

# DISEÑO DE UN SOPC (SYSTEM ON PROGRAMMABLE CHIP) PARA EL CONTROL DE UNA CÁMARA DE 5MP CON PANTALLA TÁCTIL EN EL ENTORNO DE TRABAJO DE LA TARJETA DE2-115 DE ALTERA

Memoria de TFM

**ALUMNO:** Oier Martínez Echeverría

**TUTOR:** Dr. D. Ricardo José Colom Palero

**COTUTOR:** Dr. D. Marcos Antonio Martínez Peiró



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

## Contenido

ABSTRACT .....	9
1- INTRODUCCIÓN .....	10
1.1 ESTADO DEL ARTE.....	11
Altera DE1.....	12
Altera DE0.....	13
Altera DE0-Nano .....	13
Altera DE2i-150.....	14
CPU .....	14
FPGA .....	14
Altera DE1-SoC.....	15
Altera DE0-CV .....	15
Altera DE0-Nano-SoC/Atlas SoC Kit .....	16
1.2 OBJETIVOS .....	17
2- ANÁLISIS DE LOS DISPOSITIVOS EMPLEADOS Y SU FUNCIONALIDAD.....	18
2.1 PLACA DE DESARROLLO DE2-115.....	18
2.2 CAPTURA DE IMÁGENES .....	20
2.3 PANTALLA TÁCTIL.....	26
2.2.1 Pantalla LCD.....	28
2.2.2 Panel digitalizador táctil o Touch Panel .....	28
2.3 TARJETA DE EXPANSION .....	29
3- DISEÑO HARDWARE DEL PROYECTO .....	31
3.1 Altera University Program .....	32
3.2 Bloques IP básicos del sistema .....	32
Nios II.....	32
RELOJ DEL SISTEMA (2 Opciones).....	36
System and SDRAM Clocks for DE-series Boards (Opción 1).....	36
Avalon ALTPLL (Opcion 2).....	37
SDRAM Controller.....	38
JTAG Uart.....	39
Clock Bridge y Reset Bridge .....	40
3.3 Bloques IP de la cámara.....	40
Video-In Decoder.....	41

Bayer Pattern Resampler.....	41
Video Stream Router .....	42
Test-Pattern Generator .....	43
PIO (Parallel I/O), Opcional.....	43
Video DMA Controller .....	44
OPCIÓN 1: BLOQUES DEL UNIVERSITY PROGRAM .....	47
Video Scaler .....	47
RGB Resampler .....	48
Video Dual-Clock FIFO .....	48
VGA Controller.....	49
OPCIÓN 2: BLOQUES VIP DE ALTERA.....	50
RAW to VIP Bridge .....	50
Timing Adapter 0 .....	50
Scaler II .....	51
Timing Adapter 1 .....	53
VIP to RAW Adapter Bridge .....	53
RGB Resampler .....	54
VGA Controller.....	54
3.4 Bloques IP para la pantalla táctil .....	55
LT24 Controller .....	55
3-Wire SPI .....	56
PIO (Parallel I/O).....	56
Avalon ALTPLL.....	58
Timer.....	59
3.5 Bloques IP adicionales empleados .....	60
SD Card Interface.....	60
Generic Tri-State Controller .....	60
Tri-State Conduit Bridge .....	63
Parallel Port .....	63
3.6 Modificación de la IP Audio and Video Config .....	64
3.7 Preparación del proyecto para la compilación.....	66
4- DISEÑO SOFTWARE DEL PROYECTO .....	69
4.1 Aspectos previos al diseño software .....	70

4.2	Ficheros del proyecto y su funcionalidad .....	72
4.3	Detalles específicos de la programación .....	74
	Definiciones (Ficheros: software_cam.c y gestion_flash.c) .....	74
	Variables (Ficheros: software_cam.c) .....	75
	Manejo de memoria (Ficheros: software_cam.c, gestión_flash.c y gestión_SD.c)...	75
	Funcionamiento general del sistema .....	79
5-	PRUEBAS DE RENDIMIENTO Y JUSTIFICACIONES DE DISEÑO.....	81
5.1	NIOS II .....	81
	Pruebas de rendimiento de las cachés.....	82
	Pruebas de rendimiento del Predictor de Saltos.....	88
5.2	Memoria de Vídeo .....	92
	Memoria SRAM.....	93
	Memoria SDRAM .....	99
	Memoria On-Chip .....	100
	Funcionamiento de subsistema de memoria completo.....	102
	Conclusiones y Justificación del uso de la SDRAM como memoria de programa...	105
5.3	Pruebas de rendimiento de lectura/escritura de fotografías .....	106
6-	FUNCIONAMIENTO DEL SISTEMA.....	107
	Home .....	108
	Cámara.....	108
	Galería.....	109
	Config.....	109
	Estado .....	110
7-	CONCLUSIONES.....	111
8-	REFERENCIAS .....	113
	Terasic Corp. ....	113
	Altera Corp.....	113
	Otros .....	113

## INDICE DE FIGURAS

figura 1, Diagrama de bloques básico del sistema .....	10
figura 2, Placa de desarrollo Altera DE1 .....	12
figura 3, Placa de desarrollo Altera DE0 .....	13
figura 4, Placa de desarrollo Altera DE0-Nano .....	13
figura 5, Placa de desarrollo Altera DE2i-150.....	14
figura 6, Placa de desarrollo Altera DE1 SoC .....	15
figura 7, Placa de desarrollo Altera DE0-CV .....	15
figura 8, Placa de desarrollo Altera DE0-Nano-Soc/Atlas SoC Kit .....	16
figura 9, Ejemplo de una placa de desarrollo profesional, destinada a aplicaciones demandantes de alta capacidad de computación y alto ancho de banda, en formato de tarjeta PCIe. La FPGA Arria 10 mostrada en la imagen tiene 1.150.000 LEs, la memoria RAM es manualmente intercambiable empleando ranuras SODIMM además de incorporar 36MB de memoria QDRII+ SRAM. ....	17
figura 10, Placa de desarrollo Terasic DE2-115 .....	18
figura 11, Diagrama de bloques de la placa DE2-115 facilitado por Terasic.....	19
figura 12, Cámara D5M de Terasic .....	20
figura 13, Diagrama de Bloques del sensor Micron MT9P001 .....	21
figura 14, Región Frontera o Boundary Region del Sensor (Bordes) y Región de Imagen activa (Región Central) .....	21
figura 15, A la izquierda el Patrón Bayer RGB y a la derecha la descomposición de los tres colores .....	22
figura 16, Conversión Bayer a RGB de la IP proporcionada por Terasic .....	22
figura 17, Diferentes factores de recorte en función del tamaño de sensor .....	24
figura 18, Diferentes modos de Pixel Skipping o Saltos de Píxeles .....	25
figura 19, Diferentes opciones para la realización del binning .....	25
figura 20, Configuración del sensor en función de la resolución .....	26
figura 21, Pantalla táctil Terasic LT24 .....	27
figura 22, Separación entre pantalla LCD y Panel Táctil.....	27
figura 23, Tarjeta de expansión THDB-HTG de Terasic .....	29
figura 24, Diagrama de bloques de la tarjeta de expansión THDB-HTG facilitado por Terasic .....	30
figura 25, Vista inferior de la tarjeta de expansión THDB-HTG. Se observa el conector físico HSTC que se conecta a la placa. ....	30
figura 26, Diagrama de bloques hardware del sistema. Se muestran los bloques principales sin Relojes/Resets.....	31
figura 27, Pestaña Main del NIOS II .....	33
figura 28, Pestaña Vectors del NIOS II .....	33
figura 29, Pestaña Caches and Memory Interfaces del NIOS II .....	34
figura 30, Pestaña Arithmetic Instructions del NIOS II .....	34
figura 31, Pestaña MMU and MPU settings del NIOS II .....	35
figura 32, Pestaña JTAG Debug del NIOS II.....	35
figura 33, Pestaña Advanced Features del NIOS II .....	36

figura 34, Configuración de la IP de generación de relojes.....	37
figura 35, Configuración del primer PLL (Sistema) de la IP Avalon ALTPLL.....	37
figura 36, Configuración del segundo PLL (SDRAM) de la IP Avalon ALTPLL.....	38
figura 37, Pestaña Memory Profile del controlador SDRAM.....	38
figura 38, Pestaña timing del controlador SDRAM.....	39
figura 39, Configuración del JTAG Uart .....	40
figura 40, Configuración del Reset Bridge .....	40
figura 41, Configuración del Clock Bridge para la memoria SDRAM.....	40
figura 42, Configuración del Video-In Decoder .....	41
figura 43, Configuración del Bayer Pattern Resampler .....	42
figura 44, Configuración del Video Stream Router .....	43
figura 45, Configuración de la IP Test-Pattern Generator.....	43
figura 46, Configuración de la IP PIO para el control de la exposición vía software.....	44
figura 47, Configuración del DMA Controller para escritura en memoria.....	45
figura 48, Conexión de la IP Video DMA Controller de escritura sin Video Stream Router .....	46
figura 49, Configuración del controlador DMA para la lectura de la memoria.....	46
figura 50, Configuración de la IP Video Scaler.....	48
figura 51, Configuración de la IP RGB Resampler.....	48
figura 52, Configuración de la IP Video Dual-Clock FIFO.....	49
figura 53, Configuración de la IP VGA Controller en la primera opción.....	49
figura 54, Configuración del bloque RAW to VIP Bridge .....	50
figura 55, Configuración de la IP Timing Adapter conectada entre el RAW to VIP Bridge y el Scaler II.....	51
figura 56, Configuración del bloque Scaler II.....	52
figura 57, Configuración de la IP Timing Adapter conectada al VIP to RAW Adapter.....	53
figura 58, Configuración de la IP VIP to RAW Adapter Bridge.....	53
figura 59, Configuración del RGB Resampler .....	54
figura 60, Configuración de la IP VGA Controller en la segunda opción.....	54
figura 61, Configuración de la IP 3-Wire SPI.....	56
figura 62, Configuración de la IP PIO para las interrupciones del touch.....	57
figura 63, Configuración de la IP PIO para indicar que el controlador del touch no está disponible .....	57
figura 64, Configuración de la IP PIO para el reset activo a nivel bajo del Display .....	58
figura 65, Señales de entrada/salida generadas por la IP PIO .....	58
figura 66, Configuración de la IP Avalon ALTPLL para el reloj de la pantalla táctil .....	59
figura 67, Configuración de la IP Timer .....	59
figura 68, Pestaña Signal Selection de la IP Generic Tri-State Controller .....	61
figura 69, Pestaña Signal Timing de la IP Generic Tri-State Controller .....	62
figura 70, Pestaña Signal Polarities de la IP Generic Tri-State Controller .....	62
figura 71, Configuración de la IP Parallel Port para los pulsadores anti-rebote .....	63
figura 72, Configuración de la IP Parallel Port para los LEDs Verdes .....	64
figura 73, Modificación del tcl de la IP Audio and Video Config .....	64

figura 74, Modificaciones a realizar en el fichero de auto-inicialización de la IP Audio and Video Config.....	65
figura 75, Ejemplo de un navegador que contiene un fichero con extensión .sdc.....	66
figura 76, TimeQuest Timing Analyzer en el menú Tools (izq.) y el símbolo de la barra de herramientas (dcha.) .....	66
figura 77, Barra de tareas del TimeQuest Timing Analyzer. Se debe pinchar sobre una de las opciones marcadas.....	67
figura 78, Ventana Derive Clock Uncertainty del TimeQuest.....	68
figura 79, Diagrama de bloques de la estructura general del software.....	69
figura 80, Configuración del System Clock Timer.....	70
figura 81, Ejemplo del navegador del proyecto software. ....	71
figura 82, Bloqueo de dirección de memoria en Qsys .....	72
figura 83, Diagrama de flujo de la escritura de la foto en el Buffer .....	76
figura 84, Diagrama de flujo de escritura en la tarjeta SD (izq.) y de escritura en la memoria flash (dcha.) .....	78
figura 85, Diagrama de flujo general del sistema.....	80
figura 86, Impacto en el rendimiento de la caché L1I.....	82
figura 87, Impacto en el rendimiento de la caché L1D.....	85
figura 88, Efectos de la caché L1 de Instrucciones y datos 1 .....	87
figura 89, Efectos de la caché L1 de instrucciones y datos 2 .....	88
figura 90, Efectos en el rendimiento del Predictor de Saltos con las cachés al mínimo valor .....	89
figura 91, Efectos en el rendimiento del Predictor de Saltos con las cachés al máximo valor .....	90
figura 92, Ejemplo de configuración del Patrón y el DMA para la prueba básica de SRAM .....	93
figura 93, Foto 1 de las pruebas con SRAM.....	94
figura 94, Foto 2 de las pruebas con SRAM.....	94
figura 95, Foto 3 de las pruebas con SRAM.....	95
figura 96, Información proveniente del datasheet de la IP SRAM Controller de Altera...96	
figura 97, Advertencia en el datasheet del controlador de la memoria SRAM acerca de su rendimiento .....	97
figura 98, Foto 4 de las pruebas con SRAM.....	97
figura 99, Foto 5 de las pruebas con SRAM.....	98
figura 100, Foto 6 de las pruebas con SRAM.....	98
figura 101, Foto 1 de las pruebas con SDRAM .....	99
figura 102, Foto 2 de las pruebas con SDRAM .....	100
figura 103, Foto 1 de las pruebas con memoria On-Chip.....	101
figura 104, Proceso de almacenado de un fotograma .....	103
figura 105, Resultado del Patrón de Test empleando la SRAM como memoria de vídeo en el sistema completo .....	104
figura 106, Patrón de Test (Arriba) e Imagen capturada por la cámara (Abajo).....	105
figura 107, Fragmento del informe de compilación de Eclipse.....	105

figura 108, Tiempos de almacenamiento/lectura de las Fotografías en las distintas memorias.....	106
figura 109, Controles sobre la placa .....	107
figura 110, Menú Home.....	108
figura 111, Menú Cámara .....	109
figura 112, Menú Galería.....	109
figura 113, Menú Config.....	110
figura 114, Menú Estado .....	110

## INDICE DE TABLAS

Tabla 1, Especificaciones del Sensor Micron MT9P001 .....	21
Tabla 2, Cambios en los registros de la cámara .....	23
Tabla 3, Pines del Display LCD .....	28
Tabla 4, Pines de la parte táctil de la pantalla.....	29
Tabla 5, Señales de entrada/salida generadas por la IP SDRAM Controller .....	39
Tabla 6, Señales de entrada/salida generadas por la IP Video-In Decoder .....	41
Tabla 7, Conexionado de la IP Video DMA Controller de escritura con Video Stream Router .....	45
Tabla 8, Conexionado de la IP Video DMA Controller de lectura.....	47
Tabla 9, Conexionado de la IP Video Dual-Clock Buffer .....	49
Tabla 10, Señales de entrada/salida generadas por la IP VGA Controller .....	49
Tabla 11, Conexionado de la IP VGA Controller .....	55
Tabla 12, Señales de entrada/salida generadas por la IP LT24 Controller.....	55
Tabla 13, Señales de entrada/salida generadas por la IP 3-Wire SPI.....	56
Tabla 14, Señales de entrada/salida generadas por la IP SD Card Interface .....	60
Tabla 15, Señales de entrada/salida generadas por la IP Tri-State Conduit Bridge.....	63
Tabla 16, Componentes con capacidad para enviar interrupciones a la CPU.....	70
Tabla 17, Ficheros que forman el software del sistema.....	72
Tabla 18, Ficheros importados desde "LT24 System CD" para el manejo de la pantalla táctil .....	74
Tabla 19, Efecto de la caché L1I en el rendimiento con diferencias porcentuales 1 .....	83
Tabla 20, Efecto de la caché L1I en el rendimiento con diferencias porcentuales 2 .....	83
Tabla 21, Diferencia de rendimiento entre un sistema sin L1I y una L1I = 64KB .....	84
Tabla 22, Efecto de la caché L1D en el rendimiento con diferencias porcentuales 1.....	85
Tabla 23, Efecto de la caché L1D en el rendimiento con diferencias porcentuales 2.....	85
Tabla 24, Diferencia de rendimiento entre un sistema con L1D: 512B y L1D: 16KB .....	86
Tabla 25, Diferencia de rendimiento entre un sistema con L1D: 512B y L1D: 4KB .....	86
Tabla 26, Efectos del Predictor de Saltos en el rendimiento con cachés 0 + 0,512B 1.....	90
Tabla 27, Efectos del Predictor de Saltos en el rendimiento con cachés 0 + 0,512B 2.....	90
Tabla 28, Efectos del Predictor de Saltos en el rendimiento con cachés 64KB + 16KB 1 ..91	
Tabla 29, Efectos del Predictor de Saltos en el rendimiento con cachés 64KB + 16KB 2 ..91	
Tabla 30, Diferencias de rendimiento entre un sistema con un Predictor de Saltos Estático y uno Dinámico de 8192 posiciones con una L1I = 0KB y una L1D = 512B.....	91

Tabla 31, Diferencias de rendimiento entre un sistema con Predictor de Saltos Estático y uno Dinámico de 8192 posiciones con una L1I = 64KB y una L1D = 16KB .....	91
Tabla 32, Configuración para la prueba 1 de la SRAM .....	93
Tabla 33, Configuración para la prueba 2 de la SRAM .....	94
Tabla 34, Configuración para la prueba 3 de la SRAM .....	95
Tabla 35, Configuraciones disponibles empleando la memoria SRAM.....	99
Tabla 36, Resumen de las configuraciones de memoria de vídeo disponibles.....	101

## RESUMEN DEL TRABAJO

El presente trabajo describe el desarrollo de una cámara de fotos digital con pantalla táctil sobre la placa DE2-115 de Terasic. El sistema permite la captura de fotografías a una resolución de 320 x 240 píxeles y su posterior almacenamiento en diversos tipos de memoria como memoria Flash o una tarjeta SD.

Sobre la FPGA se implementa el procesador NIOS II de Altera sobre el cual corre el software que controla la pantalla táctil y la gestión de la memoria para la captura y visualización de las imágenes. El resto del Hardware empleado consiste por una parte en las IPs (Intellectual Property) del programa universitario (University Program) de Altera, y por otra en el uso de IPs genéricas proporcionadas por Altera.

## ABSTRACT

The work described in this document contains the development of a touch screen controlled digital photo camera using the DE2-115 board from Terasic. The system allows photo capturing with a resolution of 320 x 240 pixels and is able to save the captured images on diverse type of memories, like Flash memory or an SD card.

The software that controls the touch screen and memory management for image capturing runs on a NIOS II processor from Altera, implemented on the FPGA. The rest of the Hardware uses either IPs (Intellectual Properties) inside Altera's University Program or generic IPs provided by Altera.

## 1- INTRODUCCIÓN

El proyecto pretende explorar las posibilidades de diseño a la hora de implementar la funcionalidad de cámara de fotos con pantalla táctil en la placa de desarrollo DE2-115 de Terasic.

Para ello, se cuenta con el siguiente hardware:

- Placa de desarrollo Terasic DE2-115
- Módulo de cámara Terasic D5M
- Pantalla Táctil Terasic LT24
- Tarjeta de expansión Terasic GPIO-HSMC

La finalidad es que el hardware diseñado en la FPGA emplee los bloques prediseñados por parte de Altera de forma que no sean necesarios diseños de IPs en Verilog, al menos para la estructura básica de la cámara. Se implementa también el procesador Nios II de Altera en la FPGA, de forma que se cuenta con un microprocesador capaz de ejecutar software programado en C para el desarrollo de las diferentes funcionalidades del sistema.

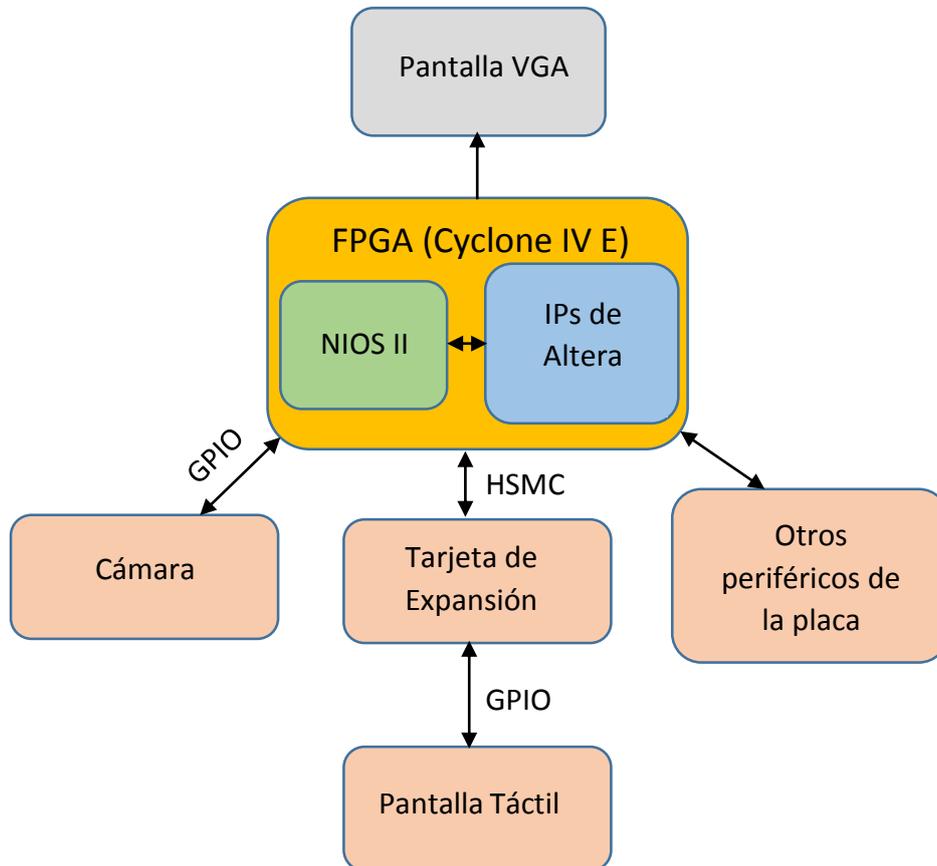


figura 1, Diagrama de bloques básico del sistema

De esta manera, se pretende que la visión de la cámara (Live View, lo que el sensor de la cámara está viendo en un momento determinado) sea mostrada en una pantalla VGA en tiempo real mientras el sistema es controlado a través de la pantalla táctil.

## 1.1 ESTADO DEL ARTE

Terasic dispone en el mercado de una serie de accesorios como complemento para sus placas de desarrollo. Como norma general sus placas de la serie “DE” ya contienen la circuitería necesaria para el mercado al que van destinadas que es el mercado educativo, de forma que lo habitual es que se empleen al natural, sin dispositivos adicionales.

Sin embargo, es interesante analizar la clase de funcionalidad alcanzable cuando estas placas de desarrollo se acompañan de una serie de complementos. Como se ha mencionado previamente en este trabajo se emplean tres de los complementos disponibles: Cámara de 5MP, Pantalla táctil de 2,4” y tarjeta de expansión GPIO. Adicionalmente a estos tres accesorios a la placa DE2-115 se le pueden insertar los siguientes accesorios agrupados por categoría:

### MULTIMEDIA

- Receptor HDMI
- Transmisor HDMI
- Tarjeta adaptadora HDMI->HSTC
- Tarjeta adaptadora HSMC->SDI

### CONVERSION DE INTERFACES

- Tarjeta adaptadora HSMC->SATA/SAS
- Tarjeta adaptadora HSMC->XTS
- Tarjeta de conversión de datos AD/DA
- Tarjeta de alta velocidad AD/DA

### VIDEO E IMAGEN

- Módulo LCD Multitouch 7”
- Módulo LCD Multitouch con Módulo de Cámara
- Tarjeta de adaptación de sensor Aptina HSMC->AHA
- Tarjeta receptora de enlace de cámaras HSMC->CLR
- Tarjeta de conversión HSMC->DVI

### REDES

- Tarjeta adaptadora HSMC->DUAL XAUI y DUAL SFP+
- Tarjeta adaptadora HSMC->ICB Altera
- Tarjeta de comunicaciones HSMC
- Tarjeta adaptadora HSMC->ETHERNET
- Tarjeta adaptadora HSMC->SFP

### RADIOFRECUENCIA

- Módulo Bluetooth BTS-TMD
- Tarjeta adaptadora HSMC->ARRADIO

La placa DE2-115 es compatible con todos esos accesorios, los cuales como se puede observar abarcan un amplio abanico de posibilidades para la FPGA. Sin embargo, a pesar de ser compatibles, muchos de esos accesorios están más destinados a otro tipo de placas de desarrollo para entornos más profesionales, generalmente empleando FPGAs de la familia Arria/Stratix en lugar de la Cyclone IV-E incorporada en la placa DE2-115.

Puesto que no es demasiado común ver placas de propósitos educativos empleando este tipo de accesorios la documentación y experiencia en el empleo de los mismos es bastante escasa, y búsquedas por internet de experiencias de otras personas suele ser un buen punto de partida a la hora de decidir si adquirir uno de estos accesorios o no.

En este caso el uso de la pantalla LT24 está bien documentado por Terasic gracias a una serie de librerías software que facilitan el trabajo junto con la IP controladora de la pantalla. No es este el caso de la cámara D5M, donde la documentación aportada por parte de Terasic sobre el funcionamiento es bastante escasa y existen también muy pocas experiencias en la red acerca de la misma, especialmente si se desea utilizar con resoluciones diferentes a la estándar o modos diferentes de funcionamiento del sensor. Por ello este trabajo abarca un estudio en profundidad sobre las posibilidades de uso de estos tres accesorios en un único proyecto, de forma que se exploran las limitaciones y las correctas utilizaciones de los mismos dentro de las posibilidades de la placa DE2-115.

Terasic dispone otra serie de placas destinadas a propósitos educativos, en cuya venta suele incluir descuentos cuando la compra la realiza una organización educativa. Otras placas disponibles (y también compatibles con los accesorios arriba listados) son las siguientes:

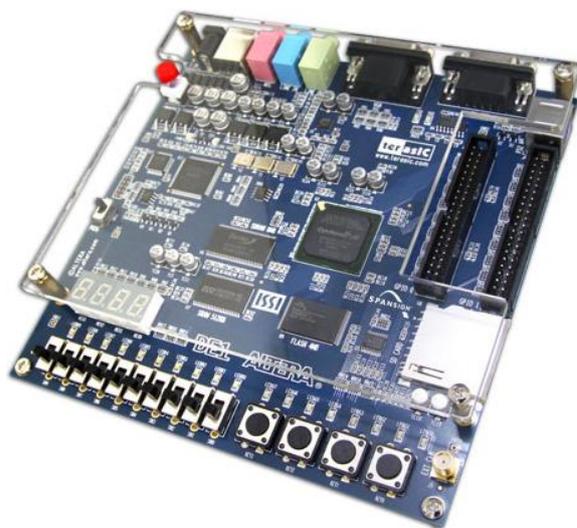


figura 2, Placa de desarrollo Altera DE1

#### Altera DE1

- Cyclone II 2C20 con 20K LEs
- 8MB SDRAM
- 4MB Flash
- 512KB SRAM
- Ranura para tarjetas SD
- 4 Pulsadores
- 8 Conmutadores
- 8 LEDs Verdes y 10 Rojos
- 4 Displays 7 Segmentos
- CODEC de Audio
- Salida VGA
- Transceptor RS232
- Entrada PS2
- 2x Conectores GPIO de 40 pines

### Altera DE0

- Cyclone III 3C16 con 15,4K LEs
- 8MB SDRAM
- 4MB Flash
- Ranura para tarjetas SD
- 3 Pulsadores
- 10 Conmutadores
- 10 LEDs Verdes
- 4 Displays 7 Segmentos
- Interfaz para una pantalla LCD 16x2 (No incluida)
- Salida VGA
- Puerto RS232 y PS2
- 2x Conectores GPIO de 40 pines

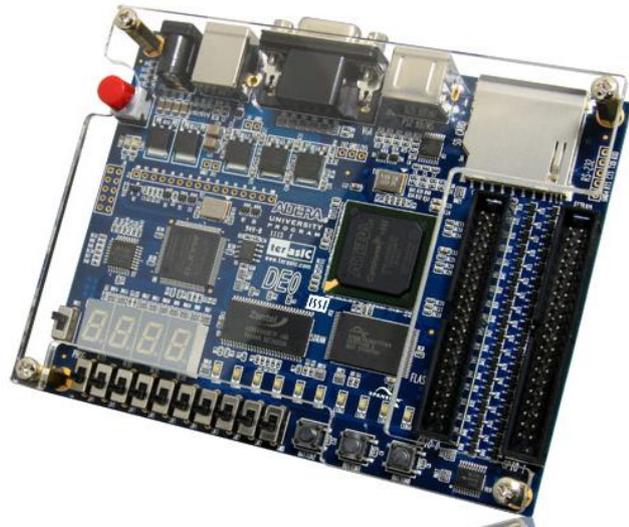


figura 3, Placa de desarrollo Altera DE0

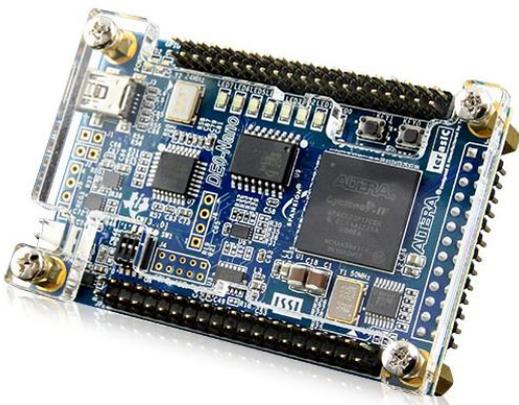


figura 4, Placa de desarrollo Altera DE0-Nano

### Altera DE0-Nano

- Cyclone IV EP4CE22F17C6N con 22,3K Les
- 32MB SDRAM
- 2Kb EEPROM
- 8 LEDs Verdes
- 2 Pulsadores
- 4 Conmutadores DIP
- Acelerómetro de 3 ejes
- ADC de 12 bits y 8 canales
- Tamaño compacto



figura 5, Placa de desarrollo Altera DE2i-150

### Altera DE2i-150

#### FPGA

- Cyclone IV GX EP4CGX150DF31 con 149K LEs
- 128MB SDRAM
- 4MB SSRAM
- 64MB Flash
- Ranura para tarjetas SD
- 4 Pulsadores
- 18 Conmutadores
- 9 LEDs Verdes y 18 Rojos
- 8 Displays 7 Segmentos
- Salida VGA
- Puerto RS232
- Conector GPIO de 40 pines, y conector HSMC para expansión adicional
- Entrada de TV y receptor IR para mando a distancia
- Puerto Ethernet
- 3x Entradas de reloj externo y Conectores SMA para salida de reloj

#### CPU

- Intel Atom N2600 (Dual-Core 1,6GHz, 1MB L2)
- Conector SO-DIMM DDR3
- Salida VGA y HDMI 1.3a
- Chipset Intel NM10 Express
- Conectividad Intel Centrino Wireless-N 135:
  - 802.11b/g/n
  - Bluetooth 4.0
  - Wi-Fi Direct
- CODEC de Audio
- Controlador Ethernet Intel 82583V GbE
- 3 LEDs de estado
- Conector de alimentación para disco duro
- Buzzer
- Conector Mini-PCIe
- Cabecera mSATA
- Limitador de corriente para el puerto USB

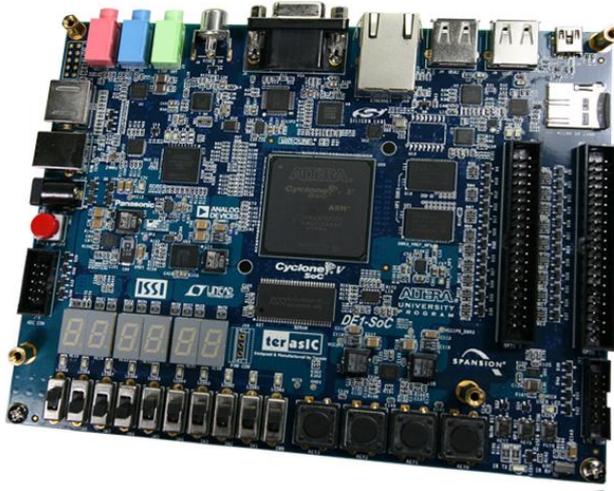


figura 6, Placa de desarrollo Altera DE1 SoC

### Altera DE1-SoC

- Cyclone V SE 5CSEMA5F31C6 con 85k LEs
- ARM Cortex A9 Dual-Core a 925MHz
- 64MB SDRAM
- 1GB DDR3 (Para la CPU ARM)
- Interfaz para tarjeta SD (Para la CPU ARM)
- 4 Pulsadores
- 10 Conmutadores
- 11 LEDs Rojos
- 7 Displays 7 Segmentos
- Doble puerto USB 2.0
- Conector Ethernet, PS2 y Emisor/Receptor de IR para mando a distancia
- 2 Conectores GPIO de 40 pines
- 1 Cabecera para entrada al ADC
- 1 Conector LTC
- Salida VGA
- CODEC de Audio

### Altera DE0-CV

- Cyclone V SE 5CEBA4F23C7N con 49K LEs
- 64MB SDRAM
- 10 LEDs Rojos
- 4 Pulsadores
- 10 Conmutadores
- 6 Displays 7 Segmentos
- Interfaz para tarjeta SD
- Salida VGA de 4 bits
- 2x Conectores GPIO de 40 pines
- Conector PS2
- Tamaño Compacto

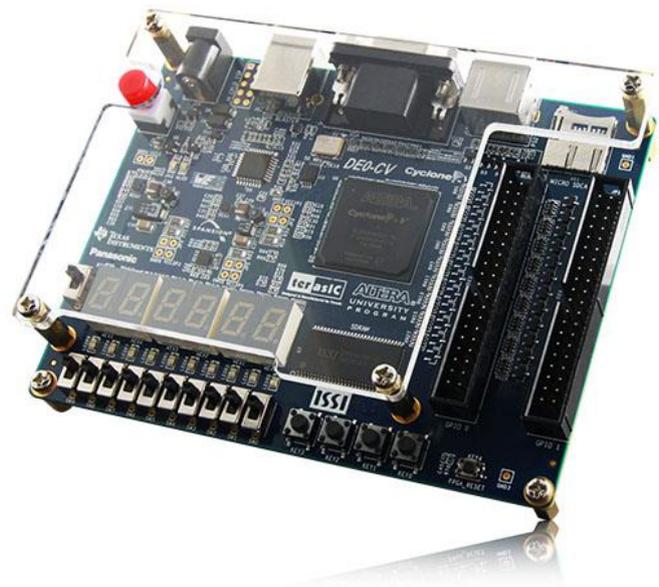


figura 7, Placa de desarrollo Altera DE0-CV

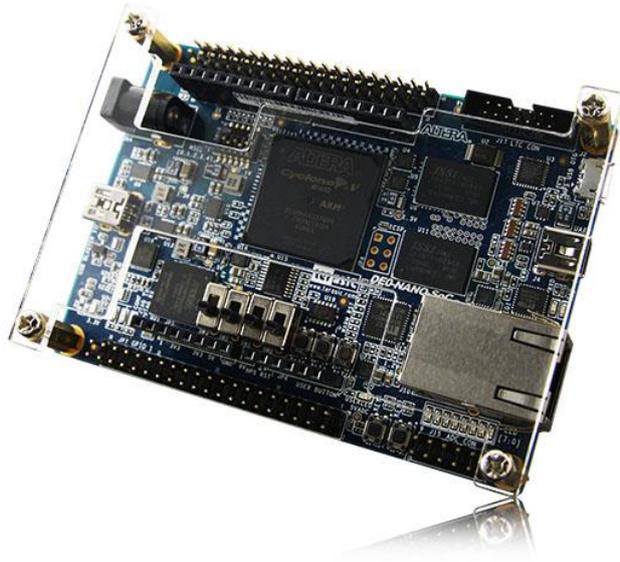


figura 8, Placa de desarrollo Altera DE0-Nano-Soc/Atlas SoC Kit

#### Altera DE0-Nano-Soc/Atlas SoC Kit

- Cyclone V SE 5CSEMA4U23C6N con 40K LEs
- ARM Cortex A9 Dual-Core a 925MHz
- 1GB DDR3 (Para el ARM)
- 2 Pulsadores
- 4 Conmutadores
- 8 LEDs Verdes
- 2 Cabeceras de 40 pines
- 1 Cabecera para Arduino
- 1 Entrada Analógica de 10 pines
- 1 ADC
- 1 Ranura para tarjetas SD (Para el ARM)
- Acelerómetro
- Conector USB
- Conector Ethernet
- Cabecera de expansión LTC 2x7 pines (Para el ARM)

La primera distinción entre una placa de desarrollo con propósitos educativos y una destinada a desarrollo de mayor nivel es, por una parte el ya mencionado descuento cuando estas placas son compradas por universidades y centros educativos, y, por otra parte, cabe mencionar que estas placas todas emplean memoria SDRAM como memoria de acceso para la FPGA (A excepción de la DE0-Nano-Soc/Atlas SoC Kit que no tiene ningún tipo de memoria conectada a la FPGA salvo la memoria EPCS128 de programación). Cualquiera de las otras placas de Terasic dispone de memorias mucho más potentes, generalmente DDR3 y en algún caso DDR2 o LPDDR2. Eso es un claro indicativo de que unas placas tienen la FPGA destinada a aplicaciones de alto rendimiento mientras que otras parecen más destinadas a aplicaciones de menor potencia y también que las placas sean económicamente más asequibles. También las placas destinadas al mundo educativo tienen en general un mayor número de periféricos de todo tipo (salida VGA, CODEC de Audio, Ethernet, RS232, PS2, etc.) mientras que aquellas no destinadas a propósitos educativos tienen periféricos más especializados como por ejemplo una placa destinada solo a vídeo, una placa destinada sólo a comunicaciones, etc.

También existen placas destinadas al mundo no educativo que tienen un diseño a modo de tarjeta con interfaz PCIe, claramente pensadas para el desarrollo de sistemas que serán integrados en un PC o en cualquier otro tipo de equipo con conector PCIe.

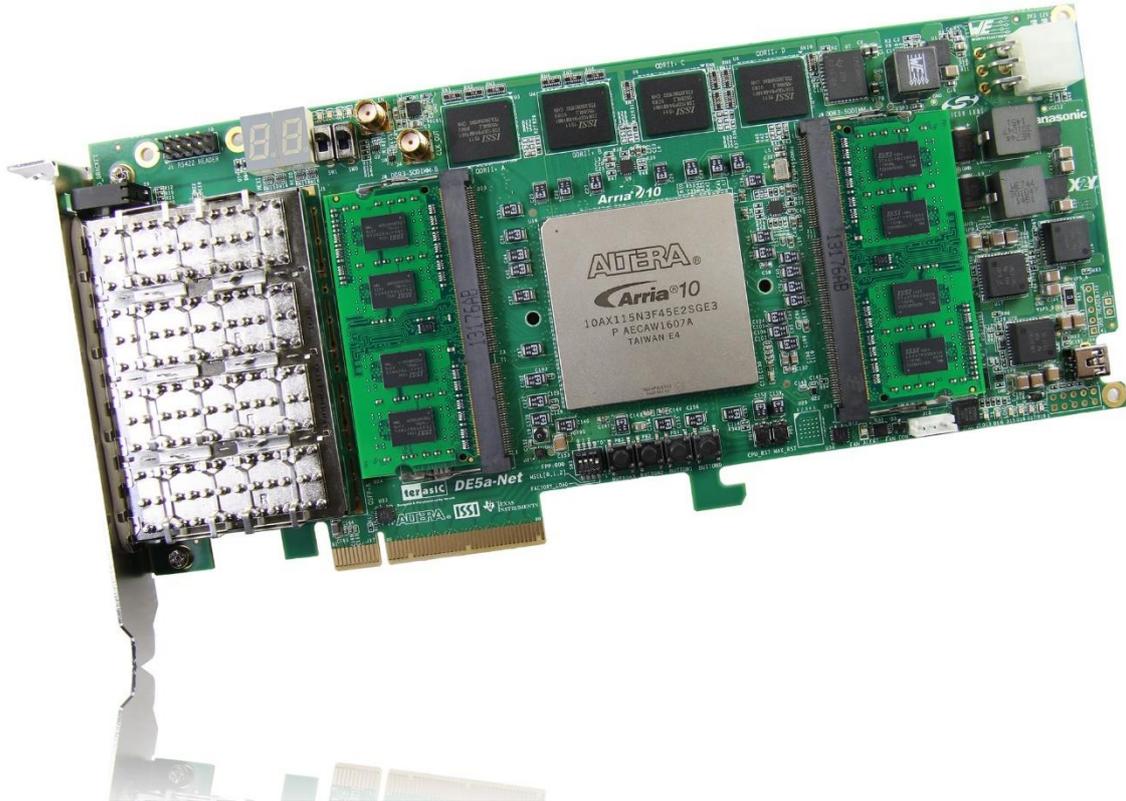


figura 9, Ejemplo de una placa de desarrollo profesional, destinada a aplicaciones demandantes de alta capacidad de computación y alto ancho de banda, en formato de tarjeta PCIe. La FPGA Arria 10 mostrada en la imagen tiene 1.150.000 LEs, la memoria RAM es manualmente intercambiable empleando ranuras SODIMM además de incorporar 36MB de memoria QDRII+ SRAM.

