

OBTENCIÓN DE CENTROIDES EN IMÁGENES ASTRONÓMICAS BASADA EN FPGA

J. TRUJILLO, M. RODRÍGUEZ, J.M. RODRÍGUEZ Y E. MAGDALENO

*Departamento de Física Fundamental y Experimental, Electrónica y Sistemas. Facultad de Física.
Universidad de La Laguna. España.*

El objetivo de este trabajo es el de introducir al alumno en el campo de las nuevas metodologías de diseño electrónico digital basadas en FPGA. Para conseguir este objetivo y como aplicación práctica se ha propuesto al alumno, como proyecto de fin de carrera, el diseño e implementación de un sistema de cálculo de centroides en imágenes astronómicas. El desarrollo de esta aplicación permite al alumno obtener un aprendizaje significativo sobre electrónica digital, procesamiento hardware de imágenes y manejo de herramientas profesionales de diseño.

1. Introducción

El uso de nuevas metodologías de diseño electrónico digital basadas en FPGA permite obtener ventajas en cuanto a velocidad de ejecución sobre DSP y microprocesadores. En el pasado, la electrónica se basó en la fabricación de un hardware único, con capacidad para ejecutar un programa almacenado en memoria, lo cual, permitiría la resolución de cada problema concreto. La llegada de las FPGA, su continuado crecimiento y la aparición de los lenguajes de descripción hardware conjuntamente con herramientas elaboradas de simulación y síntesis, han permitido que pueda abordarse la creación de hardware específico para cada algoritmo, haciendo desaparecer por tanto cualquier programa y con él, el tiempo necesario para leer y decodificar cada una de las instrucciones antes de ejecutarlas.

El objetivo de este trabajo es la implementación de un sistema de cálculo de centroides basado en el algoritmo de correlación sobre imágenes astronómicas. Este tipo de cálculo es un proceso vital en un sistema en lazo cerrado para la corrección de turbulencias atmosféricas en un telescopio. Aunque para los telescopios actuales, la capacidad de procesado que ofrecen DSP y microprocesadores es suficiente, la extrapolación a las necesidades de los futuros grandes telescopios arroja serias sospechas de inviabilidad [1], contando incluso con que se mantenga la ley de Moore [2] de crecimiento exponencial de las capacidades de los circuitos integrados. La aparición de las FPGAs y sus aplicaciones en este campo de óptica adaptativa arroja de modo directo una ganancia en el retardo de actuación, parámetro de gran importancia en un sistema de lazo cerrado para grandes telescopios.

Este trabajo fue desarrollado bajo el marco del proyecto de fin de carrera de Ingeniería Técnica Industrial especialidad en Electrónica Industrial de la Universidad de La Laguna. Por ello, la realización de un trabajo de estas características hace que el alumno adquiera un aprendizaje significativo en metodologías modernas de diseño electrónico basadas en FPGAs, HDL y procesamiento de señales.

El Trabajo fue enteramente desarrollado en VHDL con lo cual es necesario que el alumno que lo aborde tenga unos conocimientos avanzados de diseño electrónico mediante lenguajes de descripción hardware y FPGA. Aunque en la titulación de procedencia no se imparten dichos conocimientos el alumno que desarrolló este proyecto tuvo una formación intensiva de una semana tanto en VHDL como FPGA (40 horas).

2. Descripción general de funcionamiento

2.1. Características de las imágenes

Las imágenes de entrada al sistema provienen de un sensor Shack-Hartmann. Este sensor es el más usado en astronomía para óptica adaptativa en telescopios. Está formado por una matriz de microlentes y un detector de luz. Las distorsiones en el frente de onda incidente producen pendientes locales sobre cada una de las microlentes por lo que la imagen se desplaza proporcionalmente a la pendiente media del frente de onda [3, 4]. (Fig. 1a)

