

# Evaluación de módulo de estimación de movimiento basado en FPGA para el codificador de vídeo HEVC

Otoniel López-Granado, Roberto Gutierrez, Estefanía Alcocer, Hector Migallón y Manuel .P. Malumbres<sup>1</sup>

*Resumen*— La estimación de movimiento es una de las tareas más costosas del codificador de vídeo HEVC debido principalmente a, a) un mayor número de modos de particionado del Coding Tree Unit, b) la presencia de múltiples frames de referencia y c) el tamaño variable de los Coding Units en comparación con su predecesor H264/AVC. Además, HEVC utiliza una estimación de movimiento de tamaño de bloque variable para aumentar la eficiencia en la codificación. En este trabajo se ha diseñado y evaluado sobre una plataforma System-on-Chip específica, un módulo de estimación de movimiento hardware, teniendo en cuenta tanto el rendimiento en compresión cuando se aplican diferentes tamaños de Coding Tree Units, como el impacto de las transferencias del DMA en el tiempo total de codificación.

Los resultados muestran que el tiempo global de codificación se podría reducir un 77% usando nuestro módulo hardware, siendo este tipo de aceleradores, una solución de bajo coste interesante para acelerar los codificadores de vídeo de última generación.

*Palabras clave*— Codificación de vídeo, HEVC, FPGA, Estimación de movimiento, Particionado asimétrico.

## I. INTRODUCCIÓN

EL estándar de codificación de vídeo HEVC (High Efficiency Video Coding)[1] desarrollado por el Joint Collaborative Team on Video Coding (JCT-VC) en 2013, surgió para reemplazar a su predecesor H.264/AVC [2] con la intención de lidiar con las últimas tendencias del mercado multimedia como son los contenidos digitales de muy alta resolución (4K y 8K) así como con la mejora en la profundidad de color (10 bits). HEVC es capaz de comprimir el doble que su predecesor H.264/AVC para una calidad visual equivalente[3].

En lo referente a complejidad computacional, el codec HEVC es mucho más complejo que el codec H.264/AVC[4]. Al igual que en los anteriores estándares de codificación de vídeo, la Estimación de Movimiento (ME) es la parte más compleja computacionalmente hablando del codificador, requiriendo más del 90% del tiempo de codificación[5]. En HEVC, la ME es mucho más compleja, debido a las nuevas características incluidas en este estándar como son el uso de un mayor número de modos de particionado del Coding Tree Unit o la presencia de múltiples frames de referencia, en comparación con el anterior estándar de vídeo H264/AVC. Además, HEVC adopta un nuevo esquema de estimación de movimiento de tamaño de bloque variable (Variable

Block Size Motion Estimation (VBSME)) para mejorar la eficiencia en la codificación, a expensas de un incremento en el coste computacional.

En la literatura podemos encontrar varias propuestas de arquitecturas hardware para acelerar el módulo de ME de HEVC. El bloque Integer-pel Motion Estimation (IME) es el encargado de la estimación de movimiento y, la mayoría de las propuestas del estado del arte se centran en el algoritmo de búsqueda de movimiento pues requiere la mayor parte del tiempo del bloque IME. Habitualmente el algoritmo de búsqueda de movimiento más usado en las implementaciones hardware es la búsqueda completa o Full Search (FS). Este algoritmo busca en todos los puntos del área establecida de un frame de referencia y en consecuencia, obtiene el resultado óptimo, es decir, obtiene el vector de movimiento que minimiza el error residual del CTU buscado.

Diversos autores como en [5], [6], [7], [8], [9] han desarrollado bloques IME hardware usando el algoritmo FS. En [5], se propone una unidad para el cálculo del SAD (Sum of Absolute Differences) implementada en una FPGA (Field-Programmable Gate Array) capaz de evaluar todos los posibles modos de particionado de un CTU excepto los modos asimétricos. Los autores establecen un área de búsqueda menor que la establecida en el estándar HEVC obteniendo una velocidad de codificación de 30 fps para vídeos de resolución 2k. En el módulo hardware presentado en [7], el tamaño máximo de CTU se reduce a 32x32 con un área de búsqueda de  $\pm 23$  pixels. Esta arquitectura alcanza los 30 fps para resoluciones de 1080p. En [8], se estudia el efecto de diferentes áreas de búsqueda, alcanzando un frame rate máximo de 57 fps para resoluciones de 720p.