## 1.2 OBJETIVOS

- Conseguir un Proyecto funcional empleado el módulo de cámara facilitado.
- Utilizar las IPs ya disponibles
- Emplear la Pantalla Táctil facilitada como control del sistema.
- Utilizar las memorias no volátiles disponibles (Flash y SD) como almacenamiento para las fotografías.
- Estudiar las limitaciones del sistema en cuanto a capacidad.
- Estudiar las limitaciones del sistema en cuanto a velocidad
- Aprovechar los diferentes recursos/funcionalidades disponibles en la placa para conseguir el mejor resultado final.

Este proyecto está enfocado al ámbito educativo (Prácticas, Trabajos, etc) no está orientado al diseño/venta de un producto final.

## 2- ANÁLISIS DE LOS DISPOSITIVOS EMPLEADOS Y SU FUNCIONALIDAD

En este apartado se describen los tres dispositivos empleados y que forman el núcleo del trabajo. Se pretende analizar el nivel de funcionalidad que se puede alcanzar empleando estos dispositivos y explorar la complejidad a la que se puede llegar empleando la placa DE2-115.

### 2.1 PLACA DE DESARROLLO DE2-115

La placa de desarrollo DE2-115 es una placa diseñada por Altera con propósitos académicos y fabricada y distribuida por Terasic. El diseño gira en torno a una FPGA de la familia Cyclone IV E de Altera, concretamente el chip tope de gama dentro de esa familia, el cual dispone de 115K elementos lógicos, 486KB de memoria distribuida y 266 multiplicadores de 18 x 18 bits.

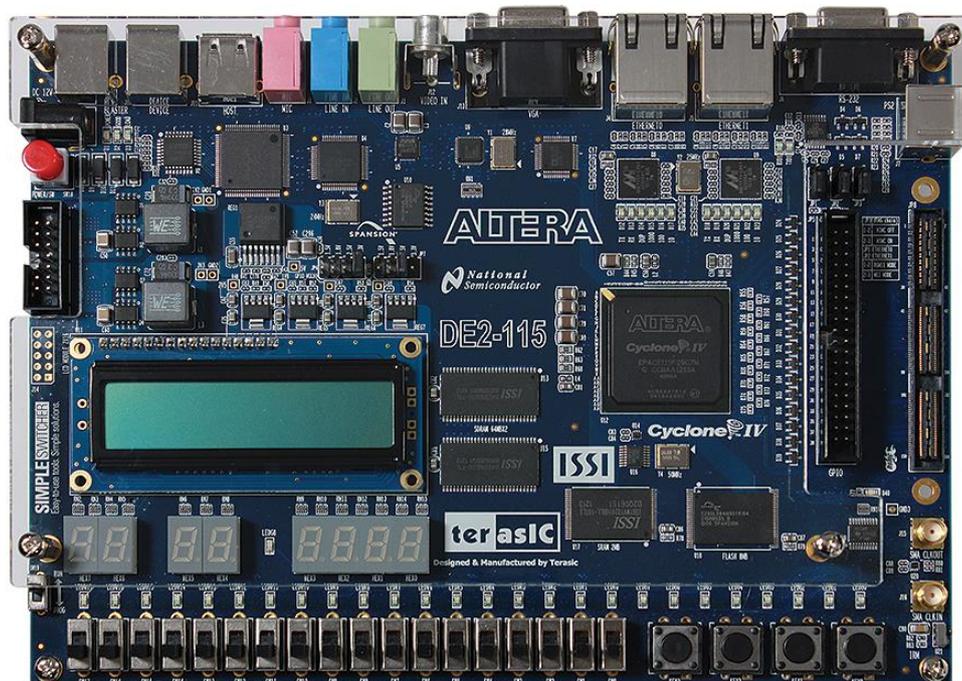


figura 10, Placa de desarrollo Terasic DE2-115

Para dotar de versatilidad al conjunto, la placa dispone de:

- 18 Conmutadores
- 4 Pulsadores
- 18 LEDs rojos
- 9 LEDs verdes
- 2 Conexiones Ethernet
- 1 Conexión USB (aparte del USB adicional para la programación)
- 2MB de Memoria SRAM
- 8MB de Memoria Flash

- 128MB de Memoria SDRAM
- 8 Displays 7-Segmentos
- 1 Display LCD de 40 caracteres
- 1 Codec de Audio
- 1 Conexión para pantalla VGA
- 1 Entrada de Vídeo Analógico
- 1 Entrada RS232
- 1 Entrada PS2
- 1 Ranura para tarjetas SD
- 1 GPIO en placa
- 1 Ranura de Expansión HSMC
- 1 Entrada y 1 Salida para Clock Externo vía conector SMA

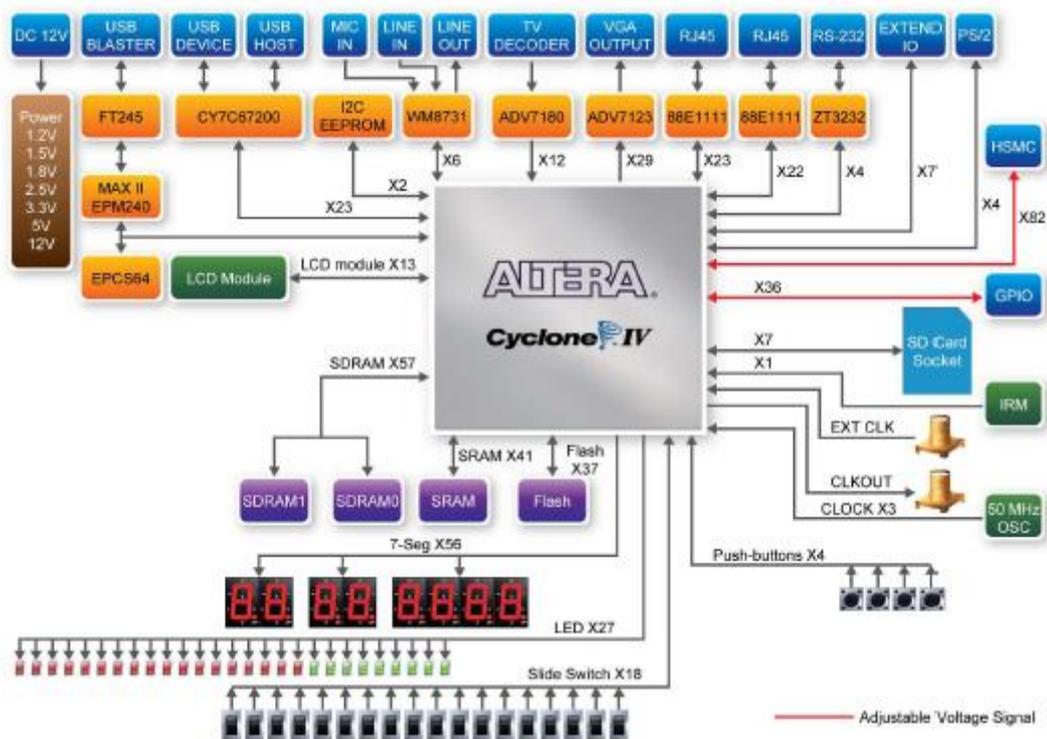


figura 11, Diagrama de bloques de la placa DE2-115 facilitado por Terasic

La cámara se conecta al conector al GPIO de la placa mientras que para emplear la pantalla táctil a la vez que la cámara, debido a que en placa solo hay 1 único conector GPIO, se debe insertar la tarjeta de expansión en el conector HSMC para disponer de hasta 3 GPIOs adicionales.

## 2.2 CAPTURA DE IMÁGENES

El corazón de este proyecto está en la toma de fotografías empleando la cámara D5M de Terasic.



figura 12, Cámara D5M de Terasic

La cámara consiste en un sensor Micron MT9P001 con un tamaño de 1/2,5" el cual es un sensor con un array de 5MP. Es un sensor de hace 10 años orientado por entonces a productos de telefonía móvil y cámaras de foto compactas y de gama baja. Sus especificaciones se muestran en la siguiente tabla:

Parámetro	Valor
Formato	4:3 (1/2,5")
Tamaño de imagen activo	5,7mm (H) x 4,28mm (V)
Píxeles activos	2592H x 1944V
Tamaño de píxel	2,2 x 2,2 $\mu$ m
Array de Filtro de Color	Patrón Bayer RGB
Tipos de Obturación	Global Reset Release (GRR) Solo Imagen Estática Electronic Rolling Shutter (ERS)
Máxima tasa de datos	96 Mp/s a 96MHz (2,8V I/O)
Tasa de Frames	5MP: 15 fps, VGA: 70 fps
Resolución del ADC	12 bits
Rango dinámico de píxel	70,1dB
Relación señal a ruido (SNR <sub>max</sub> )	38,1dB
Voltaje de funcionamiento	1,7 – 3,1V

<b>Consumo</b>	381mW a 15 fps con 5MP
<b>Temperatura de Operación</b>	-30°C a 70°C

Tabla 1, Especificaciones del Sensor Micron MT9P001

El fabricante facilita un diagrama de bloques del sensor:

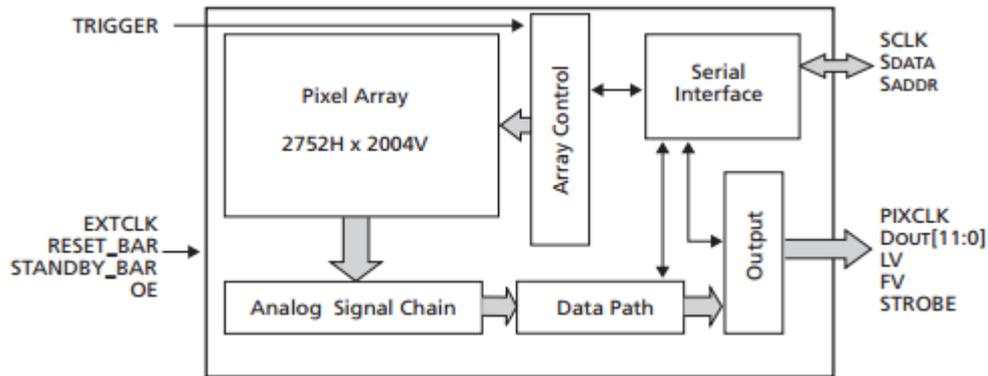


figura 13, Diagrama de Bloques del sensor Micron MT9P001

Se puede observar como en la imagen el array de píxeles tiene unos valores mayores a los anteriormente mencionados. Esto se debe a que el sensor dispone de píxeles redundantes en el borde del cuadro que no forman parte de la imagen. A la zona formada por estos píxeles se le denomina Región Frontera o Boundary Region y está compuesta por una serie de filas e imágenes con píxeles negros:

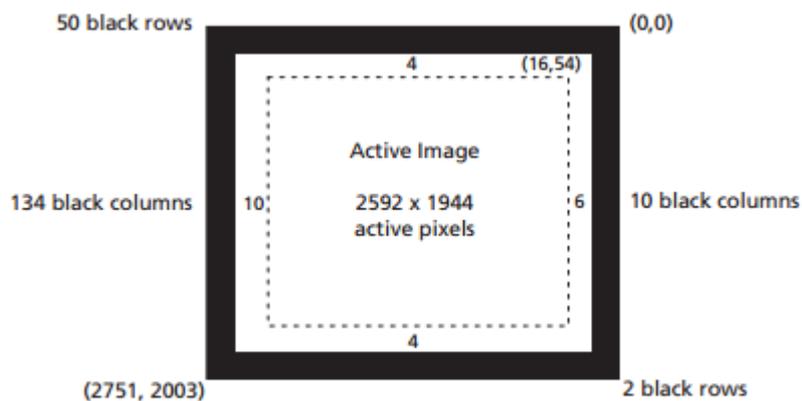


figura 14, Región Frontera o Boundary Region del Sensor (Bordes) y Región de Imagen activa (Región Central)

Cuando se realiza una lectura del sensor este solamente proporciona los datos de los píxeles de la llamada zona de imagen activa, que es la que contiene los 5 MegaPíxeles (2592 x 1944).

Por otra parte, el sensor utiliza un filtro de color con patrón Bayer. Esto es un filtro de color que hace que cada píxel del array del sensor solamente capture información relativa a un color, ya sea Verde, Rojo o Azul.

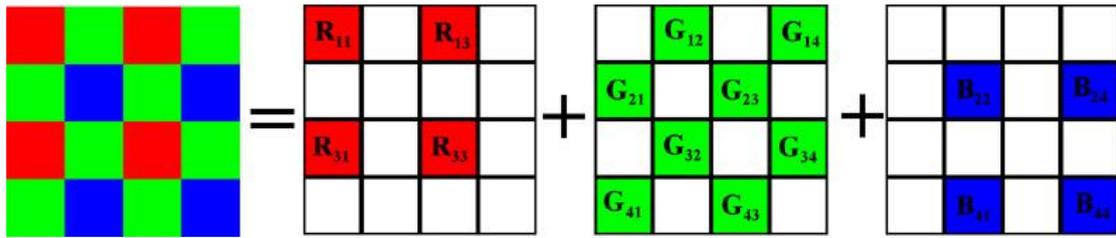


figura 15, A la izquierda el Patrón Bayer RGB y a la derecha la descomposición de los tres colores

El patrón Bayer hace que los píxeles de una fila y una columna sean siempre el patrón Rojo-Verde y la siguiente fila y columna el patrón Azul-Verde. Esto repercute en disponer el doble de píxeles de color verde que rojos y azules. El motivo de esta relación es que el ojo humano es más sensible al color verde que al rojo y al azul y se pretende “simular” de esta manera el comportamiento de la visión humana.

Que cada píxel en lugar de tres posibles colores tenga solo 1 produce la necesidad de realizar un procesado para obtener la imagen final, comúnmente denominado “Demosaicing”. Este procesado, en su versión más simple toma la información de 4 píxeles (2 Verdes, 1 Azul y 1 Rojo) y calcula el valor de los tres componentes de color de cada píxel en base al valor del píxel Azul, el Rojo y la media aritmética de los 2 valores de verde:

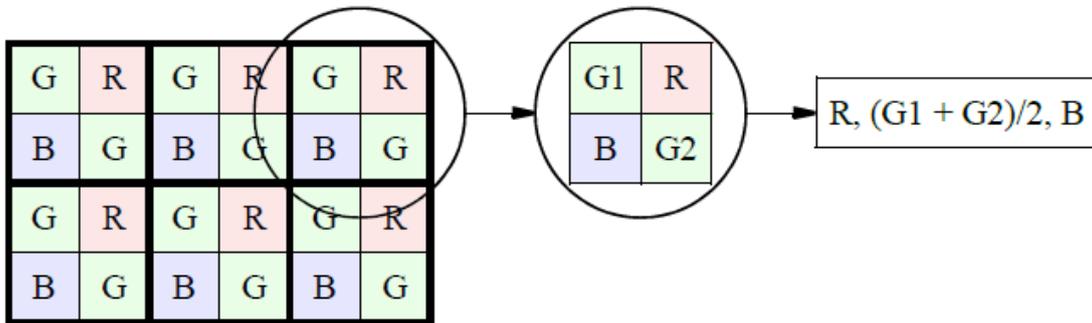


figura 16, Conversión Bayer a RGB de la IP proporcionada por Terasic

Este procesado conlleva la reducción de la resolución inicial a la mitad, de forma que si se está usando el sensor con una resolución de 2592 x 1944 la imagen obtenida tras el procesado será una imagen de 1296 x 972.

Por otra parte, el sensor cuenta con una serie de registros de configuración que permiten adaptar el funcionamiento del sensor de forma óptima para la aplicación a la que se quiere destinar. Los registros se configuran vía comunicación I2C.

En el caso concreto de este trabajo no ha sido necesario alterar el valor de muchos de los registros, aquellos que se han cambiado se muestran a continuación:

Registro	Dirección	Valor por defecto (En University Program)	Valor actual
<b>Column Start (RW)</b>	R0x002	0x0010	0x0018
<b>Shutter Width Lower (SW)</b>	R0x009	0x0797	0x027A
<b>Read Mode 2 (RW)</b>	R0x020	0xC000	0xC020

*Tabla 2, Cambios en los registros de la cámara*

Ya que la resolución nativa del sensor son 5MP en formato Bayer, es decir, 2592 x 1944 en formato Bayer o 1296 x 972 en formato RGB, surge el problema de que, de forma natural, se realizaría un zoom sobre la imagen captada por el sensor al utilizar únicamente la parte central del sensor. Es decir, cuando la cámara trabaja con la resolución nativa el sensor utiliza todos sus píxeles para formar la imagen mientras que a una resolución inferior emplearía únicamente aquellos píxeles necesarios para la resolución deseada, generalmente los de la parte central. De esta forma la imagen final estaría compuesta únicamente por la información de la región central del sensor, cuyo efecto es similar al de recortar una imagen tomada a resolución nativa.

Este efecto se da en cámaras de fotos de forma habitual, a igualdad de distancia focal en un determinado objetivo si el sensor es más pequeño el resultado es la distancia focal del objetivo multiplicado por el factor recorte del sensor.

Por ejemplo, suponiendo un objetivo para cámaras de sensor Full Frame (Formato Completo) con una distancia focal de 23mm, puesto que el objetivo y el sensor son ambos de formato completo, en una cámara de esas características la distancia focal final sería de 23mm. Sin embargo, si ese mismo objetivo se inserta en una cámara con sensor APS-C (factor recorte 1,5x o 1,6x), la distancia focal de ese mismo objetivo con este sensor sería de:  $23\text{mm} \times 1,5 = 34,5\text{mm}$ . La distancia focal de un objetivo siempre es relativa a un sensor Full Frame, por lo tanto, incluso si el objetivo fuese para sensores APS-C seguiría anunciando una distancia focal de 23mm que unida al sensor formarían una distancia focal de 34,5mm. En este último caso, si ese objetivo orientado a cámaras APS-C se insertara en una de formato completo el resultado sería que solamente cubriría una parte del sensor de formato completo y por lo tanto el resultado sería el mismo que en una cámara APS-C, con el inconveniente de que al cubrirse solo una región del sensor Full Frame la imagen capturada tendría una resolución menor, por ejemplo, serían 13MP en un sensor Full Frame de 20MP.

En la siguiente imagen se muestra un ejemplo de este fenómeno:

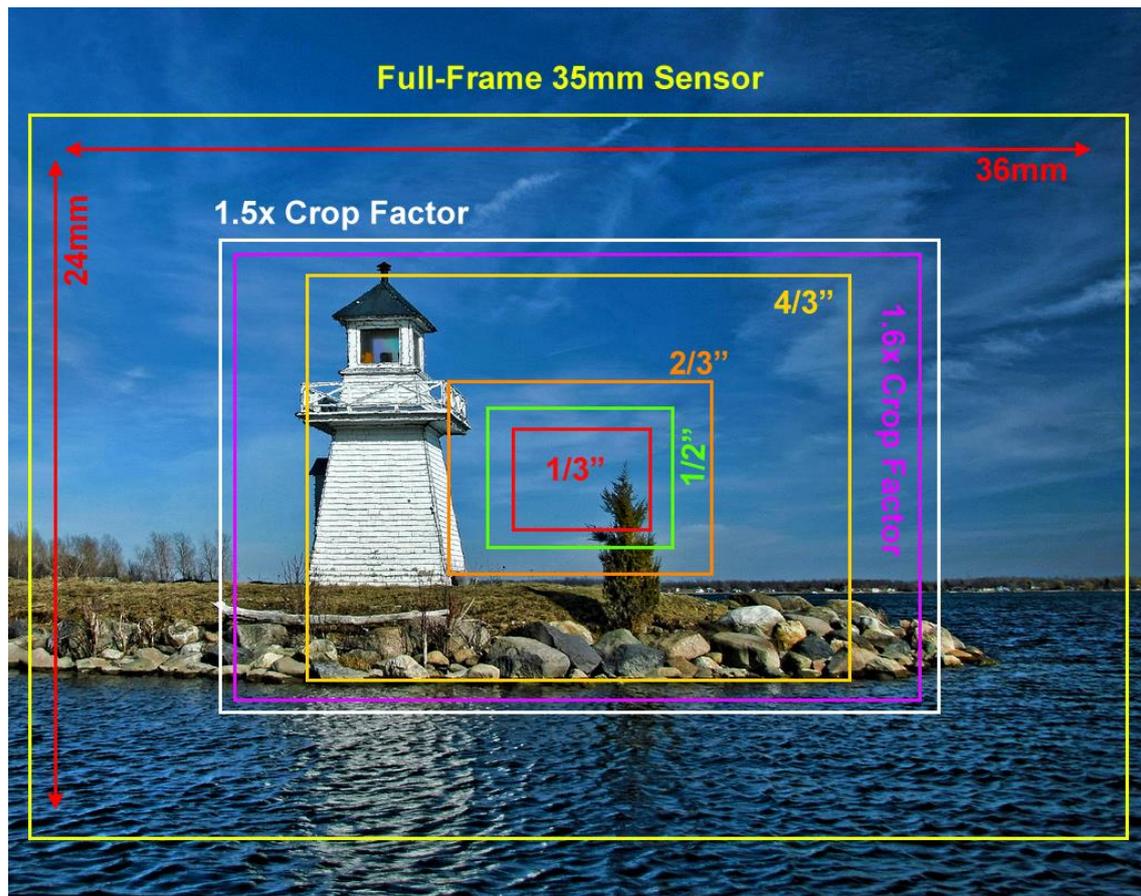


figura 17, Diferentes factores de recorte en función del tamaño de sensor

En el caso de la cámara D5M no hay ningún cambio de objetivo, pero disminuir la resolución produce el mismo efecto que el de disponer de un sensor de menor tamaño: Un factor recorte.

Sin embargo, el sensor dispone de una funcionalidad que permite mantener la misma distancia focal aun habiendo reducido la resolución. La funcionalidad se denomina pixel skipping y permite que en lugar de concentrar los píxeles en la zona central del sensor se lea el sensor completo dando saltos entre píxeles. Es decir, la imagen se lee desde el principio del sensor, de la misma forma que se haría a su resolución nativa, pero en lugar de leer todos los píxeles se leen algunos y se saltan otros.

El sensor solo permite unos modos determinados y permite elegir si se desean saltar solo filas, solo columnas, o ambas cosas.

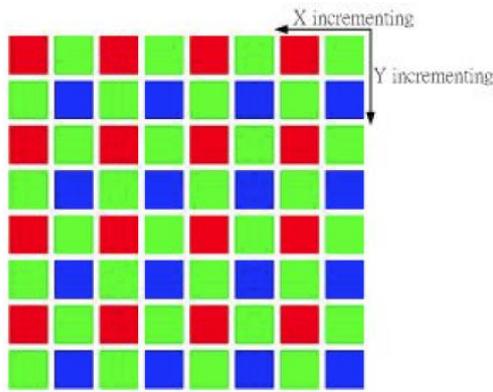


Figure 3.3: Pixel Readout (no skipping)

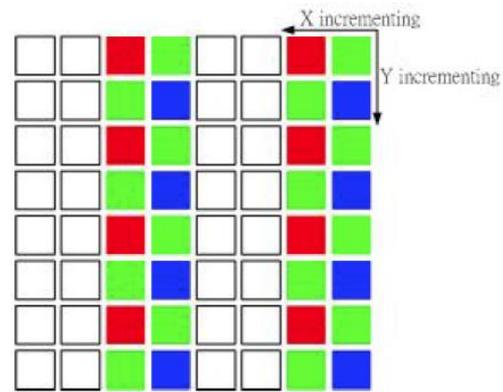


Figure 3.4: Pixel Readout (Column Skip 2X)

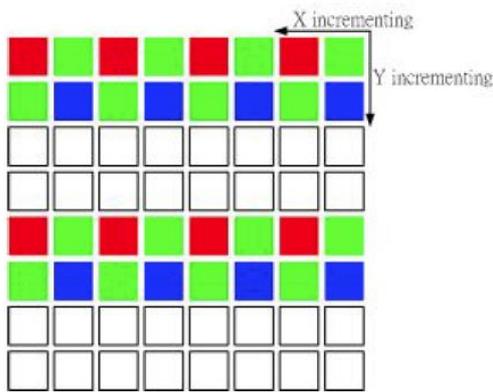


Figure 3.5: Pixel Readout (Row Skip 2X)

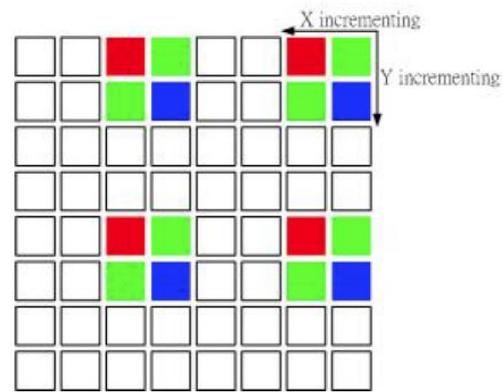


Figure 3.6: Pixel Readout (Column Skip 2X, Row Skip 2X)

figura 18, Diferentes modos de Pixel Skipping o Saltos de Píxeles

El sensor también incorpora una vuelta de tuerca a esta funcionalidad denominada Pixel Binning. Este es un caso particular del Pixel Skipping que permite la lectura de filas y columnas adicionales cercanas a aquellas que si se leen al realizar los saltos. Los valores se interpolan de forma que se pueden conseguir transiciones más suaves entre píxeles. También permite mejorar el rendimiento con baja luz. Este es el modo empleado en este trabajo.

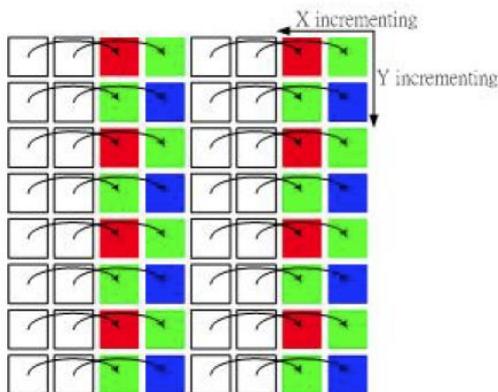


Figure 3.7: Pixel Readout (Column Bin 2X)

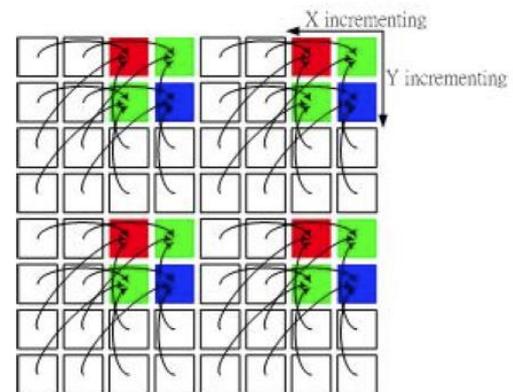


Figure 3.8: Pixel Readout (Column Bin 2X, Row Bin 2X)

figura 19, Diferentes opciones para la realización del binning

El uso del binning produce una menor tasa de frames dado que se leen más píxeles que en caso de hacer solo skipping pero mejora el rendimiento en condiciones de poca luz. El fabricante señala en el datasheet del sensor las tasas de frames por segundo alcanzables en función de la resolución y el modo de funcionamiento:

Resolution	Frame Rate	Sub-sampling Mode	Column_Size (R0x04)	Row_Size (R0x03)	Shutter_Width_Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column_Bin (R0x23 [5:4])	Column_Skip (R0x23 [2:0])
2592 x 1944 (Full Resolution)	15.15	N/A	2591	1943	<1943	0	0	0	0
2,048 x 1,536 QXGA	23	N/A	2047	1535	<1535	0	0	0	0
1,600 x 1,200 UXGA	35.2	N/A	1599	1199	<1199	0	0	0	0
1,280 x 1,024 SXGA	48	N/A	1279	1023	<1023	0	0	0	0
	48	skipping	2559	2047		0	1	0	1
	40.1	binning	2559	2047		1	1	1	1
1,024 x 768 XGA	73.4	N/A	1023	767	<767	0	0	0	0
	73.4	skipping	2047	1535		0	1	0	1
	59.7	binning	2047	1535		1	1	1	1
800 x 600 SVGA	107.7	N/A	799	599	<599	0	0	0	0
	107.7	skipping	1599	1199		0	1	0	1
	85.2	binning	1599	1199		1	1	1	1
640 x 480 VGA	150	N/A	639	479	<479	0	0	0	0
	150	skipping	2559	1919		0	3	0	3
	77.4	binning	2559	1919		3	3	3	3

Resolution	Frame Rate	Sub-sampling Mode	Column_Size (R0x04)	Row_Size (R0x03)	Shutter_Width_Lower (R0x09)	Row_Bin (R0x22 [5:4])	Row_Skip (R0x22 [2:0])	Column_Bin (R0x23 [5:4])	Column_Skip (R0x23 [2:0])
1,920 x 1,080 HDTV	34.1	N/A	1919	1079	<1079	0	0	0	0
1,280 x 720 HDTV	67.6	N/A	1279	719	<719	0	0	0	0
	67.6	skipping	2559	1439	<719	0	1	0	1
	56.4	binning	2559	1439	<719	1	1	1	1

figura 20, Configuración del sensor en función de la resolución

También un cambio en la resolución puede suponer la necesidad de cambiar el tiempo de exposición cambiando el valor de “Shutter Width Lower”, donde un valor menor supone un tiempo de exposición menor.

### 2.3 PANTALLA TÁCTIL

Junto con la captura de imágenes otro detalle importante del trabajo está en el uso de una pantalla táctil de 2,4” también fabricada por Terasic para ser empleada en sus placas de desarrollo. En este caso se emplea el modelo LT24.

Como ocurre con este tipo de dispositivos se debe visualizar realmente como dos dispositivos en uno ya que se controlan por separado: Por una parte, el array de píxeles con retroiluminación (LCD) que forma la pantalla y por otra parte la capa de digitalización que forma la parte táctil de la pantalla. A pesar de estar completamente integrados a la hora de trabajar se tratan como si fuesen dispositivos independientes el uno del otro.



figura 21, Pantalla táctil Terasic LT24

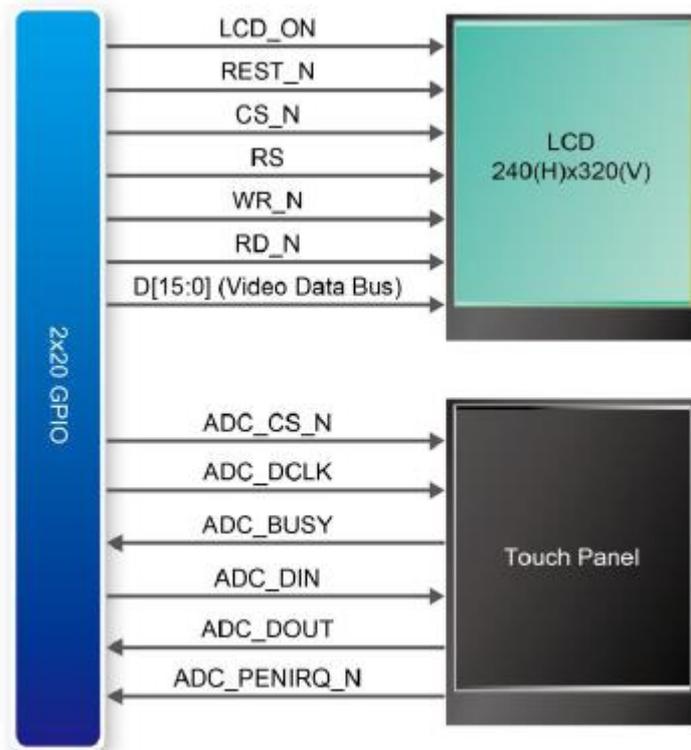


figura 22, Separación entre pantalla LCD y Panel Táctil

### 2.2.1 Pantalla LCD

La pantalla consiste en un array de 240 x 320 píxeles sobre una diagonal de 2,4". Es controlado por un chip fabricado por ILITEK modelo ILI9341 con el que se ha de mantener la comunicación necesaria para poder pintar la pantalla. Para ello se cuenta con una serie de señales para permitir dicha comunicación:

PIN	FUNCIONALIDAD
LCD_ON	Apagado y Encendido de la pantalla
RESET_N	Pin de Reset activo a nivel bajo
CS_N	Chip Select Activo a nivel bajo
RS	Selección modo dato o modo comando
WR_N	Escritura en los registros del chip
RD_N	Lectura de los registros del chip
D[15:0]	Datos de Vídeo para pintar la pantalla

Tabla 3, Pines del Display LCD



**Es importante que en el archivo “top” del proyecto en Verilog queden fijados los pines RESET N y LCD ON a “1'b1” y CS N a 1'b0, de lo contrario la pantalla no funcionará correctamente.** El resto de pines se controlan vía software a través del NIOS II y la IP facilitada por Terasic.

Las señales que se conectan al NIOS II se encuentran controladas por las librerías que facilita Terasic en un ejemplo de uso de la pantalla. El manejo lo realiza utilizando un timer a modo de referencia para el pintado de la pantalla, de forma que cuando el timer avisa el NIOS envía al chip controlador del LCD los datos a mostrar en pantalla y este los pinta. En secciones posteriores se entrará en mayor detalle acerca del manejo del dispositivo.

### 2.2.2 Panel digitalizador táctil o Touch Panel

El LCD se encuentra recubierto por un panel táctil resistivo que permite la utilización del conjunto como pantalla táctil. El panel digitalizador es capaz de enviar una señal de interrupción al micro para que actúe en consecuencia, de este modo no es necesario estar leyendo continuamente los valores numéricos que entrega el panel.

Los valores entregados por el panel son simplemente las coordenadas del punto en que se detecta un contacto, de forma que mediante software se puede actuar en consecuencia. Desde el punto de vista hardware el digitalizador es un dispositivo de tipo PIO (Parallel Input Output) de la misma forma que lo sería un interruptor o un pulsador.

En la siguiente tabla se muestra la funcionalidad de los pines de la parte táctil de la pantalla:

PIN	FUNCIONALIDAD
ADC_CS_N	Chip select del ADC del digitalizador, activo a nivel alto
ADC_DCLK	Entrada de reloj del ADC del digitalizador
ADC_BUSY	Señal que indica que el ADC está ocupado. No se utiliza en este trabajo
ADC_DIN	Entrada de datos al ADC del digitalizador
ADC_DOUT	Salida de datos desde el ADC del digitalizador
ADC_PENIRQ_N	Salida de solicitud de interrupción del ADC del digitalizador

Tabla 4, Pines de la parte táctil de la pantalla

### 2.3 TARJETA DE EXPANSION

La tarjeta de expansión utilizada es el modelo THDB-HTG de Terasic. Esta tarjeta se conecta al conector HSTC que incorpora la placa en el extremo derecho de la misma y la caja incluye tornillos y separadores que se pueden incorporar en la placa para una mejor sujeción, ya que la placa queda físicamente hacia afuera de la placa y sin sujeción existe el riesgo de que se doble y sufra daños.



figura 23, Tarjeta de expansión THDB-HTG de Terasic

El principio de funcionamiento de la tarjeta es relativamente directo, la tarjeta permite la conexión de 3 dispositivos adicionales a través de 3 interfaces GPIO. El interfaz HSTC que incorpora la placa DE2-115 y que es la entrada externa de esta tarjeta de expansión tiene 180 pines y cada GPIO son 40 pines, lo cual significa que no existe ningún tipo de limitación, la FPGA tiene control directo sobre cada uno de los pines de los 3 puertos GPIO a los que da acceso la tarjeta de expansión. Por lo tanto, para poder utilizar los GPIO solamente hay que realizar correctamente la asignación de pines en el Pin Planner del Quartus II y el funcionamiento es inmediato. En secciones posteriores se muestran los detalles de las conexiones de los pines.

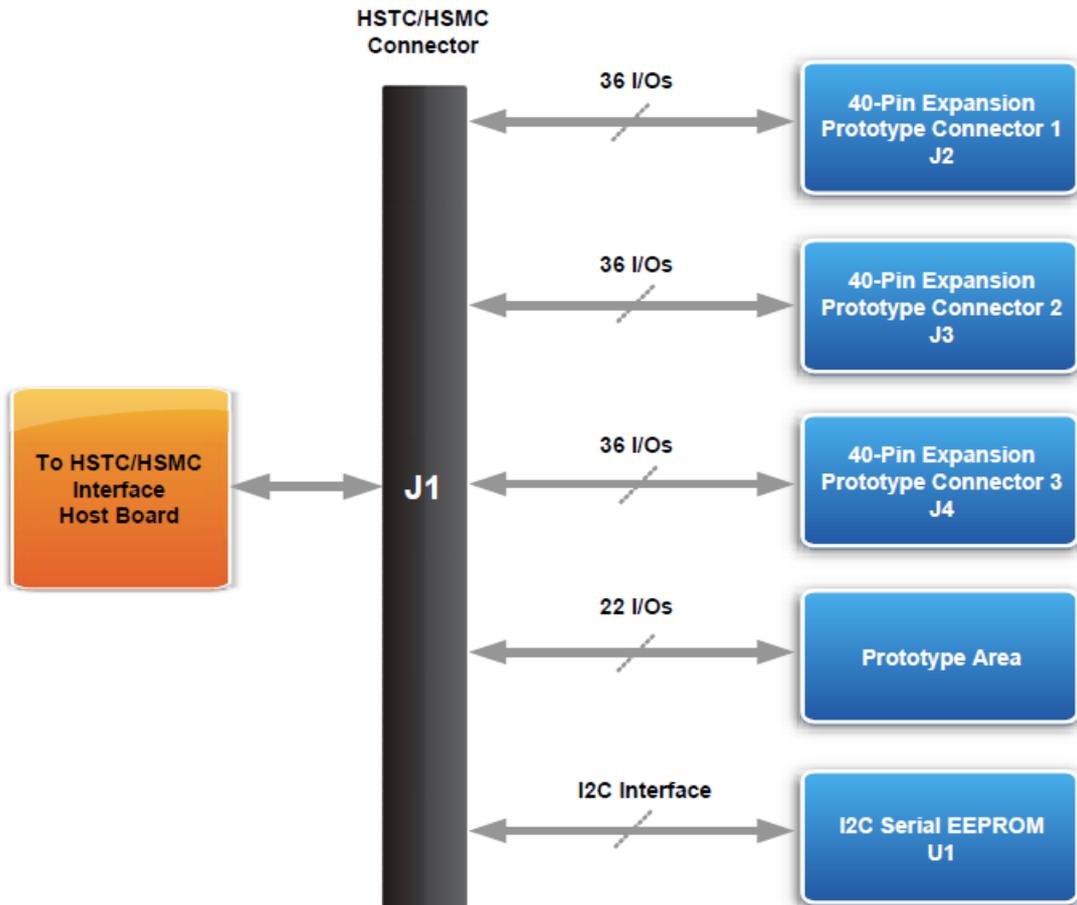


figura 24, Diagrama de bloques de la tarjeta de expansión THDB-HTG facilitado por Terasic

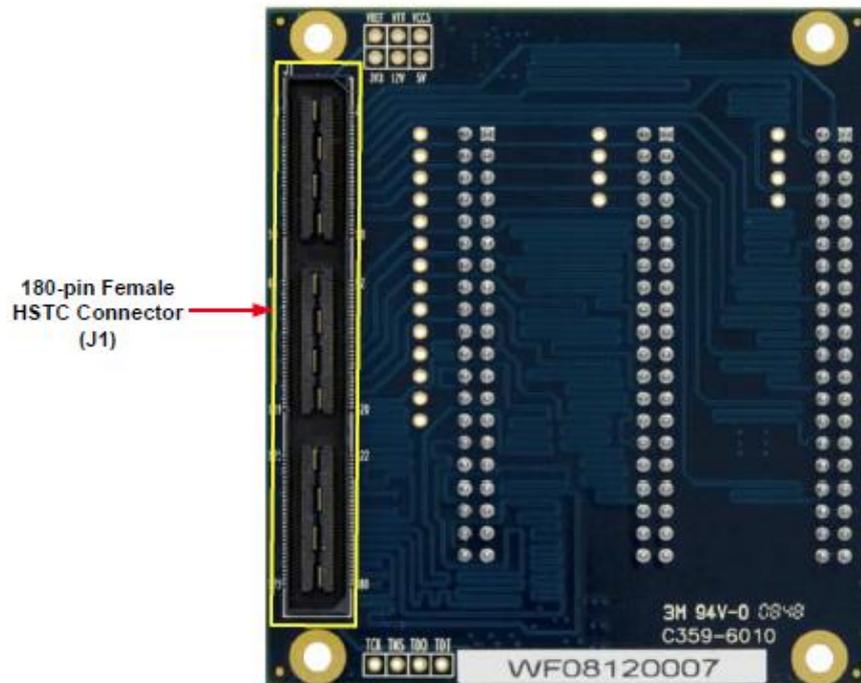


figura 25, Vista inferior de la tarjeta de expansión THDB-HTG. Se observa el conector físico HSTC que se conecta a la placa.

