

RESULTADOS Y CONCLUSIONES

6.1. INTRODUCCIÓN

En este último capítulo con el que concluye la memoria, se pretende dar una valoración del proyecto, presentando los resultados obtenidos y las posibles ampliaciones.

6.2. RESULTADOS

En primer lugar, se estudian los resultados de la implementación, esto es, los recursos consumidos por el sistema, así como la velocidad de trabajo, frecuencia de reloj que admite el diseño,... ; por otro, se comentan algunos aspectos de las soluciones adoptadas, algunos de los cuales ya se han visto a lo largo de la memoria. Asimismo, se extraerán conclusiones sobre el proceso de diseño y de depurado, así como de la facilidad de testeabilidad del sistema, veremos qué métodos y sistemas se han empleado, cuáles son sus ventajas, inconvenientes y su alcance y utilidad.



6.2.1. PRESTACIONES DEL SISTEMA

Al tratarse de un prototipo, las prestaciones son limitadas, como cabe esperar de un sistema en sus etapas iniciales de desarrollo: en ocasiones no se produce un seguimiento continuo del punto luminoso, o bien el sistema no reacciona frente a variaciones bruscas de la intensidad del punto luminoso. Existe una tasa de error relativamente alta en cuanto al funcionamiento del sistema. Otro aspecto que constituiría una interesante línea de desarrollo, es la tasa de errores del sistema en función de la distancia, es decir, la precisión.

Esta tasa de error que presenta el funcionamiento del sistema, puede deberse a la propia placa de adquisición de datos, que fue realizada íntegramente en el laboratorio. Sus pistas se diseñaron de forma que al realizar el revelado, no se deterioraran o desaparecieran del circuito. Por tanto, pueden presentar longitudes y anchuras no óptimas que afecten a las señales provocando atenuaciones y retardos no deseados.

6.2.2. RECURSOS DEL SISTEMA

Aunque el diseño del circuito lógico implementado en la FPGA se desarrolló considerando un sistema de tres cámaras, el paralelismo con el que se ideó dicho diseño permitió realizarlo y depurarlo posteriormente para cada sistema por separado y en varias etapas.

En primer lugar sólo se tuvo en cuenta a una de las cámaras y se realizó el diseño incluido en el proyecto: *Proy_1c*. A continuación, se añadió una segunda cámara, obteniéndose *Proy_2c*. Para ello se adaptaron algunos bloques, como el de configuración, se añadieron nuevo bloques, como el que promedia el valor de las coordenadas y se reutilizaron circuitos de *Proy_1c*. Finalmente se incluyó la tercera cámara, realizando *Proy_3c*, volviendo a incluir bloques de los diseños anteriores.

En los siguientes apartados se analizarán las características de cada uno de esos diseños.



6.2.2.1. PROYECTO PARA UNA CÁMARA

El sistema para una cámara consume la mitad de los recursos para los bloques lógicos de los que dispone la FPGA Virtex 300. Los recursos de memoria no se utilizan. En la siguiente tabla podemos ver el consumo de recursos.

Design Summary			

Number of errors:	0		
Number of warnings:	1		
Number of Slices:	1,637	out of 3,072	53%
Number of Slices containing unrelated logic:	0	out of 1,637	0%
Number of Slice Flip Flops:	2,613	out of 6,144	42%
Total Number 4 input LUTs:	1,409	out of 6,144	22%
Number used as LUTs:		1,398	
Number used as a route-thru:		11	
Number of bonded IOBs:	38	out of 166	22%
Number of Tbufs:	32	out of 3,200	1%
Number of GCLKs:	2	out of 4	50%
Number of GCLKIOBs:	1	out of 4	25%
Number of DLLs:	1	out of 4	25%
Number of Captures:	1	out of 1	100%
Total equivalent gate count for design:	41,563		
Additional JTAG gate count for IOBs:	1,872		
Device utilization summary:			
Number of External GCLKIOBs	1	out of 4	25%
Number of External IOBs	38	out of 166	22%
Number of SLICES	1,637	out of 3072	53%
Number of GCLKs	1	out of 4	25%
Design statistics:			
Minimum period:	19.287ns	(Maximum frequency: 51.848MHz)	
Maximum combinational path delay:	24.868ns		
Maximum net delay:	21.0245ns		

Figura 6.1: Consumo de recursos del sistema *Proy_1c*

Las estadísticas tras la ubicación y rutado de la lógica nos indican que la máxima frecuencia de funcionamiento del sistema será de 51.848Mhz, siendo el máximo retraso de 21.024ns. Esta frecuencia, superior a la del reloj del sistema (50MHz) pone de manifiesto que la velocidad no será el factor limitante del diseño.



