Framework adaptable y reconfigurable dinámicamente para procesamiento de vídeo: aplicación a la etapa de filtrado adaptativo en sistemas de compresión de vídeo H.264/AVC y SVC

T. Cervero[(1)], A. Otero[(2)], S. López[(1)], E. de la Torre[(2)], G. Callicó[(1)], T. Riesgo[(2)], R. Sarmiento[(1)]

tcervero@iuma.ulpgc.es, andresotero@upm.es, seblopez@iuma.ulpgc.es, eduardodelatorre@upm.es, gustavo@iuma.ulpgc.es, teresariesgo@upm.es, roberto@iuma.ulpgc.es.

[(1)] Instituto Universitario de Microelectrónica Aplicada, Universidad de Las Palmas de Gran Canaria.

[(2)] Centro de Electrónica Industrial, Universidad Politécnica de Madrid.

Resumen

Los sistemas basados en componentes hardware con niveles de paralelismo estático tienden a infrautilizar sus recursos lógicos, ya que se diseñan para soportar el peor escenario posible. Este hecho se acentúa cuando se trabaja con los nuevos estándares de compresión de vídeo, como son el H.264/AVC y el SVC. Estos necesitan soluciones flexibles, capaces de soportar distintos escenarios, y escalables a fin de maximizar la utilización de recursos en todo momento. Por ello, y como alternativa a las soluciones estáticas o multiprocesadoras, este artículo presenta una arquitectura hardware escalable y reconfigurable dinámicamente para el filtrado de bucle adaptativo o *Deblocking Filter*. Su funcionamiento se basa en el de los *arrays* sistólicos, y su estrategia de paralelismo maximiza el número de macrobloques que pueden ser procesos simultáneamente.

1. Introducción

Los estándares de compresión de vídeo más avanzados, como son el estándar H.264/AVC [1] y el estándar SVC [2], introducen notables mejoras en los procesos de codificación y decodificación respecto a sus predecesores. Sin embargo, estas mejoras se consiguen a costa de introducir mayor complejidad en las operaciones de compresión, lo cual dificulta la implementación de estos estándares en dispositivos puramente software. Como alternativa, las soluciones hardware-software son más adecuadas. En ellas, el hardware (HW) ejecuta las tareas con mayor coste computacional, mientras que el software (SW) se dedica al resto.

De los distintos bloques funcionales que forman parte de los estándares H.264/AVC y SVC, el *Deblocking Filter* (DF) destaca por su alto coste computacional, tal y como se recoge en [3]. Además, se utiliza indistintamente en el bucle de codificación/decodificación y apenas varía entre el estándar H.264/AVC y el SVC; motivos por los cuales suele ser implementado en HW para alcanzar prestaciones de tiempo real. En este sentido, la tendencia para acelerar la ejecución del DF es diseñar arquitecturas con cierto nivel de paralelismo estático. El inconveniente de esta medida es la infrautilización de recursos, ya que la arquitectura ha sido diseñada para soportar el peor escenario posible [4-5]. Es por ello que día a día se hace más necesario desarrollar elementos escalables, capaces de adaptarse a distintos escenarios sin necesidad de ser rediseñados. Para ello deberían permitir variar sus recursos, incluyendo o eliminando elementos de procesamiento de acuerdo a las circunstancias.

La posibilidad de escalar una arquitectura en tiempo real puede llevarse a la práctica gracias a la capacidad de reconfiguración dinámica que ofrecen algunos dispositivos hardware, como son algunas FPGAs. Por otro lado, la aceleración de las operaciones del DF en arquitecturas estáticas es limitada, mientras que en arquitecturas reconfigurables es posible adaptar el nivel de paralelismo con mayor libertad.