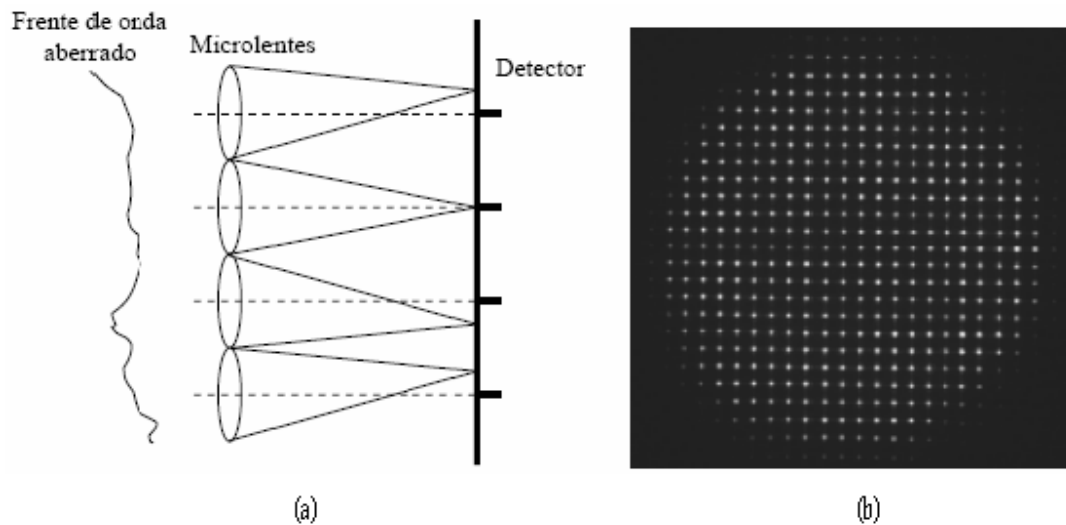


Figura 1. Diagrama del sensor Shack-Hartmann e imagen producida por él.

Las imágenes que esperará el sistema a la entrada son del tipo de la imagen mostrada en la Fig. 1b pudiendo variar el tamaño de la imagen así como el número de subpupilas que contiene.

2.2. Algoritmos de cálculo de centroides

Aunque el sistema que se ha implementado contiene algoritmos encargados de recepción y manejo de datos, control del sistema y algoritmo de cálculo de centroides, sólo nos centraremos en este último que está basado en la correlación de imágenes y su posterior ajuste cuadrático.

La correlación entre dos imágenes es uno de los mejores métodos de estimar el desplazamiento global entre ambas, este desplazamiento será la información usada para la estimación del centroide en cada imagen. Este método es además muy efectivo para eliminar el ruido de los píxeles más alejados [5].

El cálculo de la correlación, C , se efectúa entre dos imágenes, la primera, a la que nos referiremos simplemente como I , y una segunda imagen que llamaremos K (kernel), según Ec. 1.

$$C(x, y) = \sum_{i,j} I_{i,j} K(x_i + x, y_j + y) \quad (1)$$

Esto significa que el valor de cada píxel de salida representa la suma ponderada de los píxeles vecinos (Fig. 2). Además, la salida será una imagen con unas dimensiones definidas por las dimensiones de la imágenes de entrada, esto es, dimensión de la imagen menos la dimensión del kernel más uno.

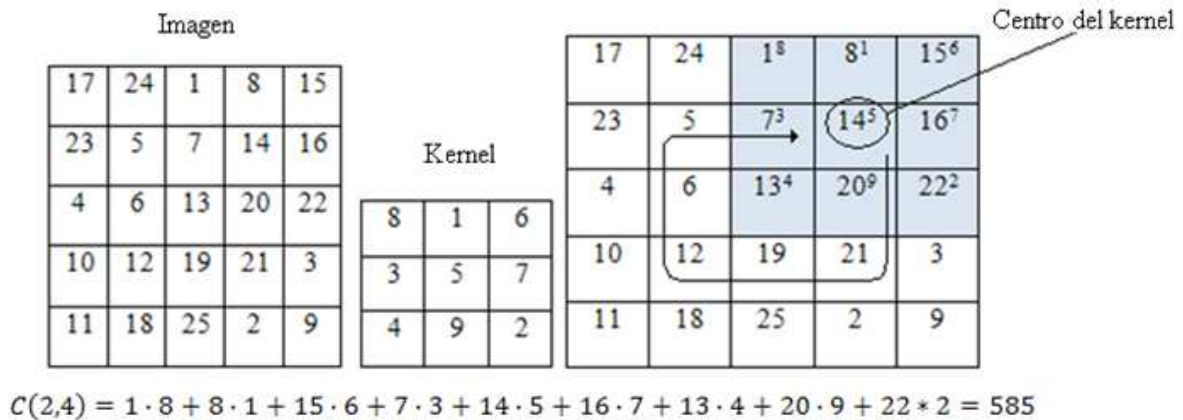


Figura 2. Obtención de la correlación entre dos imágenes.

