pwm modes with WISHBONE interface. If enabled, a rising edge of this signal will be detected and synchronized to capture tc_cnt value into tc_icr for time-stamping.
tc_int	O	Without WISHBONE – Can be used as overflow flag With WISHBONE – Controlled by three IRQ registers
tc_oc	O	Timer counter output signal

For more details on these embedded functions, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

User Flash Memory (UFM)

MachXO2-640/U and higher density devices provide a User Flash Memory block, which can be used for a variety of applications including storing a portion of the configuration image, initializing EBRs, to store PROM data or, as a general purpose user Flash memory. The UFM block connects to the device core through the embedded function block WISHBONE interface. Users can also access the UFM block through the JTAG, I^2C and SPI interfaces of the device. The UFM block offers the following features:

- Non-volatile storage up to 256 kbits
- 100K write cycles
- Write access is performed page-wise; each page has 128 bits (16 bytes)
- Auto-increment addressing
- WISHBONE interface

For more information on the UFM, please refer to TN1205, Using User Flash Memory and Hardened Control Functions in MachXO2 Devices.

Standby Mode and Power Saving Options

MachXO2 devices are available in three options for maximum flexibility: ZE, HC and HE devices. The ZE devices have ultra low static and dynamic power consumption. These devices use a 1.2 V core voltage that further reduces power consumption. The HC and HE devices are designed to provide high performance. The HC devices have a built-in voltage regulator to allow for 2.5 V V_{CC} and 3.3 V V_{CC} while the HE devices operate at 1.2 V V_{CC}.

MachXO2 devices have been designed with features that allow users to meet the static and dynamic power requirements of their applications by controlling various device subsystems such as the bandgap, power-on-reset circuitry, I/O bank controllers, power guard, on-chip oscillator, PLLs, etc. In order to maximize power savings, MachXO2 devices support an ultra low power Stand-by mode. While most of these features are available in all three device types, these features are mainly intended for use with MachXO2 ZE devices to manage power consumption.

In the stand-by mode the MachXO2 devices are powered on and configured. Internal logic, I/Os and memories are switched on and remain operational, as the user logic waits for an external input. The device enters this mode when the standby input of the standby controller is toggled or when an appropriate I^2C or JTAG instruction is issued by an external master. Various subsystems in the device such as the band gap, power-on-reset circuitry etc can be configured such that they are automatically turned “off” or go into a low power consumption state to save power when the device enters this state. Note that the MachXO2 devices are powered on when in standby mode and all power supplies should remain in the Recommended Operating Conditions.

Table 2-18. MachXO2 Power Saving Features Description

Device Subsystem	Feature Description
Bandgap	The bandgap can be turned off in standby mode. When the Bandgap is turned off, analog circuitry such as the POR, PLLs, on-chip oscillator, and referenced and differential I/O buffers are also turned off. Bandgap can only be turned off for 1.2 V devices.
Power-On-Reset (POR)	The POR can be turned off in standby mode. This monitors VCC levels. In the event of unsafe V_{CC} drops, this circuit reconfigures the device. When the POR circuitry is turned off, limited power detector circuitry is still active. This option is only recommended for applications in which the power supply rails are reliable.
On-Chip Oscillator	The on-chip oscillator has two power saving features. It may be switched off if it is not needed in your design. It can also be turned off in Standby mode.
PLL	Similar to the on-chip oscillator, the PLL also has two power saving features. It can be statically switched off if it is not needed in a design. It can also be turned off in Standby mode. The PLL will wait until all output clocks from the PLL are driven low before powering off.
I/O Bank Controller	Referenced and differential I/O buffers (used to implement standards such as HSTL, SSTL and LVDS) consume more than ratioed single-ended I/Os such as LVCMOS and LVTTL. The I/O bank controller allows the user to turn these I/Os off dynamically on a per bank selection.
Dynamic Clock Enable for Primary Clock Nets	Each primary clock net can be dynamically disabled to save power.
Power Guard	Power Guard is a feature implemented in input buffers. This feature allows users to switch off the input buffer when it is not needed. This feature can be used in both clock and data paths. Its biggest impact is that in the standby mode it can be used to switch off clock inputs that are distributed using general routing resources.

