Turkish Journal of Electrical Engineering and Computer Sciences

Volume 28 | Number 3 Article 31

1-1-2020

An optimized FPGA design of inverse quantization and transform for HEVCdecoding blocks and validation in an SW/HW environment

AHMED BEN ATITALLAH

MANEL KAMMOUN

RABIE BEN ATITALLAH

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 1656 – 1672

doi:10.3906/elk-1910-122

An optimized FPGA design of inverse quantization and transform for HEVC decoding blocks and validation in an SW/HW environment

Ahmed BEN ATITALLAH[1,2,*], **Manel KAMMOUN**[2], **Rabie BEN ATITALLAH**[3]
[1]Department of Electrical Engineering, Jouf University, Aljouf, Saudi Arabia
[2]LETI (E.N.I.S.), University of Sfax, Sfax, Tunisia
[3]Department of Computer Science, Galatasaray University, Turkey

Received: 17.10.2019 • **Accepted/Published Online:** 28.01.2020 • **Final Version:** 08.05.2020

Abstract: This paper presents an optimized hardware architecture of the inverse quantization and the inverse transform (IQ/IT) for a high-efficiency video coding (HEVC) decoder. Our highly parallel and pipelined architecture was designed to support all HEVC Transform Unit (TU) sizes: $4 \times 4, 8 \times 8, 16 \times 16$, and 32×32. The IQ/IT was described in the VHSIC hardware description language and synthesized to Xilinx XC7Z020 field-programmable gate array (FPGA) and to TSMC 180 nm standard-cell library. The throughput of the hardware architecture reached in the worst case a processing rate of up to 1080 p at 33 fps at 146 MHz and 1080 p at 25 fps at 110 MHz when mapped to FPGA and standard-cells, respectively. The validation of our architecture was conducted on the ZC702 platform using a Software/Hardware (SW/HW) enviro nment in order to evaluate different implementation methods (SW and SW/HW) in terms of power consumption and run-time. The experimental results demonstrate that the SW/HW accelerations were enhanced by more than 70% in terms of the run-time speed relative to the SW solution. Besides, the power consumption of the SW/HW designs was reduced by nearly 60% compared with the SW case.

Key words: High-efficiency video coding decoder, IDCT, inverse quantization, software/hardware environment, field-programmable gate array

1. Introduction

High-efficiency video coding decoder (HEVC) is one of the new video coding standards developed by the ITU and the ISO/IEC, specially designed to address all the crippling limitations of the H.264/AVC standard. In comparison to the AVC, HEVC [1, 2] provides a better coding efficiency, including the reduction of bitrate by half at the same picture quality with a loss of 3 times in coding blocks complexity. In this paper, we focus particularly on the HEVC decoder layer which is based on the same structure of the H.264/AVC with some improvements in each coding step. The different decoding steps are illustrated in Figure 1 below.

The HEVC coding structure replaces the traditional macroblock by a quadtree-based block partition based on coding tree block (CTB). The latter can be divided into variable units called large coding units (LCUs) of sizes 16×16 to 64×64 instead of 16×16 only. In turn, each LCU can be split recursively into coding units (CUs) with each of them representing the basic unit. At this level, a decision is made to fix whatever partition is adopted for the prediction process (inter- or intracoding). In our case, the prediction algorithm is operated at the level of prediction units (PU), the sizes of which vary from 4×4 to 64×64. Depending on the PU

*Correspondence: abenatitallah@ju.edu.sa

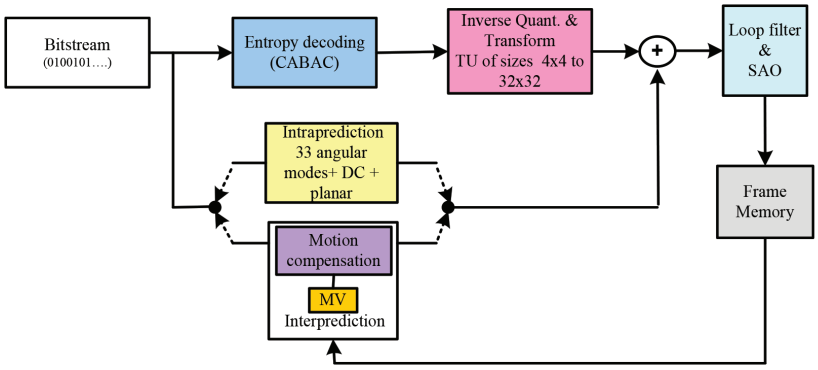

Figure 1. HEVC decoder layer.