La salida de la correlación de dos imágenes es otra imagen, pero en el caso que nos ocupa sólo consideraremos la parte de la imagen que tiene utilidad para calcular el centroide, esto es, el píxel de valor máximo de la imagen resultante y sus primeros píxeles vecinos, además sus coordenadas cartesianas. Esta información de salida es transferida a un bloque que mediante un ajuste cuadrático obtiene con resolución sub-píxel una aproximación al centroide de la imagen.

El ajuste cuadrático de los datos provenientes de la correlación tiene como objetivo obtener con precisión sub-píxel el desplazamiento de la imagen de entrada frente al kernel. Se basa en ajustar dos parábolas separadamente para los ejes x e y mediante dos puntos alrededor del máximo a lo largo de los ejes cartesianos (Fig. 3).

El ajuste cuadrático se rige según Ec. 2.

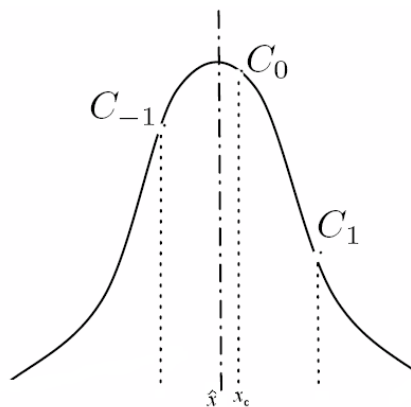


Figura 3. Representación del ajuste cuadrático, donde C_0 representa el máximo de la correlación, C_{-1} representa al píxel adyacente situado a la izquierda o bien debajo y C_1 representa al píxel adyacente situado a la derecha o arriba.

$$\frac{x}{x_c} = x_c - \frac{0.5[C_1 - C_{-1}]}{C_1 + C_{-1} - 2C_0} \quad (2)$$

2.3 Funcionamiento del sistema

El sistema recogerá las imágenes provenientes del sensor de frente de onda. Posteriormente se calcula la imagen kernel para la correlación. Esta imagen es calculada como la media de todas ellas considerando sólo la parte central de la imagen.

El siguiente paso es la obtención de la referencia para el cálculo final de los centroides. Se calcula el centroide mediante correlación entre la imagen kernel y una imagen de referencia, almacenada en memoria previamente. Con la referencia calculada puede empezar el cálculo sucesivo de los centroides de la subpupilas que contiene la imagen de entrada. A la salida se proporciona el valor del centroide calculado corregido con respecto a su referencia.

3. Arquitectura del sistema

El sistema fue diseñado enteramente en VHDL y como soporte tecnológico se ha usado una FPGA Xilinx Virtex-4. Está compuesto por una serie de módulos (Fig. 4) que serán detallados en los apartados siguientes:

- Clock manager: Contiene los componentes necesarios para la síntesis de los relojes requeridos por el sistema.
- Adquisición: Este módulo recoge la imagen de entrada almacenando en memoria únicamente la parte de la imagen perteneciente a las subpupilas y que por lo tanto contienen información útil.
- Correlador: Este módulo realiza la correlación tal como se explica en el apartado 2.2.
- Controlador: Forma una máquina de estados que gestiona todas las señales de sincronismo del sistema, así como señales de carga para registros intermedios.
- Ajuste cuadrático: Este módulo realiza el algoritmo de ajuste cuadrático. Se trata de un circuito totalmente combinacional pues ha de realizar su trabajo en el tiempo que existe entre una correlación y la siguiente.
- Restador: Realiza la corrección de los datos de salida del módulo de ajuste cuadrático frente a la referencia calculada.
- Multiplexor de salida: Se trata de un multiplexor para los datos de x e y.
- Registros intermedios: Son registros ubicados en las entradas y salidas de los módulos combinacionales.

3.1. Clock manager

Este módulo sintetiza los dos relojes necesarios para el funcionamiento del sistema, estos son, un reloj con la misma frecuencia que el reloj de entrada al sistema, y otro reloj con la mitad de frecuencia, nombrados en el código como "clk2x" y "clk180" respectivamente (Fig. 5). Está compuesto por dos tipos de primitivas específicas de la FPGA Virtex-4 empleada, estas son el PMCD (*phase matched clock divider*) y BUFG (*global clock buffer*) (Fig. 6).