En un trabajo previo [10], los autores presentaron una nueva arquitectura hardware que realiza el cómputo de IME usando tecnología FPGA. Esta nueva arquitectura se basa en dos técnicas innovadoras: La primera es una nueva estructura de árbol de sumadores para el cálculo del SAD y la segunda es una nueva forma de recorrer la memoria. Esta propuesta es capaz de codificar 116 fps y 30 fps para resoluciones 2K y 4K respectivamente. El nuevo árbol de sumadores de SAD realiza las sumas en el primer nivel del CTU quad tree, comenzando con el mayor tamaño de CTU y dividiendo la cantidad de sumas a realizar en el siguiente nivel. Esta aproximación es muy diferente al resto de propuestas encontradas en la literatura donde normalmente primero se divide el

<sup>1</sup>Dpto. de Ingeniería de Computadores, Univ. Miguel Hernández

CTU en los tamaños más pequeños para ir acumulando sucesivamente requiriendo un mayor número de pasos para obtener todos los SADs. Con esta nueva propuesta se maximiza el uso de los recursos de la FPGA obteniendo la mínima latencia posible a la hora de calcular los SADs de todas las particiones del CTU. De esta forma, los SADs que corresponden a las particiones asimétricas se obtienen de manera rápida y eficiente. Con respecto a la segunda técnica innovadora, la nueva forma de recorrer la memoria, se usan una serie de registros de desplazamiento reconfigurables y elementos de procesamiento para el almacenamiento de los pixels necesarios tanto de la referencia como del CTU actual, de manera que se mantienen siempre disponibles para calcular tanto los SADs como los vectores de movimiento (MV) de un CTU. Con esta estrategia, los autores evitan accesos a memorias externas ya que la información disponible se reutiliza reconfigurando los desplazamientos de manera eficiente.

En este trabajo, se ha implementado y evaluado el diseño del módulo IME presentado en [10] aplicado a una placa de evaluación específica. El diseño se ha variado ligeramente para obtener tres posibles implementaciones: a) una unidad que trabaja con el tamaño máximo de CTU de 32x32 pixels (CTU\_32), b) cuatro unidades CTU\_32 trabajando en paralelo y, c) una unidad que trabaja con el tamaño máximo de CTU a 64x64 pixels (CTU\_64). Todos ellos usan el área de búsqueda propuesta por el estándar de vídeo HEVC. Hemos evaluado las tres versiones analizando los recursos hardware requeridos, la frecuencia de operación, el retardo total de los módulos hardware (Transferencias DMA de entrada, cómputo de la estimación de movimiento y transferencias de DMA de salida), y el rendimiento en R/D (Rate/Distortion) de las propuestas hardware con respecto al software de referencia HEVC.

El resto del artículo se organiza de la siguiente manera. La Sección II muestra una breve descripción del diseño de la arquitectura propuesta mientras que en la Sección III, se presentan los resultados de los experimentos y se analizan los resultados de nuestro diseño hardware sobre la placa de evaluación. Finalmente, en la Sección IV se presentan las conclusiones y trabajo futuro.

## II. DESCRIPCIÓN DE LA ARQUITECTURA HARDWARE

En esta sección presentamos una breve descripción del diseño IME sobre una plataforma System-On-Chip (SoC) que consiste en dos partes bien diferenciadas, un Processing System (PS) basado en un procesador ARM y algunos periféricos como Ethernet, USB, etc. y, un FPGA Programmable Logic (PL). Nuestra arquitectura ha sido modelada en VHDL y sintetizada, simulada, implementada y evaluada sobre la placa Xilinx SoC, Zynq-7 Mini-ITX Motherboard XC7Z100 (xc7z100ffg900-2). Para asegurar el correcto funcionamiento de nuestro diseño, éste se ha contrastado con el software de referencia del codifi-

cador HEVC HM 14[11].