For more details on the standby mode refer to TN1198, Power Estimation and Management for MachXO2 Devices.

Power On Reset

MachXO2 devices have power-on reset circuitry to monitor V_{CCINT} and V_{CCIO} voltage levels during power-up and operation. At power-up, the POR circuitry monitors V_{CCINT} and V_{CCIO0} (controls configuration) voltage levels. It then triggers download from the on-chip configuration Flash memory after reaching the V_{PORUP} level specified in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet. For devices without voltage regulators (ZE and HE devices), V_{CCINT} is the same as the V_{CC} supply voltage. For devices with voltage regulators (HC devices), V_{CCINT} is regulated from the V_{CC} supply voltage. From this voltage reference, the time taken for configuration and entry into user mode is specified as Flash Download Time ($t_{REFRESH}$) in the DC and Switching Characteristics section of this data sheet. Before and during configuration, the I/Os are held in tri-state. I/Os are released to user functionality once the device has finished configuration. Note that for HC devices, a separate POR circuit monitors external V_{CC} voltage in addition to the POR circuit that monitors the internal post-regulated power supply voltage level.

Once the device enters into user mode, the POR circuitry can optionally continue to monitor V_{CCINT} levels. If V_{CCINT} drops below $V_{PORDNBG}$ level (with the bandgap circuitry switched on) or below $V_{PORDNSRAM}$ level (with the bandgap circuitry switched off to conserve power) device functionality cannot be guaranteed. In such a situation the POR issues a reset and begins monitoring the V_{CCINT} and V_{CCIO} voltage levels. $V_{PORDNBG}$ and $V_{PORDNSRAM}$ are both specified in the Power-On-Reset Voltage table in the DC and Switching Characteristics section of this data sheet.

Note that once a ZE or HE device enters user mode, users can switch off the bandgap to conserve power. When the bandgap circuitry is switched off, the POR circuitry also shuts down. The device is designed such that a minimal, low power POR circuit is still operational (this corresponds to the $V_{PORDNSRAM}$ reset point described in the paragraph above). However this circuit is not as accurate as the one that operates when the bandgap is switched on. The low power POR circuit emulates an SRAM cell and is biased to trip before the vast majority of SRAM cells flip. If users are concerned about the V_{CC} supply dropping below V_{CC} (min) they should not shut down the bandgap or POR circuit.

Configuration and Testing

This section describes the configuration and testing features of the MachXO2 family.

IEEE 1149.1-Compliant Boundary Scan Testability

All MachXO2 devices have boundary scan cells that are accessed through an IEEE 1149.1 compliant test access port (TAP). This allows functional testing of the circuit board, on which the device is mounted, through a serial scan path that can access all critical logic nodes. Internal registers are linked internally, allowing test data to be shifted in and loaded directly onto test nodes, or test data to be captured and shifted out for verification. The test access port consists of dedicated I/Os: TDI, TDO, TCK and TMS. The test access port shares its power supply with V_{CCIO} Bank 0 and can operate with LVCMOS3.3, 2.5, 1.8, 1.5, and 1.2 standards.

For more details on boundary scan test, see AN8066, Boundary Scan Testability with Lattice sysIO Capability and TN1087, Minimizing System Interruption During Configuration Using TransFR Technology.

Device Configuration

All MachXO2 devices contain two ports that can be used for device configuration. The Test Access Port (TAP), which supports bit-wide configuration and the sysCONFIG port which supports serial configuration through I^2C or SPI. The TAP supports both the IEEE Standard 1149.1 Boundary Scan specification and the IEEE Standard 1532 In-System Configuration specification. There are various ways to configure a MachXO2 device:

1. Internal Flash Download
2. JTAG
3. Standard Serial Peripheral Interface (Master SPI mode) – interface to boot PROM memory
4. System microprocessor to drive a serial slave SPI port (SSPI mode)
5. Standard I^2C Interface to system microprocessor

Upon power-up, the configuration SRAM is ready to be configured using the selected sysCONFIG port. Once a configuration port is selected, it will remain active throughout that configuration cycle. The IEEE 1149.1 port can be activated any time after power-up by sending the appropriate command through the TAP port. Optionally the device can run a CRC check upon entering the user mode. This will ensure that the device was configured correctly.