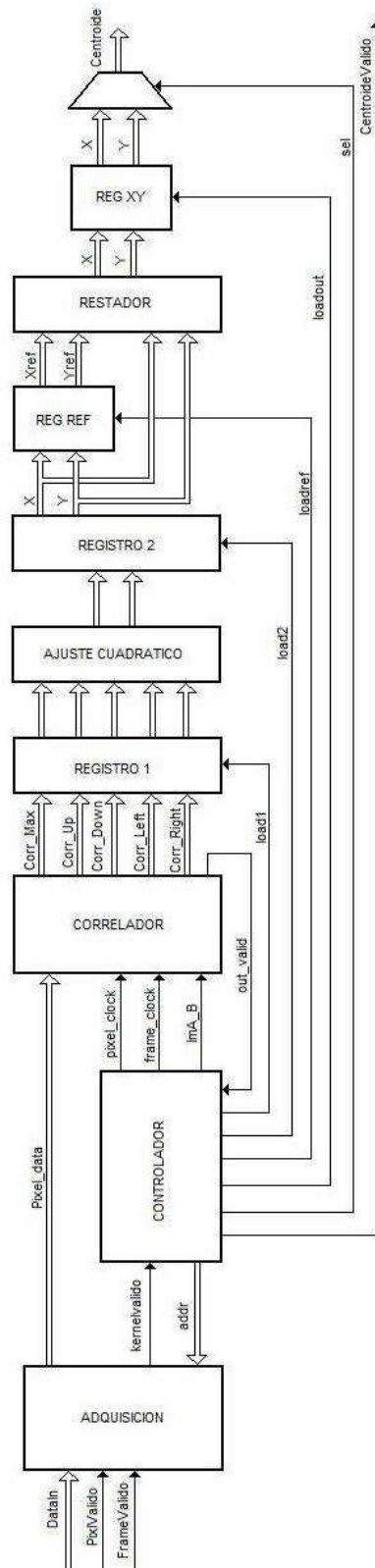


Figura 4. Esquema a nivel de módulos

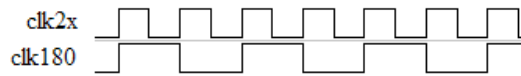


Figura 5. Señales de reloj empleadas por el sistema

- **PMCD.** Esta primitiva se emplea para obtener el reloj con frecuencia dividida por 2. Tiene la capacidad de proporcionar relojes a la salida alineados en fase entre sí, eliminando los posibles problemas de sincronismo que pueden surgir cuando las fases no están alineadas.
- **BUFG.** Este componente recibe a su entrada una señal de reloj obteniéndose a la salida una señal de reloj con alto fan-out. Se emplea en el módulo conectado a las salidas del PMCD.

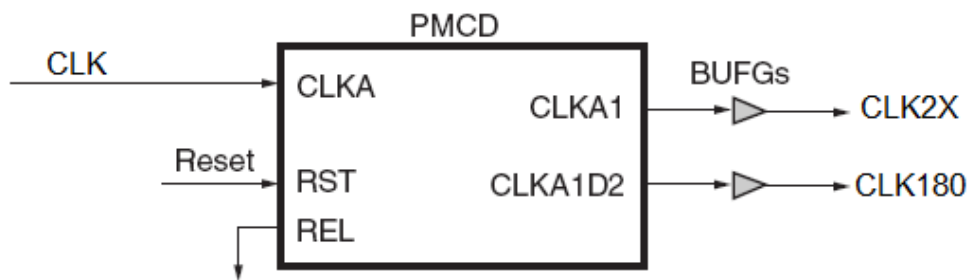


Figura 6. Esquema del módulo “clock manager”

La señal de reset actúa forzando todos los relojes de salida a 0, cuando se desactiva la señal, los relojes comienzan a oscilar un ciclo de reloj después.

3.2. Adquisición

Este módulo tiene tres funciones dentro del diseño: recepción de datos, cálculo de “kernel”, salida de datos.

3.2.1. Recepción de datos

La adquisición de la imagen de entrada se hace almacenando únicamente los datos que contienen información relevante para el cálculo que nos ocupa, esto es, los datos de las regiones de la imagen que contienen las subpupilas y no el espacio que hay entre ellas, esto permite optimizar el uso de memorias BRAM así, como una organización más sencilla de los datos que permitirá abordar cálculos posteriores con mayor sencillez.