En la arquitectura propuesta, el procesador ARM se encarga de las transferencias entre el módulo IME SAD y la memoria Double Data Rate (DDR) que almacena tanto la ventana de referencia como el CTU actual, a través de un módulo Direct Memory Access (DMA). El procesador ARM trabaja a 666.66 MHz, y la DDR a 533.33 MHz, mientras que la frecuencia de reloj del PL se ve restringido a la frecuencia máxima del módulo SAD HEVC responsable del cálculo de IME.

En lo que concierne al proceso IME, cada frame del vídeo se subdivide en unidades básicas de codificación llamadas CTUs. La estructura de codificación de HEVC consta de CUs (Coding Units) de tamaño máximo 64x64 pixels, el mismo tamaño de los CTUs (Coding Tree Units), que pueden dividirse de forma recursiva en regiones cuadradas hasta alcanzar el tamaño de 8x8 pixels. Cada coding unit (CU) consiste en Prediction Units (PUs) cuyo tamaño puede variar desde el máximo tamaño del CU hasta 4x8 o 8x4 para la predicción Inter, pudiendo dividirse en 8 modos de particionado[12]. En nuestra propuesta, el módulo SAD HEVC encargado del cálculo IME puede configurarse para trabajar con tamaños de CTU de 64x64 y 32x32. Para el caso de tamaño de CTU 64x64, el PL puede trabajar a 200 MHz mientras que con el tamaño de CTU 32x32 la frecuencia de reloj del PL queda restringida por la placa de evaluación a 250 MHz, aunque nuestro módulo podría trabajar a 333 MHz. Por lo tanto, la frecuencia máxima queda limitada a 200MHz para poder evaluar ambos tamaños de CTU.

Nuestro módulo SAD HEVC consiste en a) áreas de memoria interna para alojar los pixels del CU actual y los pixels correspondientes al área de búsqueda del frame de referencia, b) un bloque de distorsión donde los pixels de ambos CUs se restan, c) un bloque que corresponde al árbol de sumadores de SAD (Sum of Absolute Difference), y d) un bloque acumulador comparador que guarda el menor valor de SAD y su vector de movimiento (MV) correspondiente para todas las posibles particiones del CU, tal y como se muestra en la Figura 1. Para un mayor detalle del módulo véase [10].

En la Tabla I, se muestran los recursos hardware utilizados para implementar nuestro módulo SAD HEVC para tamaños máximo de CTU de 64x64 y 32x32, respectivamente, sobre una placa Zynq-7 Mini-ITX Motherboard XC7Z100 FPGA. Como se puede apreciar, nuestro módulo SAD HEVC necesita el 56.5% y el 15.3% del área total para las implementaciones CTU\_64 y CTU\_32, respectivamente. Además, la implementación 4xCTU\_32 capaz de realizar el cómputo de 4 CTUs de 32x32 al mismo tiempo, utiliza un 59.7% del área disponible.

## III. EXPERIMENTOS

La estimación de movimiento es una tarea integrada en el módulo de predicción Inter del codificador HEVC. Para los experimentos se ha utilizado