### 3- DISEÑO HARDWARE DEL PROYECTO

En esta sección se detallan los aspectos hardware de los que se compone el proyecto a alto nivel, de esta forma se describe la funcionalidad de cada una de las IPs empleadas, así como su configuración.

A continuación, se muestra un diagrama de bloques con los detalles del diseño hardware:

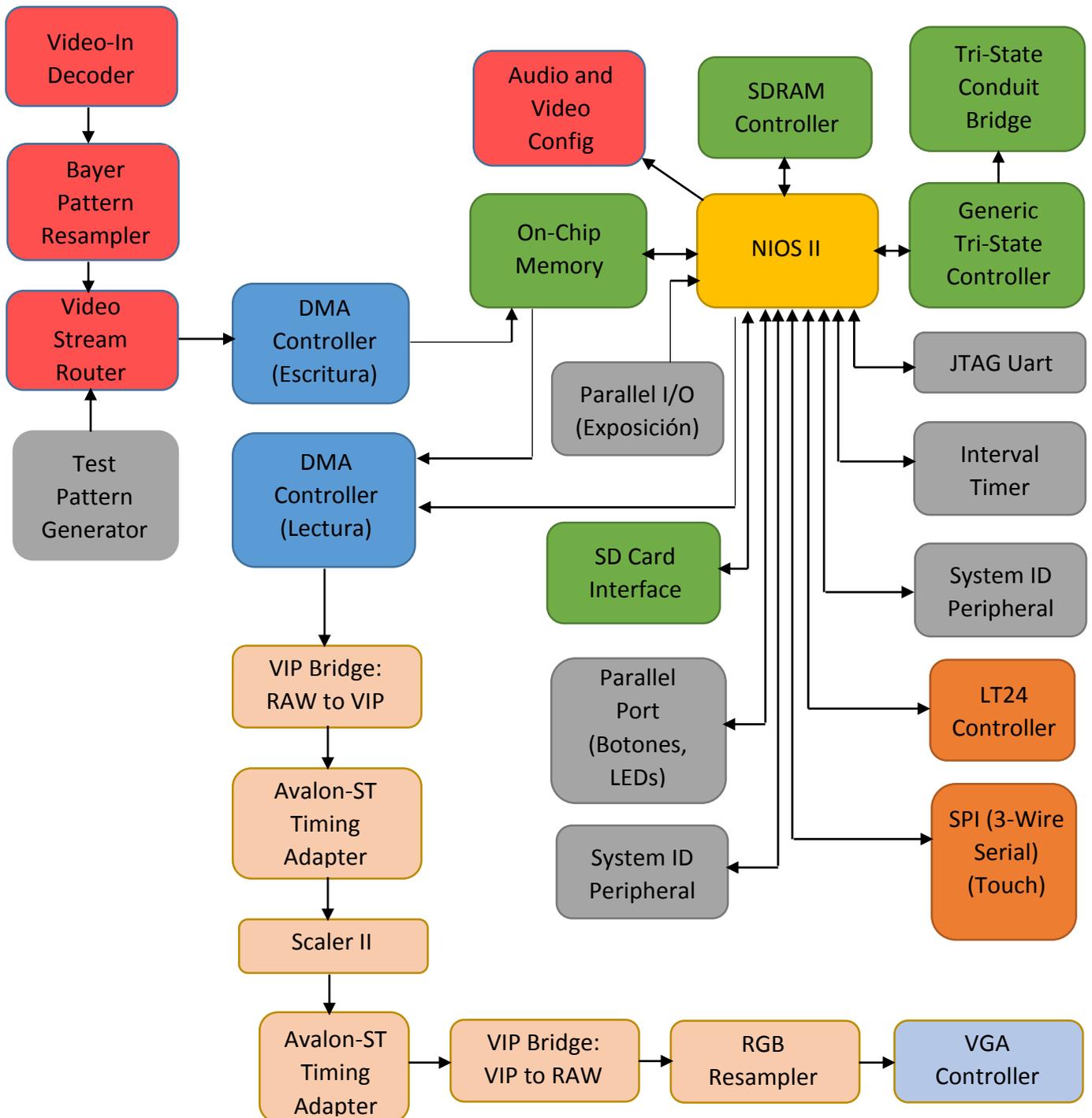


figura 26, Diagrama de bloques hardware del sistema. Se muestran los bloques principales sin Relojes/Resets

### 3.1 Altera University Program

Se denomina University Program o Programa Universitario de Altera a un paquete de IPs (Intellectual Properties o Propiedades Intelectuales) diseñados por Altera para ser empleadas en sus placas de desarrollo. Altera a través de su software ya proporciona una librería de IPs genérica que permite la generación de diversos bloques hardware válidos para diferentes usos. La particularidad del University Program es que sus IPs están pensadas para ser usadas con las placas de desarrollo “DE” de Terasic, de forma que dejan de ser IPs genéricas y pasan a ser IPs específicas para sus placas y sus componentes particulares, simplificando enormemente los parámetros de configuración necesarios para la IP.

Forma parte de los objetivos de este proyecto intentar sacarle el máximo partido a las IPs del University Program, evitando en la medida de lo posible la necesidad de diseñar nuevas IPs.

### 3.2 Bloques IP básicos del sistema

En este apartado se describen aquellos bloques que incorpora el trabajo que corresponden a la arquitectura básica del sistema. La arquitectura básica se compone el procesador NIOS II y sus periféricos habituales como son las memorias, relojes, etc.

#### Nios II

Esta es una de las IPs más importantes del proyecto. Se trata del NIOS II, la CPU soft de Altera, que se encarga de ejecutar el software del sistema. Que sea una CPU soft implica que la CPU se programa empleando los bloques lógicos de la FPGA, por lo tanto, cuando se incorpora al proyecto se produce una reducción de los recursos utilizables en la FPGA ya que una parte de los mismos los ocupa la CPU. Esto permite también cierta flexibilidad, ya que se pueden seleccionar aquellos aspectos de la CPU que se deseen activar o desactivar, ajustando ciertas características de la CPU a los detalles específicos del proyecto que se desee realizar como puede ser: Activar o desactivar la caché, activar o desactivar las multiplicaciones por hardware, el predictor de saltos, etc.

En este caso se emplea la CPU Nios II de segunda generación, IP que se incluye en versiones de Quartus II a partir de la versión 14.0.

A continuación, se muestran las distintas opciones de configuración del NIOS II en este proyecto:

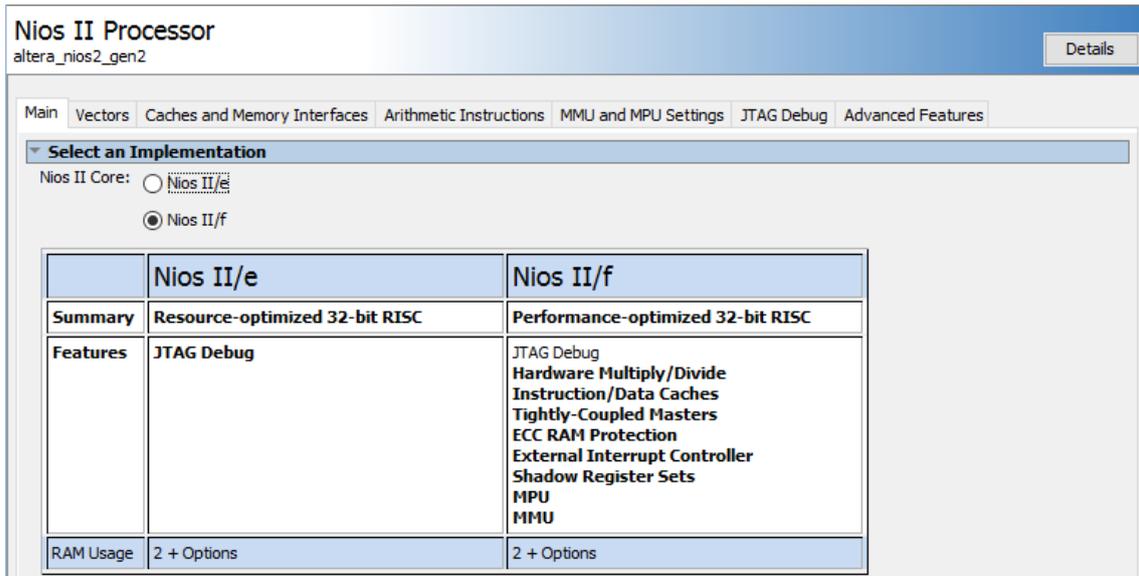


figura 27, Pestaña Main del NIOS II

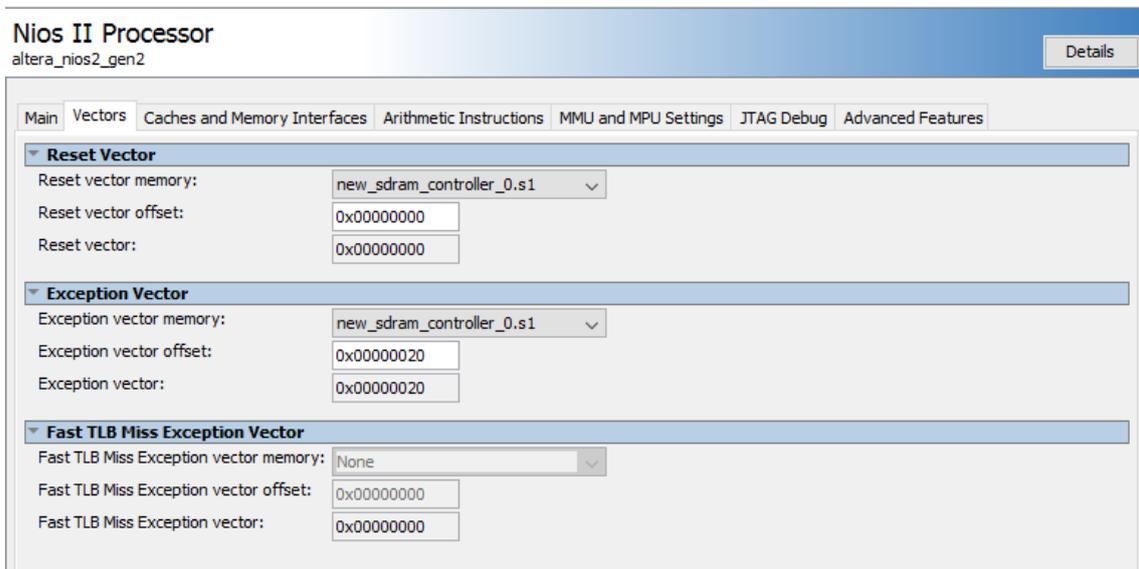


figura 28, Pestaña Vectors del NIOS II

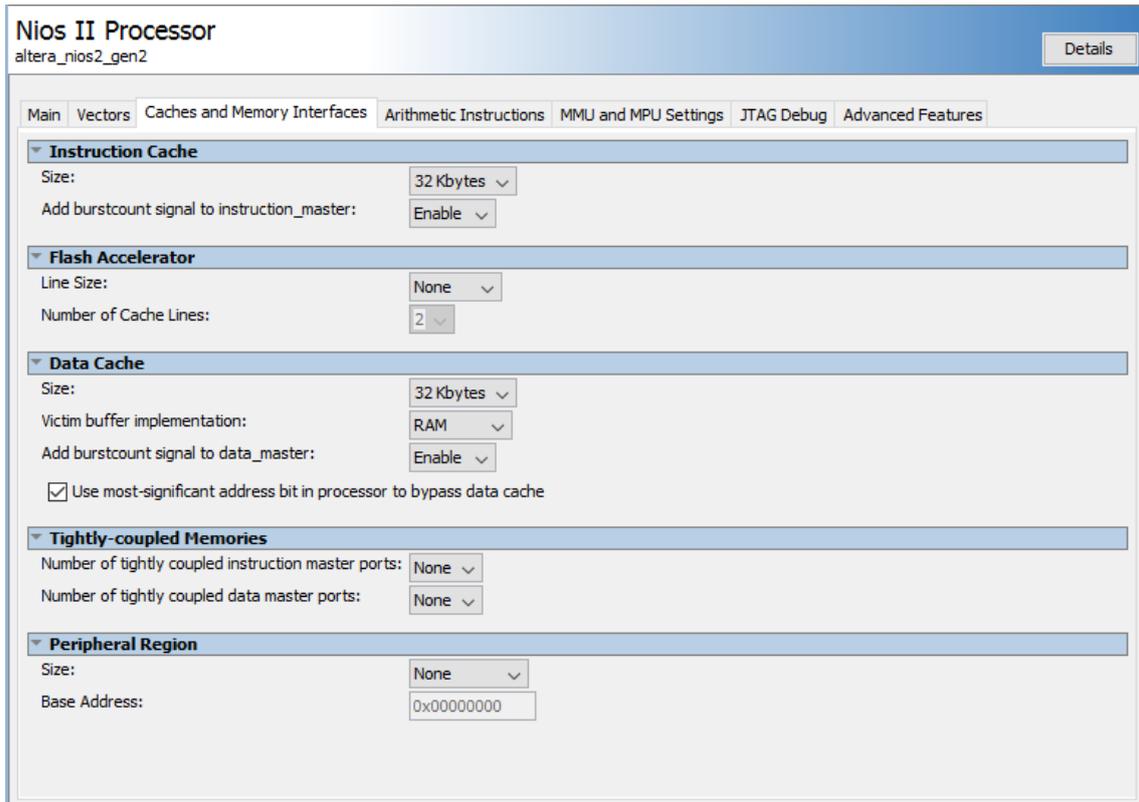


figura 29, Pestaña Caches and Memory Interfaces del NIOS II

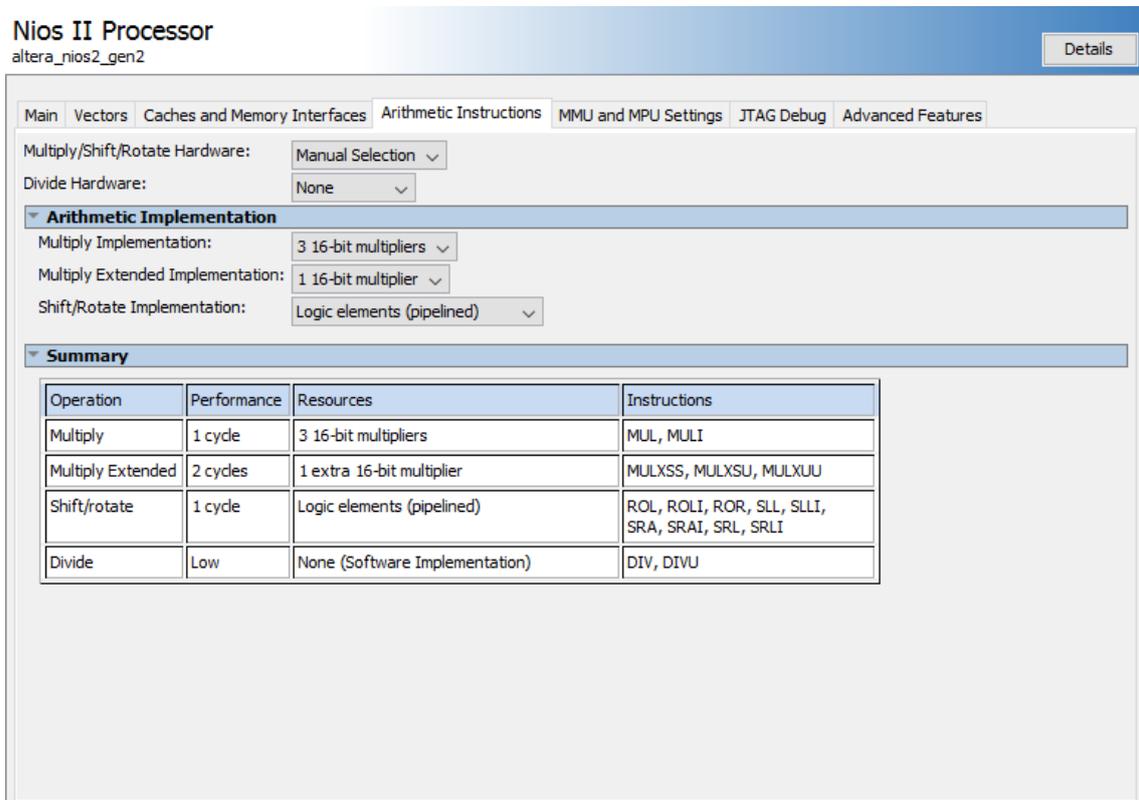


figura 30, Pestaña Arithmetic Instructions del NIOS II

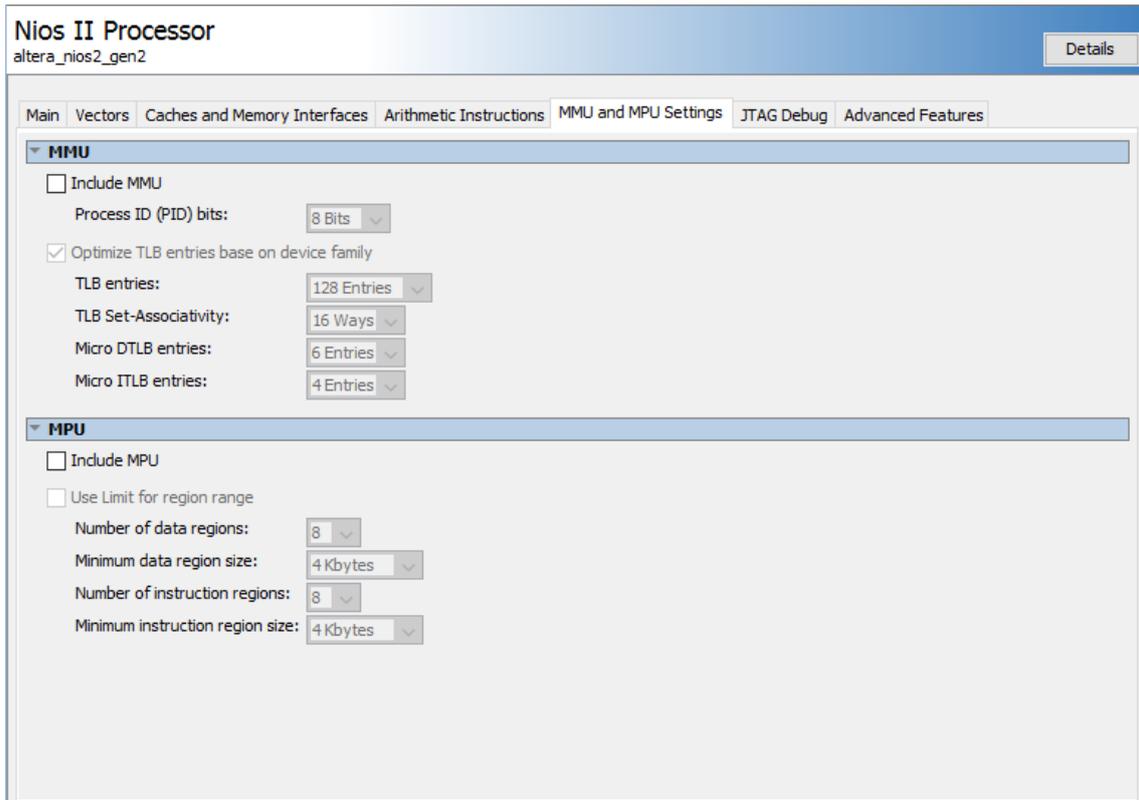


figura 31, Pestaña MMU and MPU settings del NIOS II

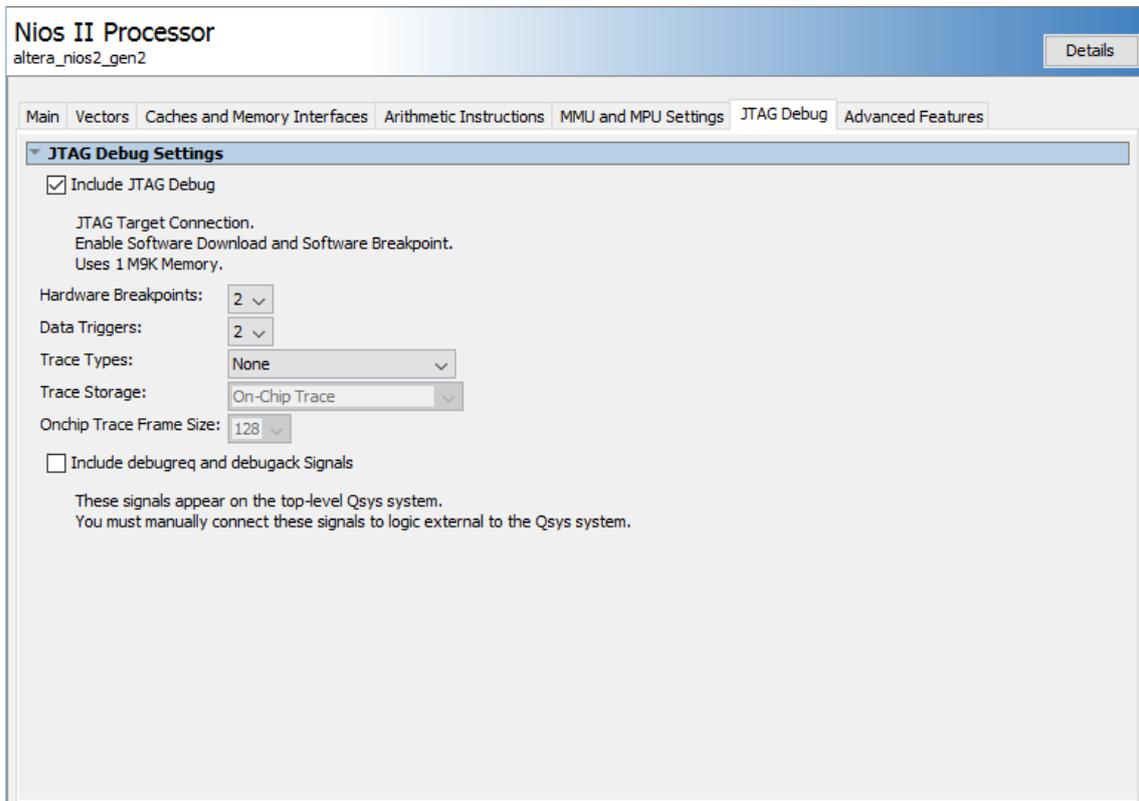


figura 32, Pestaña JTAG Debug del NIOS II

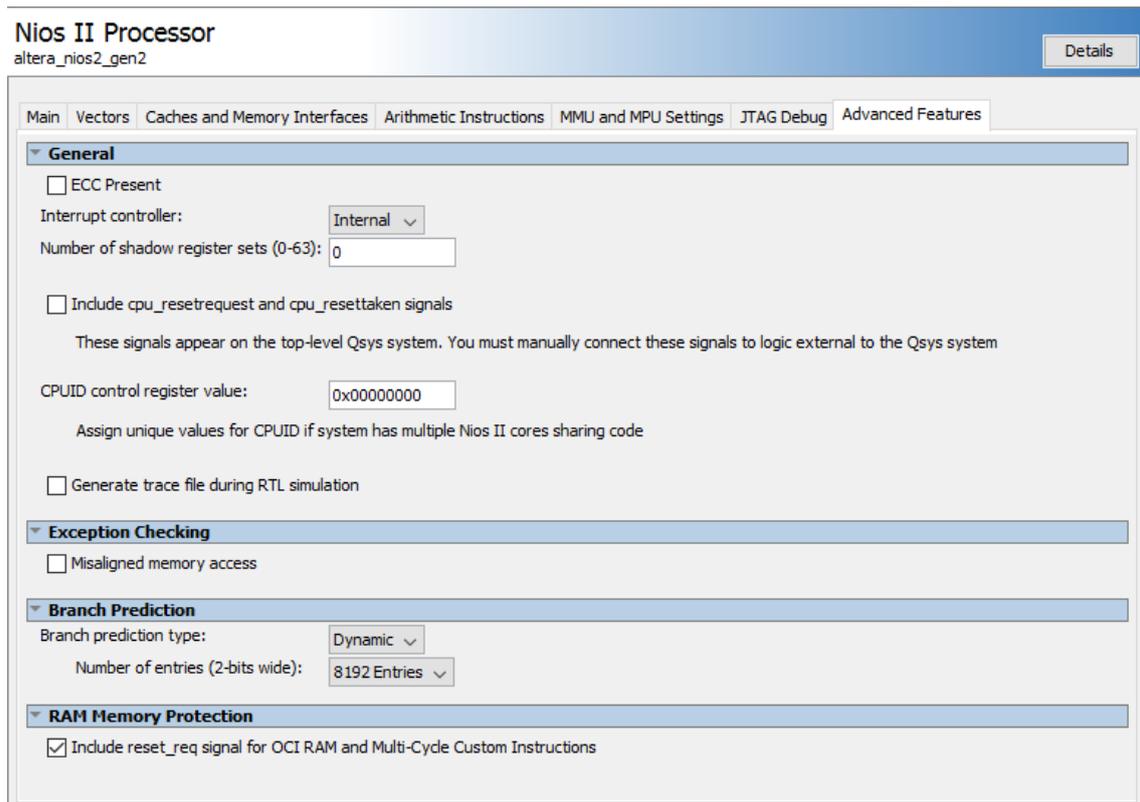


figura 33, Pestaña Advanced Features del NIOS II

Se puede observar que se está utilizando la versión orientada a rendimiento del Nios II (Nios II/f), la versión económica no ofrece suficiente potencia para esta aplicación.

La memoria del sistema es memoria SDRAM, y hay un total de 128MB disponibles. En esta memoria es donde se ejecuta la programación realizada desde Eclipse como se verá en posteriores secciones del documento.

Las cachés son importantes en esta aplicación, puesto que la memoria del sistema es la SDRAM y por lo tanto es memoria off-chip. El no incorporar cachés provoca serios problemas de rendimiento en esta aplicación (pintado lento de la pantalla táctil).

El resto de opciones no son especialmente importantes, se opta por un predictor de saltos dinámico puesto que el código ejecutado para el control de la pantalla dispone de mucho código condicional, aunque el tamaño del mismo no es especialmente crítico.

### RELOJ DEL SISTEMA (2 Opciones)

#### System and SDRAM Clocks for DE-series Boards (Opción 1)

La presente IP se incluye dentro del University Program de Altera y es un componente que engloba todos los relojes del sistema. La entrada de este componente es el reloj externo de 50MHz que incorpora la placa y lo único que es necesario configurar es indicarle el modelo concreto de placa "DE" que se está utilizando para que el componente saque los relojes adecuados, en este caso DE2-115:

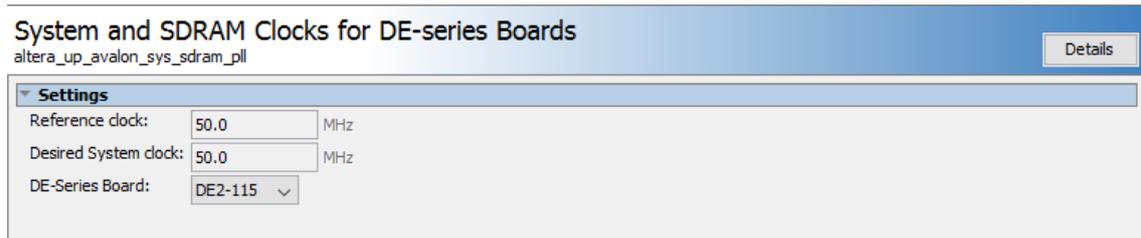


figura 34, Configuración de la IP de generación de relojes

Como se especifica el uso de esta IP es la opción en caso de que posteriormente se vayan a emplear exclusivamente las IPs de Vídeo del University Program (favoreciendo la sencillez y el área ocupada en la FPGA a costa de calidad de imagen).

### Avalon ALTPLL (Opcion 2)

En el caso de que se priorice la calidad de imagen a costa de complicar el diseño y ocupar más recursos en la FPGA se debe emplear esta segunda opción. En ese caso, se deben crear dos relojes en este bloque: Uno de 65MHz para el sistema y otro también de 65MHz, pero con un desfase de -3ns para la memoria SDRAM (El desfase se debe a la distancia entre el chip y el controlador de memoria, de forma que se compense el clock skew).

La configuración del bloque es la siguiente:

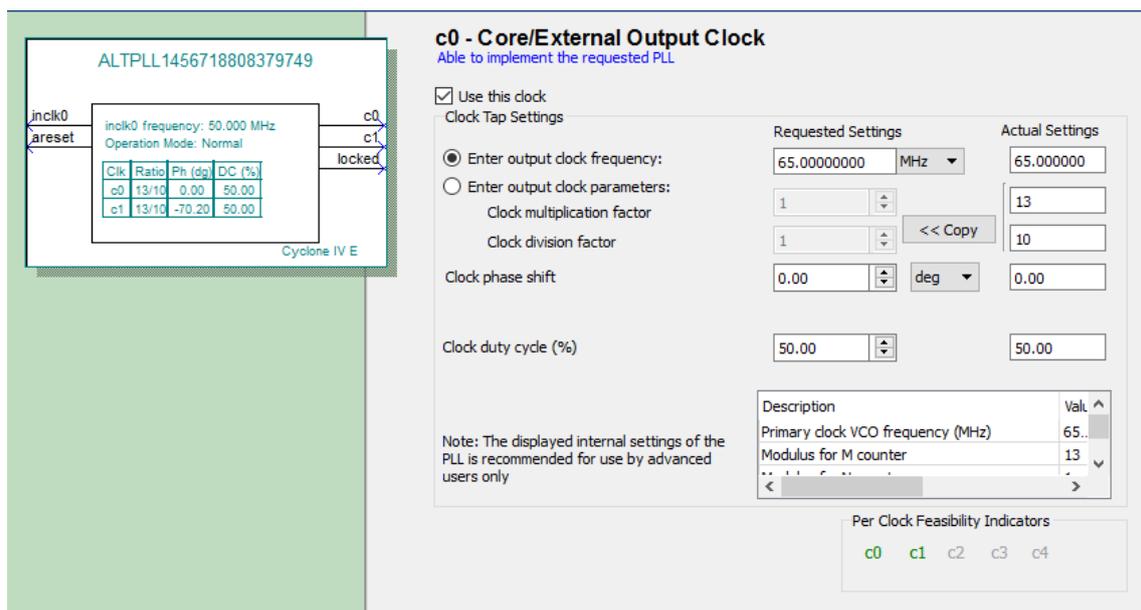


figura 35, Configuración del primer PLL (Sistema) de la IP Avalon ALTPLL

**c1 - Core/External Output Clock**  
Able to implement the requested PLL

Use this clock

Clock Tap Settings

Requested Settings	Actual Settings
Enter output clock frequency: 65.00000000 MHz	65.000000
Enter output clock parameters: Clock multiplication factor: 1	13
Clock division factor: 1	10
Clock phase shift: -3000.00 ps	-3076.92
Clock duty cycle (%): 50.00	50.00

Note: The displayed internal settings of the PLL is recommended for use by advanced users only

Per Clock Feasibility Indicators: c0 c1 c2 c3 c4

figura 36, Configuración del segundo PLL (SDRAM) de la IP Avalon ALTPLL

Las opciones de todas las demás pestañas se dejan por defecto.

En realidad, la primera opción (System and SDRAM Clock for DE-series boards) se puede también realizar empleando este bloque, configurándolos exactamente igual, pero en lugar de 65MHz introduciendo 50MHz.

### SDRAM Controller

Este componente es un componente genérico de Altera que permite la programación en la FPGA de un controlador para memorias SDRAM. La configuración para esta placa es la siguiente:

**SDRAM Controller**  
altera\_avalon\_new\_sdram\_controller

Memory Profile Timing

**Data Width**  
Bits: 32

**Architecture**  
Chip select: 1  
Banks: 4

**Address Width**  
Row: 13  
Column: 10

**Generic Memory model (simulation only)**  
 Include a functional memory model in the system testbench

Memory Size = 128 MBytes  
33554432 x 32  
1024 MBits

figura 37, Pestaña Memory Profile del controlador SDRAM

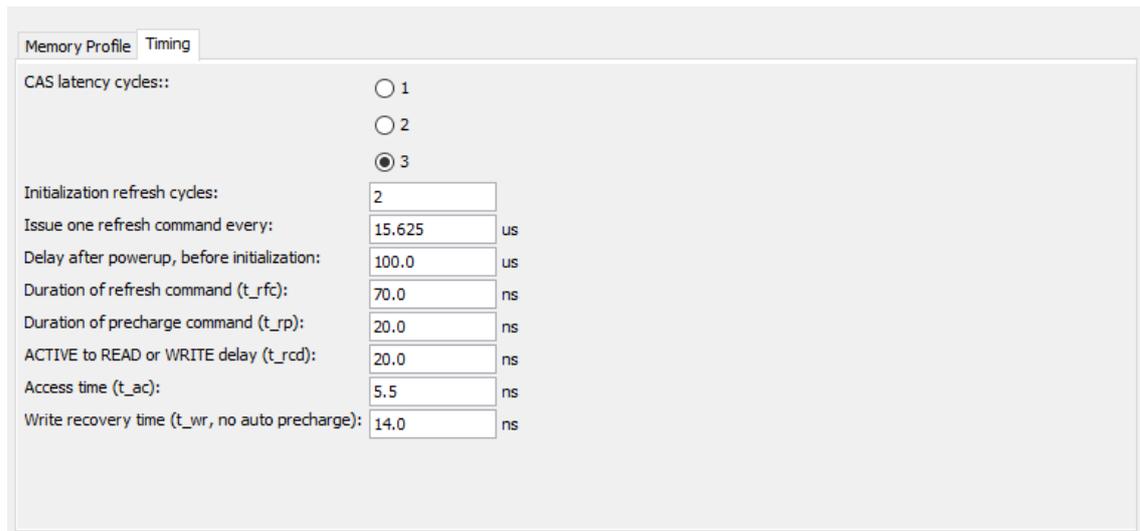


figura 38, Pestaña timing del controlador SDRAM

Puesto que este componente se comunica con memoria externa a la FPGA su instanciación genera una serie de pines de salida:

Señales de entrada/salida generadas por la IP SDRAM Controller		
Señal	Input/Output/InOut	Tamaño
DRAM_ADDR	Output	13 bits
DRAM_BA	Output	2 bits
DRAM_CAS_N	Output	1 bit
DRAM_CKE	Output	1 bit
DRAM_CS_N	Output	1 bit
DRAM_DQ	InOut	32 bits
DRAM_DQM	Output	4 bits
DRAM_RAS_N	Output	1 bit
DRAM_WE_N	Output	1 bit

Tabla 5, Señales de entrada/salida generadas por la IP SDRAM Controller

Se puede observar que falta el pin de reloj. Este componente no crea el reloj de salida hacia la memoria, por lo tanto, habrá que colocar un Clock Bridge para ello. Este componente se describe más abajo.

#### JTAG Uart

Este periférico permite la programación del Nios II así como su debug. La configuración de este componente es la siguiente (configuración por defecto):

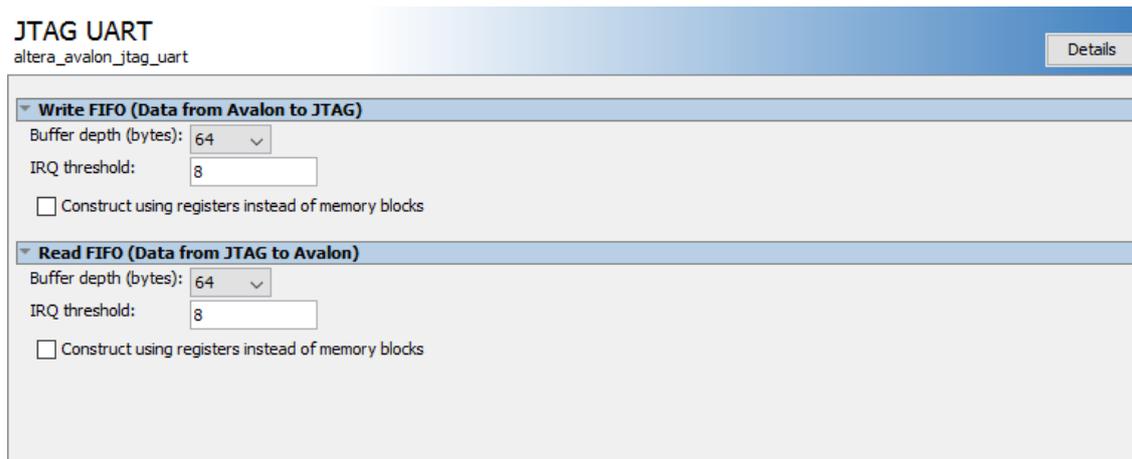


figura 39, Configuración del JTAG Uart

### Clock Bridge y Reset Bridge

Estos componentes crean puentes de salida hacia el exterior de la FPGA.

En este caso se emplea un puente de Reset para permitir un reset del sistema mediante la pulsación de un botón.

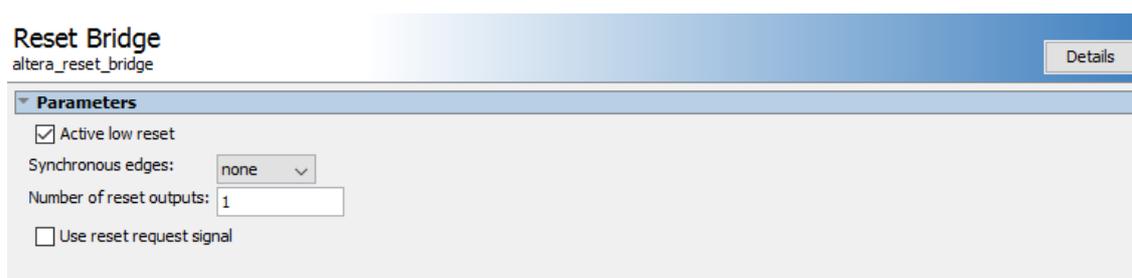


figura 40, Configuración del Reset Bridge

Por otra parte, hay ciertas IPs que al ser instanciadas no producen el reloj de salida necesario, es el caso por ejemplo de la SDRAM. Este componente requiere un Clock Bridge para generar el reloj entre la FPGA y la memoria física:



figura 41, Configuración del Clock Bridge para la memoria SDRAM

### 3.3 Bloques IP de la cámara

La posibilidad de implementar una cámara de fotos viene dada por el uso del módulo D5M de Terasic. Este módulo se conecta a una ranura GPIO de la placa de desarrollo de forma que sus pines quedan conectados directamente a la FPGA. Para poder ser utilizado, se emplean las IPs del University Program que se detallan a continuación.

### Video-In Decoder

Esta IP se comunica directamente con las señales de vídeo (pero no de datos para la configuración de los registros) de la cámara. Ofrece la posibilidad de usar la cámara D5M como es el caso, pero también permite utilizar una cámara de 1.3 Megapíxeles o el ADC de vídeo PAL que contiene la placa DE2-115.