Con el objetivo de crear una solución que aúne todas estas características, a lo largo de este artículo se presenta una arquitectura hardware para la etapa de DF escalable y reconfigurable dinámicamente, adaptada a los estándares de compresión de vídeo H.264/AVC y SVC. En la

sección 2 se explica el funcionamiento del DF como parte de los estándares mencionados. A continuación, la sección 3 describe la arquitectura propuesta; mientras que la sección 4 presenta sus características más destacables, así como algunos de los resultados obtenidos. Finalmente, la sección 5 resume las conclusiones principales de este trabajo.

2. Deblocking Filter

Toda secuencia de vídeo está formada por una sucesión de imágenes, cada una de ellas dividida en estructuras de datos más pequeñas denominadas macrobloques (MBs). Cada MB está formado por una matriz de luminancia de 16×16 píxeles, que contiene información de luminosidad y dos matrices con información del color, cada una de ellas de 8×8 píxeles. Una de ellas contiene la información de la crominancia azul (Cb), mientras que la otra de la crominancia roja (Cr). Cada MB, a su vez está estructurado en grupos de 4×4 píxeles, llamados bloques, numerados de 0 a 23 (los dieciséis primeros, de 0-15, se corresponden con la luminancia, de 16-19 con Cb, y los restantes con Cr).

Los estándares de vídeo H.264/AVC y SVC definen un proceso de filtrado en la etapa final del bucle de codificación/decodificación. Se trata de un filtrado unidimensional que debe aplicarse sobre cada uno de los bordes de bloque de cada MB. En concreto, este procedimiento consta de dos etapas. La primera consiste en procesar horizontalmente los bordes verticales (V0-V3) del MB, que en adelante se denominará filtrado horizontal, y otra que filtra verticalmente los bordes H0-H3 del MB, tal y como se muestra en la Figura 1.

El estándar determina que el filtrado horizontal ha de ser anterior al vertical, pero permite total libertad a la hora de establecer el orden de filtrado de los distintos bloques y bordes, siempre y cuando se respeten las dependencias de datos.

En cuanto a la dependencia de datos, y tal como indica la Figura 1, todo MB necesita información de su vecino inmediatamente anterior y superior. Esto implica que ambos tienen que haber sido filtrados antes de operar con el MB actual. Esta relación de dependencia entre MBs limita en gran medida el paralelismo.

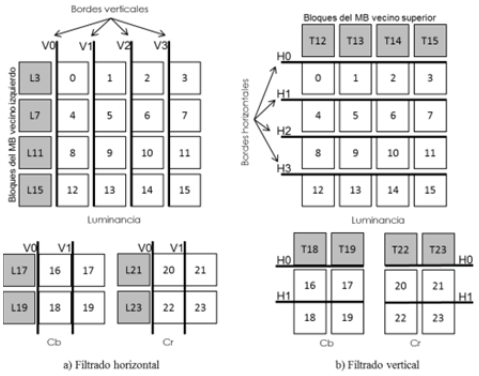

Figura 1. Bordes de MB. a) horizontales; b) verticales

El DF es un algoritmo adaptativo diseñado para reducir los efectos de bloque creados en los MBs, que se producen debido al procesado de las etapas previas dentro del proceso de codificación y decodificación. El término adaptativo se debe a que ajusta el tipo de filtrado de acuerdo a las características de los datos de entrada. Son las unidades del *Boundary Strength* (BS) y del filtro de píxeles las encargadas de calcular estos valores de ajuste. La primera de las unidades devuelve un valor comprendido entre 0 y 4. El 0 implica que no se aplica filtrado, mientras que el 4 implica que se han de aplicar las condiciones de filtrado más fuertes. Estos valores dependen del tipo de MB que se vaya a filtrar, del valor de cuantificación, de los residuos y de los vectores de movimiento. A continuación, la unidad de filtrado es quien finalmente determina si los datos de entrada deben ser filtrados, en cuyo caso se atenderá al valor del BS; o por el contrario, deben devolverse sin ser modificados. Esta decisión depende de la relación entre los píxeles del bloque actual y de su vecino, de acuerdo con unos umbrales ya tabulados.