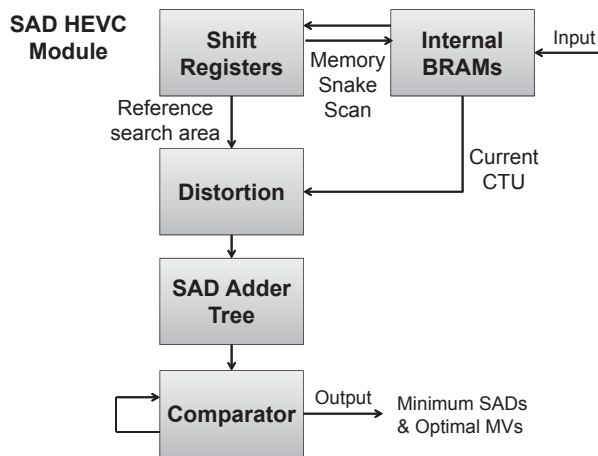


Fig. 1. Módulo hardware SAD HEVC

TABLA I  
UTILIZACIÓN DE RECURSOS SOBRE MINI-ITX

|                    | LUTs          | Flip-flops    | Block-RAMs |
|--------------------|---------------|---------------|------------|
| <b>CTU_64</b>      | 156880        | 180249        | 41,5       |
| <b>CTU_32</b>      | 42405         | 50466         | 25,5       |
| <b>4xCTU_32</b>    | 165700        | 190240        | 102        |
| <b>Disponibile</b> | <b>277400</b> | <b>554800</b> | <b>755</b> |

el modo de codificación Low Delay P (LD-P). En el modo LD-P, el primer frame se codifica como un cuadro Intra, mientras que el resto de frames se codifican como frames Inter unidireccionales. Este tipo de codificación es la más utilizada para sistemas de vídeo conferencia, y está diseñado para una comunicación en tiempo real.

Usando el modo de codificación anterior, hemos realizado varios experimentos de nuestro módulo HEVC IME para ver cómo impacta el tamaño del CTU en la eficiencia de la codificación (R/D), así como en el tiempo de codificación del codificador HEVC. Se han elegido los tamaños máximos de CTU de 64x64 y 32x32, y sus correspondientes áreas de búsqueda (SR) al 100% del tamaño del CTU. Se han utilizado cuatro secuencias de vídeo elegidas de entre las sugeridas en las condiciones comunes de test para HEVC: RaceHorses (832x480-30 fps), ParkScene (1920x1080-24 fps), Traffic (2560x1600-30 fps) y PeopleOnStreet (2560x1600-30 fps). Para realizar estos test hemos usado el software de referencia HEVC HM 14 [11]. Este software de referencia se ha compilado utilizando Visual Studio 2015 y se ha ejecutado sobre una plataforma Intel Core i7-6800K CPU 3.40GHz con 16GB RAM.

Primeramente se ha analizado el impacto del tamaño máximo de CTU en términos de R/D usando la métrica Bjontegaard (BD-rate) [13]. Para obtener el valor de BD-rate se han comprimido todas las secuencias de vídeo a cuatro niveles de compresión (QP): 22, 27, 32, y 37. La curva R/D de referencia es la obtenida con el software de referencia usando un CTU de tamaño 64x64. Como podemos ver en la Tabla III se obtiene mejor calidad cuando usamos el CTU de tamaño 64x64. El uso de un CTU de

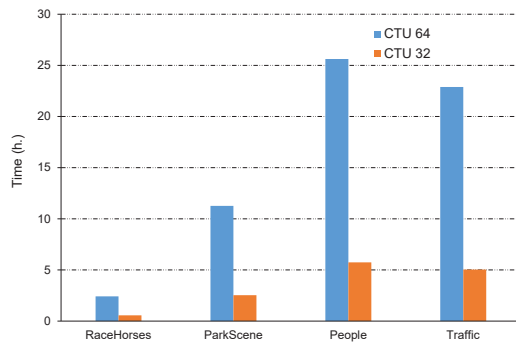


Fig. 2. Tiempo de codificación (30 frames) para CTU de 64x64 y 32x32 para todas las secuencias de vídeo.