Para discernir entre los datos que se han de almacenar y los que no, se emplea una máscara del mismo tamaño que toma valores binarios, 1 para datos almacenables y 0 para los desechables, dicha máscara ha de crearse teniendo en cuenta las posiciones de cada subpupila (Fig. 7). Se procede leyendo los datos de entrada y los de la máscara al mismo tiempo, luego cuando el dato de la máscara es 1 se almacenará el dato proveniente de la cámara, cuando sea 0 se omitirá.

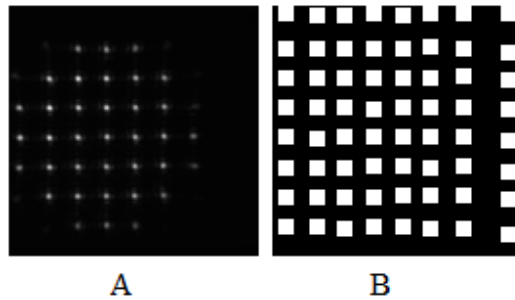


Figura 7. Imagen de entrada (A), máscara (B)

La memoria de entrada, en la que se almacenan los datos de entrada se organiza escribiendo los datos de las subpupilas una a continuación de la siguiente, y no como entran al sistema, es decir, por filas independientes (Fig. 8). Además en esta memoria se reserva un espacio para el “kernel” y se inicializa la imagen de referencia

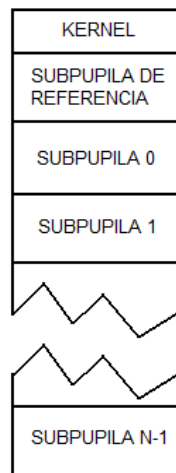


Figura 8. Organización de datos en “memoria_entrada”

Esta memoria, pues ha de inicializarse empleando una subpupila de referencia, esto quiere decir que se ha de conocer la imagen proveniente del sensor de frente de onda en un estado ideal en el que no haya turbulencias. En el archivo de inicialización de la memoria se ha de dejar el espacio suficiente al comienzo de la memoria para almacenar el “kernel” posteriormente.

3.2.2. Cálculo del “kernel”

El cálculo del “kernel” se realiza calculando la media de la parte central de todas las subpupilas que componen la imagen. Este cálculo se realiza durante la carga de datos de la imagen de entrada, se acomete empleando un conjunto de tantos acumuladores como píxeles contenga el “kernel”, finalizada la carga de la imagen, se divide el valor a la salida del acumulador por el número de subpupilas. Nótese que el número de subpupilas es potencia de 2, luego la división se realiza mediante un desplazamiento binario hacia la derecha de $\log_2(\text{núm. subpupilas})$. Al terminar la escritura de los datos de las subpupilas se

escriben los datos del “kernel”, cuando esta última escritura concluye se activa la salida “kernel”valido” indicando que los datos en memoria están listos para procesar.

3.2.3. Salida de datos

Con la señal “kernel_valido” en valor alto, cualquier dirección en rango válido en la entrada “addrin” obtendrá a la salida “datos_salida” el dato perteneciente a la dirección de entrada.

3.3. Módulo de correlación

La realización del módulo de correlación en VHDL, se basó en el documento “Especificaciones del módulo de correlación directa para FPGA” [1].

El módulo debe esperar al inicio la llegada de dos imágenes. La primera será la que se emplee como “kernel”, y la segunda será la imagen sobre la que se debe hacer la correlación. Una vez se extrae el primer resultado, el módulo está listo para recibir una nueva imagen aunque también puede volver a recibir una nueva imagen tipo “kernel” y utilizarla para el cálculo de la correlación a partir de este momento.

3.4. Controlador

Este módulo está compuesto por cuatro máquinas de estados de Mealy que controlan varios aspectos del diseño. Controla las direcciones de lectura de la memoria del módulo “adquisicion”, las señales de sincronismo del “correlador”, las señales de carga de los registros intermedios, el control del multiplexor de salida y la señal “centroidevalido”.