3. Arquitectura propuesta

La arquitectura hardware propuesta es escalable porque permite modificar la cantidad de elementos que la conforman. La escalabilidad introduce ciertas ventajas, como son: mayor flexibilidad, ya que consigue optimizar la utilización de los recursos; y mayores niveles de paralelismo a la hora de procesar los MBs.

En lo que a la flexibilidad se refiere, la arquitectura propuesta explota las capacidades de algunas FPGAs modernas que permiten reconfigurarse parcial y dinámicamente. Esta característica hace posible que la arquitectura se adapte en tiempo real a las necesidades impuestas desde el exterior.

En lo que se refiere al paralelismo, esta arquitectura procesa los datos siguiendo la política de un *array* sistólico. Es decir, cada vez que se inicia el filtrado horizontal, nuevos elementos de procesamiento empiezan a filtrar MBs siguiendo el patrón de un frente de ondas, tal y como muestra la Figura 2.

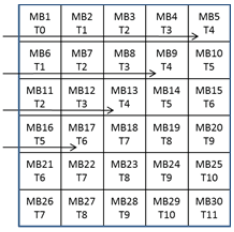

Figura 2. Paralelismo a nivel de MB como un frente de ondas.

El diseño de esta arquitectura escalable se ha realizado en base a una estructura modular de elementos homogéneos, en la que se diferencian claramente dos regiones según su capacidad de adaptación. La primera se trata de la región estática, que siempre está presente en el diseño, y permanece inalterada sea cual sea la configuración que se decida implementar. Por el contrario, en la segunda, la región reconfigurable, la cantidad y distribución de elementos presentes puede variar.

3.1. Descripción básica de elementos

Los elementos que componen la arquitectura se pueden clasificar según la región que ocupan, y conforme a la función principal que desempeñan en ella.

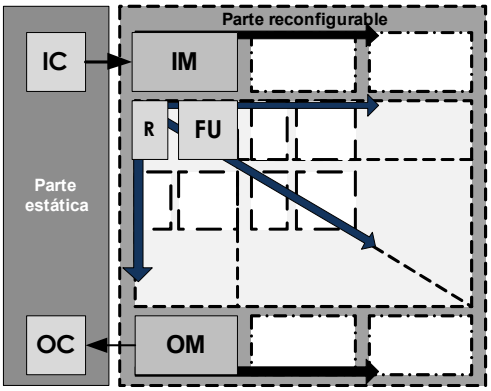

Figura 3. Esquemático de la arquitectura propuesta

3.2. Región estática

Está formada por dos elementos que operan como interfaz entre la región dinámica y el resto del decodificador. Su función principal es la de leer los MBs sin filtrar de la memoria principal y, una vez procesados, almacenarlos nuevamente en ella. Los dos elementos son el control de entrada (IC – *Input Controller*) y el control de salida (OC – *Output Controller*). El primero de ellos lee MBs de la memoria principal, adapta su formato a las necesidades de la arquitectura, y los envía a la región reconfigurable para que sean procesados. La lectura de memoria se realiza MB a MB, pero no necesariamente en orden, ya que eso depende directamente de la configuración seleccionada. Con el fin de cumplir con esta característica del diseño, el IC incluye un mecanismo de cálculo que genera la secuencia válida de lectura de los MBs de memoria principal. Por su parte, el OC recoge los MBs filtrados de la región reconfigurable, y los envía a la memoria de reconstrucción en orden.

3.3. Región reconfigurable

Está formada por cuatro tipos de elementos (IM, OM, R y FU), que pueden variar en número y distribución según se indica en la Figura 3.

Esta región se puede estructurar en matrices de $m \times n$ elementos de procesamiento, donde m representa el número de columnas, y n el número de filas de cada matriz. De todos los elementos que componen la región reconfigurable,

únicamente las unidades funcionales (FU – *Functional Units*) procesan MBs. Los demás tienen como función principal distribuirlos.