En la siguiente tabla se muestran los recursos consumidos por cada uno de los bloques que forman el diseño *Proy_1c*.

<i>Bloque</i>	<i>Slice(total:3072) (%)</i>	<i>Número de Puertas equivalentes</i>	<i>Frecuencia Máx(MHz)</i>	<i>Máximo Retraso(ns)</i>
<i>Configuración</i>	143 (4%)	2054	63.012	16.148
<i>Coordenadas_xy</i>	36 (1%)	700	101.906	5.184
<i>Calcula_coord</i>	604 (19%)	13610	97.267	9.222
<i>Cal_valor_medio</i>	54 (1%)	1216	83.872	3.402
<i>Top_VGA1cam</i>	72 (2%)	1298	84.331	7.113
<i>PC_topc1</i>	53 (1%)	1198	83.445	8.926

Tabla 6.1: Recursos utilizados por los diferentes bloques

Las únicas características destacables son la elevada frecuencia del bloque que determina la fila y columna de cada píxel: *coordenadas_xy* y el área ocupada por el circuito encargado de calcular las coordenadas: *calcula_coord.*, que es bastante superior a la que ocupan los demás bloques.

6.2.2.2. PROYECTO PARA DOS CÁMARAS

Este diseño está formado, prácticamente, por dos diseños como el realizado para una de las cámaras. Salvo algunos bloques que son comunes, como el que se encarga de la configuración, y los que controlan los interfaces con el PC y el monitor VGA.

Para obtener este diseño, se modifica el bloque de configuración y se añaden otros circuitos, de modo que los recursos empleados por *Proy_2c* son los siguientes:



Device utilization summary:		
Number of External GCLKIOBs	1 out of 4	25%
Number of External IOBs	53 out of 166	31%
Number of SLICES	3017 out of 3072	98%
Number of CAPTURES	1 out of 1	100%
Number of DLLs	1 out of 4	25%
Number of GCLKs	2 out of 4	50%
Number of TBUFs	48 out of 3200	1%
Design statistics:		
Minimum period:	18.344ns (Maximum frequency: 54.514MHz)	
Maximum combinational path delay:	30.835ns	
Maximum net delay:	8.833ns	

Figura 6.2: Consumo de recursos del sistema *Proy_2c*

Se puede observar un aumento del área ocupada del 45% con respecto a *Proy_1c*. Además, la frecuencia del sistema también es mayor que para el caso de una sola cámara. De nuevo se tiene que la velocidad no supone ninguna limitación. Sin embargo, el área sí lo es, más bien, lo puede llegar a ser, ya que para incluir la tercera de las cámaras se necesitarán más recursos de los que dispone la Virtex 300. Más adelante se enunciará una posible solución.

Este aumento de área se debe principalmente a los bloques añadidos:

- Un bloque *coordenadas_xy*: supone un aumento del 1% (ver tabla 6.1).
- Dos bloques *calcula_coord.*: cada uno contribuye con el 19%.
- Tres bloques *promedia2*: con una ocupación del 1% cada uno.

Sólo estos tres circuitos aumentan el área en un 42%. Si a esto añadimos las ampliaciones que hay que realizar de algunos circuitos, como es el caso de *top_vga2cam*, o *PC_top*, que del 1% pasan al 2%, se obtiene la cantidad de 44%, muy próxima a 45%.



6.2.2.3. PROYECTO PARA TRES CÁMARAS

Tal y como cabía esperar, al añadir los bloques necesarios para procesar los datos de la tercera cámara, los recursos de la Virtex 300 son insuficientes. Necesitaríamos un 30% más de los recursos que dispone la FPGA para los circuitos lógicos.