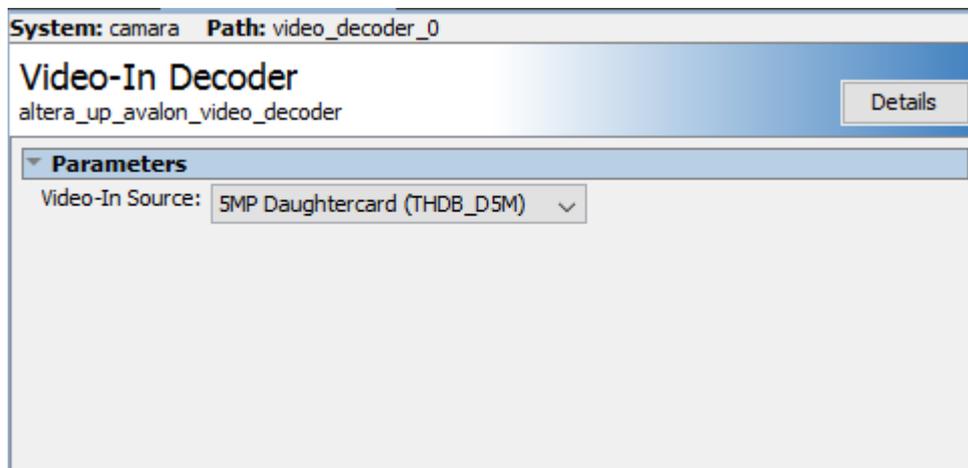


figura 42, Configuración del Video-In Decoder

Una vez se ha configurado la IP para ser usada con la cámara D5M seleccionando “5MP Daughtercard (THDB-D5M)” como fuente, esta IP generará las siguientes señales de entrada/salida que deberán ser configuradas en el Pin Planner del Quartus II:

Señales de entrada/salida generadas por la IP Video-In Decoder		
Señal	Input/Output/Inout	Tamaño
PIXEL_CLK	Output	1 bit
LINE_VALID	Input	1 bit
FRAME_VALID	Input	1 bit
PIXEL_CLK_RESET	Output	1 bit
PIXEL_DATA	Input	12 bits

Tabla 6, Señales de entrada/salida generadas por la IP Video-In Decoder

**La salida de este componente** es conectada al siguiente componente o IP que es el **Bayer Pattern Resampler**.

### Bayer Pattern Resampler

Esta IP realiza la conversión del formato de vídeo con filtro Bayer que proporciona el Video-In Decoder y convierte dicha señal de vídeo a formato RGB de 24 bits (8 bits x 3 planos). Esta conversión hace que la resolución de salida en formato RGB sea la mitad que la resolución que entra desde el Video-In Decoder. En este caso la señal pasa de 640 x 480 con 8 bits de color a 320 x 240 con 24 bits RGB.

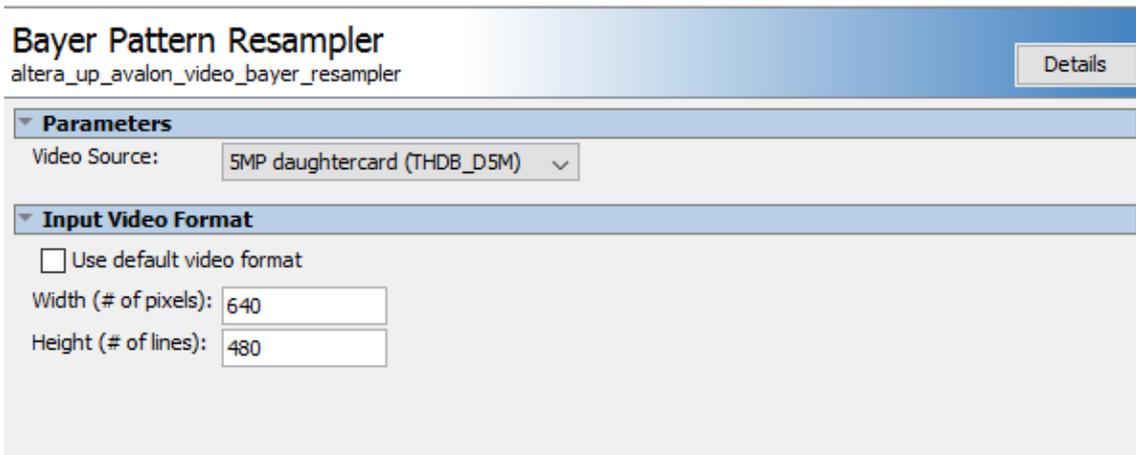


figura 43, Configuración del Bayer Pattern Resampler

Esta IP no requiere ninguna configuración adicional y su funcionamiento reside completamente en el interior de la FPGA por lo que esta IP no produce ningún pin de salida.

**La salida de este componente puede ir conectada directamente a la controladora DMA de escritura**, sin embargo, en este caso irá conectado a un **enrutador de vídeo (Video Stream Router)** insertado en el proyecto con propósitos de debugging. Puede ir conectado a cualquiera de las dos entradas del enrutador.

### Video Stream Router

Esta es una IP opcional que en este caso se ha introducido para ser empleada a modo de debug. Esta IP funciona a modo de multiplexor como selector de una señal de vídeo saliente a partir de dos posibles señales de vídeo entrantes. El componente dispone de una entrada de selección para permitir la visualización de una u otra señal de vídeo. En este caso se conecta a una de las entradas el Bayer Pattern Resampler del apartado anterior y a la otra un componente de Altera denominado Video Test Pattern que genera un patrón de colores similar al de una carta de ajuste. Esta conexión permite, en caso de fallo, detectar si el problema reside en el Hardware anterior, es decir, en la IP Video-In Decoder o en la IP Audio and Video Config que se describe más adelante (en el Bayer Pattern Resampler es muy improbable que exista un error pues apenas tiene parámetros de configuración) o en el hardware posterior que es básicamente el DMA y la salida de vídeo VGA.



figura 44, Configuración del Video Stream Router

**La salida de este componente se conecta a la IP Video DMA Controller.**

### Test-Pattern Generator

Esta IP solo es necesaria si se desea utilizar una imagen secundaria a la de la cámara con propósitos de debug, para lo cual habrá sido necesario añadir el componente “Video Stream Router” que permite la selección entre ambos. La configuración de este bloque es muy simple:

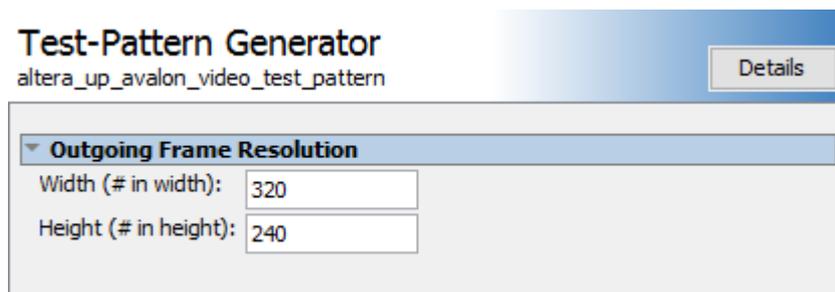


figura 45, Configuración de la IP Test-Pattern Generator

**La salida de este componente, al igual que el “Bayer Pattern Resampler” puede ir conectado a cualquiera de las dos entradas del enrutador de vídeo (Video Stream Router).**

### PIO (Parallel I/O), Opcional

Se instancia un puerto PIO para el control del tiempo de exposición del sensor de la cámara vía software. No es estrictamente necesario hacerlo, en este caso se ha decidido controlar la exposición vía software cuyo valor se altera con los botones antirebote de la placa (KEY [3] y KEY [2]). Si no se hace así se puede dejar fijado un valor en el archivo “TOP” de Quartus para la exposición: Por ejemplo, para una tasa de frames de 60fps ese valor sería 16’d633, a partir de aquí aumentar este valor aumenta el tiempo de exposición (y reduce la tasa de fotogramas por segundo como consecuencia), y reducirlo lo reduce (y como consecuencia se aumenta la tasa de frames por segundo).

La configuración del puerto PIO para el control del tiempo de exposición es la siguiente:

<b>Basic Settings</b>	
Width (1-32 bits):	16
Direction:	<input type="radio"/> Bidir <input type="radio"/> Input <input type="radio"/> InOut <input checked="" type="radio"/> Output
Output Port Reset Value:	0x0000000000000000
<b>Output Register</b>	
<input type="checkbox"/> Enable individual bit setting/clearing	
<b>Edge capture register</b>	
<input type="checkbox"/> Synchronously capture	
Edge Type:	RISING
<input type="checkbox"/> Enable bit-clearing for edge capture register	
<b>Interrupt</b>	
<input type="checkbox"/> Generate IRQ	
IRQ Type:	LEVEL
<b>Level:</b> Interrupt CPU when any unmasked I/O pin is logic true <b>Edge:</b> Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled	
<b>Test bench wiring</b>	
<input type="checkbox"/> Hardwire PIO inputs in test bench	
Drive inputs to:	0x0000000000000000

figura 46, Configuración de la IP PIO para el control de la exposición vía software



**El control de la exposición no se puede hacer vía hardware (en verilog) directamente.**

Esto se debe a que la inicialización de los parámetros del sensor lo realiza la IP Audio and Video Config y únicamente los hace una vez, cuando el bloque se inicia. Vía software dicho bloque se puede resetear de forma que se reinician los parámetros del sensor, por eso el control de la exposición se realiza vía software, alterando los valores con los pulsadores antirebote y realizando un reset al bloque Audio and Video Config (programado en C, en el software ejecutado en el NIOS II) cada vez que se altera el valor de exposición.

**Video DMA Controller**

Esta IP es una controladora DMA para aplicaciones de vídeo que se incluye en el University Program. El propósito de uso de esta IP consiste en tomar la señal de vídeo entrante e ir almacenándola en una memoria. Posteriormente otra instanciación del mismo componente DMA lee dicha memoria y envía los datos al hardware de salida de vídeo.

El objetivo principal de esta arquitectura consiste en que una vez el DMA que escribe en la memoria ha escrito esos datos y se encuentran almacenados en dicha memoria, estos puedan ser manipulados a través de una CPU que también dispone de acceso a esa memoria. De esta forma, una vez los datos se encuentran en la memoria la CPU puede desactivar el DMA que lee los datos, alterarlos y, posteriormente, volver a activar el DMA de lectura para que los datos alterados sean mostrados por pantalla. Esta arquitectura también permite visualizar en pantalla vídeo sin intervención de una CPU, con el ahorro de recursos y rendimiento que ello supone.

Es importante que el valor “Default Buffer Start Address” coincida con la dirección de la memoria que se va a utilizar para la escritura/lectura de los DMAs.

En este caso se requieren dos instancias de este componente, en ambos casos con la misma configuración exceptuando la opción “DMA Direction” que en una de las instancias deberá estar configurado como “From Stream to Memory” en el caso del DMA que escribe en memoria y en “From Memory to Stream” en el caso del DMA que lee la memoria y envía los datos a otro componente para que la imagen sea mostrada por una pantalla VGA.

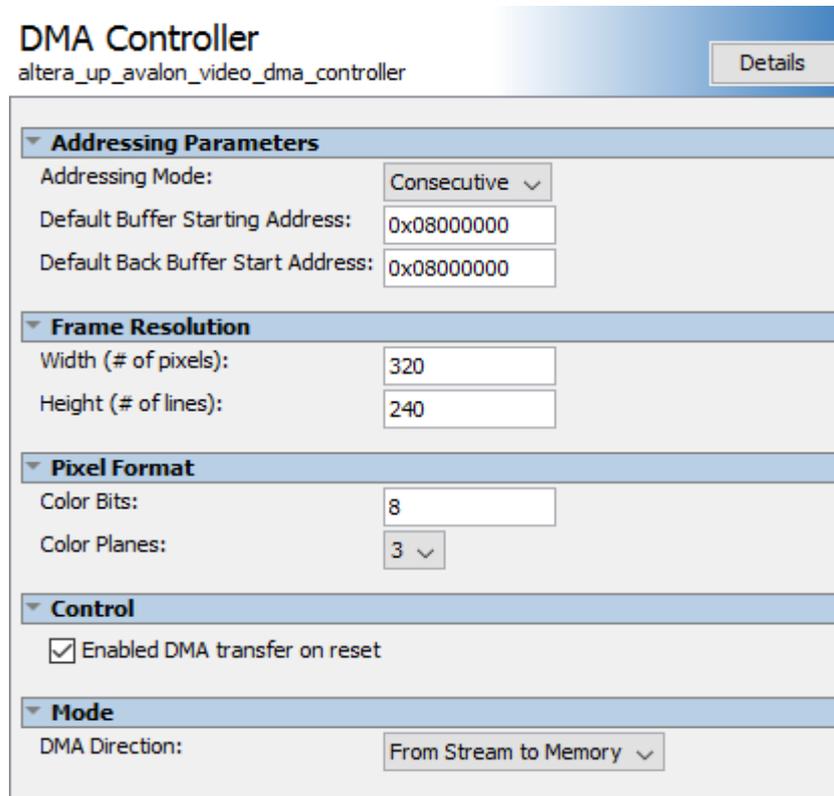


figura 47, Configuración del DMA Controller para escritura en memoria

El conexionado de este componente para el caso de la configuración “From Stream to Memory” si se utiliza el Video Stream Router es:

IP Salida	Salida	IP Entrada	Entrada
Video Stream Router	avalon_stream_router_source	Video DMA Controller	avalon_dma_sink
Nios II	data_master	Video DMA Controller	avalon_dma_control_slave
Video DMA Controller	avalon_pixel_source	Video RGB Resampler	avalon_rgb_sink
Video DMA Controller	avalon_dma_master	On-Chip Memory (u otra memoria)	s1

Tabla 7, Conexionado de la IP Video DMA Controller de escritura con Video Stream Router

En caso de que no se utilice el video stream router a la conexión avalon\_dma\_sink entra directamente la salida del Video Bayer Resampler:

IP Salida	Salida	IP Entrada	Entrada
Video Bayer Resampler	avalon_bayer_source	Video DMA Controller	avalon_dma_sink
Nios II	data_master	Video DMA Controller	avalon_dma_control_slave
Video DMA Controller	avalon_pixel_source	Video RGB Resampler	avalon_rgb_sink
Video DMA Controller	avalon_dma_master	On-Chip Memory (U otra memoria)	s1

figura 48, Conexión de la IP Video DMA Controller de escritura sin Video Stream Router

A continuación, se muestra la configuración del segundo DMA, en este caso este DMA lee los datos de la memoria donde el primero ha escrito los datos y los envía a la salida de vídeo:

figura 49, Configuración del controlador DMA para la lectura de la memoria

A continuación, se muestra el conexión de este componente, que debe estar en modo "From Memory to Stream":

IP Salida	Salida	IP Entrada	Entrada
On-Chip Memory (U otra memoria)	s1	Video DMA Controller	avalon_dma_master
Nios II	data_master	Video DMA Controller	avalon_dma_control_slave
Video DMA Controller	avalon_pixel_source	Video RGB Resampler	avalon_rgb_sink

Tabla 8, Conexión de la IP Video DMA Controller de lectura

Una vez el DMA es funcional existen dos alternativas a la hora de sacar la señal por pantalla como se verá a continuación.

- Bloques University Program: El uso de esta opción implica utilizar los bloques Video Scaler y Video Dual-Clock FIFO del University program. Para ello la frecuencia de reloj del sistema será de 50MHz para una resolución de salida VGA de 640 x 480. La máxima tasa de frames por segundo alcanzables es de 30 fps en el monitor VGA.
- Bloques de Altera VIP: En este caso es necesario usar los bloques Video Raw to VIP y Video VIP to Raw del University Program, así como dos bloques Timing Adapter y un bloque Scaler II de Altera. La frecuencia de reloj del sistema en este caso debe ser de 65 MHz para una salida VGA de 1024 x 768. La máxima tasa de frames por segundo alcanzable en este caso es de 60fps en el monitor VGA.

La mayor calidad de imagen, especialmente en cuanto a fluidez de la misma en pantalla se consigue con la segunda opción, a costa eso si de un sistema más complejo que ocupa mucha más área en la FPGA.

A continuación, se describen las dos opciones.

## OPCIÓN 1: BLOQUES DEL UNIVERSITY PROGRAM

### Video Scaler

Esta IP es un simple escalador de video que permite desde una resolución determinada aumentar o reducir el tamaño de la imagen empleando unos factores de escalado predeterminados. En este caso se va a escalar la imagen, que actualmente es de 320 x 240 píxeles a 640 x 480 píxeles simplemente para que la visualización en pantalla sea más grande. La configuración de este componente es la siguiente:

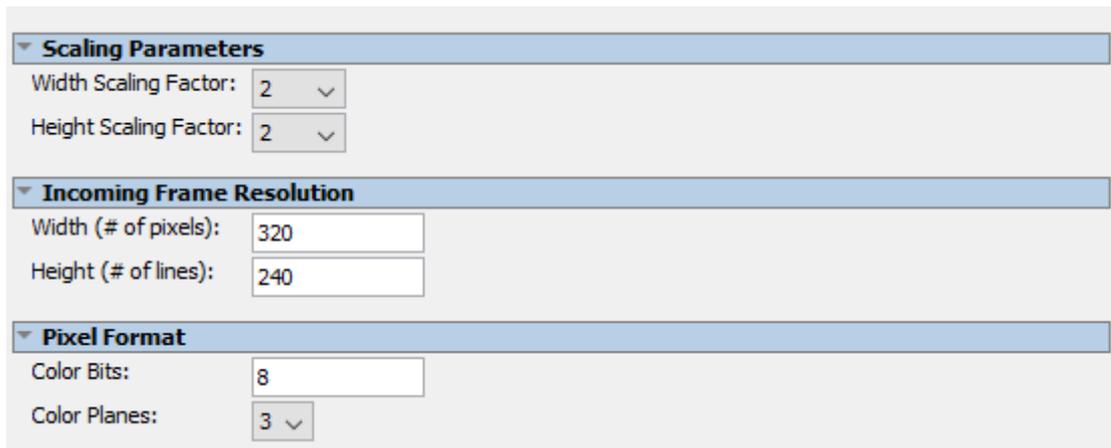


figura 50, Configuración de la IP Video Scaler

La salida de este componente (**avalon\_scaler\_source**) se conecta a la entrada de la IP RGB Resampler.

### RGB Resampler

Esta IP contenida dentro del University Program permite realizar conversiones de formato de color. A la salida del Video Scaler es necesario realizar una conversión a formato RGB de 30 bits (hasta ahora el formato es de 24 bits) ya que el controlador VGA que va a sacar la imagen proveniente de la cámara trabaja con este formato. La configuración de este bloque es la siguiente:

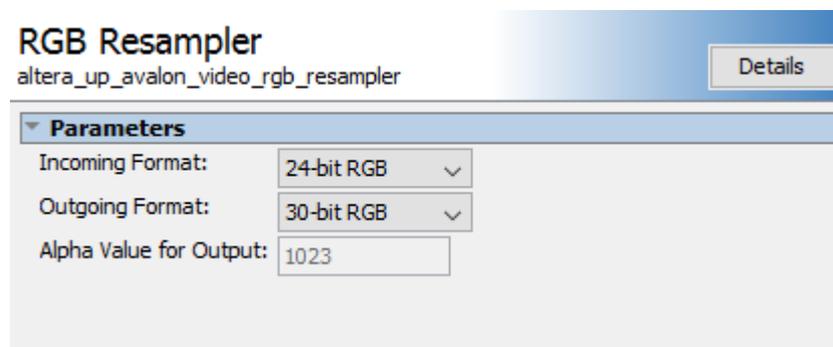


figura 51, Configuración de la IP RGB Resampler

La salida de este componente (**avalon\_rgb\_source**) se conecta a la entrada de la IP Video Dual-Clock FIFO

### Video Dual-Clock FIFO

Esta IP es una cola FIFO que permite el cruce de datos entre dos dominios de reloj. Este componente es necesario puesto que las IPs hasta ahora empleadas están funcionando de forma síncrona con el reloj del sistema (50 MHz), sin embargo, el último componente de la cadena, el controlador VGA, emplea un reloj de 25MHz para una resolución de salida de 640 x 480 píxeles. Esto implica que los datos entran a 50MHz en la FIFO y salen a 25MHz, dando como resultado que uno de cada dos frames es desechado, por lo tanto, en el monitor VGA solamente se representa la mitad de los frames que llegan desde la fuente de imagen.



figura 52, Configuración de la IP Video Dual-Clock FIFO

El conexionado de esta IP es el siguiente:

Entrada a la IP Dual-Clock FIFO	IP desde la que se conecta
<b>Clock Stream In</b>	Sys Clock o PLL del sistema (50 MHz)
<b>Reset Stream In</b>	CLK Reset del reloj principal
<b>Clock Stream Out</b>	PLL de Vídeo (25MHz)
<b>Reset Stream Out</b>	CLK Reset del reloj principal
<b>Avalon DC Buffer Sink</b>	RGB Resampler

Tabla 9, Conexión de la IP Video Dual-Clock Buffer

La salida de este componente (**avalon\_dc\_buffer\_source**) se conecta a la entrada de la IP **VGA Controller**.

#### VGA Controller

Esta es la última IP de la cadena de vídeo. Se trata del controlador VGA, que controla el DAC de salida de vídeo que integra la placa de forma que se muestre la imagen a través de una pantalla conectada al conector VGA de la placa.

La configuración de la IP es la siguiente:



figura 53, Configuración de la IP VGA Controller en la primera opción

La instanciación de esta IP genera las siguientes señales de entrada/salida:

Señales de entrada/salida generadas por la IP VGA Controller		
Señal	Input/Output/Inout	Tamaño
VGA_CLK	Output	1 bit
VGA_HS	Output	1 bit
VGA_VS	Output	1 bit
VGA_BLANK_N	Output	1 bit
VGA_SYNC_N	Output	1 bit
VGA_R	Output	8 bits
VGA_G	Output	8 bits
VGA_B	Output	8 bits

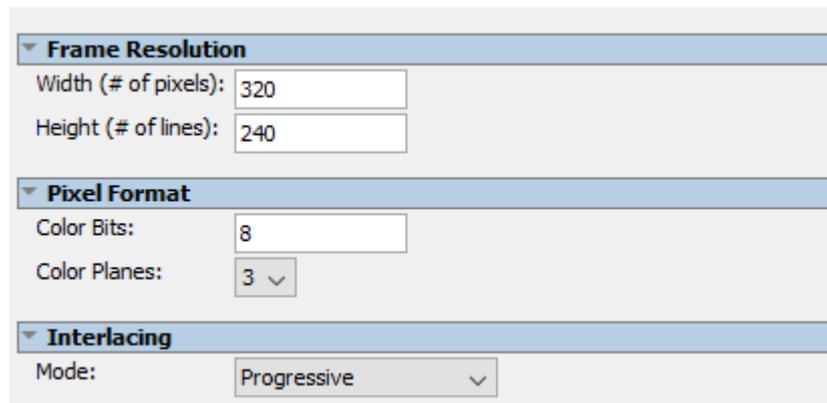
Tabla 10, Señales de entrada/salida generadas por la IP VGA Controller

## OPCIÓN 2: BLOQUES VIP DE ALTERA

### RAW to VIP Bridge

Puesto que se van a emplear las IPs del paquete VIP de Altera es necesario realizar primero una conversión entre el formato RAW con el que trabajan los bloques del University Program y el formato VIP de Altera. Para ello se debe colocar un puente que realice esta conexión (y posteriormente será necesario otro que haga la conversión opuesta y convierta de VIP a RAW).

Este componente se encuentra entre las IPs del University Program y su configuración es la siguiente:



▼ Frame Resolution	
Width (# of pixels):	320
Height (# of lines):	240

▼ Pixel Format	
Color Bits:	8
Color Planes:	3 ▼

▼ Interlacing	
Mode:	Progressive ▼

figura 54, Configuración del bloque RAW to VIP Bridge

**La entrada a este componente** (avalon\_raw\_to\_vip\_sink) proviene del bloque **Video DMA Controller** de lectura de la memoria de vídeo. **La salida** (avalon\_raw\_to\_vip\_source) se conecta a la IP **Timing Adapter**.

### Timing Adapter 0

La latencia de la señal Ready del RAW to VIP Bridge y del Scaler II no son iguales, por lo tanto, es necesario insertar un componente que realice dicha adaptación. Esta IP es de Altera y se denomina Timing Adapter. Su configuración cuando se sitúa entre el RAW to VIP Bridge y el Scaler II es la siguiente:

figura 55, Configuración de la IP Timing Adapter conectada entre el RAW to VIP Bridge y el Scaler II

La **entrada** (in) de este componente proviene del **Raw to Vip Bridge** (avalon\_raw\_to\_vip\_source) mientras que la **salida** (out) se conecta a la IP **Scaler II** de Altera (din).

### Scaler II

Esta IP se encuentra dentro del conjunto de IPs denominado DSP, y dentro del subgrupo Video and Image Processing. Su función es prácticamente la misma que el Scaler del University Program sin embargo esta IP es mucho más avanzada. Mientras que el Scaler del University Program solo permite unos valores determinados de escalados (0.25, 0.5, 0.75, 1, 2, 4 y 8) el Scaler II permite prácticamente cualquier combinación. Esto es muy importante si se desean mostrar por pantalla los 60 fotogramas por segundo que es capaz de entregar el sensor de la cámara cuando se encuentra funcionando a 640 x 480.

La configuración de esta IP es la siguiente:

Video Data Format	
Bits per symbol:	8
Symbols in parallel:	3
Symbols in sequence:	1
<input type="checkbox"/> Enable runtime control of output frame size and edge/blur thresholds	
Maximum input frame width:	320
Maximum input frame height:	240
Maximum output frame width:	1024
Maximum output frame height:	768
<input type="checkbox"/> 4:2:2 video data	
<input checked="" type="checkbox"/> No blanking in video	
Algorithm Settings	
Scaling algorithm:	POLYPHASE
<input type="checkbox"/> Enable post scaling sharpen	
<input type="checkbox"/> Always downscale or pass-through	
<input type="checkbox"/> Share horizontal and vertical coefficients	
Vertical filter taps:	8
Vertical filter phases:	4
Horizontal filter taps:	8
Horizontal filter phases:	4
Default edge threshold:	7
Default upper blur limit (per color plane):	15
Default lower blur limit (per color plane):	0
Precision Settings	
<input checked="" type="checkbox"/> Vertical coefficients signed	
Vertical coefficient integer bits:	1
Vertical coefficient fraction bits:	7
<input checked="" type="checkbox"/> Horizontal coefficients signed	
Horizontal coefficient integer bits:	1
Horizontal coefficient fraction bits:	7
Fractional bits preserved:	0
Coefficient Settings	
<input type="checkbox"/> Load scaler coefficients at runtime	
Vertical coefficient banks:	1
Vertical coefficient function:	LANCZOS_3
Vertical coefficients file:	<enter file name (including full path)>
Horizontal coefficient banks:	1
Horizontal coefficient function:	LANCZOS_3
Horizontal coefficients file:	<enter file name (including full path)>
Pipelining	
<input type="checkbox"/> Add extra pipelining registers	

figura 56, Configuración del bloque Scaler II

Puesto que en este caso el componente VGA Controller se encuentra configurado para una resolución de 1024 x 768 el escalador realiza un upscale desde los 320 x 240 de entrada a los 1024 x 768 de salida. Para hacer esto con bloques del university program haría falta emplear bien 2 bloques scaler o bien 1 bloque scaler y 1 bloque clipper, sin embargo, eso no funciona correctamente puesto que para poder colocar dos bloques en serie estos deberían funcionar al doble de la frecuencia de reloj del sistema y estas IPs no permiten funcionamiento asíncrono. Por eso la solución es usar un bloque capaz de hacer el upscale en lugar de usar varios bloques en serie.

La **entrada** de este bloque (din) proviene del **timing adapter** (out) mencionado previamente, mientras que la **salida** se conecta a un segundo **timing adapter** (in).

## Timing Adapter 1

Este segundo timing adapter realiza la conversión inversa del primero antes de poner el puente complementario (VIP to RAW Adapter) para poder volver al formato de datos de vídeo del University Program.

La configuración de este bloque es la siguiente:

**Input Interface Parameters**

Support Backpressure with the ready signal  
Ready Latency:   
 Include Valid Signal

**Output Interface Parameters**

Support Backpressure with the ready signal  
Ready Latency:   
 Include Valid Signal

**Common to Input and Output Interface Parameters**

Data Bits Per Symbol:   
 Include Packet Support  
When packets are supported, the startofpacket, endofpacket, and empty signals are used.  
Include Empty Signal:   
Data Symbols Per Beat:   
Channel Signal Width (bits):   
Max Channel:   
Error Signal Width (bits):   
Error Signal Description:

figura 57, Configuración de la IP Timing Adapter conectada al VIP to RAW Adapter

La **entrada** de este componente (in) proviene del bloque **Scaler II** (dout), mientras que **su salida** (out) se conecta a la entrada (avalon\_vip\_to\_raw\_sink) de la IP **Video VIP to RAW Adapter Bridge**.

## VIP to RAW Adapter Bridge

Este bloque realiza la conversión opuesta al primer puente mencionado en esta segunda opción. Se realiza la conversión del formato de datos de vídeo (VIP) utilizado por el Scaler II de Altera al formato RAW que emplea el University Program.

La configuración de este bloque es la siguiente:

**Pixel Format**

Color Bits:   
Color Planes:

figura 58, Configuración de la IP VIP to RAW Adapter Bridge

La **entrada** a este componente (avalon\_vip\_to\_raw\_sink) proviene de la salida de la IP **Timing Adapter 1** (out) mientras que **la salida** (avalon\_vip\_to\_raw\_source) se conecta al **RGB Resampler** (avalon\_rgb\_sink).

#### RGB Resampler

Este es el mismo bloque conversor mencionado en la anterior opción. La configuración es la misma, se convierte de formato de color RGB de 24 bits a RGB de 30 bits, requerido por la IP VGA Controller.

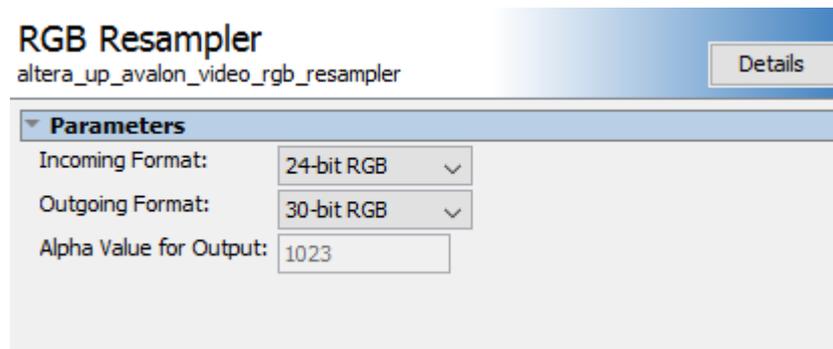


figura 59, Configuración del RGB Resampler

La **entrada** a este componente (avalon\_rgb\_sink) proviene de la salida (avalon\_vip\_to\_raw\_source) de la IP **VIP to RAW Adapter Bridge**, mientras que **la salida** (avalon\_rgb\_source) se conecta directamente a la entrada (avalon\_vga\_sink) del bloque **VGA Controller**.

#### VGA Controller

Este es el mismo componente mencionado en la opción anterior que controla el DAC de salida al monitor VGA. Sin embargo, en este caso se configura de forma diferente como se muestra a continuación:



figura 60, Configuración de la IP VGA Controller en la segunda opción

Esta IP requiere de una frecuencia de reloj diferente en función de la resolución empleada. El motivo de haber utilizado en este caso esta resolución se debe a que es la más cercana a la frecuencia típica del sistema (50MHz), en este caso son 65MHz. Una resolución intermedia (800 x 600) requiere un reloj de 40MHz, inferior a la frecuencia típica del sistema (50MHz), y el bloque JTAG Uart requiere una frecuencia mínima de 50MHz, por eso se elige como resolución 1024 x 768.

Esta configuración permite que todo el sistema funcione a una frecuencia ligeramente superior a la típica (50MHz -> 65MHz) y se mantienen los 60 fps proporcionados por la cámara ya que el escalado lo realiza un único bloque (Scaler II).

La instanciación de esta IP genera las siguientes señales de entrada/salida:

Señales de entrada/salida generadas por la IP VGA Controller		
Señal	Input/Output/Inout	Tamaño
VGA_CLK	Output	1 bit
VGA_HS	Output	1 bit
VGA_VS	Output	1 bit
VGA_BLANK_N	Output	1 bit
VGA_SYNC_N	Output	1 bit
VGA_R	Output	8 bits
VGA_G	Output	8 bits
VGA_B	Output	8 bits

Tabla 11, Conexionado de la IP VGA Controller



**En caso de utilizar esta segunda opción ya no se puede emplear la IP System and SDRAM Clocks for DE boards como en el caso anterior. Para poder usar la Opción 2 se debe instanciar una IP Avalon ALTPLL y configurar 2 salidas de reloj de 65MHz:** Una normal y otra con un desfase de -3ns (esta última para la memoria SDRAM) como se ha explicado en el apartado 3.2.

### 3.4 Bloques IP para la pantalla táctil

También la pantalla táctil requiere de ciertos componentes Hardware para funcionar.

#### LT24 Controller

Esta IP se puede descargar desde la página web de Terasic, concretamente viene incluida en el kit software denominado “System CD” que puede encontrarse al navegar por la web de Terasic y buscando el apartado donde está la pantalla LT24.

La IP LT24 Controller no tiene nada configurable, es una IP bastante sencilla cuya única labor consiste en conectar los pines de datos del display al NIOS II para poder manejar esas señales vía software (internamente es un puerto PIO).

La instanciación de esta IP genera las siguientes señales:

Señales de entrada/salida generadas por la IP LT24 Controller		
Señal	Input/Output/Inout	Tamaño
LT24_CS_N	Output	1 bit
LT24_RS	Output	1 bit
LT24_D	Output	16 bits
LT24_RD_N	Output	1 bit
LT24_WR_N	Output	1 bit
LT24_LCD_ON	Output	1 bit

Tabla 12, Señales de entrada/salida generadas por la IP LT24 Controller



**La señal LT24 LCD ON debe ser controlada manualmente.** Se puede fijar con 1'b1 en el archivo “TOP” de Quartus, o bien se controla vía un switch o vía software conectando este pin a un puerto PIO previamente instanciado. De lo contrario la retroiluminación de la pantalla estará apagada y no se visualizará nada.

### 3-Wire SPI

El controlador de la parte táctil de la pantalla (touchpanel) emplea un control mediante SPI. Se requiere instanciar esta IP para ello. La configuración de la IP es la siguiente:

The screenshot shows the configuration interface for the 3-Wire SPI IP block. It is divided into four sections:

- Master/Slave:** Type is set to Master. Number of select (SS\_n) signals is 1. SPI clock (SCLK) rate is 32000 Hz. Actual clock rate is 31996.0 Hz. Target delay and Actual delay are both 0.0 ns.
- Data register:** Width is 8 bits. Shift direction is MSB first.
- Timing:** Clock polarity is 0. Clock phase is 0.
- Synchronizer Stages:** Insert Synchronizers is unchecked. Depth is 2.

figura 61, Configuración de la IP 3-Wire SPI

La instanciación de esta IP genera las siguientes señales:

Señales de entrada/salida generadas por la IP 3-Wire SPI		
Señal	Input/Output/Inout	Tamaño
TOUCH_SPI_MISO	Input	1 bit
TOUCH_SPI_MOSI	Output	1 bit
TOUCH_SPI_SCLK	Output	1 bit
TOUCH_SPI_SS_n	Output	1 bit

Tabla 13, Señales de entrada/salida generadas por la IP 3-Wire SPI

### PIO (Parallel I/O)

Se requiere la instanciación de 3 puertos PIO, dos de ellos para el touch y uno para el LT24.

El primero de ellos se encarga de generar interrupciones cuando se detecta que se ha tocado la parte táctil de la pantalla:

**Basic Settings**

Width (1-32 bits):

Direction:

Bidir

Input

InOut

Output

Output Port Reset Value:

---

**Output Register**

Enable individual bit setting/clearing

---

**Edge capture register**

Synchronously capture

Edge Type:

Enable bit-clearing for edge capture register

---

**Interrupt**

Generate IRQ

IRQ Type:

**Level:** Interrupt CPU when any unmasked I/O pin is logic true  
**Edge:** Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled

---

**Test bench wiring**

Hardwire PIO inputs in test bench

Drive inputs to:

figura 62, Configuración de la IP PIO para las interrupciones del touch

El segundo se emplea como indicador de que el controlador del touch no está disponible (Busy):

**Basic Settings**

Width (1-32 bits):

Direction:

Bidir

Input

InOut

Output

Output Port Reset Value:

---

**Output Register**

Enable individual bit setting/clearing

---

**Edge capture register**

Synchronously capture

Edge Type:

Enable bit-clearing for edge capture register

---

**Interrupt**

Generate IRQ

IRQ Type:

**Level:** Interrupt CPU when any unmasked I/O pin is logic true  
**Edge:** Interrupt CPU when any unmasked bit in the edge-capture register is logic true. Available when synchronous capture is enabled

---

**Test bench wiring**

Hardwire PIO inputs in test bench

Drive inputs to:

figura 63, Configuración de la IP PIO para indicar que el controlador del touch no está disponible

Por último, se emplea un PIO adicional como reset para el display activo a nivel bajo:

figura 64, Configuración de la IP PIO para el reset activo a nivel bajo del Display

Al insertar estas IPs en el sistema se generan las siguientes señales de reloj desde el punto de vista del Display y el Touchscreen:

Señales de entrada/salida generadas por la IP PIO		
Señal	Input/Output/Inout	Tamaño
TOUCH_PENIRQ_n	Input	1 bit
TOUCH_BUSY	Input	1 bit
LT24_RESET_n	Output	1 bit

figura 65, Señales de entrada/salida generadas por la IP PIO

### Avalon ALTPLL

Todas las IPs anteriores no van conectadas al reloj del sistema. Es necesario generar una señal de reloj adicional empleando la IP Avalon ALTPLL, en este caso será un reloj de 150MHz. La configuración del bloque es la siguiente:

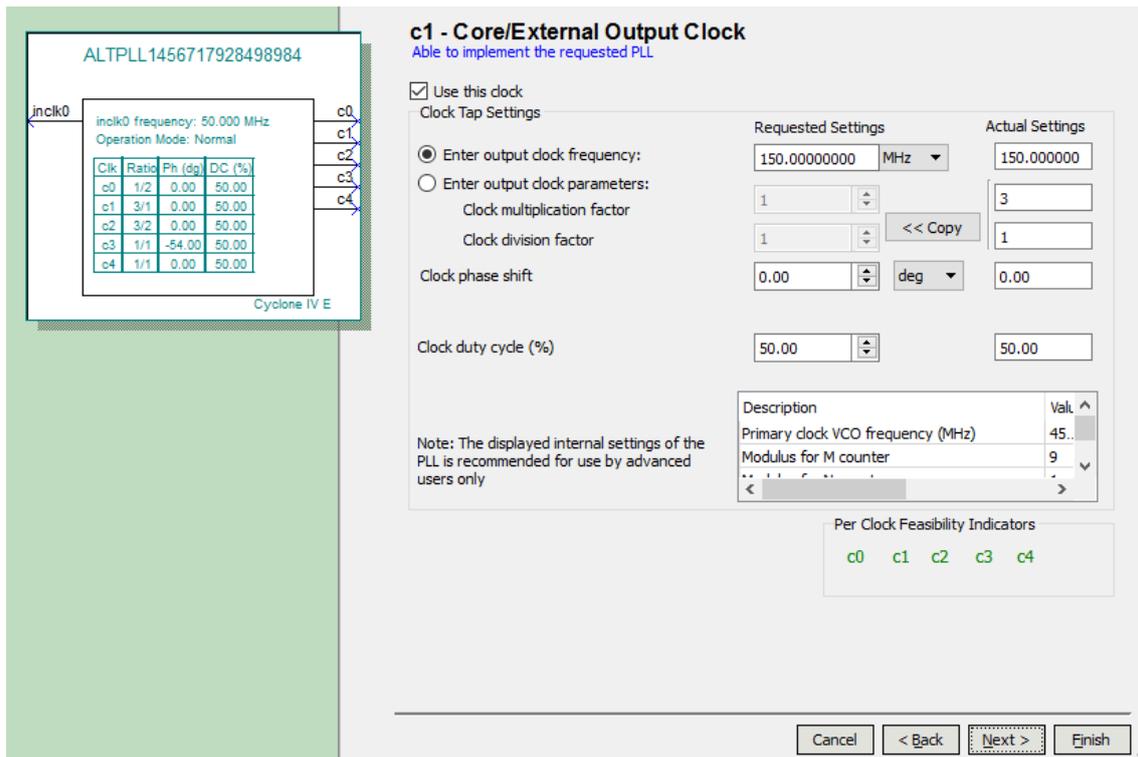


figura 66, Configuración de la IP Avalon ALTPLL para el reloj de la pantalla táctil

El resto de pestañas de la IP se dejan por defecto, solamente es necesario configurar la mostrada en la figura.

### Timer

Para poder controlar correctamente la pantalla táctil vía software es necesario instanciar un temporizador, puesto que se usa como referencia temporal del ADC del touch como se verá más adelante.