Cada FU se comporta como un DF por sí misma, pudiendo también asociarse en estructuras más complejas para procesar más rápidamente una imagen. Esta asociación de FUs es posible gracias a su diseño homogéneo, que permite comunicar fácilmente una FU con otra, y con los demás elementos de la región reconfigurable. Sin embargo, además de su función principal de filtrado de MBs, toda FU es responsable de mantener la sincronización con el resto del sistema, así como de enviar y recibir gran cantidad de datos, entre los que se encuentran los MBs semifiltrados y los totalmente filtrados.

En cuanto a los elementos de distribución, existen tres tipos:

- **Memoria de entrada** (IM – *Input Memory*): Se trata del primer elemento presente en cada columna de la matriz de procesamiento. Se basa en una FIFO que almacena los MBs sin filtrar que llegan desde el IC, y que posteriormente alimentarán a las FUs. La carga de datos se inicia repartiendo los MBs entre los distintos IMs. Cada IM almacena sólo los MBs que le pertenecen, dejando pasar los demás hacia su IM vecino. Cuando todas las FIFOs están completas, cada IM transmite los MBs a las FUs de su propia columna. En cuanto a la capacidad de reconfiguración, el IM sólo puede crecer en una dimensión, tal y como muestra la Figura 3.
- **Memoria de salida** (OM – *Output Memory*): Este elemento es similar al IM, salvo porque en este caso su función es la de recoger los datos filtrados provenientes de su columna de FUs, y enviarlos hacia su vecino izquierdo, hasta llegar al OC.
- **Distribuidor** (R – *Router*): Este elemento es la interfaz de comunicación entre el resto de la región reconfigurable y la FU, de manera que la pareja distribuidor-FU podría entenderse como un único bloque más complejo. En la etapa previa al procesamiento, cada distribuidor recoge el primer MB que le llega y lo almacena en su FU, mientras que los MBs restantes los deja pasar hacia el vecino inferior. Cuando cada FU ha terminado de procesar un MB, es este elemento el que lo envía a la matriz para que llegue a su OM correspondiente.

Todos los elementos de la arquitectura incluyen lógica de control distribuida para gestionar la transmisión de datos y mantener el sistema sincronizado. Esta estrategia permite gestionar con mayor facilidad la escalabilidad, ya que simplifica la transición de una configuración a otra sin necesidad de rediseñar ni reprogramar el control. Esta ventaja también se debe en gran medida a la homogeneidad de todos los elementos, así como de las comunicaciones entre ellos.

4. Caracterización y resultados

Atendiendo a todo lo expuesto hasta el momento, es posible clasificar esta propuesta para implementar el DF como una arquitectura hardware de grano grueso, escalable, reconfigurable dinámicamente y con un alto grado de paralelismo. Se dice que es de grano grueso a nivel de datos puesto que trabaja con el mayor formato de datos permitido, a nivel de MB. La escalabilidad hace posible incluir o eliminar elementos fácilmente de la región reconfigurable según lo requiera el sistema, en gran medida gracias a la homogeneidad de los elementos implicados. Mientras que la capacidad de reconfiguración dinámica hace posible que estos cambios de configuración se realicen en tiempo real, sin necesidad de detener la ejecución del resto del decodificador. En referencia al paralelismo, la política implementada se aprovecha de las ventajas que ofrecen las estructuras basadas en *arrays* sistólicos; además de utilizar una estrategia de transferencia de datos semifiltrados entre las FUs que permite acceder a los MBs vecinos sin tener que de acceder a memoria principal.

4.1. Resultados de síntesis

La arquitectura ha sido descrita a nivel RTL, haciendo uso de lenguajes de descripción hardware. Con el fin de explotar al máximo las capacidades de reconfiguración dinámica parcial, se ha elegido utilizar como dispositivo hardware una FPGA de Xilinx de tamaño mediano de la Virtex-5: la LX110T. En la Tabla I se recogen los resultados de síntesis de la configuración más sencilla que se puede implementar con esta arquitectura. Se trata de una configuración 1×1, compuesta por un elemento de cada tipo.