Design Summary:			
Number of errors:	1		
Number of warnings:	1		
Number of Slices:	4,040	out of 3,072	131%
Number of Slices containing unrelated logic:	303	out of 4,040	7%
Number of Slice Flip Flops:	7,437	out of 6,144	121%
Total Number 4 input LUTs:	3,496	out of 6,144	56%
Number used as LUTs:		3,476	
Number used as a route-thru:		20	
Number of bonded IOBs:	67	out of 166	40%
Number of Tbufs:	48	out of 3,200	1%
Number of GCLKs:	2	out of 4	50%
Number of GCLKIOBs:	1	out of 4	25%
Number of DLLs:	1	out of 4	25%
Number of Captures:	1	out of 1	100%
Total equivalent gate count for design:	101,551		
Additional JTAG gate count for IOBs:	3,264		

Figura 6.3: Consumo de recursos del sistema *Proy_3c*

El número de recursos necesarios puede reducirse a costa de disminuir el grado de paralelismo conseguido. Para cada procesador de vídeo se tienen dos bloques *calcula_coord.* (ver *figura 5.40*), donde, como se vio en el Capítulo V, primero se obtiene la suma de las coordenadas x_i (o y_i) y el área, y luego se calcula el cociente de ambas cantidades (ver *figura 5.41*). Si extraemos el circuito divisor del bloque *calcula_coord.* y lo utilizamos de forma compartida para las tres cámaras, en lugar de tener 6 procesos de división realizándose de forma paralela, se calcularían esas 6 divisiones de forma secuencial, mejorando en área la implementación. Lo más importante es que esta solución no afectaría al funcionamiento en tiempo real que se le exige al sistema:



La división se produce al final de cada pantalla, durante el periodo de *blanking* vertical, que es de 1.6ms. Sabiendo que cada división tarda en realizarse 29 ciclos de reloj y suponiendo que el máximo retardo del circuito *calcula_coord.*: 9.222ns, se produce en el bloque divisor, se tiene que el tiempo necesario para realizar las 6 divisiones es:

$$6 \text{ div} \times \left[29 \frac{T_{CLK}}{\text{div}} \times 20 \frac{\text{ns}}{T_{CLK}} + 9.222\text{ns} \right] = 3535.332 \text{ ns} = 3.535\mu\text{s} < 1.6 \text{ ms}.$$

Por tanto, esta solución no afectaría a la velocidad de respuesta del sistema. En caso de seguir con el diseño original, se necesitaría una FPGA superior a la Virtex 300.

6.3. CONCLUSIONES

A lo largo de la realización del proyecto las dificultades encontradas han sido muchas y de distinta índole.

En primer lugar, la elaboración de un proyecto basado en sistemas de visión requiere de conocimientos multidisciplinares (Visión Artificial, Geometría de formación de imágenes, Tratamiento Digital de Imágenes, etc.). Si además el proyecto requiere la realización de una placa de circuito impreso y la utilización de FPGAs para la implementación hardware del diseño, entonces hay que añadir a la lista anterior disciplinas como la Electrónica y la Microelectrónica.

Por otro lado, la búsqueda de información resulta complicada y harto pesada, no por escasez de la misma, sino por todo lo contrario. Sobre el tema de Sistemas de Visión 3D existen grandes volúmenes de información y en muchas ocasiones, no resultan muy útiles. Cuando se consultan los sistemas de visión comerciales, la mayoría de las veces se obtienen las características de éstos pero no la descripción de los procedimientos que siguen. Además hay que señalar que no se encontró ningún sistema



con características similares al nuestro. Es decir, ninguno de los sistemas consultados emplea sólo una FPGA para el procesamiento de las imágenes. Todos utilizan sofisticadas tarjetas formadas por DSPs, FPGAs, microprocesadores,... De ahí que la dificultad fuese aún mayor.

En la elaboración de la tarjeta de adquisición de datos, la principal dificultad fue conseguir los osciladores de cristal de 24.576MHz de frecuencia, pese a que su utilización en circuitos de vídeo es muy común.

En cuanto al sistema hardware implementado en la FPGA, se encontraron dificultades con el espacio disponible en la FPGA, ya que como se ha comentado en el apartado anterior, para el sistema de las tres cámaras, la estructura en paralelo diseñada requiere más recursos de los ofrecidos por la FPGA. Además, lo más destacable fue la modificación del circuito encargado del cálculo del valor medio, debido al control automático de ganancia realizado por las cámaras y cuya existencia desconocíamos hasta el momento de la depuración.