El timer se configura de la siguiente manera:



figura 67, Configuración de la IP Timer

Todas estas IPs van conectadas al puerto data master del NIOS II, de forma que su control se hace vía software. Este bloque se conecta al reloj del sistema, no al ALTPLL del apartado anterior.

### 3.5 Bloques IP adicionales empleados

Hasta ahora se han descrito las IPs que conforman la estructura básica del sistema NIOS II, de la parte de imagen y la pantalla táctil. Adicionalmente a estas IPs se han empleado algunas IPs adicionales para aportar una mayor funcionalidad al proyecto.

#### SD Card Interface

Se pretende utilizar una tarjeta SD para almacenar fotografías realizadas. Para ello se instancia en Qsys la IP del University Program SD Card Interface. Esta IP no requiere configuración alguna.

La instanciación de esta IP genera las siguientes señales:

Señales de entrada/salida generadas por la IP SD Card Interface		
Señal	Input/Output/Inout	Tamaño
SD_cmd	Inout	1 bit
SD_dat	Inout	1 bit
SD_dat3	Inout	1 bit
SD_clock	Output	1 bit

Tabla 14, Señales de entrada/salida generadas por la IP SD Card Interface

Este interfaz se conecta al bus data master del NIOS II para su control vía software.

#### Generic Tri-State Controller

Se desea utilizar también la memoria Flash de la placa para almacenar fotografías, como almacenamiento alternativo y más veloz que la tarjeta SD. El uso de la memoria flash requiere el instanciado de dos componentes, el primero es el Generic Tri-State Controller que se configura de la siguiente manera:

Signal Selection | Signal Timing | Signal Polarities

Address width:

Data width:

Byteenable width:

Bytes per word:

**Enable the following signals:**

Refer to the Avalon Interface Specifications for definitions of these signals: [http://www.altera.com/literature/manual/mnl\\_avalon\\_spec.pdf](http://www.altera.com/literature/manual/mnl_avalon_spec.pdf)

readdata

writedata

read

write

begintransfer

byteenable

chipselect

lock

address

waitrequest

writebyteenable

outputenable

resetrequest

irq

reset output

---

**Parameters**

Is memory device

---

**Module Assignments**

Parameter	Value
embeddedsw.CMacro.SETUP_VALUE	0
embeddedsw.CMacro.WAIT_VALUE	0
embeddedsw.CMacro.HOLD_VALUE	0
embeddedsw.CMacro.TIMING_UNITS	ns
embeddedsw.CMacro.SIZE	8388608u
embeddedsw.memoryInfo.MEM_INIT_DATA_WIDTH	8

+ -

---

**Parameters**

Use the module assignments to identify your components to downstream embedded software tools. A value of 1 identifies the parameter as true, a value of 0 identifies it as false.

Note: For memory devices, the module assignment **embeddedsw.CMacro.SIZE = Memory Size in Bytes** must be defined and it must equal the size of the memory. Value should be an unsigned integer.

---

**Avalon Connection Point Assignments**

Parameter	Value
embeddedsw.configuration.isFlash	1
embeddedsw.configuration.isMemoryDevice	1
embeddedsw.configuration.isNonVolatileStorage	1

+ -

figura 68, Pestaña Signal Selection de la IP Generic Tri-State Controller

Signal Selection | **Signal Timing** | Signal Polarities

Read wait time: 160

Write wait time: 160

Setup time: 60

Data hold time: 60

Maximum pending read transactions: 3

Turnaround time: 2

Timing units: Nanoseconds ▾

Read latency: 2

Chipselect through read latency

*figura 69, Pestaña Signal Timing de la IP Generic Tri-State Controller*

Signal Selection | Signal Timing | **Signal Polarities**

**Enable active low polarity on the following signals:**

read

lock

write

chipselect

byteenable

outputenable

writebyteenable

waitrequest

begintransfer

resetrequest

irq

reset output

*figura 70, Pestaña Signal Polarities de la IP Generic Tri-State Controller*

**La salida de esta IP (tcm) se conecta a la entrada de la IP Tri-State Conduit Bridge (tcs).**

### Tri-State Conduit Bridge

Esta es la IP que genera los pines físicos del bloque anterior. No tiene nada configurable, su instanciación genera las siguientes señales:

Señales de entrada/salida generadas por la IP Tri-State Conduit Bridge		
Señal	Input/Output/Inout	Tamaño
TCM_ADDRESS_OUT	Output	23 bits
TCM_READ_N_OUT	Output	1 bit
TCM_WRITE_N_OUT	Output	1 bit
TCM_DATA_OUT	Input	8 bits
TCM_CHIPSELECT_N_OUT	Output	1 bit

Tabla 15, Señales de entrada/salida generadas por la IP Tri-State Conduit Bridge



Existen dos señales adicionales para 2 pines de la memoria Flash: RST\_n y WP\_n. El primero es el Reset del chip, el segundo el Write Protect. **Estos pines o bien se fijan con 1'b1 en el archivo "TOP" del proyecto o bien se manejan vía software/hardware** (puede ser un botón en el caso del Reset y un Switch en el caso del Write Protect), **pero es importante que queden controlados manualmente de alguna manera, sino la memoria no funcionará.**

### Parallel Port

Existen en el University Program una IP que instancia diversos dispositivos de la placa directamente. Esta IP permite instanciar muy fácilmente los LEDs, Switches, Pulsadores, 7 Segmentos, Display Alfanumérico, etc.

En este caso se realiza una instanciación de los LEDs Verdes (que sirven como indicador del valor de la exposición) y de los Pulsadores (para el control de la exposición). Esta IP es realmente un puerto PIO normal pre-configurado para los dispositivos de la placa.

Las configuraciones son las siguientes:

**Configurations**  
DE-Series Board: DE2-115  
 Create custom parallel port

**Presets**  
I/O device: Pushbuttons  
LEDs Colour: Green  
Seven Segment Digits: 3 to 0  
Expansion Header: GPIO 0 (JP1)

**Basic Settings (Preset)**  
Data Width: 4  
Port Direction: Input only

**Edge Capture Register**  
 Synchronously Capture  
Capture on which edge: Rising  
 Generate IRQ

figura 71, Configuración de la IP Parallel Port para los pulsadores anti-rebote

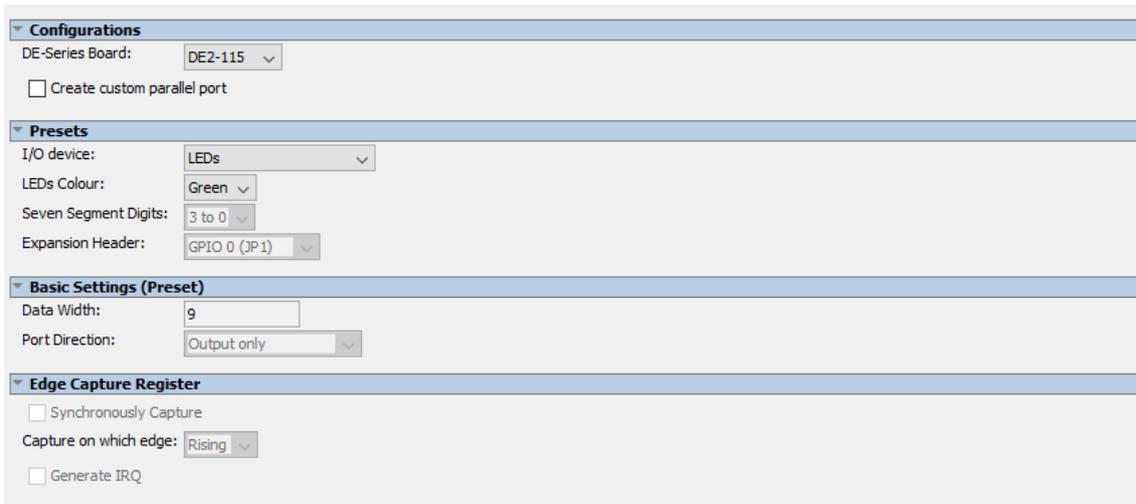


figura 72, Configuración de la IP Parallel Port para los LEDs Verdes

Estas IPs simplemente generan cada una una señal del mismo tamaño del que se han instanciado: 4 bits para los pulsadores y 9 bits para los LEDs.

### 3.6 Modificación de la IP Audio and Video Config

Para la obtención de la mayor calidad de imagen y mejor tasa de frames por segundo es recomendable realizar una pequeña modificación en la IP "Audio and Video Config".

El acceso a los archivos que la componen se encuentra en la carpeta de Altera en la ruta de instalación del sistema (Ej: C:\altera\_lite\15.1), y, una vez dentro, se debe acceder a IP\Altera\University\_Program\audio\_video\altera\_up\_avalon\_audio\_and\_video\_config.

Hay que modificar dos archivos. En primer lugar, se debe abrir el fichero "altera\_up\_avalon\_audio\_and\_video\_config\_hw.tcl" y modificar el "else" que se encuentra en la línea 614. Está indicado con un comentario de # "640 x 480". Debe quedar como se indica a continuación:

```

613     } else {
614         # "640 x 480"
615         set d5m_column_size [ format "D5M_COLUMN_SIZE:16'd2559" ]
616         set d5m_row_size    [ format "D5M_ROW_SIZE:16'd1919" ]
617         set d5m_column_bin  [ format "D5M_COLUMN_BIN:16'h0033" ]
618         set d5m_row_bin     [ format "D5M_ROW_BIN:16'h0033" ]
619     }

```

figura 73, Modificación del tcl de la IP Audio and Video Config

Esta modificación hará que el sensor funcione en modo binning 4x, lo cual permite tasas de frames por segundo superiores a 60fps con tiempos de exposición bajos y mejor calidad de imagen con poca luz.

Una vez guardados los cambios debe modificarse a continuación el fichero "altera\_up\_av\_config\_auto\_init\_d5m.v". Deben modificarse 3 valores en el campo "always (\*)" que comienza en la línea 102. Este always debe modificarse como se indica a continuación:

```

101 // Internal Assignments
102 always @(*)
103 begin
104     case (rom_address)
105     0 : data    <= {8'hBA, 8'h00, 16'h0000};
106     1 : data    <= {8'hBA, 8'h20, 16'hc020}; // Mirror Row and Columns // Def: hc000
107     2 : data    <= {8'hBA, 8'h09, exposure}; // Exposure
108     3 : data    <= {8'hBA, 8'h05, 16'h038A}; // H_Blanking
109     4 : data    <= {8'hBA, 8'h06, 16'h0019}; // V_Blanking
110     5 : data    <= {8'hBA, 8'h0A, 16'h8000}; // change latch
111     6 : data    <= {8'hBA, 8'h2B, 16'h000b}; // Green 1 Gain
112     7 : data    <= {8'hBA, 8'h2C, 16'h000f}; // Blue Gain
113     8 : data    <= {8'hBA, 8'h2D, 16'h000f}; // Red Gain
114     9 : data    <= {8'hBA, 8'h2E, 16'h000b}; // Green 2 Gain
115     10 : data   <= {8'hBA, 8'h10, 16'h0051}; // set up PLL power on
116     11 : data   <= {8'hBA, 8'h11, 16'h1407}; // PLL_m_Factor<<8+PLL_n_Divider Def: h1807
117     12 : data   <= {8'hBA, 8'h12, 16'h0002}; // PLL_p1_Divider Def: h0002
118     13 : data   <= {8'hBA, 8'h10, 16'h0053}; // set USE PLL
119     14 : data   <= {8'hBA, 8'h98, 16'h0000}; // disable calibration
120     `ifdef ENABLE_TEST_PATTERN
121     15 : data   <= {8'hBA, 8'hA0, 16'h0001}; // Test pattern control
122     16 : data   <= {8'hBA, 8'hA1, 16'h0123}; // Test green pattern value
123     17 : data   <= {8'hBA, 8'hA2, 16'h0456}; // Test red pattern value
124     `else
125     15 : data   <= {8'hBA, 8'hA0, 16'h0000}; // Test pattern control
126     16 : data   <= {8'hBA, 8'hA1, 16'h0000}; // Test green pattern value
127     17 : data   <= {8'hBA, 8'hA2, 16'h0FFF}; // Test red pattern value
128     `endif
129     18 : data   <= {8'hBA, 8'h01, 16'h0036}; // set start row   Def: 36
130     19 : data   <= {8'hBA, 8'h02, 16'h0018}; // set start column  Def: 10
131     20 : data   <= {8'hBA, 8'h03, DSM_ROW_SIZE}; // set row size
132     21 : data   <= {8'hBA, 8'h04, DSM_COLUMN_SIZE}; // set column size
133     22 : data   <= {8'hBA, 8'h22, DSM_ROW_BIN}; // set row mode in bin mode
134     23 : data   <= {8'hBA, 8'h23, DSM_COLUMN_BIN}; // set column mode in bin mode
135     24 : data   <= {8'hBA, 8'h49, 16'h01A8}; // row black target
136     default : data <= {8'h00, 8'h00, 16'h0000};
137     endcase
138 end

```

figura 74, Modificaciones a realizar en el fichero de auto-inicialización de la IP Audio and Video Config

- En el case 1 se debe cambiar el valor “hc000” por defecto por el valor “hc020” en el último campo de la concatenación. Esto habilita la suma de columnas, permitiendo en modo binning una mayor ganancia, es decir, mejor rendimiento con poca luz, lo cual permite conseguir la misma luminosidad con tiempos de exposición menores.
- En el case 11 se debe cambiar el valor por defecto “h1807” por el valor “h1407”. Esto aumenta la frecuencia del reloj de píxeles (Pixel Clock) de 85MHz a 95MHz, llevándolo cerca del límite de 96MHz.
- En el case 19 debe cambiarse el valor “h0010” por “h0018” para permitir que el modo binning funcione correctamente. Este es el valor mínimo cuando se usa el modo binning 4x como indica el datasheet del sensor ( $16n + 8$ , siendo n un número entero con valor mínimo de 1).

Una vez realizadas estas modificaciones debe volver a compilarse el Qsys y el proyecto de Quartus completo en caso de haber creado el proyecto sin estas modificaciones previamente.

### 3.7 Preparación del proyecto para la compilación

En todos los pasos anteriores se ha explicado la configuración de los distintos bloques dentro de Qsys. Una vez se ha completado el sistema en Qsys se debe generar el sistema pulsando en “Generate -> Generate HDL”. Una vez acabe, Qsys habrá creado un archivo con extensión “.qip” en la ruta que se le haya indicado. Hecho esto se puede proceder a preparar el proyecto dentro de Quartus II.

Hay que tener dos cosas en cuenta en caso de modificar el proyecto.

La primera es que dentro del proyecto de Quartus debe estar incluido el archivo con extensión “.qip” creado en el paso anterior. Simplemente haciendo doble click sobre “Files” aparece una ventana desde donde se puede añadir el archivo.

La segunda cosa a tener en cuenta es que este proyecto es especialmente sensible a los relojes, ya que se emplean bastantes PLLs para generar las diferentes señales de reloj para las distintas IPs. Por ello, cada vez que se modifica algún reloj en Qsys y se regenera el sistema, se deben seguir los siguientes pasos en Quartus:

1. Si existe en el navegador de archivos algún fichero con extensión “.sdc” este debe ser eliminado.

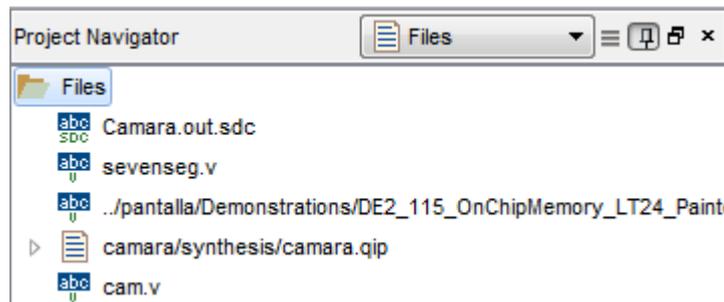


figura 75, Ejemplo de un navegador que contiene un fichero con extensión .sdc

2. Una vez eliminado se compila el proyecto completo o se puede también pinchar únicamente sobre “Analysis and Synthesis” y sobre “Fitter (Place and Route)” y, una vez compilado, se inicia la aplicación Timequest Timing Analyzer pinchando sobre un símbolo con forma de reloj o en el menú tools:

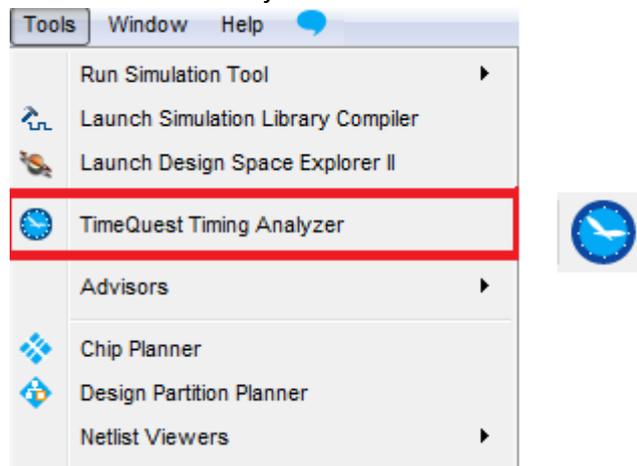


figura 76, TimeQuest Timing Analyzer en el menú Tools (izq.) y el símbolo de la barra de herramientas (dcha.)

3. Una vez abierto el TimeQuest, debe pincharse bien sobre “Report Clocks” o bien sobre “Report All Summaries” en la barra “Tasks” que aparece en la parte inferior izquierda de la pantalla:

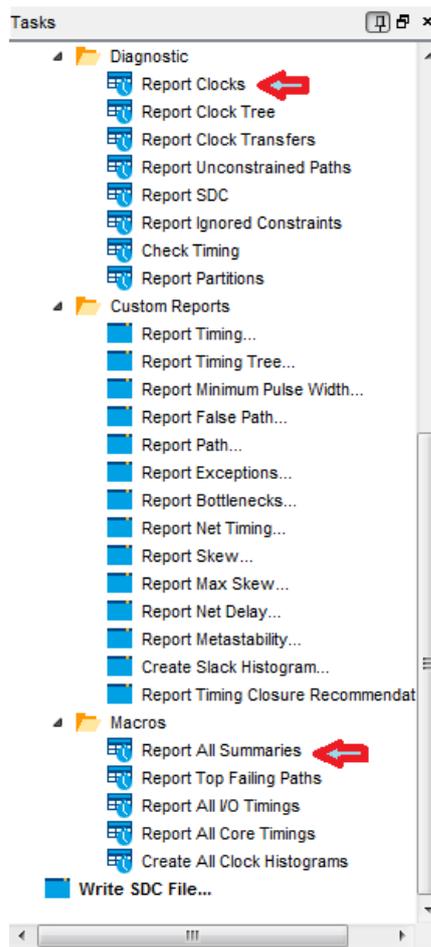


figura 77, Barra de tareas del TimeQuest Timing Analyzer. Se debe pinchar sobre una de las opciones marcadas.

4. Aparecerán los relojes del sistema que deberían ser correctos. Si no lo son, se puede intentar el pinchar en la opción “Constraints” de la barra de tareas superior, y dentro “Derive PLL Clocks...” con las opciones por defecto. Si aun así los relojes son incorrectos es posible que el fichero “.qip” haya quedado obsoleto y en ese caso es recomendable regenerar el proyecto en Qsys y recompilar el proyecto de Quartus.
5. Una vez los relojes son correctos se debe pinchar en “Constraints” y una vez dentro se debe seleccionar “Derive Clock Uncertainty”. En la ventana que se abre se seleccionan ambas opciones y se hace click en “Run”:

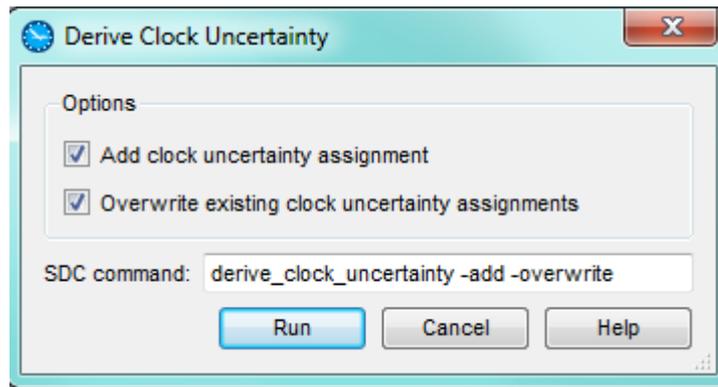


figura 78, Ventana Derive Clock Uncertainty del TimeQuest

6. Una vez acabado se selecciona “Write SDC File en la barra inferior izquierda mostrada anteriormente”. Se generará un fichero con extensión “.sdc” que se deberá añadir al proyecto de la misma forma que el “.qip” explicado anteriormente.

Si no se genera un fichero “.sdc” correcto suelen aparecer problemas con el módulo de la cámara como por ejemplo que no se muestre el Live View por pantalla. Por eso es importante generar correctamente este fichero “.sdc”.

## 4- DISEÑO SOFTWARE DEL PROYECTO

En este apartado se describen los detalles respecto al software que es ejecutado sobre el NIOS II. Esta CPU tiene acceso a todos los componentes del sistema que se encuentren conectados al puerto "data master" de la CPU (y puede ejecutar software desde aquellos conectados al puerto "instruction master").

A continuación, se muestra un diagrama de bloques con la estructura general del software:

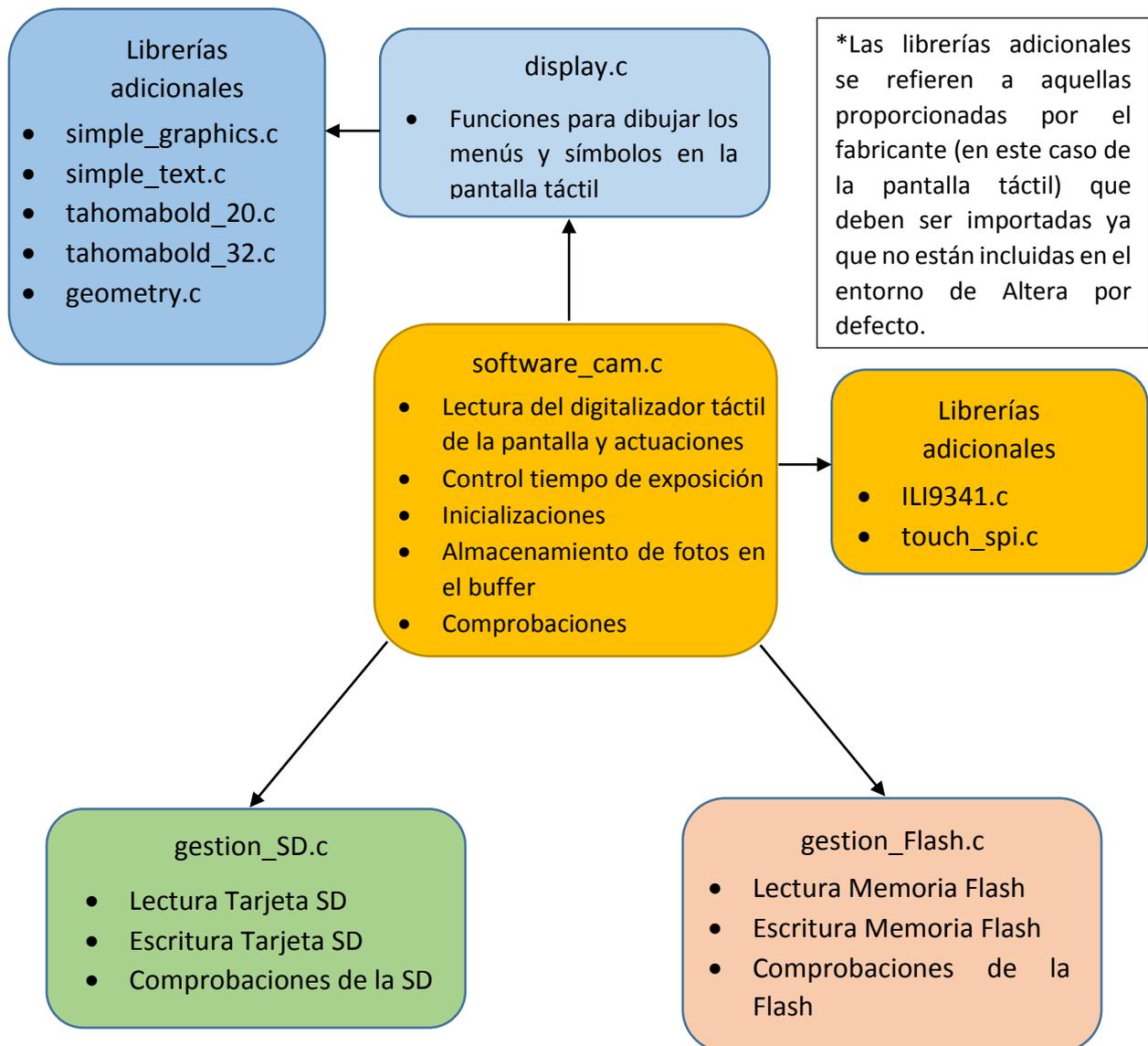


figura 79, Diagrama de bloques de la estructura general del software

Adicionalmente, algunos componentes pueden enviar interrupciones a la CPU, lo cual evita comprobaciones intermitentes hacia los dispositivos, traduciéndose en un mejor rendimiento global. Para ello hay algunos componentes conectados al controlador de interrupciones de la CPU:

Componentes con capacidad para enviar interrupciones a la CPU	
Componente	Prioridad
Timer (timer_0)	0
3-Wire SPI (touch spi)	1
PIO (touch_pen_irq_n)	2
PIO (botones)	3
JTAG Uart	4

Tabla 16, Componentes con capacidad para enviar interrupciones a la CPU

Una vez se encuentre el NIOS II instanciado en el archivo de mayor nivel de Quartus II, se puede proceder con el software empleando Eclipse.

#### 4.1 Aspectos previos al diseño software

Para que todo funcione correctamente hay que asegurarse previamente de que el entorno Eclipse está correctamente configurado. Es especialmente importante que el timer\_0 esté configurado como el timer del sistema. Para ello se debe hacer click derecho sobre la parte bsp del proyecto, en este caso:

Click Izquierdo en **cam\_soft\_bsp** -> **BSP Editor...** -> **Settings** -> **Common** -> **hal** -> **sys\_clk\_timer: Timer\_0**.

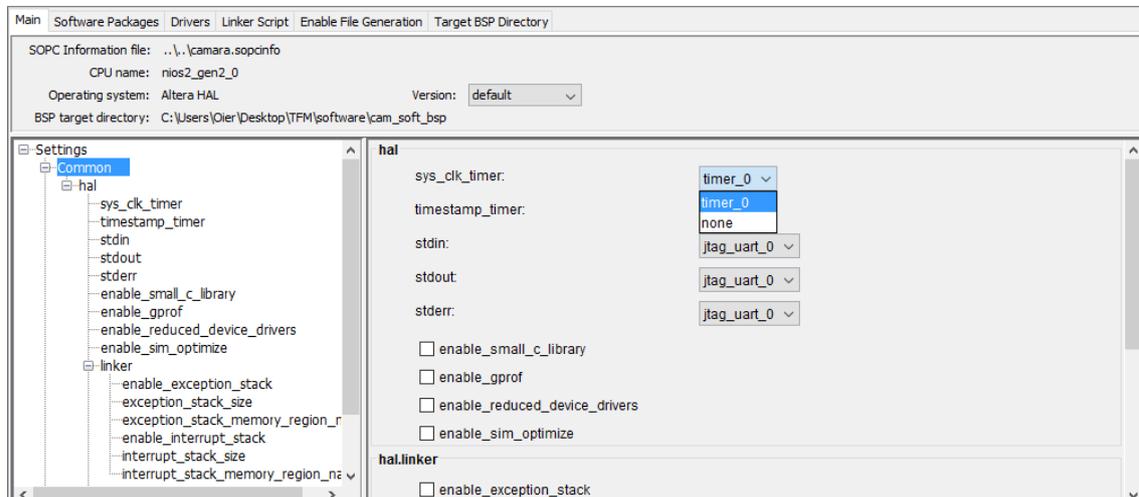


figura 80, Configuración del System Clock Timer

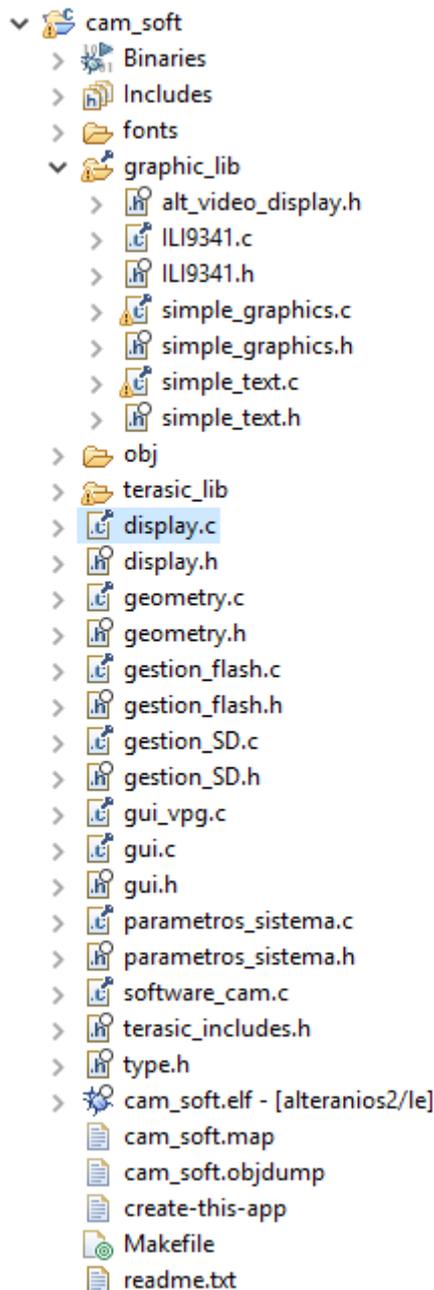


figura 81, Ejemplo del navegador del proyecto software.

Por otra parte, para el diseño de los gráficos de la pantalla táctil Terasic proporciona una serie de librerías que facilitan enormemente la labor, además de que al ser el Display y la parte táctil controladas vía software también Terasic proporciona las funciones necesarias para ello. Estos ficheros se encuentran en la web de Terasic, en el apartado “Resources” de “LT24 Card”. Una vez descomprimido, se pueden añadir al proyecto de la siguiente manera:

Click izquierdo en **cam\_soft** -> **Import...** -> **Existing Projects into Workspace** y ahí se selecciona la ruta donde se encuentra el proyecto descomprimido.

Posiblemente aparecerán una serie de errores debido a que esos ficheros apuntan al proyecto anterior, además de porque algunos componentes no se llamarán igual.

Para solucionar esto simplemente se debe indicar las rutas correctas en los #include de los ficheros.

Para adaptar los ficheros, se procede de la siguiente manera:

- Si el fichero que se desea incluir está en el directorio raíz del proyecto se incluye directamente. Por ejemplo, en el ejemplo visual de arriba los archivos: display.c, display.h, geometry.c, geometry.h, etc, se encuentran en el directorio raíz. Para incluir por ejemplo display.h y geometry.h se incluyen directamente:

```
#include "display.h"
```

```
#include "geometry.h"
```

- Si el fichero buscado se encuentra dentro de una carpeta del directorio raíz se pone la ruta completa. En este caso, dentro de graphic\_lib están los archivos:

alt\_video\_display.h, ILI9341.h, etc. Para incluir estos archivos se ha de escribir la ruta completa:

```
#include "graphic_lib/alt_video_display.h"
```

```
#include "graphic_lib/ILI9341.h"
```

A veces ocurre que cuando se va a inicializar un dispositivo empleando el HAL de Altera no lo encuentra a pesar de que system.h se encuentre incluido en el fichero. En ese caso se puede copiar el #define correspondiente al fichero (copiar – pegar desde system.h) o bien se puede copiar directamente el valor numérico de ese #define dentro de la función. No obstante, se ha de tener cuidado porque si se cambia algo en Qsys que provoque un cambio en la dirección de memoria de ese componente, dejará de funcionar. Para evitar esto es mejor bloquear en Qsys la dirección de memoria del componente si ya funciona correctamente:



figura 82, Bloqueo de dirección de memoria en Qsys

#### 4.2 Ficheros del proyecto y su funcionalidad

Puesto que todas las funciones en un único fichero ocuparían muchas líneas de código y sería difícil de leer el proyecto se ha ido estructurando en diferentes ficheros que definen funciones diferentes, además de los que se han importado desde el “LT24 System CD”. A continuación, se ofrece una breve descripción de cada uno de los ficheros que componen la parte software del proyecto:

Ficheros que forman el software del sistema	
Nombre del fichero	Descripción
<ul style="list-style-type: none"> <li>software_cam.c (main)</li> </ul>	Fichero principal del proyecto
<ul style="list-style-type: none"> <li>display.c</li> <li>display.h</li> </ul>	Ficheros que disponen de las funciones relativas a los menús de la pantalla táctil
<ul style="list-style-type: none"> <li>gestion_SD.c</li> <li>gestion_SD.h</li> </ul>	Ficheros que incorporan funciones para el manejo de la tarjeta SD
<ul style="list-style-type: none"> <li>gestión_flash.c</li> <li>gestión_flash.h</li> </ul>	Ficheros que incorporan funciones para el manejo de la memoria Flash

Tabla 17, Ficheros que forman el software del sistema

A continuación, se muestra una breve descripción de los ficheros importados al proyecto desde el “LT24 System CD”:

Ficheros importados desde el "LT24 System CD"		
Nombre del fichero	Ruta	Descripción
<ul style="list-style-type: none"> <li>• geometry.c</li> <li>• geometry.h</li> </ul>	Raíz	Incorpora algunas funciones simples de tratamiento geométrico. Se importan, pero no se usan en este proyecto (se pueden eliminar).
<ul style="list-style-type: none"> <li>• terasic_includes.h</li> </ul>	Raíz	Se trata de un fichero de cabecera donde se realizan diversos #includes para las funciones de Terasic. Lo emplean algunos ficheros de Terasic.
<ul style="list-style-type: none"> <li>• type.h</li> </ul>	Raíz	Es un fichero de cabecera donde se definen algunos tipos de variables para ser usadas por los ficheros de fuentes (para dibujar caracteres en el display).
<ul style="list-style-type: none"> <li>• fonts.h</li> </ul>	/fonts	Fichero para la inclusión de ficheros de fuentes
<ul style="list-style-type: none"> <li>• tahomabold_20.c</li> <li>• tahomabold_30.c</li> </ul>	/fonts	Ficheros de fuentes. En ellos se incluyen todos los caracteres que se pueden pintar en el display.
<ul style="list-style-type: none"> <li>• alt_video_display.h</li> </ul>	/graphic_lib	Fichero donde se instancia la estructura básica para el uso de la pantalla táctil.
<ul style="list-style-type: none"> <li>• ILI9341.c</li> <li>• ILI9341.h</li> </ul>	/graphic_lib	Fichero de control de la pantalla táctil. Estos ficheros son el driver software de la pantalla, y se comunican con el controlador de pantalla hardware ILI9341.
<ul style="list-style-type: none"> <li>• simple_graphics.c</li> <li>• simple_graphics.h</li> </ul>	/graphic_lib	Estos ficheros incorporan un surtido de funciones que permiten pintar muy fácilmente figuras y colores en el display.
<ul style="list-style-type: none"> <li>• simple_text.c</li> <li>• simple_text.h</li> </ul>	/graphic_lib	Ficheros que incluyen una función para pintar frases y otra para pintar caracteres en el display (empleando las fuentes definidas en fonts.h).
<ul style="list-style-type: none"> <li>• debug.c</li> <li>• debug.h</li> <li>• I2C.c</li> <li>• I2C.h</li> <li>• queue.c</li> <li>• queue.h</li> </ul>	/terasic_lib	Ficheros para debug y para comunicación I2C. Son usados por algunos ficheros de Terasic.

<ul style="list-style-type: none"> <li>• touch_spi.c</li> <li>• touch_spi.h</li> </ul>	/terasic_lib	Ficheros que incluyen funciones para el control de la parte táctil de la pantalla. Se puede considerar el controlador software de la misma.
--	--------------	---

Tabla 18, Ficheros importados desde "LT24 System CD" para el manejo de la pantalla táctil

### 4.3 Detalles específicos de la programación

Si se mira el código que forma el software del proyecto pueden surgir dudas sobre el porqué de algunas decisiones. En este apartado se detallan los motivos de las decisiones que se han ido tomando a lo largo de la programación del software.

#### Definiciones (Ficheros: software\_cam.c y gestion\_flash.c)

El fichero software\_cam.c dispone de las siguientes definiciones:

```
#define TOUCH_PEN_IRQ_N_BASE 0x8081470
#define TOUCH_PEN_IRQ_N_IRQ 2
#define TIMER_1_BASE 0x8081400
#define ON_CHIP 305000
#define ALTURA 320
#define ANCHURA 240
#define TAM_BUFF 76250
#define NUM_BUFF 6
#define RETARDO 800000
```

Las cuatro primeras son solamente copy-paste tomadas del fichero system.h. A pesar de que dicho fichero está incluido en el proyecto, por alguna razón desconocida el entorno Eclipse no encuentra los valores correspondientes, por eso se vuelven a copiar. También podría simplemente escribirse la dirección a la que apuntan esas definiciones, pero dejándolas al principio del fichero es más cómodo, sobre todo si hay varias funciones que emplean esa dirección ya que definiéndolas solamente hay que cambiar el #define, de la otra manera hay que cambiar la dirección manualmente en todas las funciones. En el caso de ON\_CHIP se ha cambiado el nombre para hacerlo más corto, pero su valor corresponde a los #define ON\_CHIP\_MEMORY2\_0\_SIZE\_VALUE o #define ONCHIP\_MEMORY2\_0\_SPAN de system.h, cualquiera de los dos es válido.

ALTURA y ANCHURA son definiciones empleadas por las funciones que inicializan la pantalla táctil y corresponden a su resolución, que en este caso es 320 x 240.

El trabajo permite, además de almacenar las fotos en la SD y en la Flash, almacenarlas en una memoria volátil, que en este caso es la propia memoria RAM (SDRAM). Por eso hay definido un array bidimensional a modo de buffer. El valor de TAM\_BUFF no debe ser alterado (salvo que se cambie la resolución en Qsys), mientras que el NUM\_BUFF se puede modificar libremente para permitir capturar más o menos fotos y almacenarlas en la RAM. Por defecto tiene valor 6.



El valor TAM\_BUFF se obtiene dividiendo por 4 el valor de ON\_CHIP (el valor en bytes de la memoria de video, en este caso  $305.000/4=76.250$ ). El motivo de esto es que cada posición de esa memoria tiene un tamaño de palabra de 1 byte. Puesto que el array bidimensional es de tipo INT, cada posición del mismo es capaz de almacenar 4 bytes (32 bits), por lo tanto, en cada posición del array se copian 4 posiciones desde la memoria ON\_CHIP (4 posiciones = 4 bytes).

Por último, la definición del RETARDO es solamente una forma de esperar a la hora de disparar en ráfaga, de forma que haya tiempo suficiente para vaciar la memoria de video y volverla a llenar con una nueva imagen.

En el caso de gestión\_flash.c solamente hay dos definiciones:

```
#define offset_base 8192
#define sectores_foto 305000
```

En el caso de offset\_base ese es el valor en bytes del primer sector (8KB). El primer sector de la Flash está reservado para almacenar ahí el nº de fotos que se encuentran almacenados en la flash a modo de cabecera, de esta forma solo con leer el primer sector es suficiente para informar del nº de fotos almacenadas en la Flash y el nº de fotos que aun caben en ella.

El segundo valor es el número de bytes a almacenar, que se corresponde con el tamaño de ON\_CHIP Memory al igual que en el caso del fichero anterior.

#### Variables (Ficheros: software\_cam.c)

Aclaraciones respecto a las siguientes variables:

```
int x=0;
int y=0;

short ign=0
```

Las dos primeras variables almacenan las coordenadas leídas por la pantalla táctil. Cada vez que se lee un valor estas variables deben ser puestas de nuevo a 0 una vez leídas, sino la ejecución podría bloquearse.

Por otro lado, la variable ign reduce a la mitad la velocidad de adquisición de datos desde la pantalla táctil. Si no se hace así una pulsación puede ser leída por el software como 2, 3 o 4 pulsaciones.

#### Manejo de memoria (Ficheros: software\_cam.c, gestión\_flash.c y gestión\_SD.c)

El funcionamiento a la hora de leer o escribir en las diferentes memorias con las que se trabaja es prácticamente el mismo con pequeñas diferencias.