Los resultados (IC&OC) hacen referencia a los recursos conjuntos de los elementos IC y OC, puesto que ambos forman la región estática. La columna *Router&FU* asocia los resultados de ambos elementos, considerándolos como una única entidad con el fin de facilitar la representación y la futura adaptación a la reconfiguración dinámica.

Recursos Lógicos	Elementos de la arquitectura			
	IC&OC	IM	OM	Router&FU
Slices reg.	473	172	124	2004
Slices LUTs	552	134	226	2386
BRAMs	1	2	2	8
Frecuencia (MHz)	236	400	286	124

Tabla I. Recursos de una configuración 1×1

De los resultados de síntesis, cabe mencionar la notable diferencia entre los recursos necesarios para implementar la pareja Router-FU y el resto de elementos. De hecho, esta pareja supone más del 80% del total de los recursos utilizados por la arquitectura.

En cuanto a la velocidad de procesamiento conjunto, ésta viene determinada por el elemento más lento. Es por ello que la frecuencia máxima de operación se corresponde con el valor de la FU, que alcanza un máximo de 124 MHz.

4.2. Reconfiguración dinámica parcial

Hasta el momento los resultados presentados se han obtenido sintetizando configuraciones de la arquitectura de manera estática. Es decir, es necesario detener la ejecución y sintetizar la nueva configuración para poder operar con ella. Sin embargo, el objetivo final es aprovechar las ventajas que la reconfiguración dinámica ofrece. Por ello, este apartado se centra en cómo adecuar la arquitectura para que esto sea posible. En este sentido, la modularidad de esta arquitectura facilita la adaptación de todos los elementos a los flujos de diseño específicos para explotar la reconfiguración dinámica parcial.

Siguiendo las estrategias clásicas seguidas para la adaptación a la reconfiguración dinámica, es necesario introducir la utilización de bus macros (BMs) en el diseño. Estos son puntos de comunicación unidireccionales que se insertarán en cada elemento, con el objetivo de permitir la transferencia de datos, o señalización de control a través de puntos fijos dentro de la FPGA.

Una vez que todos los elementos hayan sido adaptados, cada uno de ellos se tratará como un elemento reconfigurable independiente a fin de obtener su bitstream de configuración. Considerando que las distintas configuraciones $m \times n$ se basan en réplicas de los distintos elementos, bastará con generar una única versión de cada uno de ellos para su posterior reubicación. De este modo, tal y como representa la Figura 4, todo IM incluirá una serie de BMs asociados a cada una de las entradas y salidas de datos. Las líneas de la izquierda de la imagen se corresponden con los BMs que permiten la recepción de datos desde su vecino izquierdo. Las líneas de la parte derecha de la imagen se corresponden con los BMs que permiten transmitir datos a su IM vecino derecho, mientras que los BMs inferiores transfieren datos a la columna de FUs.

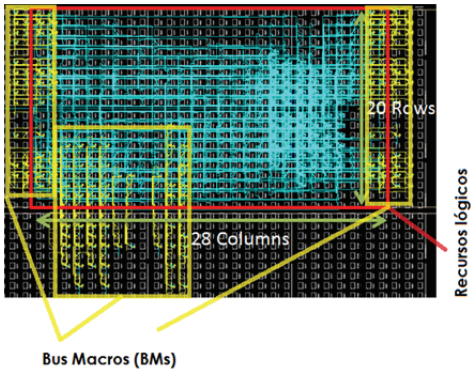

Figura 4. Floorplanning de una IM

Con el fin de simplificar la adaptación, y dado que todo elemento distribuidor está siempre asociado a una FU, ambos elementos se han considerado como una única entidad reconfigurable. Los BMs se han asignado a las distintas señales de entrada salida, algunas de ellas dedicadas a la transferencia de MBs semifiltrados entre las FUs vecinas, y otras al control y transferencia de MBs entre los routers vecinos.