3.5. Ajuste cuadrático

Se creó un módulo de ajuste cuadrático que debía cumplir con el objetivo de realizar este cálculo en el menor tiempo posible, ya que entra en funcionamiento en el tiempo intermedio entre la salida de dos correlaciones, que debido a la flexibilidad del diseño es variable. Es por esto que se decidió que el módulo fuera completamente combinacional, esto es, que no dependa del reloj del sistema el tiempo necesario para efectuar el cálculo, con el inconveniente añadido del mayor uso de recursos de la FPGA que requiere esta metodología.

Este módulo está formado a alto nivel por 3 componentes:

- Doblador de imagen: Este módulo se encarga de realizar la corrección de los datos de salida del módulo de correlación cuando el máximo se sitúa en un borde de la imagen resultante y por tanto no se puede proporcionar el valor de uno de los píxeles adyacentes
- Control: Este módulo calcula los datos de entrada al correlador, mediante comparaciones de los píxeles adyacentes pertenecientes al mismo eje.
- División (replicado para x e y): Éste es un algoritmo de división síncrono para números naturales codificados en binario [6], cuando el dividendo tiene mayor número de bits que el divisor. Sea la operación “ $q=x/y$ ”, donde m es el número de bits de “x” (y por tanto de “c”) y “n” el número de bits de “y”. El algoritmo se basa en la implementación de “m” pasos de división (Fig. 9).

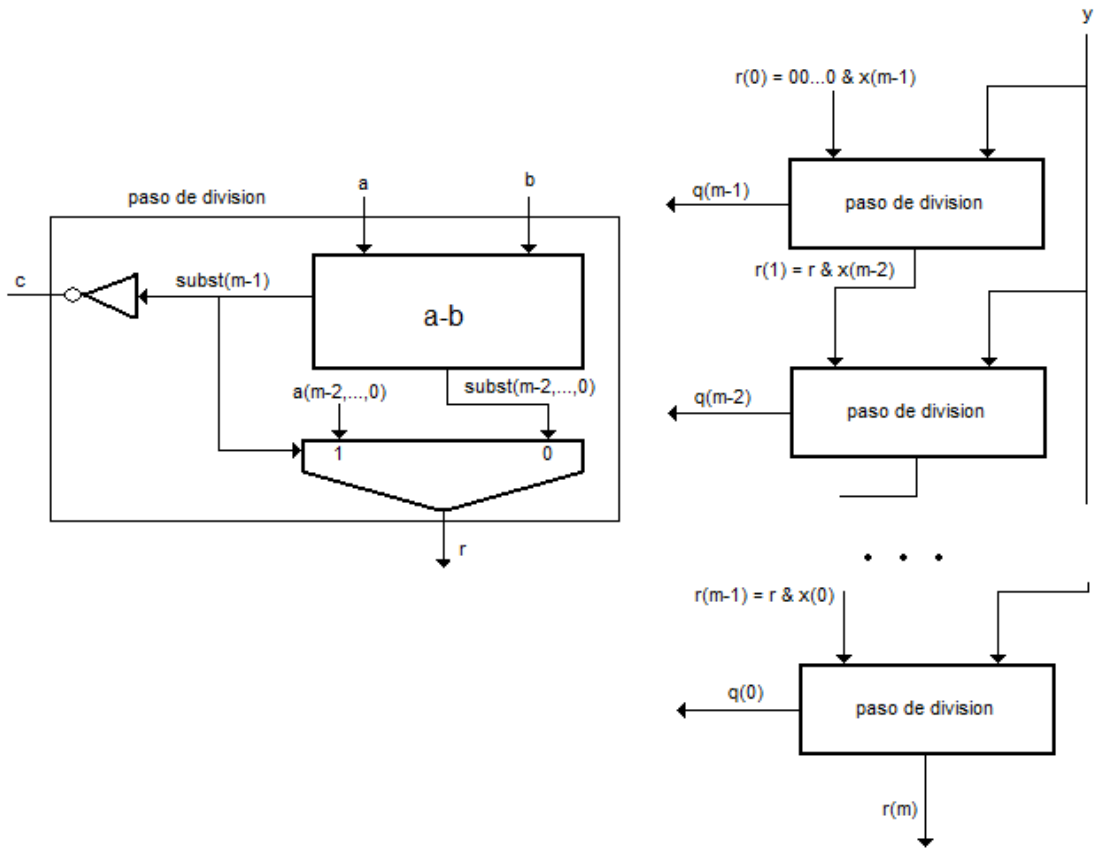


Figura 9. Esquema a nivel de bloques del módulo de división

Al tratarse de un módulo puramente combinacional, se debe tener en cuenta el tiempo máximo que puede tardar en procesar los módulos hasta obtener una salida válida. Este valor ha de influir en el momento en que se aplican las señales de carga a los registros intermedios que operan junto a este módulo y que se ha tenido en cuenta en el diseño del módulo “controlador”.