Las escrituras en los diferentes almacenamientos siguen el procedimiento descrito en los siguientes diagramas de flujo.

En el caso de las lecturas/escrituras en el buffer (memoria volátil):

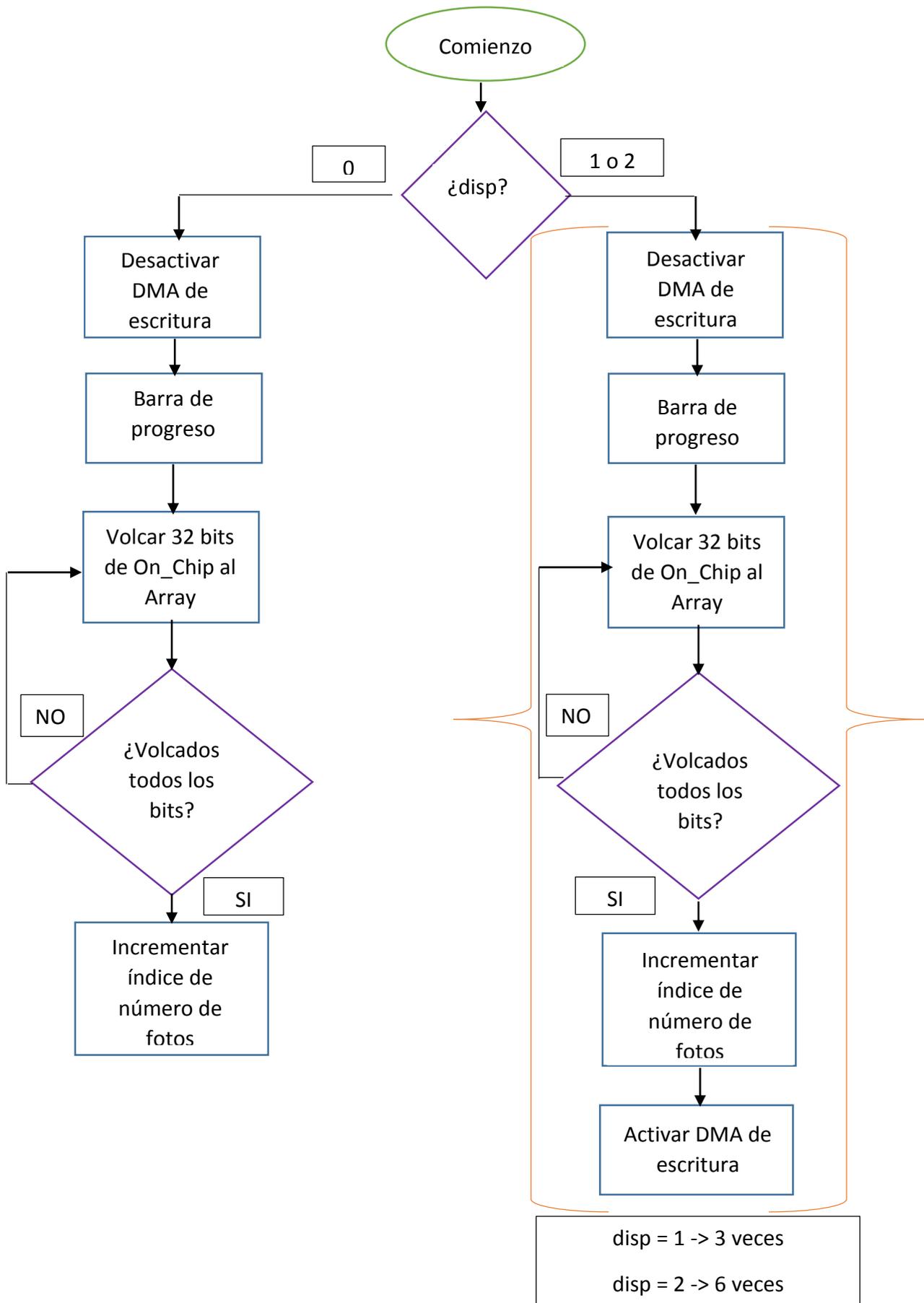


figura 83, Diagrama de flujo de la escritura de la foto en el Buffer

El diagrama de flujo superior resume la ejecución cuando el almacenamiento seleccionado es el Buffer. Si solo se selecciona la toma normal de 1 foto se ejecuta la rama de la izquierda, si se seleccionan ráfagas de 3 o 6 fotos se ejecuta la rama derecha, repitiéndose el proceso de dicha rama 3 o 6 veces en función del modo seleccionado.

En el caso del almacenamiento en la tarjeta SD el proceso es muy similar al anterior, la única diferencia radica en que existe una limitación de 64KB por cada fichero y por lo tanto una foto ocupa varios ficheros, lo cual supone crearlos e inicializar la variable que apunta al fichero que se desea utilizar.

En el caso de la memoria flash el proceso se divide en dos partes. La primera parte es exactamente la rama izquierda del diagrama de flujo de la memoria del buffer, donde se realiza el volcado de lo que hay en la memoria de vídeo a un array que lo almacena temporalmente en la memoria RAM. Posteriormente empleando las funciones del HAL de Altera se vuelca el array directamente a la memoria Flash, dicho volcado se realiza con una única función. El HAL permite el manejo por grano grueso o por grano fino. En este caso se emplea el manejo por grano grueso.

Cuando se emplean las funciones del HAL para control por grano grueso las propias funciones se encargan de ocupar los bloques suficientes en la memoria Flash para que quepan aquellos datos que se desean almacenar. Si la función que escribe el dato apunta directamente a un bloque que ya tiene datos estos son sobrescritos por el nuevo dato y, por lo tanto, el dato anterior es eliminado y no se puede recuperar.

Por otro lado, si se desease el control de la memoria Flash a nivel de bloque se emplearían las funciones de grano fino. Estas funciones permiten leer/escribir la memoria flash de bloque en bloque, de forma que se tiene un control de la misma a más bajo nivel.

En este caso no es necesaria la gestión de la flash a nivel bajo, por eso se emplean las funciones de grano grueso.

En las siguientes páginas se muestran los diagramas de flujo de la escritura de las fotos en la tarjeta SD y de la gestión de la memoria flash.

El proceso de lectura es igual, pero a la inversa en lugar de leerse la memoria On-Chip se lee el almacenamiento y se escribe en la memoria On-Chip, esto se aplica en los tres casos y leyendo el código se ve muy fácilmente.

En el caso de la tarjeta SD se escribe un fichero info.txt que contiene el número de fotos que hay almacenados en la tarjeta.

En la memoria flash la información relativa al número de fotos almacenadas se encuentra en la dirección 0 de la misma, en el primer sector.

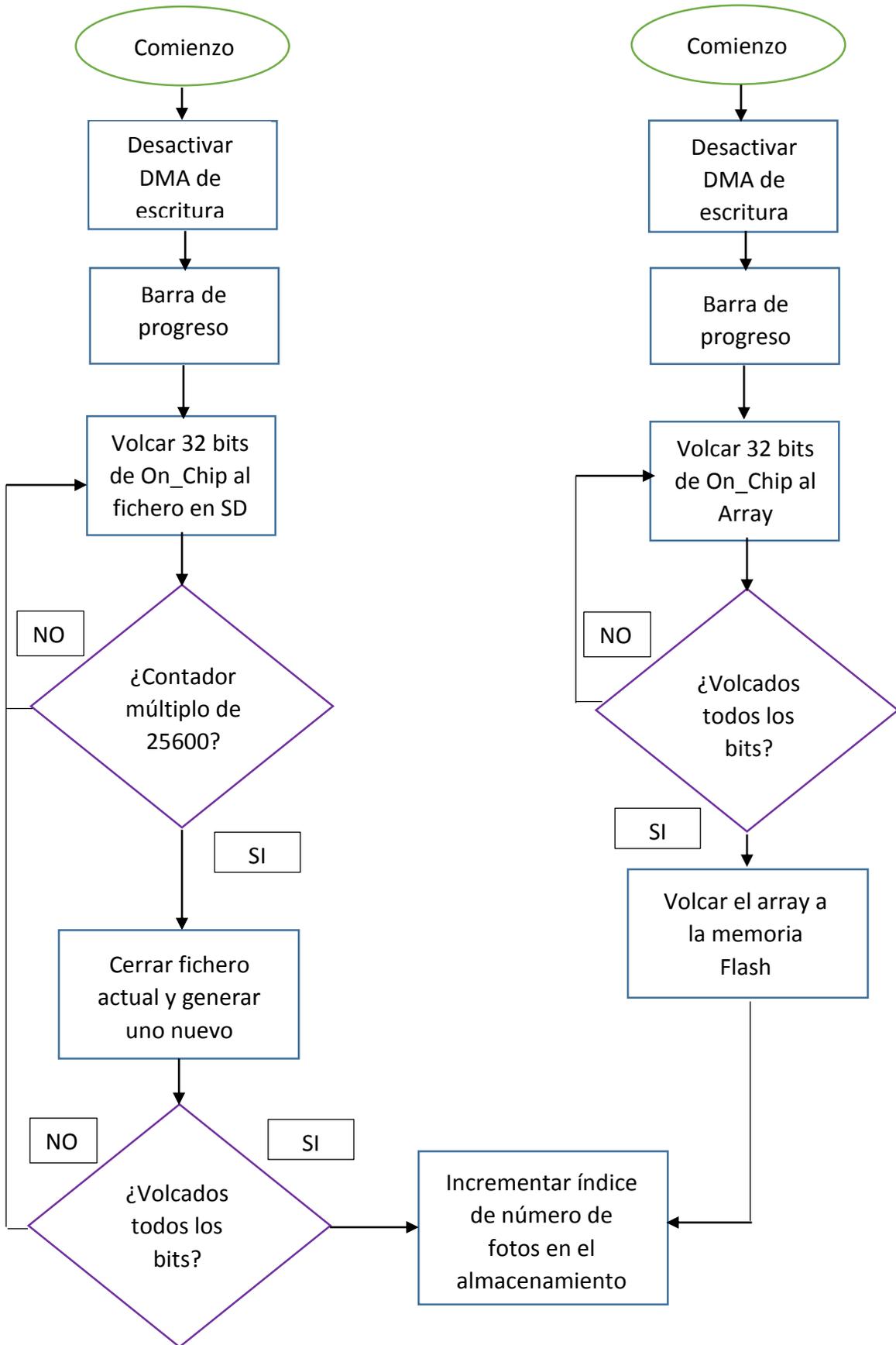


figura 84, Diagrama de flujo de escritura en la tarjeta SD (izq.) y de escritura en la memoria flash (dcha.)

### Funcionamiento general del sistema

El fichero que incluye el main (software\_cam.c) sigue todo el procedimiento de ejecución del software. Las interrupciones están habilitadas para la pantalla táctil y para los botones KEY 2 y 3.

En la ejecución, el primero paso es la inicialización de los diferentes dispositivos a los que la CPU tiene acceso, además de algunas comprobaciones como la existencia de una tarjeta SD conectada a la placa. Inmediatamente después se lee el número de fotos almacenadas en el dispositivo de almacenamiento seleccionado.

Una vez hecho esto se registran las interrupciones y la CPU permanece en un estado infinito de recepción de interrupciones. Toda la ejecución relativa al interfaz de usuario forma parte de la rutina de atención a la interrupción de la pantalla táctil.

La rutina de atención a las interrupciones de la pantalla táctil es un código bastante complejo que consta de condicionales en los que se va entrando en función de las coordenadas devueltas por la pantalla táctil.

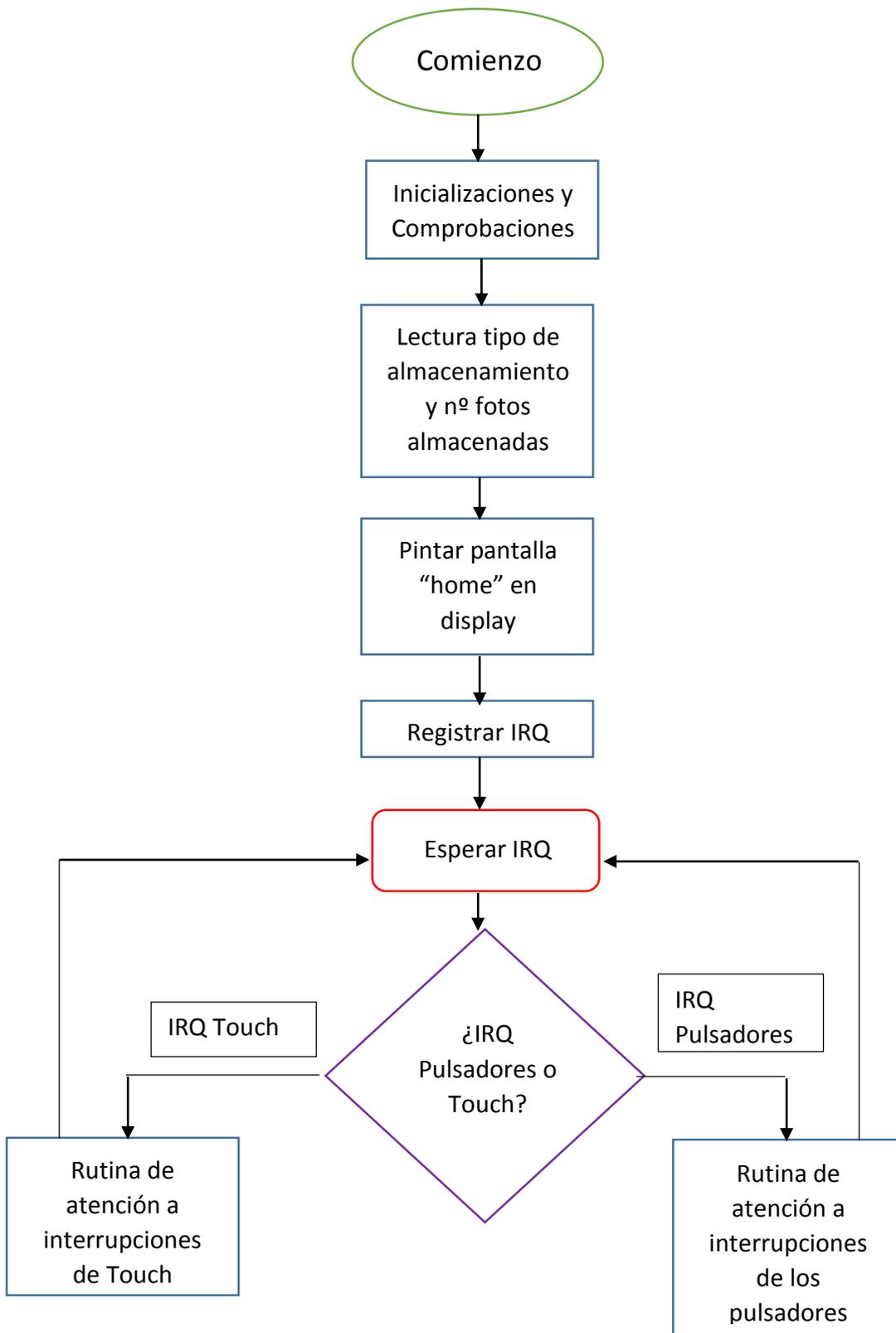


figura 85, Diagrama de flujo general del sistema

## 5- PRUEBAS DE RENDIMIENTO Y JUSTIFICACIONES DE DISEÑO

A continuación, se describen las pruebas que se han realizado a la hora de configurar algunos componentes del sistema, así como los motivos de la selección de algunos de ellos.

### 5.1 NIOS II

El componente principal desde el punto de vista del rendimiento es el NIOS II. El NIOS II dispone de varias posibilidades de configuración, las interesantes en este caso son las cachés y el predictor de saltos.

Las cachés permiten realizar copias de segmentos de la memoria principal (RAM) en una memoria cercana a la CPU. Esta memoria, al estar cerca de la CPU, es mucho más rápida, gracias a eso se agiliza el rendimiento del sistema, especialmente cuando la memoria de programa es una memoria externa. En esta aplicación la caché es crítica.

Existen dos tipos de memorias caché: Instrucciones y Datos. La primera almacena copias de instrucciones localizadas en la memoria de programa, mientras que la segunda almacena resultados de instrucciones ejecutadas que serán escritos de nuevo en la memoria. Hay que tener en cuenta que la caché ocupa elementos de memoria de la FPGA (M9K en este caso). Puesto que la memoria de video se encuentra también usando estos elementos, el máximo tamaño de la caché estará limitado a los recursos disponibles en la FPGA. En la práctica es posible cualquier combinación donde la suma de ambas cachés de un tamaño de 80KB como máximo, lo cual en la práctica sería 64+16, 16+64 o 32+32 (Instrucciones + Datos).

En el caso del predictor de saltos se trata de una unidad que intenta predecir el camino que seguirán a través del pipeline de la CPU las instrucciones de tipo branch (condicionales). El código empleado tiene bastantes estructuras “for” e “if”, por lo que también se realiza un pequeño estudio para cuantificar el impacto que esta unidad tiene en el rendimiento.

Para poder realizar estas pruebas de rendimiento se ha añadido al sistema el componente “Performance Counter” contenido dentro de las IPs de Altera. Este componente permite medir el tiempo que tarda en ejecutarse una porción del código, mostrando los resultados en un informe.

Para estas pruebas de rendimiento se realizan las siguientes medidas:

- Tiempo que se tarda en pintar la primera pantalla “home” en el display.
- Tiempo que se tarda en pintar la pantalla “camara” en el display.
- Tiempo que se tarda en pintar la pantalla “config” en el display.
- Tiempo que se tarda en pintar la pantalla “galeria” en el display, mostrando 6 fotos tomadas.
- Tiempo que se tarda en realizar un disparo en ráfaga de 6 fotografías.
- Tiempo que se tarda en pintar la pantalla “estado” en el display.

Se realizan precisamente esas pruebas en base a lo que sería el manejo del sistema en una situación práctica (acceso al menú de configuración, disparo en ráfaga, entrada en la galería, etc).

Se puede observar que la mayoría de las medidas consisten en calcular cuánto se tarda en pintar en el display. Esta es la tarea que más procesado necesita por parte de la CPU, y cuanto más compleja sea la imagen a pintar mayor tiempo se requiere.

Por otra parte, el disparo en ráfaga mide el rendimiento del NIOS II a la hora de realizar copias entre la memoria On-Chip y la memoria de programa.

#### Pruebas de rendimiento de las cachés

Puesto que esta es la parte más crítica desde el punto de vista del rendimiento de la CPU se han probado diversas configuraciones y se ha realizado un estudio completo para comprobar el impacto de la misma en esta aplicación. Para realizar estas pruebas se ha configurado el predictor de saltos en la opción de máximo rendimiento (Dinámico de 8192 posiciones).

No se ha podido probar el impacto de una caché de datos de 0KB puesto que el sistema se colgaba, por lo que en ese caso la mínima prueba ha sido con una caché de datos de 512 Bytes.

En la siguiente gráfica se muestra el impacto del tamaño de la caché L1 de instrucciones en las diferentes pruebas de rendimiento, manteniendo fijada la L1 de datos en 512 Bytes:

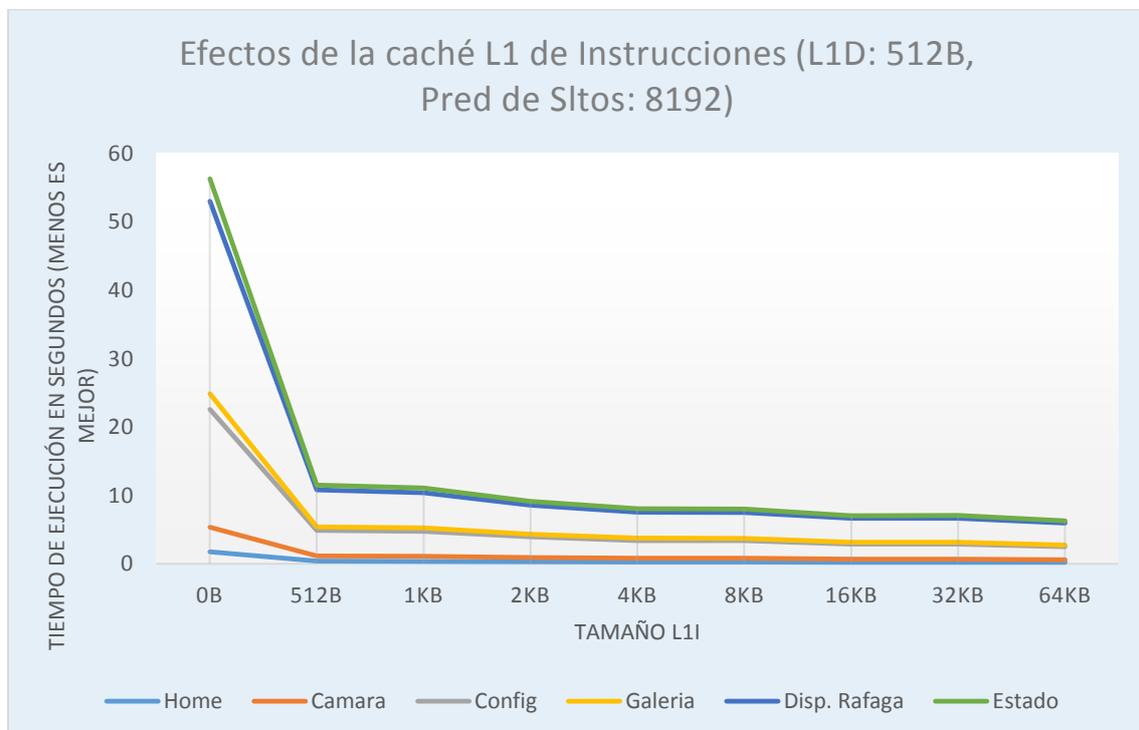


figura 86, Impacto en el rendimiento de la caché L1I

Se observa como el mayor impacto se produce simplemente entre tener y no tener caché de instrucciones. A partir de ahí va mejorando el rendimiento cada vez menos. Matemáticamente los resultados son los siguientes:

<b>Tamaño L1</b>	<b>Home</b>	<b>Dif. (%)</b>	<b>Cámara</b>	<b>Dif. (%)</b>	<b>Config</b>	<b>Dif. (%)</b>
0B	1,755		3,624		17,207	
512B	0,394	345,43%	0,753	381,27%	3,766	356,90%
1KB	0,37	6,49%	0,752	0,13%	3,649	3,21%
2KB	0,322	22,36%	0,603	24,88%	3,015	24,91%
4KB	0,283	13,78%	0,531	13,56%	2,604	15,78%
8KB	0,282	0,35%	0,535	-0,75%	2,565	1,52%
16KB	0,248	13,71%	0,454	17,84%	2,188	17,23%
32KB	0,247	0,40%	0,454	0%	2,191	-0,14%
64KB	0,215	14,88%	0,405	12,10%	1,875	16,85%

Tabla 19, Efecto de la caché L1 en el rendimiento con diferencias porcentuales 1

<b>Tamaño L1</b>	<b>Galeria</b>	<b>Dif. (%)</b>	<b>Disp. Raf</b>	<b>Dif. (%)</b>	<b>Estado</b>	<b>Dif. (%)</b>
0B	2,291		28,16		3,266	
512B	0,498	360,04%	5,413	420,23%	0,711	359,35%
1KB	0,483	3,11%	5,163	4,84%	0,697	2,01%
2KB	0,395	26,08%	4,248	27,42%	0,57	24,74%
4KB	0,341	15,84%	3,797	11,88%	0,493	15,62%
8KB	0,345	-1,16%	3,784	0,34%	0,491	0,41%
16KB	0,294	17,35%	3,464	9,24%	0,412	19,17%
32KB	0,294	0%	3,467	-0,09%	0,423	-2,60%
64KB	0,254	15,75%	3,206	8,14%	0,358	18,16%

Tabla 20, Efecto de la caché L1 en el rendimiento con diferencias porcentuales 2

En las tablas superiores se muestran las diferencias porcentuales existentes a medida que se aumenta el tamaño de la caché. Se ven diferencias enormes cuando se incorpora la caché L1 más pequeña de 512B, en el caso el disparo en ráfaga la diferencia se dispara por encima del 400% que como ya se ha mencionado antes es una prueba que estresa mucho el subsistema de memoria del sistema haciendo copias entre la memoria RAM y la memoria de video On-Chip.

Si hubiera que buscar un equilibrio en cuanto a recursos y rendimiento se podría hablar de 4KB en caso de priorizar recursos o de 16KB en caso de priorizar rendimiento manteniendo una ocupación de memoria embebida controlada. En este trabajo se opta por una L1 de 64KB puesto que existen recursos suficientes en la FPGA, opción que ofrece entre alrededor de un 8 - 18% de mejora adicional en el rendimiento.

Se observa también que los aumentos de rendimiento son escasos con tamaños de 8KB y de 32KB, a veces incluso se llega a perder algo de rendimiento al pasar a un tamaño superior, lo cual en principio no es muy lógico. Este comportamiento posiblemente tenga que ver con el algoritmo empleado por la caché para seleccionar las partes de la memoria RAM que se copian en la caché. La relación de tamaño de las cachés viene motivada por el tamaño que ocupan los diferentes segmentos de código, podría ocurrir que una caché de 16KB suponga un aumento de rendimiento pero al aumentarla a 32KB el algoritmo de las cachés no encuentra en memoria segmentos útiles para rellenar ese espacio extra,

bien porque no existen (poco probable) o bien por que los existentes son demasiado grandes como para caber y haría falta aumentar la caché todavía más para poder tener esos segmentos adicionales almacenados en la misma.

En cualquier caso, no parece tener ningún sentido el emplear cachés de 8KB o de 32KB en esta aplicación, es mucho más eficiente utilizar los tamaños anteriores y posteriores a esos ya que el rendimiento es prácticamente el mismo, pero se ocupan menos recursos en la FPGA.

Las ganancias totales acumuladas entre no tener caché de instrucciones y tener una caché de 64KB son (con una L1D: 512B y Pred de Saltos: 8192):

<b>Prueba</b>	<b>Reducción de tiempo (s)</b>	<b>Porcentaje de mejora</b>
<i>Home</i>	1,540	716,279%
<i>Cámara</i>	3,219	694,815%
<i>Config</i>	15,332	717,707%
<i>Galería</i>	2,037	701,969%
<i>Disp. Ráfaga</i>	24,954	678,353%
<i>Estado</i>	2,908	712,291%

*Tabla 21, Diferencia de rendimiento entre un sistema sin L1I y una L1I = 64KB*

Hasta ahora se ha estudiado el impacto únicamente de la caché L1 de instrucciones, manteniendo la caché de datos fija en 512 Bytes. A continuación, se estudia si se puede mejorar aún más el rendimiento en base a la caché de datos. Para estas pruebas se ha fijado la L1 de instrucciones en los 64KB elegidos anteriormente, esto limita a 16KB el tamaño máximo de la misma debido a la limitación de recursos (memoria embebida) en la FPGA. Aun así, en posteriores pruebas adicionales también se prueba a rebajar la caché de instrucciones y aumentar la caché de datos para comprobar si el rendimiento es mejor o peor así.

En la página siguiente se muestra la gráfica con el impacto de la caché de datos cuando se mantiene en 64KB la caché de instrucciones.

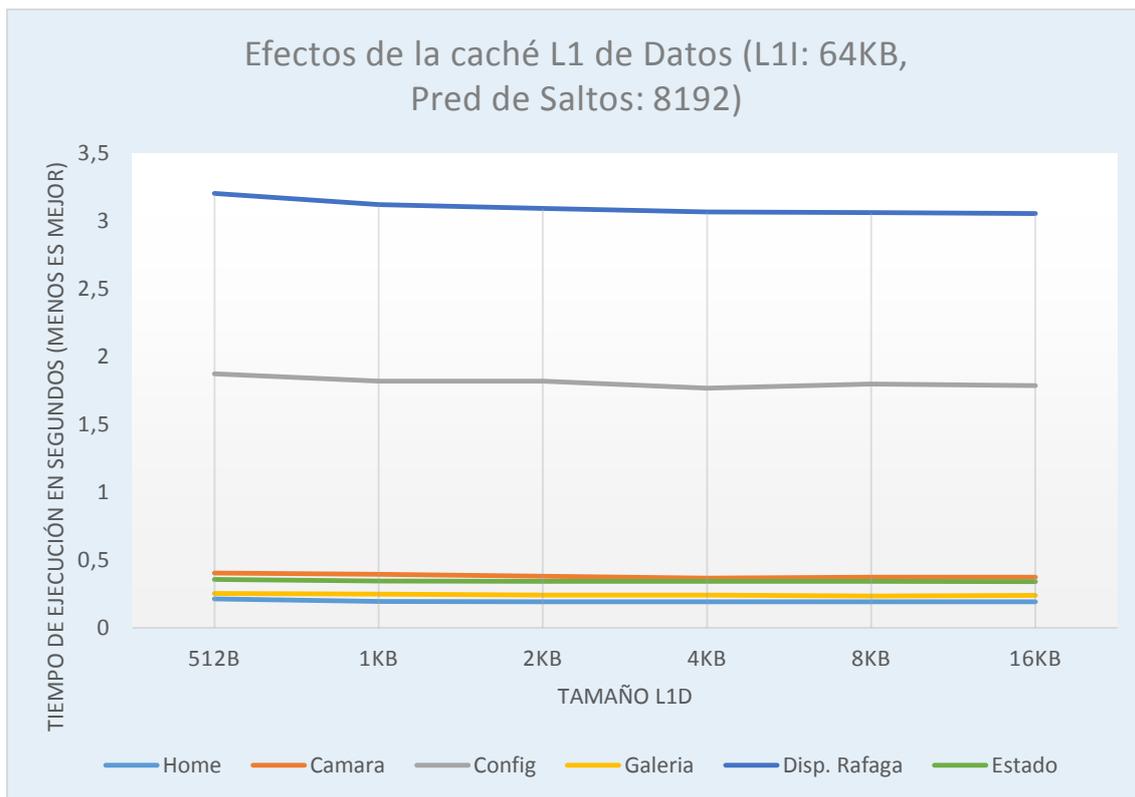


figura 87, Impacto en el rendimiento de la caché L1D

En la gráfica se observa que existe mejora, pero es muy ligera. A continuación, se muestran los valores numéricos:

Tamaño L1D	Home	Dif. (%)	Cámara	Dif. (%)	Config	Dif. (%)
512B	0,215		0,405		1,875	
1KB	0,196	9,69%	0,395	2,53%	1,822	2,91%
2KB	0,194	1,03%	0,382	3,40%	1,822	0%
4KB	0,193	0,52%	0,367	4,09%	1,77	2,94%
8KB	0,193	0%	0,375	-2,13%	1,8	-1,67%
16KB	0,193	0%	0,375	0%	1,787	0,73%

Tabla 22, Efecto de la caché L1D en el rendimiento con diferencias porcentuales 1

Tamaño L1D	Galeria	Dif. (%)	Disp. Raf.	Dif. (%)	Estado	Dif. (%)
512B	0,254		3,206		0,358	
1KB	0,249	2,01%	3,123	2,66%	0,346	3,47%
2KB	0,243	2,47%	3,095	0,90%	0,343	0,87%
4KB	0,242	0,41%	3,07	0,81%	0,344	-0,29%
8KB	0,235	2,98%	3,064	0,20%	0,343	0,29%
16KB	0,241	-2,49%	3,057	0,23%	0,342	0,29%

Tabla 23, Efecto de la caché L1D en el rendimiento con diferencias porcentuales 2

Las mejoras son más ligeras que en el caso de la caché de instrucciones. Cuando en el apartado anterior se pasaba de una L1I de 16/32 a 64KB todavía existía una mejora notable en algunas pruebas (Config y Disparo en Ráfaga), indicando que aun con el máximo tamaño de la L1I el cuello de botella bien está en la L1I o bien está en las capacidades de la propia CPU.

La caché de esta CPU utiliza la estructura más simple de caché, que es la caché de mapeado directo (Direct Mapped), si se emplearán cachés asociativas es posible que se alcanzara mayor rendimiento con menores tamaños. También existe el problema de que hay una diferencia muy grande de rendimiento entre la RAM y la L1, podría ser interesante disponer de una memoria caché L2, que sea más grande que la L1, pero más rápida que la RAM, sin esta L2 el rendimiento se frena considerablemente cuando el dato a acceder no se encuentra en la L1 y se produce un fallo de caché (L1 cache miss).

Como se observa en los resultados apenas hay ganancia a partir de una caché de datos de más de 4KB, son ganancias muy ligeras de menos del 1% salvo en un caso que es del 1,3%. En este caso se ha optado por usar 16KB simplemente porque se cuentan con los recursos suficientes, en caso de tener que optimizar los recursos se hubiese optado por dejar el tamaño de la L1D en 4KB.

El aporte de rendimiento de tener esta L1D en 16KB respecto al caso de tenerlo en 512B se muestra a continuación:

<b>Prueba</b>	<b>Reducción de tiempo (s)</b>	<b>Porcentaje de mejora</b>
<i>Home</i>	0,022	11,40%
<i>Cámara</i>	0,03	8%
<i>Config</i>	0,088	4,92%
<i>Galería</i>	0,013	5,39%
<i>Disp. Ráfaga</i>	0,149	4,87%
<i>Estado</i>	0,016	4,68%

Tabla 24, Diferencia de rendimiento entre un sistema con L1D: 512B y L1D: 16KB

De nuevo se observa la mayor ganancia en aquella prueba que más accesos realiza al subsistema de memoria (Disparo en ráfaga), seguida de aquella que más elementos pinta en pantalla (Config).

Ya que hay tan poca diferencia entre 16KB y 4KB a continuación se muestran las diferencias que habría en caso de haber empleado la segunda opción:

<b>Prueba</b>	<b>Reducción de tiempo (s)</b>	<b>Porcentaje de mejora</b>
<i>Home</i>	0.022	11,40%
<i>Cámara</i>	0.038	10,35%
<i>Config</i>	0.105	5,93%
<i>Galería</i>	0.012	4,96%
<i>Disp. Ráfaga</i>	0.136	4,43%
<i>Estado</i>	0.014	4,07%

Tabla 25, Diferencia de rendimiento entre un sistema con L1D: 512B y L1D: 4KB

Existe incluso algún caso en donde se obtiene mejor rendimiento con 4KB, aunque este tipo de resultados entran dentro del margen de error, pasar las pruebas una y otra vez da resultados ligeramente diferentes pero muy similares. Con estos resultados queda claro que 4KB es un valor más interesante en lo que respecta a rendimiento/recursos.

En las siguientes páginas se muestran algunas pruebas alternativas realizadas.

Cabe destacar que, en cámaras comerciales que disponen de la funcionalidad de disparo en ráfaga se suelen proporcionar datos de velocidad en fps indicando en este modo de funcionamiento cuantas fotos es capaz de realizar la cámara en cada segundo. En este caso se puede calcular el valor dividiendo 6 por el tiempo que se tarda en pasar la prueba del disparo en ráfaga, de esta forma tendríamos un rendimiento en este modo que iría desde los 0,21fps cuando no hay caché de instrucciones, hasta 1,96fps en el caso de emplear una L1 de 64 + 16. Esta prueba está claramente limitada por el rendimiento de las cachés y el subsistema de memoria, con cachés más grandes o una memoria RAM más veloz se podrían alcanzar mayores velocidades.

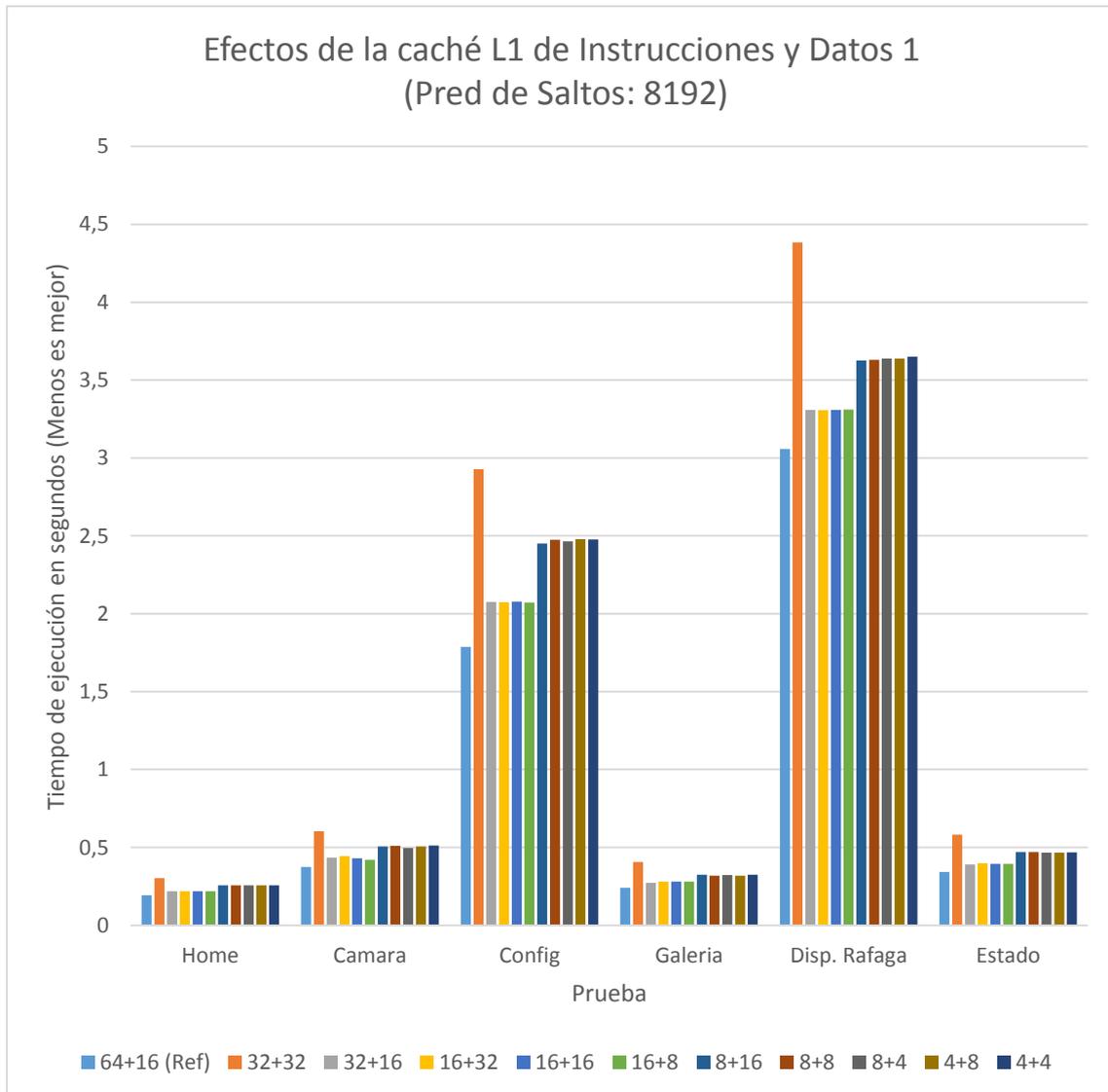


figura 88, Efectos de la caché L1 de Instrucciones y datos 1

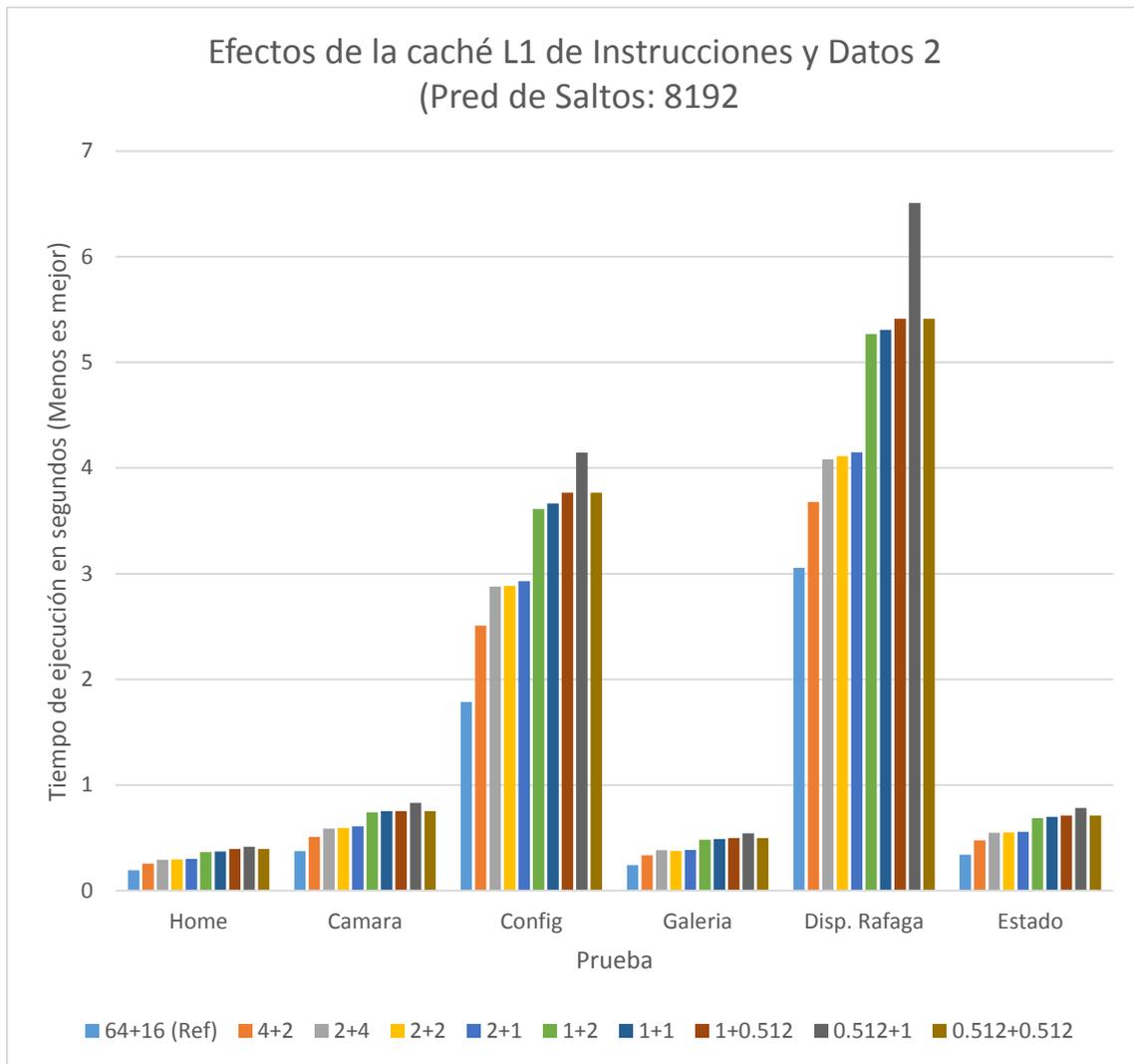


figura 89, Efectos de la caché L1 de instrucciones y datos 2

### Pruebas de rendimiento del Predictor de Saltos

En códigos donde se utilizan muchas estructuras de tipo condicional (Bucles for o if) se generan muchas instrucciones de tipo branch al realizar la traducción a código ensamblador. Cuando este tipo de instrucciones se ejecutan en el pipeline de la CPU se pueden producir drásticas reducciones de rendimiento ya que si no se tiene en cuenta la dirección que toma la instrucción es necesario vaciar todo el pipeline de la CPU y volver a empezar desde la dirección a la que apunta el registro PC (Program Counter) actualizada por la instrucción condicional. Para paliar esta situación las CPUs incluyen una unidad que intenta predecir el camino que seguirá la instrucción a la que se denomina predictor de saltos.