Siguiendo este procedimiento con todos y cada uno de los elementos, es posible configurar cualquier tamaño de la arquitectura, siempre y cuando los recursos de la FPGA utilizada lo permitan. A modo de ejemplo, la Figura 5 representa el floorplanning de una configuración 1×2, en la que se ha omitido la región estática. En ella se observan el IM de entrada, dos FUs con sus

routers correspondientes, y finalmente el OM. Las líneas ubicadas a la izquierda de la imagen se corresponden con los BMs que deberían comunicarse con los elementos de control de la región estática.

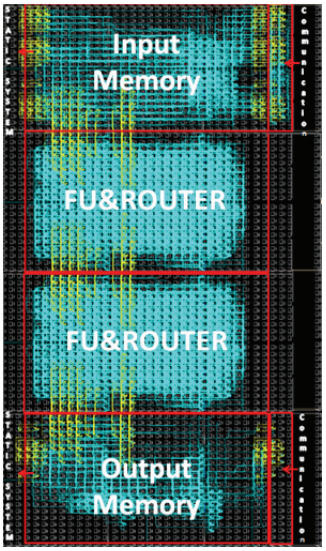

Figura 5. Floorplanning de una configuración 1×2

A pesar de las ventajas que aporta un sistema dinámicamente reconfigurable, la adaptación supone un incremento de recursos hardware debido a la inclusión de los puntos fijos de comunicación (BMs), tal y como indican los valores de la Tabla III. En esta tabla se recogen los resultados de síntesis de la región reconfigurable para una configuración 1×1, incluyendo los recursos lógicos del diseño y los BMs de cada módulo. En este caso, los resultados de IC y OC no se muestran porque coinciden con los valores recogidos en la Tabla I, ya que los resultados de la región estática no se ven afectados por la inclusión de BMs.

Recursos lógicos	**Elementos de la arquitectura**		
	IM	**Router&FU**	**OM**
Slices regs	2080	4160	2080
Slices LUTs.	2080	4160	2080
BRAMs	4	8	4

Tabla III. Impacto en los recursos tras la adaptación a la escalabilidad dinámica

La adaptación supone un incremento de más del 900% de los recursos para los IMs y OMs. Mientras que este porcentaje se reduce a un 180% en el caso de la pareja Router-FU.

5. Conclusiones

La alta carga computacional del DF, junto con las altas prestaciones asociadas a los decodificadores de vídeo más avanzados, hace que las soluciones hardware sean las más utilizadas para implementar este algoritmo. Sin embargo, no basta con implementar la funcionalidad del DF, sino que también debe utilizar el menor número de recursos posibles, y ser capaz de hacer frente a las variaciones del entorno. Tratando de abordar todas estas cuestiones, el presente artículo describe una arquitectura hardware para el DF de grano grueso, escalable y reconfigurable dinámicamente que cumple con los estándares de compresión de vídeo H.264/AVC y SVC. Entre otras características, esta arquitectura es capaz de adaptar su capacidad de procesamiento en tiempo-real, incluyendo o eliminando FUs.

Referencias

[1] M. Horowitz, A. Joch, F. Kossentini and A. Hallapuro; "H.264/AVC baseline profile decoder complexity analysis". IEEE Trans. On Circuit Systems for Video Tech., 13(7), 704-716. (2003)

[2] M. Wien, H. Schwarz and T. Oelbaum, "Performance Analysis of SVC", ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-U141, (2006).

[3] I. Werda,, T. Grandpierre, M. Ayed and N. Masmoudi. "Real-time H.264/AVC baseline decoder implementation on TMS320C6416". Journal of Real-Time Image Processing Springer Berlin, 1-18, (2010).

[4] F. Tobajas, G.M. Callicó, P. A. Pérez, V. de Armas and R. Sarmiento, "An efficient double-filter hardware architecture for H.264/AVC Deblocking filtering", IEEE Trans. on Consumer Electronics, 54(1), (2008).

[5] M. Torabi, A. Vafaee and N. Movahhedinia, "A fast architecture for Deblocking filter in H.264/AVC using clock cycles saving process", IMPACT, (2009).