En el depurado del sistema tuvimos problemas con el alcance de las herramientas de las que disponíamos, lo que nos obligó a diseñar varios sistemas de prueba para comprobar el correcto funcionamiento bloque a bloque.

Realizando una división porcentual y aproximada del tiempo empleado en la elaboración del proyecto se tiene que:

- La realización de la placa de circuito impreso consumió un 15% .
- En la búsqueda de información, clasificación y estudio de la documentación obtenida se empleó un 25%.
- En la definición del proyecto, elaboración de una solución y en la implementación hardware de la misma, un 25%.
- Y el 35% restante en el depurado del sistema hardware, incluidas las realimentaciones y modificaciones de la implementación hardware.



6.3.1. FUTURAS LÍNEAS DE DESARROLLO

A continuación se tratarán algunos aspectos del sistema que son susceptibles de mejora.

En primer lugar, tal y como se ha propuesto anteriormente, sería interesante modificar el proyecto para las tres cámaras de modo que el área ocupada se ajuste al tamaño de la FPGA Virtex 300.

En lo referente a la precisión del sistema, proponemos una solución que permitirá aumentarla, aunque en un principio pensamos que no sería posible introducir este cálculo en la V300 y que se necesitaría de una FPGA mayor. Se trataría de realizar el cálculo de los centros de gravedad como:

$$\bar{x}_1 = \frac{1}{n} \sum_i v_{i1} x_{i1} \quad \bar{y}_1 = \frac{1}{n} \sum_i w_{i1} y_{i1}$$

siendo v_{i1} y w_{i1} los valores de intensidad del píxel i -ésimo (dado por la cámara 1 en este caso). De este modo cada coordenada (x_{i1}, y_{i1}) estará ponderada de distinto modo dependiendo de la intensidad luminosa y no como ocurre en nuestro sistema, donde $v_{i1}=1$ y $w_{i1}=1$.

Una interesante línea de desarrollo también sería el estudio detallado sobre la tasa de errores del sistema en función de la distancia, es decir, determinar la precisión con la que el sistema calcula la posición del punto luminoso.

Por último, el siguiente paso será determinar la posición de más de un punto luminoso. Con toda seguridad, el sistema será complejo y requerirá una FPGA mayor.



6.3.2. APLICACIONES

En este último apartado se pretenden resumir algunos campos y aplicaciones concretas en los que el sistema desarrollado podría tener interés.

El prototipo se ha diseñado para ser aplicado en la industria, en el control de robots en soldadura de arco. La comparación entre la trayectoria real y la que en teoría debe seguir el robot se utilizará para determinar el grado de control del brazo del robot.

Otro campo de aplicación podría ser la Aeronáutica, por ejemplo, para analizar el comportamiento de una maqueta de un prototipo de avión, sometida a distintas condiciones de turbulencias en un túnel de viento. Para ello, en los puntos críticos del avión, se colocarían reflectores de modo que, dentro del túnel, se vieran como puntos luminosos. Para este tipo de aplicación, el sistema tendría que ser capaz de determinar la trayectoria de cada uno de los puntos por separado. Analizando las trayectorias de esos puntos se podría conocer el comportamiento del avión.

De forma similar, pero en la Disciplina Deportiva, se podrían analizar y corregir los movimientos de los deportistas. Es decir, colocando los reflectores en las principales articulaciones del cuerpo de, por ejemplo, un tenista, se puede analizar el movimiento que realiza en el saque y determinar los aspectos que se puedan mejorar.

Una posible aplicación, esta vez de la parte más física del sistema: placa de adquisición y placa UNSHADES-1, sería la de clasificar frutas (por ejemplo naranjas) en verdes y maduras. En aplicaciones como ésta, es fundamental el uso de imágenes en color, lo cual se puede implementar perfectamente en nuestro sistema mediante la utilización de cámaras a color y programando los decodificadores para ese tipo de formato.

En otro ámbito, donde sería aplicable el sistema es en el reconocimiento de formas. Por ejemplo, el reconocimiento de letras, aplicable en la detección de matrículas.