En este caso, no es posible desactivarla, pero sí que se puede variar la complejidad de la unidad. La IP en Qsys permite emplear un predictor de saltos estático, o bien uno dinámico cuyo tamaño se puede seleccionar (256, 4096 o 8192 entradas). En este apartado se realiza un breve estudio acerca del efecto que esta unidad tiene en el rendimiento de la CPU.

Se ha realizado una batería de pruebas en los valores mínimos de caché (Sin L1 de Instrucciones y una L1 de datos de 512 Bytes) y otra con los valores de cachés fijados en el apartado anterior (L1I: 64KB y L1D: 16KB).

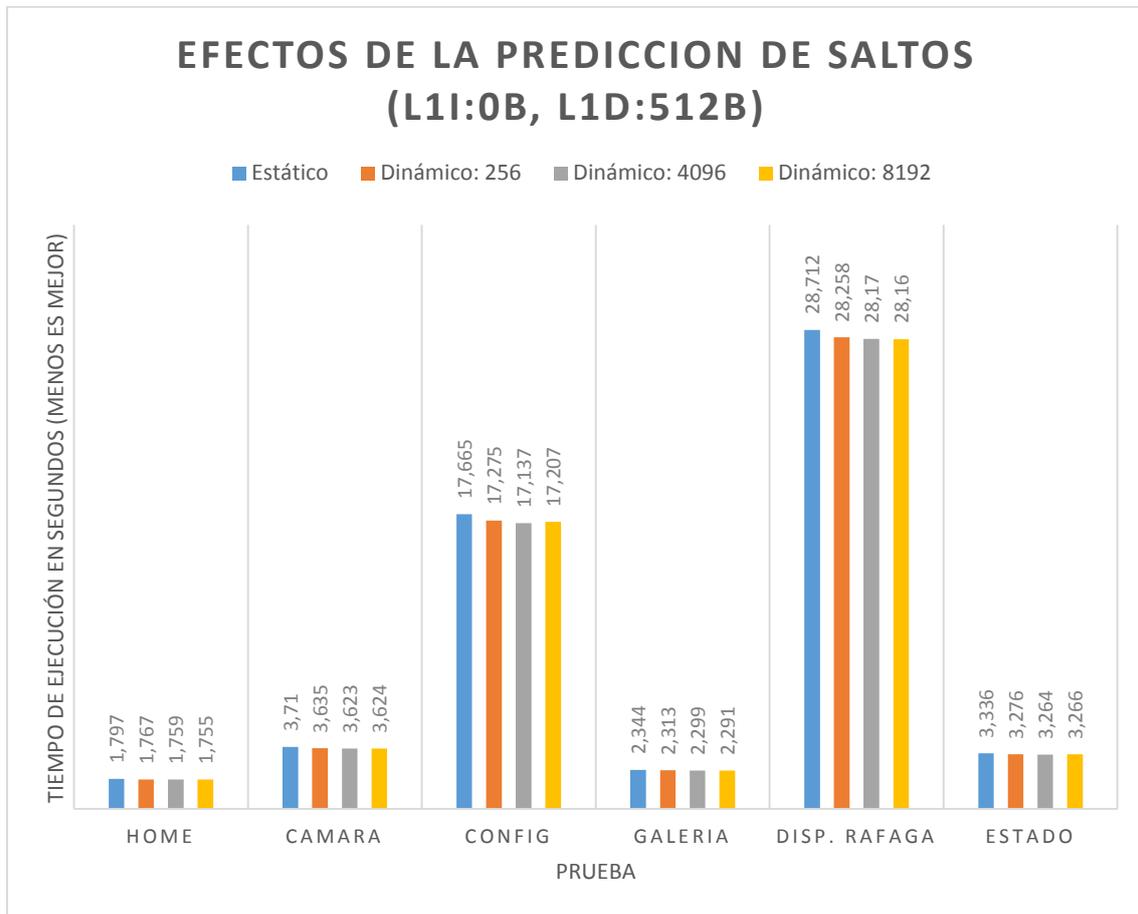


figura 90, Efectos en el rendimiento del Predictor de Saltos con las cachés al mínimo valor

En esta prueba la intención es comprobar si en una situación de un elevado cuello de botella con la memoria RAM (al carecer de L1I) el predictor de saltos tendría algún efecto. Puede verse que las diferencias son sumamente ligeras como consecuencia del inmenso cuello de botella que produce el subsistema de memoria.

En las pruebas de la página siguiente se reduce drásticamente el cuello de botella empleando cachés, aunque los resultados siguen teniendo muy poca variación con una predicción de saltos más avanzada, lo cual es indicativo de que realmente el código condicional no está mermando demasiado el rendimiento de la CPU incluso en el caso de no integrar cachés en el sistema.

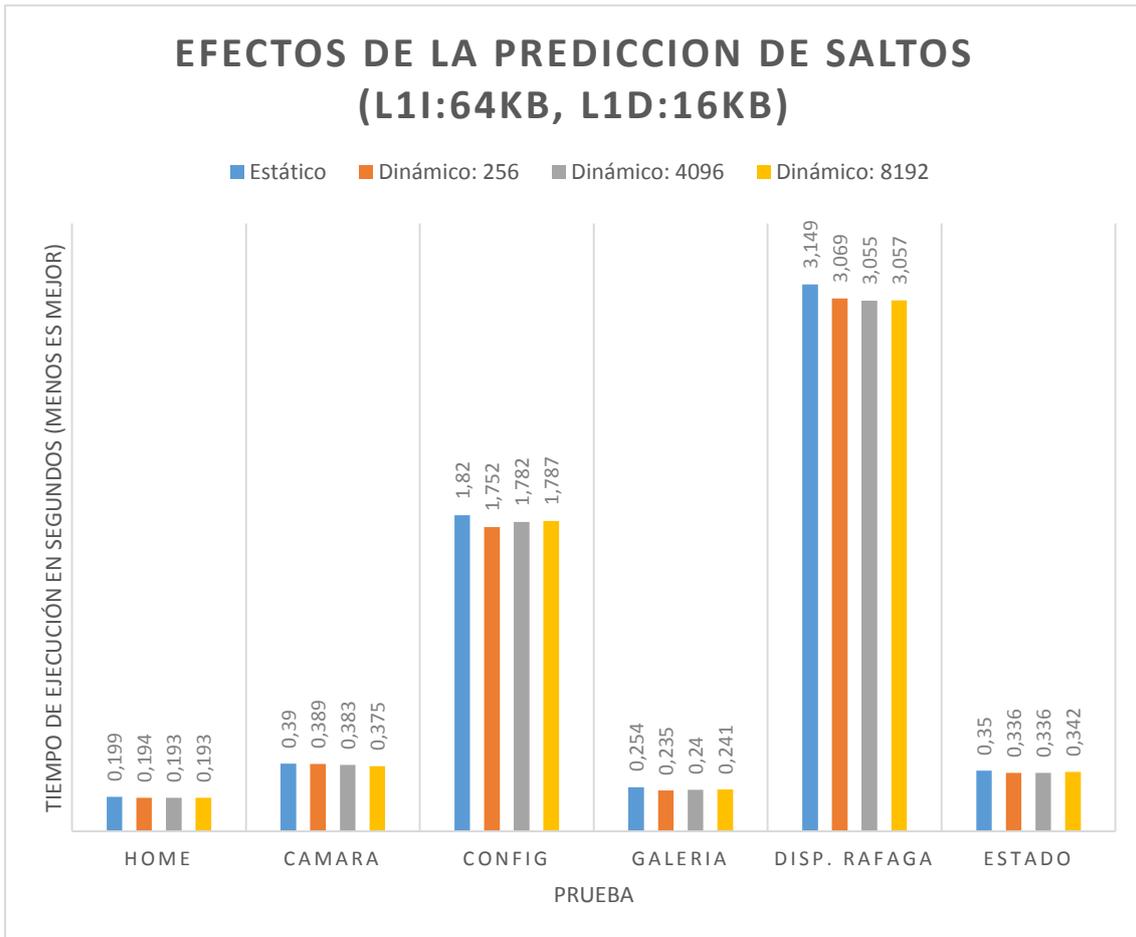


figura 91, Efectos en el rendimiento del Predictor de Saltos con las cachés al máximo valor

A continuación, se muestran los resultados matemáticos.

Con L1I = 0B y L1D = 512B:

<b>Pred. Saltos</b>	<b>Home</b>	<b>Dif. (%)</b>	<b>Cámara</b>	<b>Dif. (%)</b>	<b>Config</b>	<b>Dif. (%)</b>
<i>Estático</i>	1.797		3.71		17.665	
<i>256 Entradas</i>	1.767	1.70%	3.635	2.06%	17.275	2.26%
<i>4096 Entradas</i>	1.759	0.45%	3.623	0.33%	17.137	0.81%
<i>8192 Entradas</i>	1.755	0.23%	3.624	-0.03%	17.207	-0.41%

Tabla 26, Efectos del Predictor de Saltos en el rendimiento con cachés 0 + 0,512B 1

<b>Pred. Saltos</b>	<b>Home</b>	<b>Dif. (%)</b>	<b>Cámara</b>	<b>Dif. (%)</b>	<b>Config</b>	<b>Dif. (%)</b>
<i>Estático</i>	2.344		28.712		3.336	
<i>256 Entradas</i>	2.313	1.34%	28.258	1.61%	3.276	1.83%
<i>4096 Entradas</i>	2.299	0.61%	28.17	0.31%	3.264	0.37%
<i>8192 Entradas</i>	2.291	0.35%	28.16	0.04%	3.266	-0.06%

Tabla 27, Efectos del Predictor de Saltos en el rendimiento con cachés 0 + 0,512B 2

En el caso de emplear una L1I = 64KB y L1D = 16KB:

<b>Pred. Saltos</b>	<b>Home</b>	<b>Dif. (%)</b>	<b>Cámara</b>	<b>Dif. (%)</b>	<b>Config</b>	<b>Dif. (%)</b>
<i>Estático</i>	0.199		0.39		1.82	
<i>256 Entradas</i>	0.194	2.58%	0.389	0.26%	1.752	3.88%
<i>4096 Entradas</i>	0.193	0.52%	0.383	1.57%	1.782	-1.68%
<i>8192 Entradas</i>	0.193	0.00%	0.375	2.13%	1.787	-0.28%

Tabla 28, Efectos del Predictor de Saltos en el rendimiento con cachés 64KB + 16KB 1

<b>Pred. Saltos.</b>	<b>Home</b>	<b>Dif. (%)</b>	<b>Cámara</b>	<b>Dif. (%)</b>	<b>Config</b>	<b>Dif. (%)</b>
<i>Estático</i>	0.254		3.149		0.35	
<i>256 Entradas</i>	0.235	8.09%	3.069	2.61%	0.336	4.17%
<i>4096 Entradas</i>	0.24	-2.08%	3.055	0.46%	0.336	0.00%
<i>8192 Entradas</i>	0.241	-0.41%	3.057	-0.07%	0.342	-1.75%

Tabla 29, Efectos del Predictor de Saltos en el rendimiento con cachés 64KB + 16KB 2

Con cierta similitud a las cachés, las mayores diferencias se registran entre el predictor de Saltos estático y el dinámico más pequeño de 256 entradas. A partir de ahí las diferencias son marginales. Por lo tanto, se puede concluir que la opción más eficiente sería un predictor de 256 entradas. En este caso de nuevo, como hay recursos disponibles en la FPGA, se compila el trabajo con un predictor dinámico de 8192 entradas. A continuación, se muestran las ganancias combinadas.

Con L1I = 0KB y L1D = 512B:

<b>Prueba</b>	<b>Reducción de tiempo (s)</b>	<b>Porcentaje de mejora</b>
<i>Home</i>	0,042	2,39%
<i>Cámara</i>	0,086	2,37%
<i>Config</i>	0,458	2,66%
<i>Galería</i>	0,053	2,31%
<i>Disp. Ráfaga</i>	0,552	1,96%
<i>Estado</i>	0,07	2,14%

Tabla 30, Diferencias de rendimiento entre un sistema con un Predictor de Saltos Estático y uno Dinámico de 8192 posiciones con una L1I = 0KB y una L1D = 512B

Con L1I = 64KB y L1D = 16KB:

<b>Prueba</b>	<b>Reducción de tiempo (s)</b>	<b>Porcentaje de mejora</b>
<i>Home</i>	0,006	3,11%
<i>Cámara</i>	0,015	4%
<i>Config</i>	0,033	1,85%
<i>Galería</i>	0,013	5,39%
<i>Disp. Ráfaga</i>	0,092	3,01%
<i>Estado</i>	0,008	2,34%

Tabla 31, Diferencias de rendimiento entre un sistema con Predictor de Saltos Estático y uno Dinámico de 8192 posiciones con una L1I = 64KB y una L1D = 16KB

Observando las últimas tablas con las ganancias totales es llamativo que en el caso de cachés muy pequeñas la ganancia es consistente entre pruebas, con diferencias en casi todas las pruebas en el entorno del 2,3%, mientras que en el caso de cachés grandes las ganancias son más dispares en función de la prueba que se esté ejecutando. Esto puede ser consecuencia de que en el primer caso el cuello de botella está claramente en el subsistema de memoria, y lo está en todas las pruebas, impidiendo ganancias, en este caso, de más de un 2,66%, mientras que en el segundo caso parece que el subsistema de memoria limita el rendimiento en menos ocasiones y, como consecuencia, el predictor de saltos tiene un mayor efecto en el rendimiento, provocando también una mayor variabilidad entre las distintas pruebas.

## 5.2 Memoria de Vídeo

El segundo elemento clave en este sistema es el conjunto memoria-dma. El flujo de datos que contiene los fotogramas provenientes de la cámara son controlados por dos bloques DMA que almacenan y leen esos datos utilizando para ello una memoria. La placa contiene 3 tipos de memoria utilizables en este caso (la memoria Flash se descarta por su limitado número de ciclos de borrado/escritura):

- Memoria SRAM
- Memoria SDRAM
- Memoria On-Chip (Interna en la FPGA)

La viabilidad de cada una de esas memorias depende tanto de su capacidad como de su velocidad (ancho de banda y latencia).

Las pruebas para este apartado se han hecho sobre un proyecto de Quartus diferente y donde solo se usan los bloques necesarios. Se ha hecho de esta manera para poder aislar los componentes probados (memorias) y comprobarlos de manera individual, sin que otros componentes puedan interferir en las pruebas.

## Memoria SRAM

La memoria que inicialmente parece más adecuada para esta función inicialmente parece la memoria SRAM. Las memorias SRAM son las memorias externas más rápidas que se pueden comprar y suelen emplearse junto con microcontroladores/procesadores como memoria externa de baja capacidad y alta velocidad. También se emplea como caché L1/L2/L3 en las CPUs de alto rendimiento, aunque en ese caso son memorias SRAM On-Chip.

El chip implementado en la placa permite frecuencias de hasta 100MHz y su ancho de palabra es de 16 bits. Eso en un caso ideal le dota de un ancho de banda de:

$(16/8) \text{ bytes} * 100 = 200 \text{ MB/s}$  (Sería la mitad en el caso de un reloj de 50MHz).

La prueba más básica es la siguiente:

CONFIGURACIÓN		
Componente	Resolución	Color
Patrón Test	320 x 240	24 bit RGB (3 x 8 bit)
DMAs	320 x 240	24 bit RGB (3 x 8 bit)

Tabla 32, Configuración para la prueba 1 de la SRAM

The image shows two configuration panels side-by-side. The left panel is titled 'Test-Pattern Generator' (altera\_up\_avalon\_video\_test\_pattern) and has a section 'Outgoing Frame Resolution' with input fields for 'Width (# in width): 320' and 'Height (# in height): 240'. The right panel is titled 'DMA Controller' (altera\_up\_avalon\_video\_dma\_controller) and has several sections: 'Addressing Parameters' with 'Addressing Mode' set to 'Consecutive', 'Default Buffer Starting Address' and 'Default Back Buffer Start Address' both set to '0x00000000'; 'Frame Resolution' with 'Width (# of pixels): 320' and 'Height (# of lines): 240'; 'Pixel Format' with 'Color Bits' set to '8' and 'Color Planes' set to '3'; and 'Control' with a checked checkbox 'Enabled DMA transfer on reset'.

figura 92, Ejemplo de configuración del Patrón y el DMA para la prueba básica de SRAM

Se puede comprobar como el resultado es correcto:



figura 93, Foto 1 de las pruebas con SRAM

Se pasa a comprobar el resultado con una resolución mayor:

CONFIGURACIÓN		
Componente	Resolución	Color
Patrón Test	640 x 480	24 bit RGB (3 x 8 bit)
DMAs	640 x 480	24 bit RGB (3 x 8 bit)

Tabla 33, Configuración para la prueba 2 de la SRAM

El resultado no es correcto:



figura 94, Foto 2 de las pruebas con SRAM

Por lo tanto, no es posible emplear esta resolución con la memoria SRAM.

La memoria SRAM tiene una capacidad de 2MB, así que se puede calcular cuánto espacio ocupa un fotograma con la configuración indicada en la tabla:

640 pixeles x 480 pixeles = 307.200 pixels \* 24 bits por pixel = 7.372.800 bits = 0,87 MB.

En la práctica las posiciones de memoria con las que se trabaja son de 32 bits, por lo tanto, a pesar de trabajar con 24 bits RGB se ocupará la posición de memoria completa por cada píxel, rellenando con ceros los píxeles no utilizados, eso quiere decir que, a efectos prácticos, lo que ocurre es lo siguiente:

640 pixeles x 480 pixeles = 307.200 pixels \* 32 bits por pixel = 9.830.940 bits = 1,17 MB.

Se puede observar que el tamaño del fotograma continúa por debajo del tamaño de la memoria, lo cual es un indicativo de que el problema no radica en que falte capacidad de almacenamiento, sino que lo que falta es velocidad.

Si se reduce la calidad del color a 16 bits RGB es posible mantener la resolución y disminuir el flujo de datos para comprobar si escribiendo menos datos el sistema funciona.

CONFIGURACIÓN		
Componente	Resolución	Color
Patrón Test	640 x 480	24 bit RGB (3 x 8 bit)
DMAs	640 x 480	16 bit RGB (1 x 16 bit) *

Tabla 34, Configuración para la prueba 3 de la SRAM

\*Para la realización de esta prueba se debe emplear la IP RGB Resampler antes de los DMAs para convertir de 24 bits a 16 bits.

En este caso el resultado es correcto, pero la calidad de imagen se resiente:

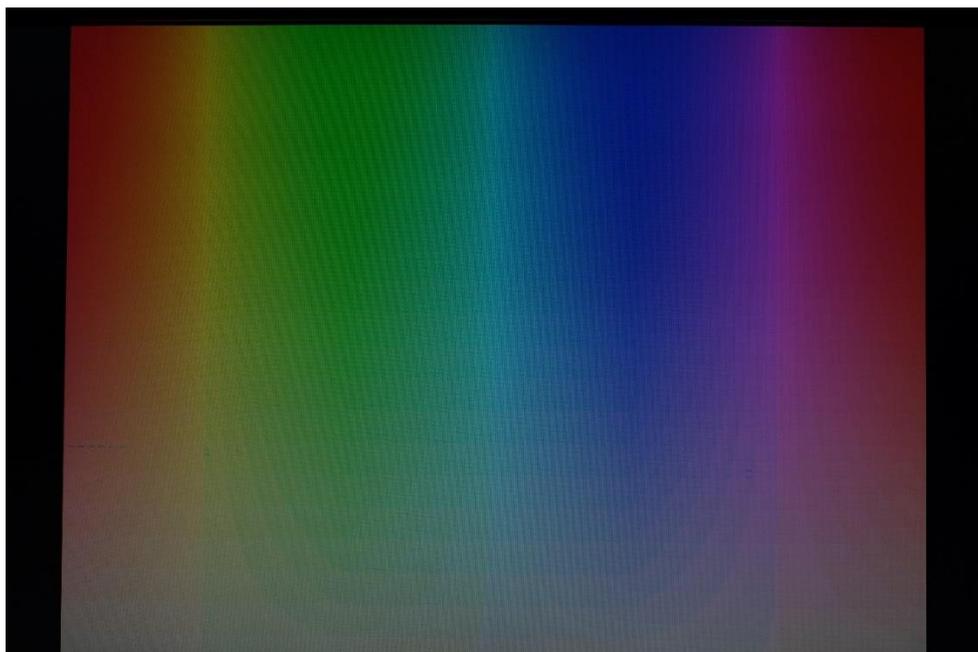


figura 95, Foto 3 de las pruebas con SRAM

Dado que la calidad de imagen a 640 x 480 16 bits RGB es inferior a la obtenida en 320 x 240 24bits RGB concluyen las pruebas con la memoria SRAM dejando como única opción el empleo de esta última configuración.

Puesto que el problema parece ser de rendimiento la siguiente prueba puede ser aumentar la frecuencia de funcionamiento de la memoria. No obstante, en el manual de la IP del controlador de la memoria SRAM se indica que solo se garantiza el funcionamiento de este bloque a 50MHz:

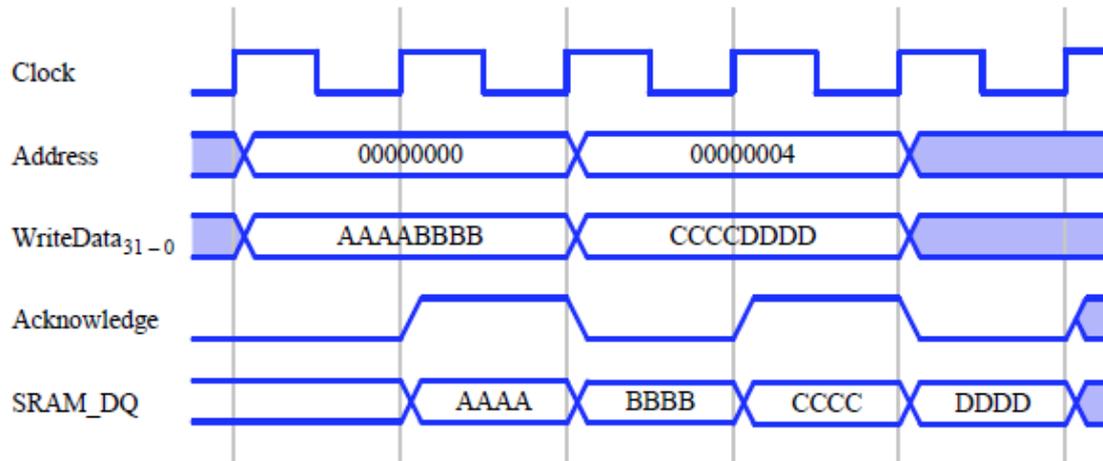


Figure 1. Timing diagram of writing 32-bit data to the SRAM

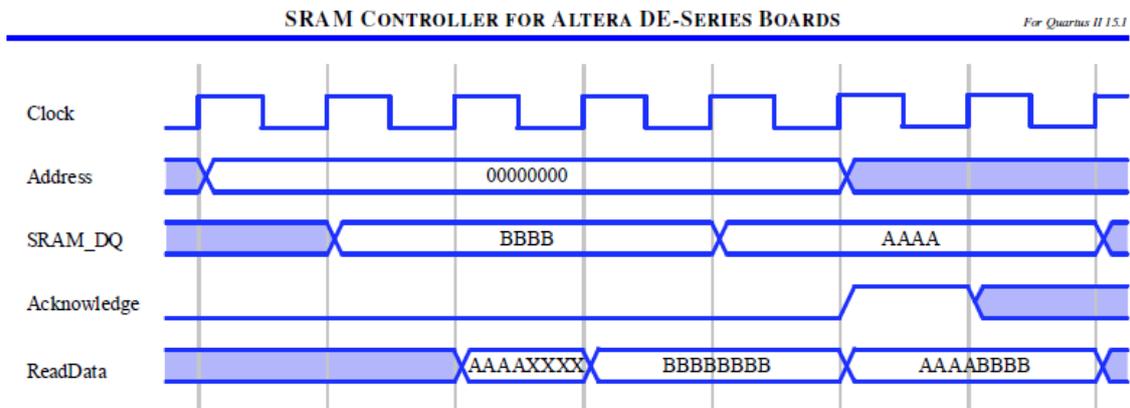


Figure 2. Timing diagram of reading 32-bit data from the SRAM

The SRAM Controller supports a clock frequency of 50 MHz, which is readily available on the DE2-115 Board.

*figura 96, Información proveniente del datasheet de la IP SRAM Controller de Altera*

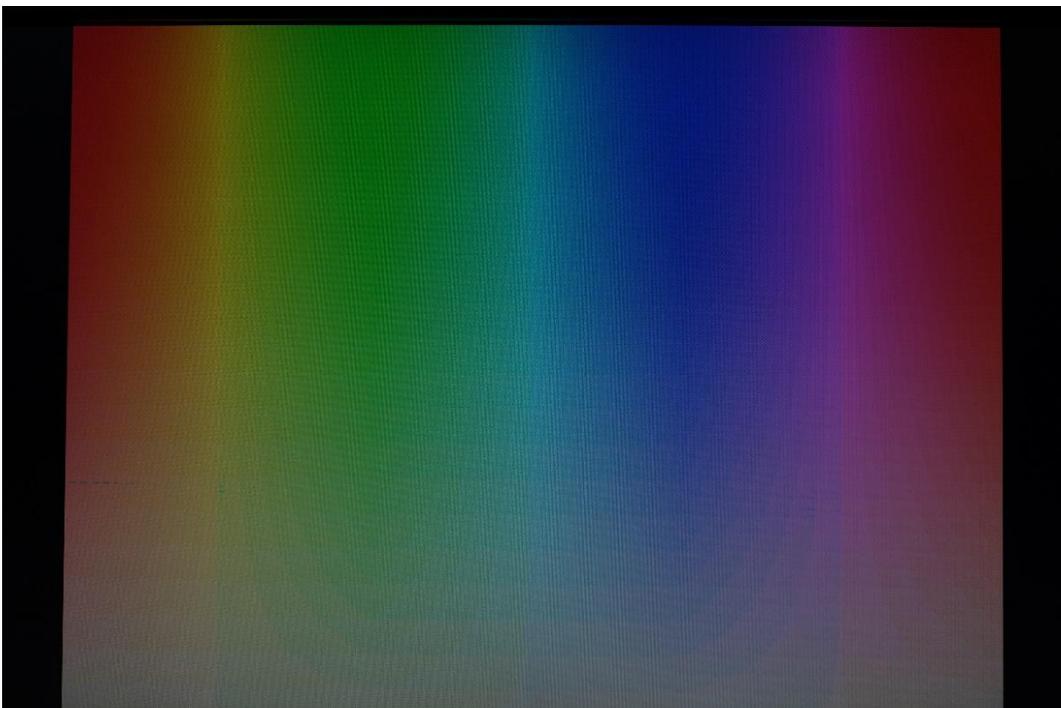
También en el documento se indica que una lectura/escritura de 32 bits por parte de un dispositivo maestro a la SRAM es dividido en dos escrituras de 16 bits, por lo tanto, se requieren 2 ciclos para leer/escribir un dato de 32 bits. En el documento se recomienda no usar este controlador para aplicaciones que requieran una respuesta rápida por parte de la memoria:

### 3 Instantiating the Core

The SRAM Controller can be instantiated in a system using Qsys or as a standalone component from the IP Catalog within the Quartus II software. There is no need to configure the controller. Once instantiated the user can use the SRAM in the same way as using an On-Chip Memory. Any read or write operation to an address within SRAM Controller's address range will be read or written to the SRAM on the DE2-115 boards. Note that the SRAM Controller has a longer read/write latency than the On-Chip Memory and needs two transfers for 32-bit data, hence it is not recommended for designs that require fast memory response.

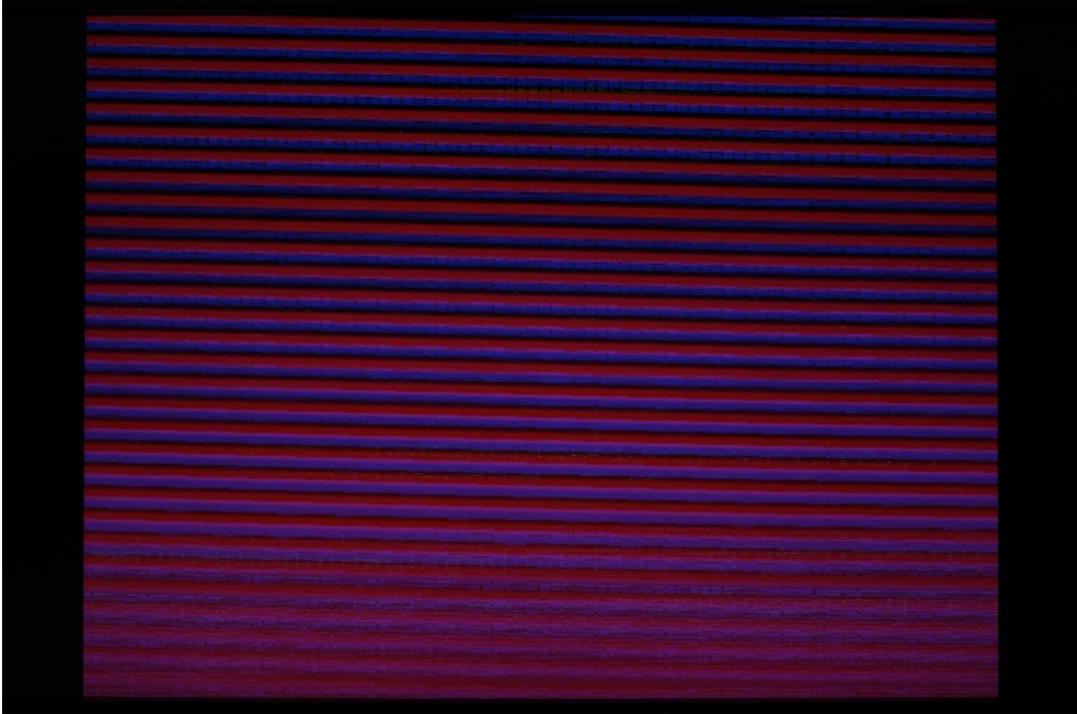
*figura 97, Advertencia en el datasheet del controlador de la memoria SRAM acerca de su rendimiento*

Aun así, se decide hacer la prueba para saber el resultado, en este caso empleando la última configuración probada (640 x 480 16 bits RGB, que funciona correctamente con 50MHz). En este caso se hace lo mismo con un reloj para todo el sistema de 75 MHz, resultado en la siguiente imagen:



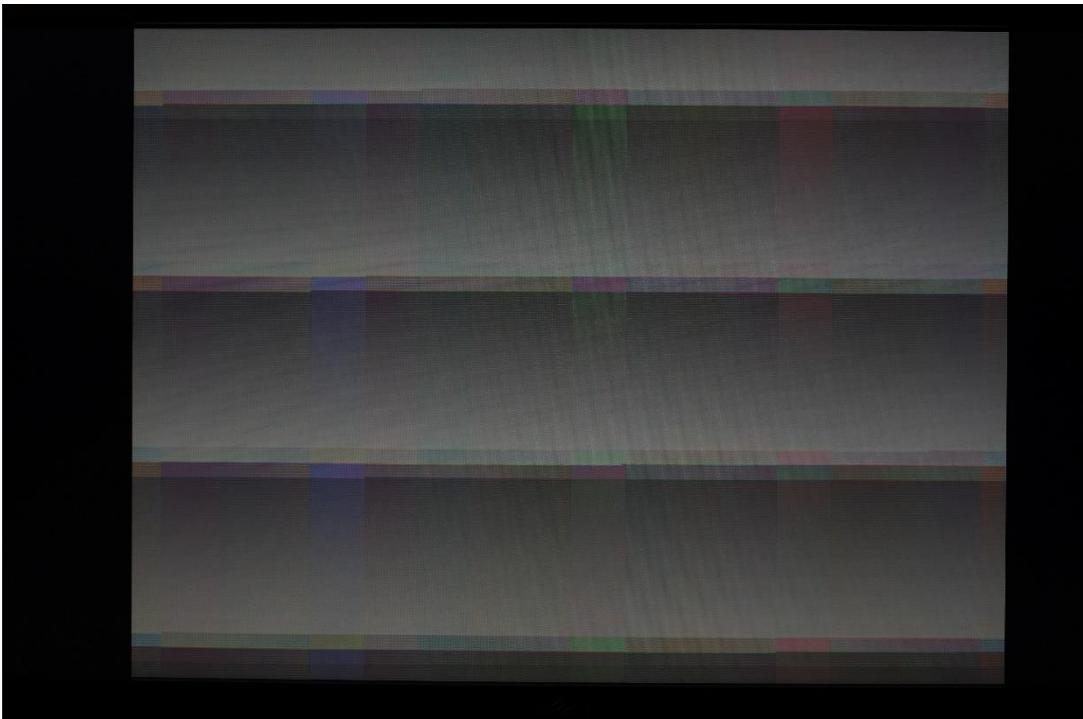
*figura 98, Foto 4 de las pruebas con SRAM*

El resultado es correcto, por lo que se pasa a probar 640 x 480 con 24 bits RGB, resultando en la siguiente imagen:



*figura 99, Foto 5 de las pruebas con SRAM*

Se observa cómo la imagen en este caso no es correcta, por lo tanto, se intenta aumentar de nuevo la frecuencia del sistema, en este caso a 90MHz y se vuelve a 640 x 480 16 bits que es una configuración que funciona correctamente:



*figura 100, Foto 6 de las pruebas con SRAM*

En este caso el controlador SRAM no es capaz de funcionar correctamente, por lo tanto, al menos con esta IP, no es posible hacer funcionar la memoria a 90MHz.

Como conclusión, después de esta serie de pruebas se puede concluir que empleando la memoria SRAM como memoria de vídeo solo es posible funcionar con las siguientes configuraciones:

Resolución	Color
<b>320 x 240</b>	24 bit RGB (3 x 8 bit)
<b>640 x 480</b>	16 bit RGB (1 x 16 bit)

*Tabla 35, Configuraciones disponibles empleando la memoria SRAM*

### Memoria SDRAM

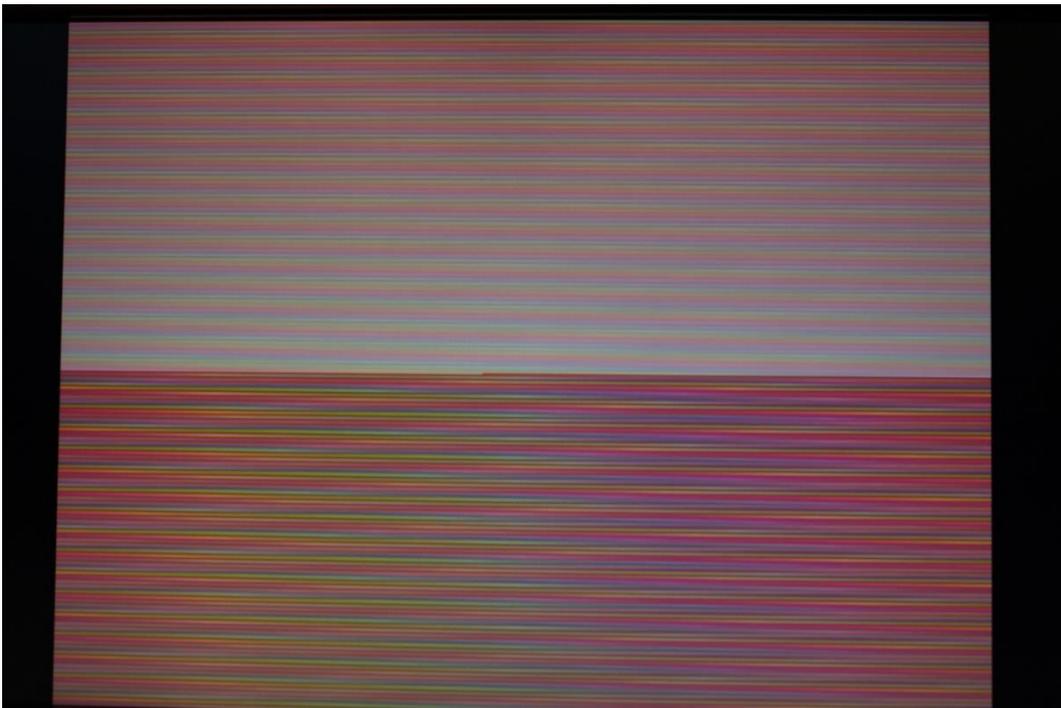
En este caso se dispone de una capacidad de 128MB en dos chips capaces de funcionar hasta 133 MHz con un ancho de palabra de 32 bits, lo cual idealmente da un ancho de banda de:

$$(32/8) \text{ bytes} * 133 \text{ MHz} = 532 \text{ MB/s}$$

Con un reloj de 50MHz ese ancho de banda se reduce a 200 MB/s.