The sysCONFIG port has 10 dual-function pins which can be used as general purpose I/Os if they are not required for configuration. See TN1204, MachXO2 Programming and Configuration Usage Guide for more information about using the dual-use pins as general purpose I/Os.

Lattice design software uses proprietary compression technology to compress bit-streams for use in MachXO2 devices. Use of this technology allows Lattice to provide a lower cost solution. In the unlikely event that this technology is unable to compress bitstreams to fit into the amount of on-chip Flash memory, there are a variety of techniques that can be utilized to allow the bitstream to fit in the on-chip Flash memory. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

The Test Access Port (TAP) has five dual purpose pins (TDI, TDO, TMS, TCK and JTAGENB). These pins are dual function pins - TDI, TDO, TMS and TCK can be used as general purpose I/O if desired. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

TransFR (Transparent Field Reconfiguration)
TransFR is a unique Lattice technology that allows users to update their logic in the field without interrupting system operation using a simple push-button solution. For more details refer to TN1087, Minimizing System Interruption During Configuration Using TransFR Technology for details.

When implementing background programming of the on-chip Flash, care must be taken for the operation of the PLL. For devices that have two PLLs (XO2-2000U, -4000 and -7000), the system must put the RPLL (Right-side PLL) in reset state during the background Flash programming. More detailed description can be found in TN1204, MachXO2 Programming and Configuration Usage Guide.

Security and One-Time Programmable Mode (OTP)
For applications where security is important, the lack of an external bitstream provides a solution that is inherently more secure than SRAM-based FPGAs. This is further enhanced by device locking. MachXO2 devices contain security bits that, when set, prevent the readback of the SRAM configuration and non-volatile Flash memory spaces. The device can be in one of two modes:

1. Unlocked – Readback of the SRAM configuration and non-volatile Flash memory spaces is allowed.

2. Permanently Locked – The device is permanently locked.

Once set, the only way to clear the security bits is to erase the device. To further complement the security of the device, a One Time Programmable (OTP) mode is available. Once the device is set in this mode it is not possible to erase or re-program the Flash and SRAM OTP portions of the device. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

Dual Boot
MachXO2 devices can optionally boot from two patterns, a primary bitstream and a golden bitstream. If the primary bitstream is found to be corrupt while being downloaded into the SRAM, the device shall then automatically re-boot from the golden bitstream. Note that the primary bitstream must reside in the on-chip Flash. The golden image MUST reside in an external SPI Flash. For more details, refer to TN1204, MachXO2 Programming and Configuration Usage Guide.

Soft Error Detection
The SED feature is a CRC check of the SRAM cells after the device is configured. This check ensures that the SRAM cells were configured successfully. This feature is enabled by a configuration bit option. The Soft Error Detection can also be initiated in user mode via an input to the fabric. The clock for the Soft Error Detection circuit is generated using a dedicated divider. The undivided clock from the on-chip oscillator is the input to this divider. For low power applications users can switch off the Soft Error Detection circuit. For more details, refer to TN1206, MachXO2 Soft Error Detection Usage Guide.

TraceID

Each MachXO2 device contains a unique (per device), TraceID that can be used for tracking purposes or for IP security applications. The TraceID is 64 bits long. Eight out of 64 bits are user-programmable, the remaining 56 bits are factory-programmed. The TraceID is accessible through the EFB WISHBONE interface and can also be accessed through the SPI, I^2C, or JTAG interfaces.

Density Shifting

The MachXO2 family has been designed to enable density migration within the same package. Furthermore, the architecture ensures a high success rate when performing design migration from lower density devices to higher density devices. In many cases, it is also possible to shift a lower utilization design targeted for a high-density device to a lower density device. However, the exact details of the final resource utilization will impact the likely success in each case. When migrating from lower to higher density or higher to lower density, ensure to review all the power supplies and NC pins of the chosen devices. For more details refer to the MachXO2 migration files.