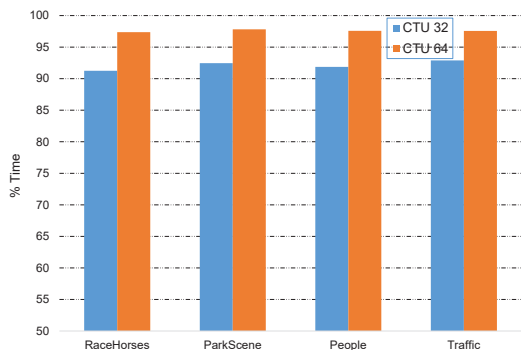


Fig. 3. Porcentaje del tiempo total del SW de referencia requerido por el módulo SAD en el algoritmo Full-Search.

menor tamaño tiene una penalización máxima de un 4.3% en términos de BD-rate, siendo esta penalización mayor en las secuencias de vídeo de alta resolución.

Después de la evaluación del R/D, evaluamos el impacto del tamaño de CTU en el tiempo de codificación. En la Figura 2 se muestra el tiempo total de codificación para comprimir 30 frames (1 segundo) de todas las secuencias de vídeo evaluadas usando tamaños de CTU de 64x64 y 32x32 y para a un QP 37. Como se puede ver, el tiempo de codificación para el tamaño máximo de CTU de 64x64 es más de 4 veces mayor que el necesario para el caso de CTU de tamaño 32x32, necesitando más de 22 horas para comprimir una secuencia de resolución 4k. También, en la Figura 3, se muestra el porcentaje de tiempo total requerido por el algoritmo Full Search para realizar ME deshabilitando la optimización del R/D. Dependiendo del tamaño máximo de CTU y del parámetro de cuantización (QP), el porcentaje de tiempo utilizado por el proceso de ME va desde el 85% hasta el 97.8% en nuestros test. Estos resultados concuerdan los los obtenidos en [5]. Como era de esperar, el codificador requiere más tiempo en el módulo IME cuando el tamaño de CTU es mayor. Por lo tanto, tiene sentido la implementación de un diseño hardware para el cómputo del IME con el fin de reducir el tiempo total de codificación lo máximo posible.

Con el fin de evaluar el impacto de la inclusión de nuestro IME hardware en el HEVC, hemos medido el número de ciclos requeridos para a) la transferen-

TABLA II  
CICLOS NECESARIOS Y TIEMPO DEL MÓDULO HARDWARE IME

|        | DMA Envío | DMA Recepción | SADs y MVs Cómputo | Ciclos totales | Tiempo Total (ms) a 200MHz |
|--------|-----------|---------------|--------------------|----------------|----------------------------|
| CTU_32 | 10069     | 151           | 4139               | 14359          | 0.07179                    |
| CTU_64 | 44276     | 605           | 16400              | 61281          | 0.30640                    |

TABLA III  
COMPARACIÓN DE % BD-RATE ENTRE TAMAÑOS DE CTU DE 64X64 Y 32X32

| Secuencia de vídeo | % BD-Rate |
|--------------------|-----------|
| RaceHorses         | 2.4       |
| ParkScene          | 3.8       |
| PeopleOnStreet     | 3.7       |
| Traffic            | 4.3       |

cia tanto del CTU como de la ventana de referencia, b) el procesamiento del IME y c) la transferencia de los SADs y vectores de movimiento de todos los PUs. En la Tabla II se puede ver el número de ciclos requeridos para la realización del proceso IME para un determinado CTU, donde, como se ha comentado previamente, todos los SADs y MVs de todos los PUs incluidas las particiones asimétricas se calculan a la vez. De esta manera, el tiempo requerido para procesar un CTU será el tiempo para transferir el CTU y la ventana de búsqueda, el tiempo de procesamiento del módulo HEVC SAD, y el tiempo necesario para devolver los resultados. Como podemos ver, la mayor parte del tiempo necesario para un CTU está en las transferencias de DMA tanto del CTU como de la ventana de búsqueda desde la memoria DDR a la memoria interna del módulo hardware IME. En la Tabla II vemos el tiempo total requerido por nuestro módulo hardware IME para realizar la estimación de movimiento de un CTU. Considerando estos tiempos, el módulo hardware IME es capaz de procesar 13959 CTUs por segundo para el tamaño de 32x32 y 3264 para el tamaño de 64x64 a 200MHz.