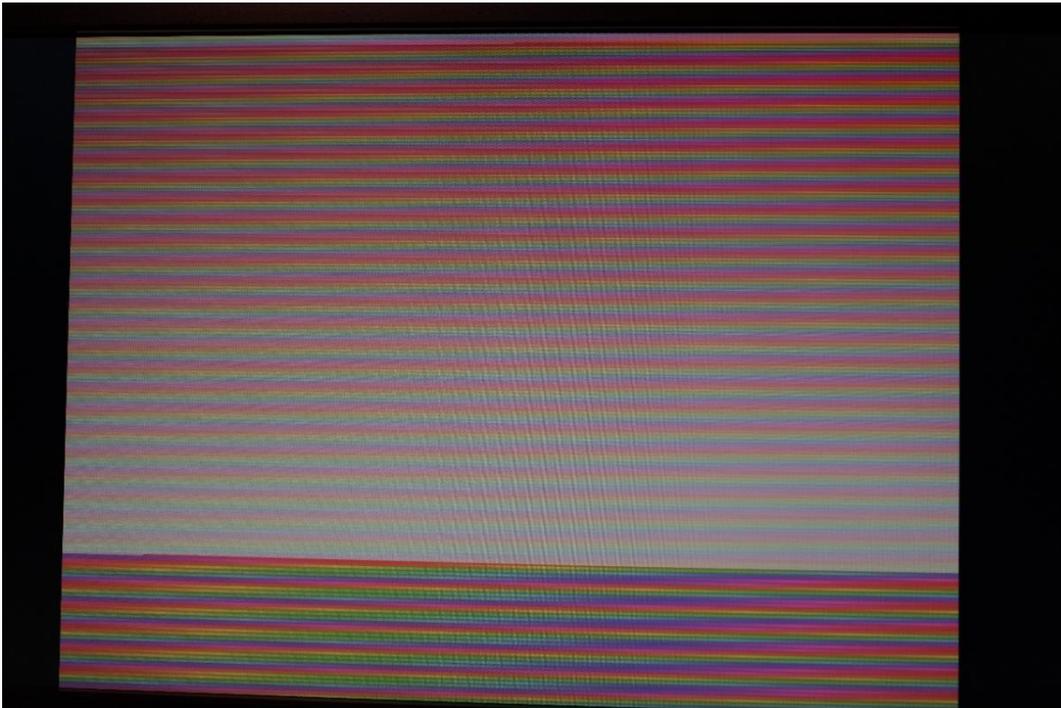
En este caso concreto el ancho de banda de la memoria SDRAM es mayor que el de la SRAM, aunque no es el caso de la latencia.

Al igual que en el caso anterior se comienza la prueba con una configuración de bajos requisitos como es 320 x 240 con 24 bits RGB. El resultado es el siguiente:



*figura 101, Foto 1 de las pruebas con SDRAM*

Se observa que no se consigue el funcionamiento esperado. Se intenta incrementando la frecuencia de reloj de la memoria SDRAM de 50 MHz a 100 MHz, pero el resultado sigue siendo el mismo:



*figura 102, Foto 2 de las pruebas con SDRAM*

Se realizan pruebas disminuyendo la latencia CAS a 2 ciclos manteniendo los 100 MHz de frecuencia y también aumentando la frecuencia de reloj de la memoria al máximo de 133 MHz con latencia CAS a 3 ciclos y el resultado continúa siendo el mismo, por lo tanto, parece que no es posible emplear la memoria SDRAM como memoria de vídeo.

El problema puede deberse a la propia latencia de la SDRAM que es mayor a la de la SRAM, en combinación con alguna limitación de las IPs de los DMAs del University Program, donde todas las IPs son bastante básicas.

Por lo tanto, en vista de los resultados, la memoria SDRAM queda descartada como memoria de vídeo.

#### Memoria On-Chip

El último tipo de memoria que se puede utilizar es la memoria embebida en la propia FPGA. Esta es la memoria más rápida, aunque también es la más limitada en cuanto a capacidad, por lo tanto, incluso antes de probarla se da por hecho que no se podrá incrementar mucho la resolución. También hay que tener en cuenta que el NIOS II utiliza este tipo de memoria como caché y para sus registros internos, al igual que el resto de IPs implementadas en la FPGA que también pueden estar utilizando parte de esta memoria, con lo cual la capacidad utilizable es aún más reducida.

La IP de la memoria On-Chip permite seleccionar varios tipos de ancho de palabra. En este caso se utilizan 32 bits, por lo tanto, el ancho de banda teórico será:

$$(32/8) \text{ bytes} \times 50 \text{ MHz} = 200 \text{ MB/s}$$

Este modelo de Cyclone IV tiene 486 KB de memoria embebida. Según los cálculos hechos en la sección de las pruebas de la memoria SRAM no hay espacio suficiente para el uso de

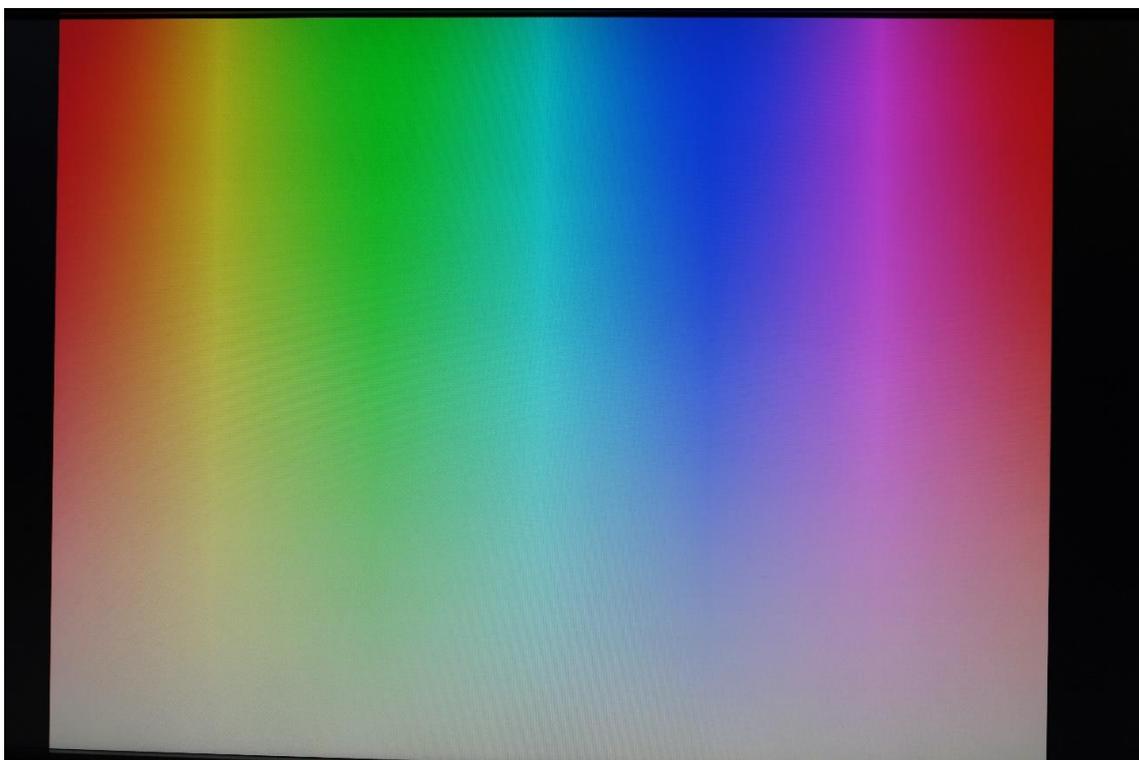
fotogramas con una resolución de 640 x 480 píxeles, por lo tanto, en este caso el límite será de 320 x 240. Suponiendo una calidad de color de 24 bits RGB, cada fotograma ocupa:

$$320 \text{ píxeles} \times 240 \text{ píxeles} \times 24 \text{ bits/píxel} = 225 \text{ KB}$$

Sin embargo, igual que ocurría con la SRAM, las transferencias provenientes de las IPs anteriores seguirán empleando un ancho de palabra de 32 bits, por lo tanto, en la práctica los 8 bits que sobran serán rellenos con ceros y se consideran empleados. El resultado de esto es:

$$320 \text{ píxeles} \times 240 \text{ píxeles} \times 32 \text{ bits/píxel} = 300 \text{ KB}$$

En la práctica se están ocupando 300 KB de memoria On-Chip a pesar de que la información útil sea de 225 KB.



*figura 103, Foto 1 de las pruebas con memoria On-Chip*

Se puede observar cómo, al igual que con la SRAM, el funcionamiento es correcto con este tipo de memoria. En este caso la configuración también se queda como 640 x 480 con 24 bits RGB.

Se puede concluir que estas son las configuraciones posibles en lo que respecta a la memoria de vídeo:

Tipo de Memoria	On-Chip	SRAM
Configuración 1	320 x 240 16 bits RGB	
	320 x 240 24 bits RGB	
Configuración 2	-	640 x 480 16 bits RGB

*Tabla 36, Resumen de las configuraciones de memoria de vídeo disponibles*

### Funcionamiento de subsistema de memoria completo

Una vez se ha comprobado el correcto funcionamiento del sistema por separado se debe comprobar si es correcto el funcionamiento del subsistema de memoria una vez se encuentra conectado al sistema. En este caso, la memoria de vídeo se conecta tanto a los DMAs como al NIOS II. El NIOS II, en un momento dado, debe parar el DMA de escritura para volcar los datos del fotograma que en ese momento se encuentra en la memoria y, una vez se ha realizado la copia del fotograma en otro tipo de memoria la CPU debe reanudar de nuevo el funcionamiento del DMA que había desactivado. Este funcionamiento se describe visualmente en la siguiente página.

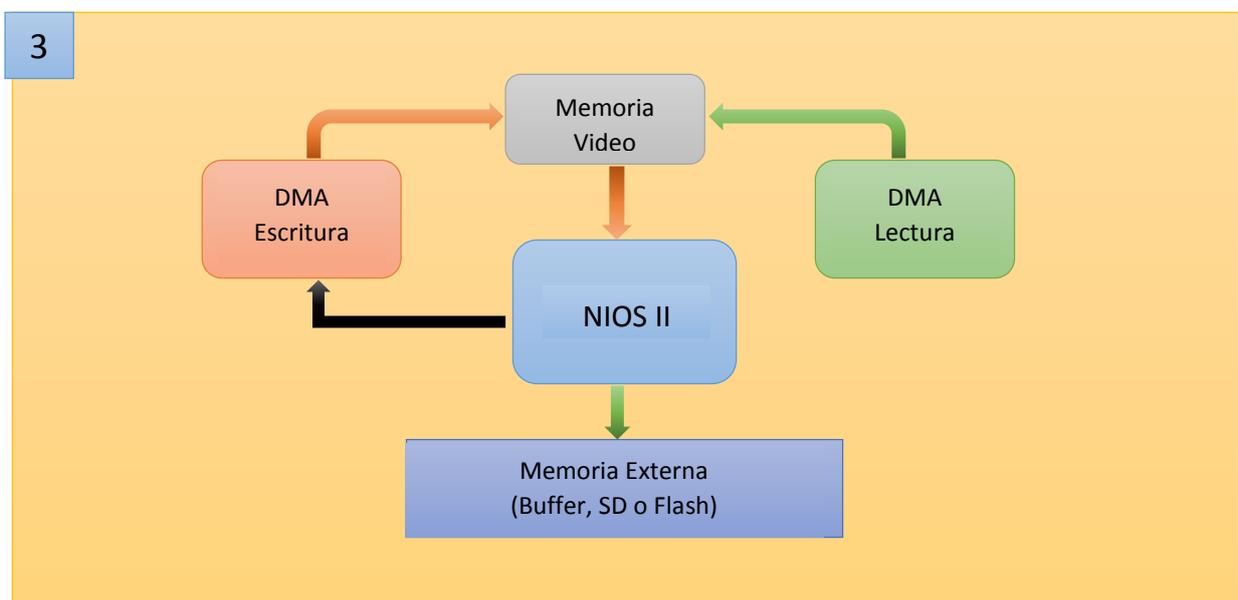
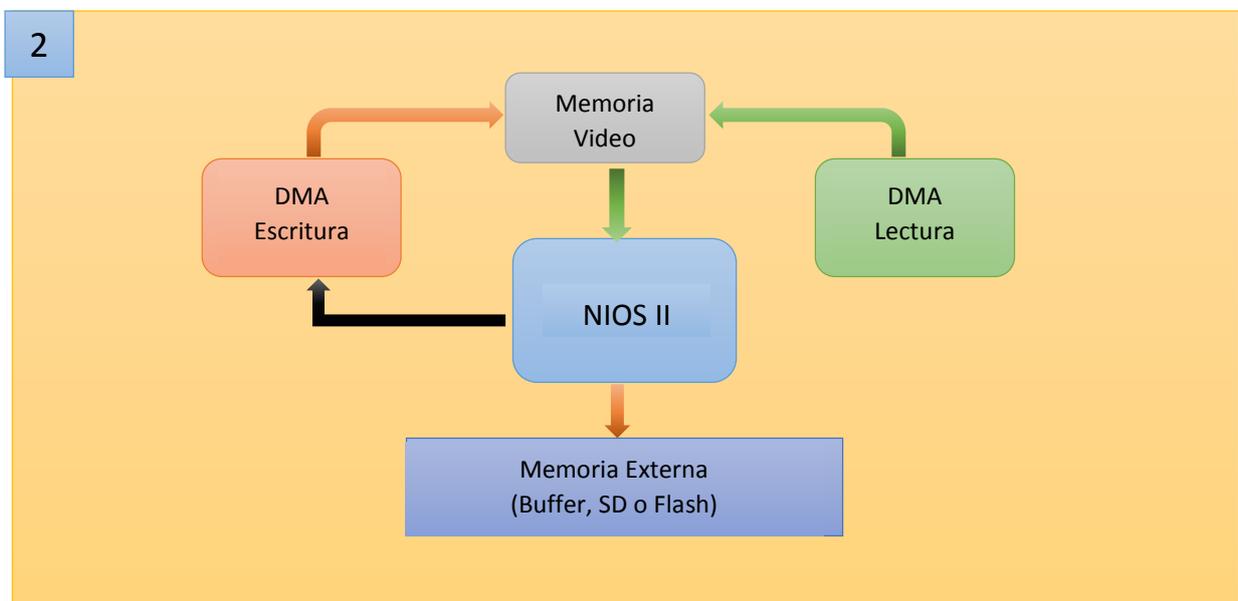
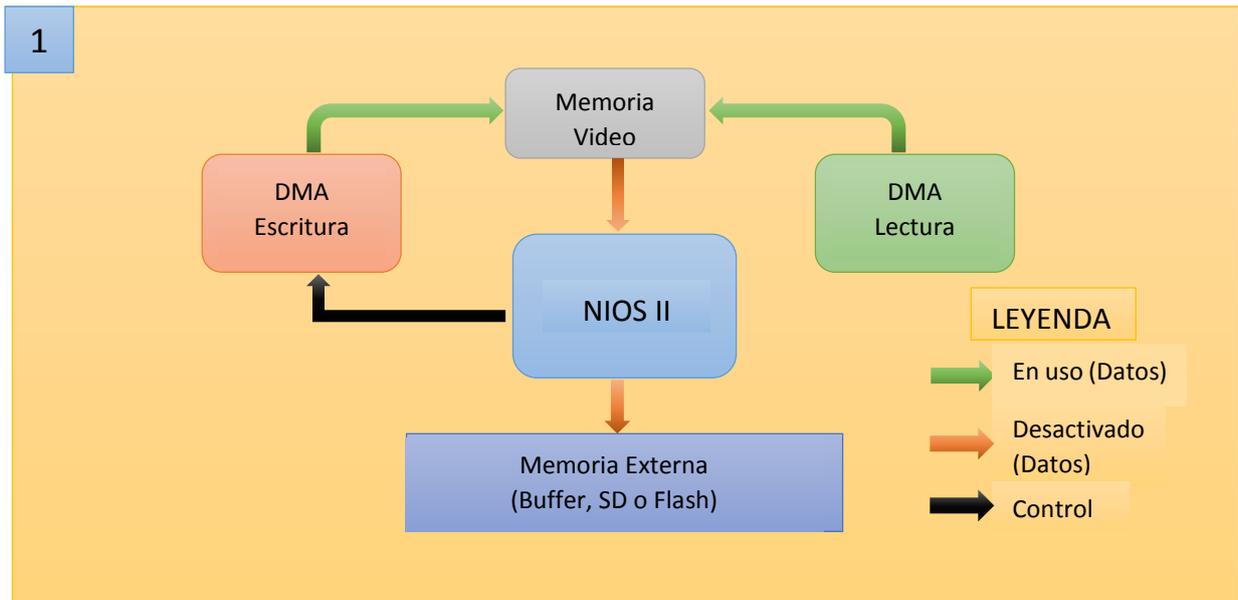
Los pasos para la toma de una fotografía son los siguientes:

1. El DMA de escritura escribe los datos en una memoria que el DMA de lectura lee. La CPU no interviene. Por pantalla se muestra la imagen Live View, lo que en ese momento está capturando el sensor de la cámara.
2. A través de la línea de control la CPU detiene el DMA de escritura. El DMA de escritura se detiene cuando ha terminado de escribir en la memoria el fotograma que se encontraba escribiendo en el momento en que le llega la instrucción de detención de la CPU. El DMA de lectura continúa activo, en ese momento se muestra por pantalla el último fotograma escrito en memoria que será la fotografía que se acaba de tomar (Y que en ese momento la CPU comienza a leer).
3. La CPU lee el contenido de la memoria y lo copia a otra memoria. Cuando ha terminado vuelve a activar el DMA de escritura y se vuelve a mostrar la imagen Live View por pantalla. El sistema entonces está listo para tomar otra fotografía.

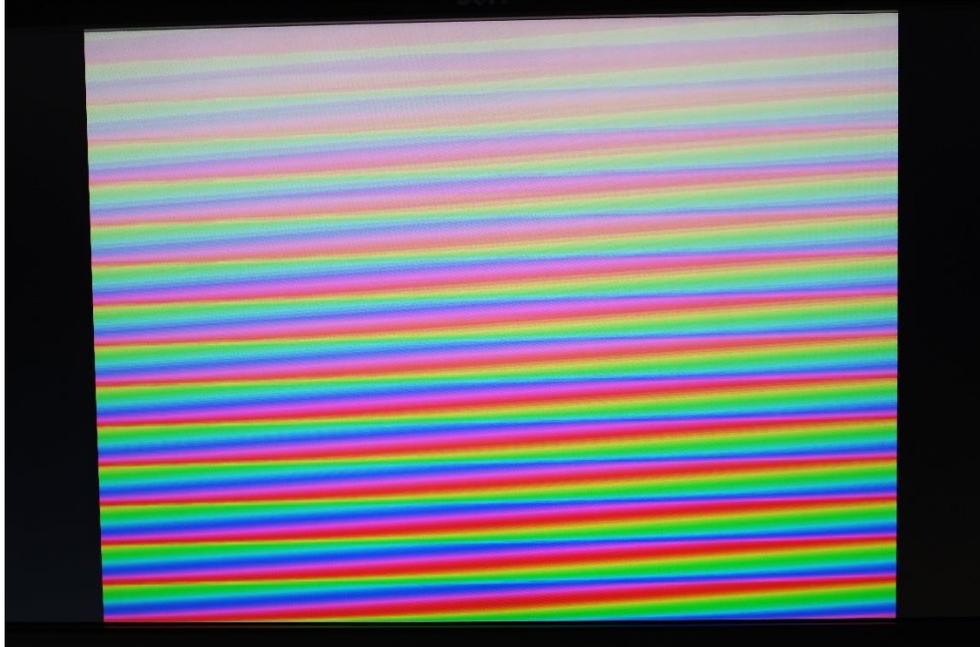
Teniendo en cuenta estos pasos a la hora de tomar la fotografía hay que comprobar como es el comportamiento de esa memoria de video durante este proceso. Se comprueba que todos los pasos se realizan correctamente.

En el segundo capítulo de este documento se describían dos posibles opciones a la hora de diseñar el proyecto. En este caso se considera que se está empleando la segunda de ellas dadas las ventajas de la misma en el funcionamiento final.

figura 104, Proceso de almacenado de un fotograma

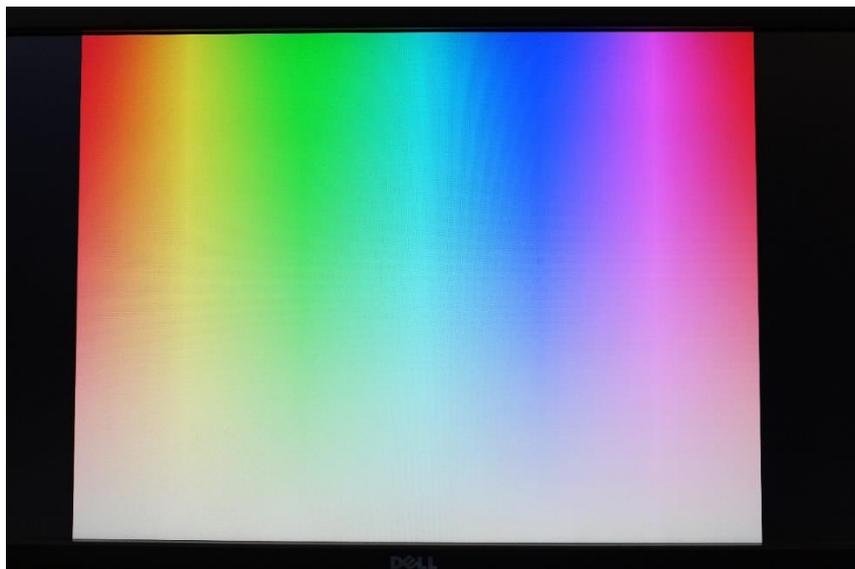


En las pruebas que se han realizado con la memoria SRAM en el apartado 5.2, se veía como la memoria SRAM funcionaba correctamente a 75MHz en el proyecto de pruebas empleado para ese apartado. Sin embargo, en el sistema completo no ha sido posible hacer funcionar correctamente la SRAM a esa frecuencia:



*figura 105, Resultado del Patrón de Test empleando la SRAM como memoria de vídeo en el sistema completo*

Por lo tanto, de las dos opciones empleadas solo queda la posibilidad de emplear memoria On-Chip como memoria de vídeo. En ese caso, el funcionamiento es correcto:



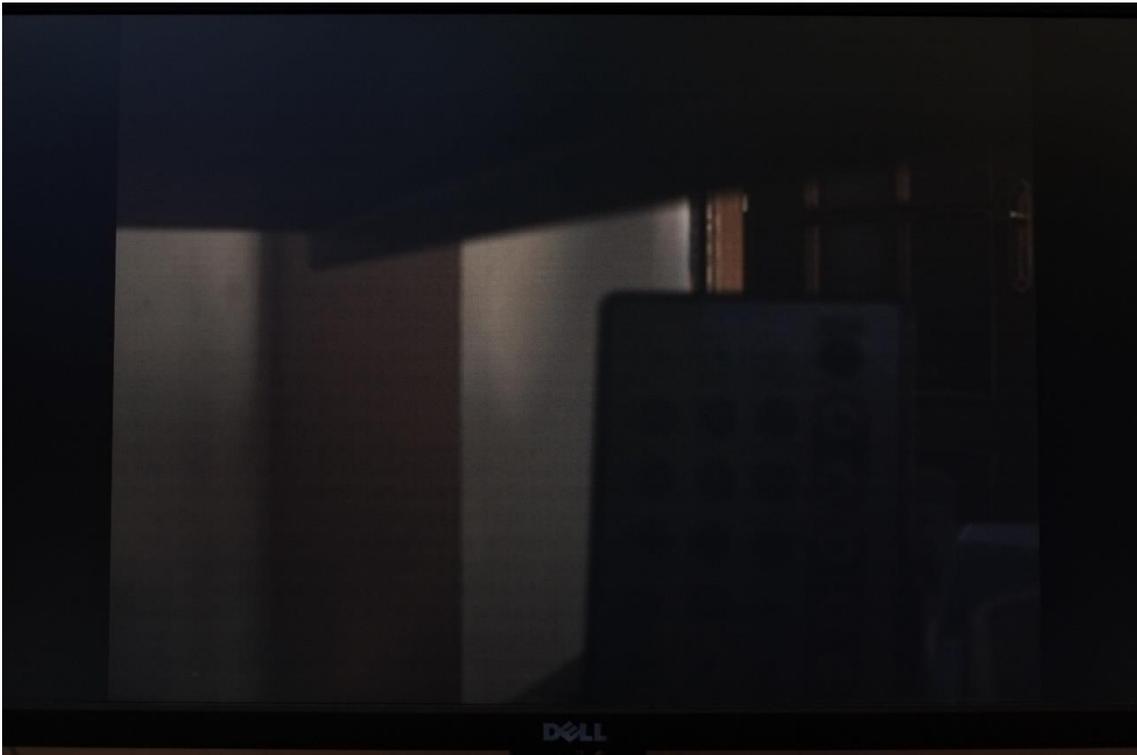


figura 106, Patrón de Test (Arriba) e Imagen capturada por la cámara (Abajo)

Por lo tanto y a modo de conclusión, la memoria on-chip es la elegida como memoria de vídeo.

#### Conclusiones y Justificación del uso de la SDRAM como memoria de programa

Como se ha ido viendo tanto en la descripción hardware como software del sistema, la memoria de programa asociada al NIOS II es la memoria SDRAM. El motivo de esta decisión es la falta de capacidad tanto de la memoria SRAM como de la memoria On-Chip (en ésta se encuentran el buffer de vídeo y las cachés, pero si estuviese vacía tampoco serviría).

Cuando todo el software que corre en el NIOS II se compila y está listo para ser descargado al micro, Eclipse muestra la cantidad de espacio que ocupa el programa en memoria:

```
Info: (cam_soft.elf) 4451 KBytes program size (code + initialized data).
Info: 126606 KBytes free for stack + heap.
Info: Creating cam_soft.objdump
nios2-elf-objdump --disassemble --syms --all-header --source cam_soft.elf >cam_soft.objdump
[cam_soft build complete]

23:43:11 Build Finished (took 5s.314ms)
```

figura 107, Fragmento del informe de compilación de Eclipse

El programa necesita:

4451 KB = 4,35MB de Espacio.

- SDRAM: 128MB
- SRAM: 2MB
- On-Chip: 0,47MB

Por lo tanto, el programa únicamente puede ser ejecutado desde la memoria SDRAM ya que el resto no son lo suficientemente grandes. La placa dispone también de una memoria Flash de 8MB que podría ejecutar el programa, pero al ser una memoria Flash tiene ciclos limitados de borrado de celdas por lo tanto no es adecuada para usarla como memoria del sistema puesto que su vida se vería acortada considerablemente.

Debido a todas estas pruebas, en el proyecto final se trabaja con una resolución de 320 x 240 píxeles debido a:

- El bajo rendimiento de las memorias disponibles.
- El Software ocupa demasiado espacio en la memoria de programa, solo cabe en la memoria SDRAM, inhabilitándola para ser usada como buffer de vídeo. Aun así, se han realizado pruebas también sobre esta memoria para caracterizar su rendimiento y se ha podido constatar que no es adecuada para ser usada como memoria de vídeo.
- La memoria On-Chip es demasiado pequeña en cuanto a tamaño, solo es lo suficientemente grande como para trabajar a 320 x 240 con 32 bits RGB.

Por lo tanto, que el sensor de la cámara capture a 640 x 480 ha sido una decisión tomada a partir de estas pruebas. 640 x 480 en el sensor significan 320 x 240 una vez convertido de Bayer a RGB, de esta manera no es necesario ningún tipo de conversión intermedia, más allá del escalado final para mostrar la imagen Live View en el monitor VGA.

### 5.3 Pruebas de rendimiento de lectura/escritura de fotografías

En este apartado se muestra la velocidad de lectura/escritura de fotografías en los 3 tipos de almacenamientos disponibles para fotografías: Tarjeta SD, Memoria Flash y Memoria RAM (Buffer, incluido en la memoria de programa que es la SDRAM).

	<i>Disparo único</i>	<i>Ráfaga, 3 Fotos</i>	<i>Ráfaga, 6 Fotos</i>	<i>Lectura</i>
<i>SD</i>	22,8s	-	-	6,62s
<i>Flash</i>	15,12s	-	-	1,07s
<i>Buffer</i>	1s	2,11s	3,55s	0,76s

*figura 108, Tiempos de almacenamiento/lectura de las Fotografías en las distintas memorias*

Se observa como lo más rápido es tomar y leer fotos desde el Buffer. La tarjeta SD y la memoria Flash no disponen de la función de disparo en ráfaga habilitada debido a que los disparos son muy lentos y, por lo tanto, pierde el sentido esta funcionalidad, cuyo objetivo es capturar fotos lo más rápido posible para poder dar sensación de movimiento, fotos capturadas con esperas de 15 o 22 segundos entre tomas impiden conseguirlo.

## 6- FUNCIONAMIENTO DEL SISTEMA

En el presente apartado se muestra el funcionamiento del sistema, explicando también las diferentes opciones de configuración que se han incluido en el mismo y el modo de empleo del sistema una vez se ha programado en la placa.



No hay que olvidar que si se emplea la versión gratuita de Quartus la placa permanece continuamente comunicada con el PC, de forma que Quartus apagará la CPU después de 1 hora. Si la placa permanece conectada durante más de una hora es posible que deje de responder y haya que volver a programarla desde Quartus.

Los controles de la placa para el proyecto son los siguientes:

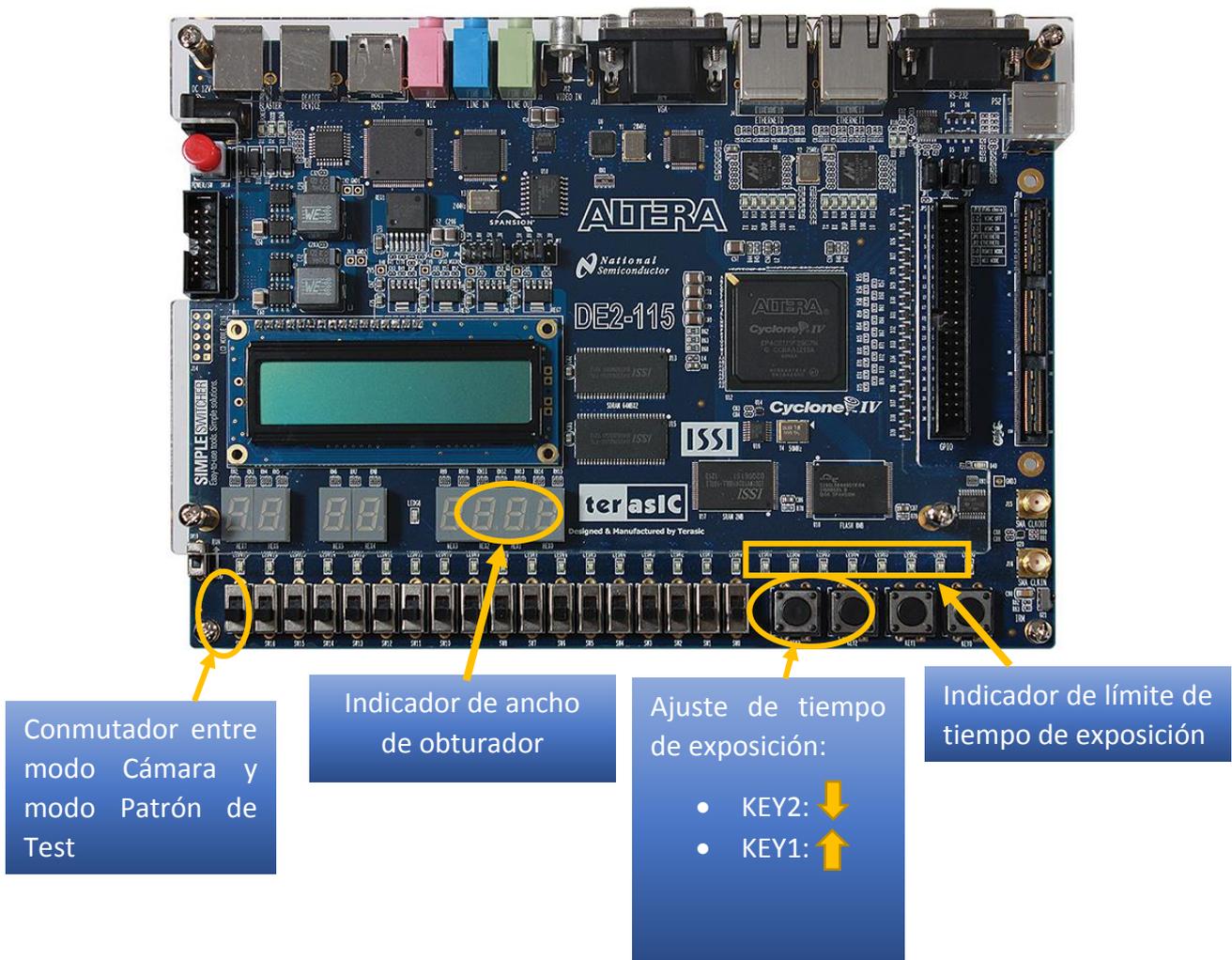


figura 109, Controles sobre la placa

El resto de funciones no mostradas en la figura anterior se controlan a través de la pantalla táctil. Los menús se muestran a continuación.

## Home

Es el punto de partida de todo el sistema. Desde él se puede acceder a los 4 sub-menús disponibles, solo hay que pulsar sobre ellos.



figura 110, Menú Home

Hay tres símbolos que se muestran en pantalla en todo momento en la parte superior para proporcionar información al usuario. Los posibles símbolos son los siguientes:

-  Indica que el DMA de escritura está activado.
-  Indica que el DMA de escritura está desactivado. Normalmente esto ocurre cuando el sistema está almacenando una fotografía. Si se muestra este símbolo, por pantalla no se muestra la visión Live View.
-  Indicación de que el sistema se encuentra en modo de fotografía única. Al disparar solo se toma una única foto.
-  Indicación de que el sistema se encuentra en modo de fotografía en ráfaga triple. Al disparar se toman 3 fotografías consecutivas.
-  Indicación de que el sistema se encuentra en modo de fotografía en hexaráfaga. Al disparar se toman 6 fotografías consecutivas.
-  Indicación de que se ha seleccionado la tarjeta SD como almacenamiento.
-  Indicación de que se ha seleccionado la memoria interna como almacenamiento. Esta memoria es volátil por lo que las fotos aquí almacenadas serán eliminadas una vez se apague la placa.
-  Indicación de que se ha seleccionado la memoria Flash como almacenamiento.

## Cámara

Este sub-menú permite tomar fotografías. Simplemente con pulsar sobre “Tomar Foto” se toma la fotografía y se almacena en el almacenamiento que haya sido seleccionado, el

cual se muestra en la esquina superior derecha de la pantalla. Pulsando sobre “Volver” se regresa al menú Home.



*figura 111, Menú Cámara*

### Galería

Desde aquí se pueden visualizar las imágenes tomadas. Se muestran un máximo de hasta 6 fotografías desde el almacenamiento seleccionado. Pulsando sobre la foto se muestra en la pantalla VGA conectada a la placa.



*figura 112, Menú Galería*

### Config

En este menú se puede configurar el almacenamiento que se desea utilizar, así como el modo de disparo que se desea utilizar. Al modificar estas opciones los símbolos en la parte superior de la pantalla son actualizados al modo que se esté utilizando.



figura 113, Menú Config

### Estado

En este menú se muestra información acerca del almacenamiento utilizado y el almacenamiento disponible en cada una de las memorias utilizables. Esto es de gran ayuda para conocer si algunas de las memorias están llenas, sobretodo en el momento en que se enciende la placa o se programa.



figura 114, Menú Estado

## 7- CONCLUSIONES

La idea detrás de este proyecto era estudiar las posibilidades que permite el módulo de cámara D5M como accesorio de la placa de desarrollo DE2-115 para investigar al punto al que se puede llegar y cuáles son las limitaciones.

Inicialmente fue bastante difícil conseguir que la imagen capturada por la cámara se mostrase por pantalla hasta que se descubrió que era culpa de una mala asignación del pin RESET del sensor. Llegado a un cierto punto de complejidad en el diseño, el sistema dejaba aleatoriamente de funcionar y se descubrió que era debido a que los clocks no tenían la asignación de incertidumbre (Uncertainty assignment) descrita en el segundo capítulo de este documento.

En cuanto a la pantalla táctil y los menús su programación ha sido sencilla, dado que Terasic facilita un proyecto de ejemplo con una serie de librerías muy útiles que permiten la creación de gráficos en pantalla rápidamente.

Se ha estudiado el rendimiento del sistema en las pruebas de cachés de la CPU y en las pruebas con las memorias, dado que era de especial interés el intentar capturar y visualizar imágenes de mayor resolución a la empleada, desgraciadamente bien por limitación de velocidad/capacidad de las memorias o bien por limitación de las IPs no ha sido posible lograrlo. Sin embargo, sí que se ha logrado la visualización Live View de la pantalla a 60fps, con lo cual, a pesar de estar empleando una resolución baja en el sensor, se obtiene la máxima fluidez posible (en caso de emplear el sensor con resoluciones mayores la tasa de frames por segundo desciende por limitaciones del propio módulo de la cámara). Esto último es imposible de alcanzar con las IPs del University Program de Altera, las cuales durante el desarrollo del proyecto han demostrado que sacrifican bastante calidad a favor de la facilidad de uso.

También se ha estudiado el correcto almacenado/lectura de las fotografías en la memoria SD, ya que la IP para el manejo de la tarjeta SD facilitada por Altera y también incluida en el University Program está muy limitada, limitada en velocidad y también limitada en cuanto a tamaño máximo de archivo, ya que no permite el almacenamiento de archivos de tamaño superior al de clúster del formato de archivos de la tarjeta. Esto es un motivo más para insistir en que las IPs del University Program tienen serias limitaciones y, como su nombre indica, parece que no tienen mucha utilidad fuera del ámbito educativo.

Como conclusión final, se han cumplido los objetivos:

- ✓ Conseguir un Proyecto funcional empleado el módulo de cámara facilitado.

El sistema captura fotografías correctamente controlado a través de la pantalla táctil.

- ✓ Utilizar las IPs ya disponibles.

Se utilizan las IPs del University Program para todo salvo el re-escalado de vídeo que emplea las IPs de Vídeo de Altera destinadas a DSP.

- ✓ Emplear la Pantalla Táctil facilitada como control del sistema.

La pantalla táctil funciona como se esperaba, su velocidad a la hora de dibujar depende del rendimiento de la CPU que la controla y el fabricante facilita una serie de librerías que facilitan el trabajo.

- ✓ Uso de las memorias no volátiles disponibles como almacenamiento para las fotografías.

La tarjeta SD es la más lenta tanto en lectura como en escritura mientras que emplear la memoria de programa (SDRAM + Cachés de la CPU) es lo más veloz. La memoria Flash es casi igual de lenta que la tarjeta SD a la hora de escribir, pero mucho más veloz a la hora de leer.

- ✓ Estudio de las limitaciones del sistema en cuanto a capacidad.

Por falta de capacidad de la memoria SRAM y memoria On-Chip se emplea la memoria SDRAM como memoria de programa. Cualquiera de las dos anteriores son más veloces para esta aplicación, pero no disponen del tamaño suficiente, Eclipse indica que el programa ocupa 4451KB (4,37MB, no cabe en ninguna memoria salvo la SDRAM).

La memoria On-Chip no es lo suficientemente grande como para ser usada como buffer de vídeo para resoluciones mayores de 320 x 240 24 bits.

- ✓ Estudio de las limitaciones del sistema en cuanto a velocidad

La falta de velocidad en la memoria SDRAM impide emplearla como buffer de vídeo en cualquier caso. La falta de velocidad en la memoria SRAM impide emplearla como buffer de vídeo a resoluciones superiores a 320 x 240 24 bits o 640 x 480 16 bits.

- ✓ Aprovechar los diferentes recursos/funcionalidades disponibles en la placa para conseguir el mejor resultado final.

El sistema ocupa alrededor de 18K LEs, únicamente un 16% de los elementos lógicos de la placa. Se emplea el 81% de la memoria distribuida (395KB) y 2 de los 4 PLLs disponibles. Este chip es muy grande y tiene muchos recursos, sin embargo, en esta placa son difíciles de explotar debido a las carencias que tiene en cuanto a velocidad/capacidad de las memorias disponibles.

## 8- REFERENCIAS

### Terasic Corp.

- 1- *TRDB - D5M User Guide*
- 2- *TRDB - D5M Hardware Specification V0.2*
- 3- *DE2-115 User Manual*
- 4- *THDB - HTG User Manual*
- 5- *LT24 User Manual*

### Altera Corp.

- 6- *NIOS II Classic Software Developer's Handbook*
- 7- *Embedded Peripherals IP User Guide*
- 8- *SDRAM Controller Core*
- 9- *SRAM Controller for Altera DE-Series Boards*
- 10- *Video IP Cores for Altera DE-Series Boards*
- 11- *Altera University Program Secure Data Card IP Core*
- 12- *Performance Counter Core*
- 13- *Video and Image Processing Suite User Guide*

### Otros

- 14- (ISSI Corp.) *42-45R-S\_86400D-16320D-32160D Datasheet*
- 15- (ISSI Corp.) *IS61LV25616 Datasheet*
- 16- (Spansion Corp.) *S29GL-N MirrorBit® Flash Family Datasheet*