3.6. Restador

Se implementó un restador en punto fijo combinacional con objeto de sustraer la referencia al dato calculado.

Al igual que el módulo de ajuste cuadrático, estamos ante un módulo combinacional, luego habrá de tenerse en cuenta, el tiempo máximo posible de este módulo en alcanzar una salida válida, pues este valor influirá en el diseño.

3.7. Multiplexor de salida

Los datos de salida deben seguir la secuencia $X_1Y_1 X_2Y_2 X_3Y_3... X_NY_N$, donde N es el número de subpupilas. Los datos de las coordenadas x e y utilizan el mismo puerto, luego debe existir un multiplexor que seleccione el dato a presentar a la salida. La señal de control de este multiplexor es una salida del módulo “controlador”.

3.8. Registros intermedios

Son registros implementados mediante flip-flop tipo D sin reset, activados mediante un flanco de subida en su señal de carga.

Registro 1. Almacenamiento de los datos de salida del “correlador”.

Registro 2. Almacenamiento de los datos de salida del módulo de ajuste cuadrático.

Registro de referencia. Almacena el centroide de la subpupila de referencia.

Registro de salida. Este registro almacena los datos finales a la entrada del multiplexor de salida.

4. Resultados y conclusiones

La síntesis del diseño se ha hecho empleando el programa Xilinx ISE 8.2i, sobre una FPGA Virtex-4 XC4VSX35-10C. En la tabla 1 se desglosa el uso de recursos, incluyendo el tiempo de procesado hasta obtener el desplazamiento de los centroides de todas las subpupilas para imágenes de entrada con dimensiones ascendentes en potencias de 2, desde 128x128 hasta 512x512, aumentando proporcionalmente el tamaño de las subpupilas. Nótese que para tamaños superiores a 256x256 píxeles de imagen de entrada, el tiempo necesario sobrepasa los 10ms de estabilidad de la turbulencia en el visible, haciendo ineficaz la corrección en lazo cerrado.

Recursos	128x128	256x256	512x512	Disponibles
	Subp:8x8 Ker:4x4	Subp:16x16 Ker:8x8	Subp:32x32 Ker:16x16	
Slices	4447	5431	9932	15360
Slice flip flop	1542	2096	5981	30720
LUTs	8250	10161	18607	30720
IOBs	29	29	29	448
FIFO16/RAMB16	7	22	73	192
GCLKs	16	16	16	32
DSP48s	4	5	6	192
PMCDs	1	1	1	4
Tiempo(100 MHz)	664.2 μ s	2.89 ms	18.1 ms	

Tabla 1. Uso de recursos y tiempos en una FPGA Virtex-4 XC4VSX35-10C

Agradecimientos

Este trabajo ha sido realizado en el marco del Programa Nacional de Diseño y Producción Industrial (Project DPI 2006-07906) del Ministerio de Educación y Ciencia y por el “European Regional Development Fund” (ERDF).

Referencias

- [1] L. F. Rodríguez Ramos. *Especificaciones del módulo de correlación directa para FPGA*. Instituto de astrofísica de Canarias (2006).
- [2] Gordon E. Moore. *Cramming More Components Onto Integrated Circuits*. *Electronics Volume 38, N° 8* (1965)
- [3] R. H. Hudin. *J. Opt. Soc. Am.* N° 67, 375 (1977).
- [4] D. I. Fried. *J. Opt. Soc. Am.* N° 67, 370 (1977).
- [5] S. Thomas, T. Fusco, A. Tokovinin, M. Nicolle, V. Michau y G. Rousset. *Comparison of centroid computation algorithms in a Shack-Hartmann sensor*. *Monthly Notices of the Royal Astronomical Society* (2006).
- [6] J. P. Deschamps, G. J. A. Bioul, G. D. Sutter. *Synthesis of arithmetic circuits*. Ed. Wiley-Interscience (2006).