TABLA IV  
FRAMES DE REFERENCIA USADOS PARA UN GOP

|         |              |
|---------|--------------|
| Frame 1 | -1 -5 -9 -13 |
| Frame 2 | -1 -2 -6 -10 |
| Frame 3 | -1 -3 -7 -11 |
| Frame 4 | -1 -4 -8 -12 |

TABLA V  
CÓMPUTO EN FRAMES POR SEGUNDO DEL MÓDULO HARDWARE IME A 200 MHz

| resolución vídeo | CTU_32 |      | 4xCTU_32 | CTU_64 |      |
|------------------|--------|------|----------|--------|------|
|                  | CTUs   | fps  | fps      | CTUs   | fps  |
| 832x480          | 390    | 8.93 | 35.71    | 104    | 7.85 |
| 1920x1080        | 2040   | 1.70 | 6.83     | 510    | 1.60 |
| 2560x1600        | 4000   | 0.87 | 3.48     | 1000   | 0.82 |

Como se ha comentado previamente, en el modo

LD-P, el primer frame se codifica como Intra-frame y el resto como frames tipo P. Los frames P realizan la estimación de movimiento usando frames previamente codificados y decodificados como frames de referencia. En HEVC, los frames tipo P pueden utilizar múltiples frames de referencia. En este artículo, un GOP (group of pictures) consiste en 4 frames P donde cada uno de ellos puede usar hasta 4 frames de referencia. En la Tabla IV se muestran las referencias usadas por cada frame P de un GOP. Al comienzo de la codificación de la secuencia de vídeo, el primer frame P tiene solo una referencia disponible, el frame I (-1). El segundo, tercero y cuarto frame tipo P usaran hasta 2 frames de referencia. Sin embargo, tras codificar varios GOPs, los frames P dentro de los subsiguientes GOPs usaran todas las posibles 4 referencias. Los tiempos mostrados en la Tabla II, son los requeridos para realizar la estimación de movimiento de un CTU sobre un único frame de referencia. Por lo tanto, cuando se usen varios frames de referencia, ese proceso de estimación de movimiento se repetirá para cada uno de los frames de referencia usados, reduciéndose hasta en 4 veces el número de CTUs por segundo que se pueden procesar si se usasen 4 frames de referencia.

Para poder compensar esta situación, hemos desarrollado un módulo SAD HEVC que contiene 4 módulos CTU\_32, cada uno con su propio canal de DMA, y que es capaz de realizar el procesamiento de 4 CTUs de 32x32 a la vez. El poder realizar la estimación de movimiento sobre los cuatro posibles frames de referencia al mismo tiempo reduciría el tiempo de codificación total un 77% de media. La tabla V muestra el número de frames por segundo (fps) que pueden procesarse para las cuatro resoluciones de vídeo evaluadas si se usase nuestro módulo IME. También se muestra en la Tabla V, el número de CTUs por frame a procesar dependiendo tanto de la resolución del vídeo como del tamaño máximo de CTU. Como se puede apreciar, usando el módulo 4xCTU\_32, podemos procesar 4 veces más fps, logrando una codificación de tiempo real para la menor resolución de vídeo.

#### IV. CONCLUSIONES

En este trabajo hemos presentado una arquitectura hardware IME. Hemos analizado como impacta el tamaño de CTU máximo en el rendimiento del codificador HEVC tanto en R/D como en tiempo de procesamiento. Como se ha mostrado hay pequeñas diferencias en R/D, siendo el máximo incremento en BD-rate un 4.3% cuando el tamaño máximo de CTU es 32x32 para los vídeos de muy alta resolución. En

cuanto a la complejidad, el tamaño del CTU tiene un gran impacto en el tiempo total de codificación, siendo mucho más rápido cuando usamos el tamaño de CTU de 32x32. Cabe indicar que el módulo de estimación de movimiento ocupa entre el 85% y el 97.8% del tiempo total de codificación, con lo que nuestro módulo hardware IME reducirá ese tiempo significativamente.

En cuanto a la evaluación de nuestro módulo hardware IME, hemos medido el tiempo requerido para las transferencias a través del DMA así como el tiempo necesario para los cálculos. Los resultados muestran que nuestro módulo puede procesar 13928 y 3263 CTUs por segundo para tamaños máximo de CTU 32x32 y 64x64, respectivamente trabajando a 200MHz. Se puede apreciar que el cuello de botella del sistema es el proceso de transferencia del DMA, necesitando más del 70% del tiempo total para el procesamiento de un CTU.

Además hemos presentado un módulo SAD HEVC para el tamaño de CTU 32x32 que contiene 4 módulos IME, cada uno con su propio canal de DMA. Este nuevo módulo es capaz de procesar 4 CTUs de 32x32 a la par. Con este nuevo módulo integrado en HEVC y para el modo de codificación LD-P podemos reducir el tiempo total de codificación un 77% por término medio, porque este módulo es capaz de realizar la estimación de movimiento de todos los 4 posibles frames de referencia al mismo tiempo.

Como trabajo futuro, queremos reducir los tiempos de transferencia del DMA, reutilizando parte del área de búsqueda para el cálculo del IME de CUs contiguos, transfiriendo únicamente los nuevos pixels de referencia necesarios en cada momento y por otra parte empaquetando en una palabra de 64 bits ocho pixels del área de referencia.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia, Innovación y Universidades con referencia RTI2018-098156-B-C54 cofinanciado con fondos FEDER (MINECO/FEDER/UE).

#### REFERENCIAS

- [1] B. Bross, W.J. Han, J.R. Ohm, G.J. Sullivan, Y-K Wang, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 10," *Document JCTVC-L1003 of JCT-VC*, Geneva, January 2013.
- [2] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," *ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC) version 16*, 2012, 2012.
- [3] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1648–1667, December 2012.
- [4] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1685–1696, 2012.
- [5] Ahmed Medhat, Ahmed Shalaby, Mohammed S Sayed, and Maha Elsabrouty, "A highly parallel sad architecture for motion estimation in hevc encoder," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS'14)*, Ishigaki, Nov. 2014, pp. 280–283.
- [6] J. Byun, Y. Jung, and J. Kim, "Design of integer motion estimator of hevc for asymmetric motion-partitioning

- mode and 4k-uhd," *Electronics Letters*, vol. 49, no. 18, pp. 1142–1143, 2013.
- [7] Xu Yuan, Liu Jinsong, Gong Liwei, Zhang Zhi, and Robert K.F. Teng, "A high performance vlsi architecture for integer motion estimation in hevc," in *IEEE 10th International Conference on ASIC (ASICON'13)*, Shenzhen, Oct. 2013, pp. 1–4.
- [8] Thomas D'huys, "Reconfigurable data flow engine for hevc motion estimation," in *IEEE International Conference on Image Processing (ICIP'14)*, Paris, Oct. 2014, pp. 1223–1227.
- [9] Antonio Navarro Purnachand Nalluri, Luis Nero Alves, "High speed sad architectures for variable block size motion estimation in hevc video coding," in *IEEE International Conference on Image Processing (ICIP'14)*, Paris, Oct. 2014, pp. 1233–1237.
- [10] Estefania Alcocer, Roberto Gutierrez, Otoniel Lopez-Granado, and Manuel P. Malumbres, "Design and implementation of an efficient hardware integer motion estimator for an hevc video encoder," *Journal of Real-Time Image Processing*, pp. 1–11, 2016.
- [11] HEVC software repository HM-14.0 reference model, <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-14.0>, .
- [12] I. Kim, J. Min, T. Lee, W. Han, and J. Park, "Block partitioning structure in the hevc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1697–1706, Dec 2012.
- [13] G. BjÅ,ntegaard, "Document vceg-m33: Calculation of average psnr differences between rd-curves," Tech. Rep., ITU-T VCEG Meeting, Austin, Texas, USA, Tech. Rep, 2001.