partition adopted, inverse quantization and transform (IQ/IT) algorithms are performed at the level of those at (TUs) whose size can be 4×4, 8×8, 16×16, and 32×32. This variable size approach particularly helps to attain HD video resolutions. In fact, larger TU sizes achieve better energy compaction. Yet, they increase the computational complexity exponentially.

However, IQ/IT are heavily used in the HEVC decoder [3]. It accounts for 18% of the computational complexity of an HEVC video decoder. Since the standardization of the HEVC codec, researchers have worked continuously to reduce decoder complexity by adopting hardware acceleration as a solution. In fact, Martuza et al. [4] present a shared architecture that supports both 1D-IDCT 8×8 for HEVC and H.264/AVC using a new mapping technique. This approach is able to decode 1080 p at 67 fps using CMOS 180 nm technology. However, Chiang et al. exploit in [5] an optimized method for 2D-IDCT which generates new coefficients transformed from the already calculated coefficients. This architecture has since not only reduced the occupied area significantly and also met the throughput of 3840×2160 at 30 fps using CMOS 90 nm technology. However, Liang et al. [6] present an architecture that supports the 2D integer inverse discrete sine transform (2D-IDST) and 2D integer inverse discrete cosine transform (2D-IDCT) using two 1D-IDCT/IDST units and memory block. This architecture calculates 4 residual pixels in parallel in each clock cycle. This design can decode 7680×4320 at 30 fps. Furthermore, Goebel et al. [7] illustrate an efficient multisize hardware for DCT transform algorithm dedicated to support all HEVC TU sizes 4×4, 8×8, 16×16, and 32×32. The synthesis results using a Nangate 45nm standard-cell library allow a processing rate of 1080 p at 30 fps. In addition, Chen et al. [8] design a 2D inverse transform architecture that supports all TU sizes. This architecture can compute two rows in parallel during the 1D-IDCT instead of only one. In this case, the maximum throughput achieved is about 4K at 54 fps with the Xilinx Zynq platform. In [9], Ercan et al. propose an energy reduction technique that helps to lower the computational complexity present in the IDST and IDCT algorithms for all transform core sizes. In the worst of cases, this architecture can process 4K at 48 fps. Finally, in [10], Mohamed et al. provide a system-on-chip FPGA platform based on Xilinx Zynq to integrate the DCT coding block as an accelerator. The proposed design is capable to perform the coding of 1080 at 30 fps.

In this paper, we propose an optimized hardware architecture for IQ/IT algorithm. The IQ/IT architecture is developed using the VHSIC hardware description language (VHDL) [11] and integrated as an intellectual property (IP) core into a system-on-programmable-chip (SoPC) platform for the improvement of the treatment process [12]. However, the Xilinx Zynq-based FPGA platform [13] is an example of such circuit embedding a dual Cortex A9 processor as a processing system (PS) and programmable logic (PL) gate arrays of the FPGA

Series 7. Xilinx adopted the Advanced eXtensible Interface AXI4 of the Advanced Microcontroller Bus Architecture (AMBA) protocols to exchange data between the PS and the PL in an efficient and flexible way. In this context, designers could better exploit the existing partitioning for software/hardware (SW/HW) [14, 15], resulting in unrivaled levels of system performance, flexibility, scalability while providing system benefits in terms of power reduction, lower cost, and faster time to market.

In this work, we present an overview of the IQ/IT algorithms for the HEVC decoder in Section 2. Section 3 describes the proposed HW architecture for the IQ/IT block and illustrates the comparison of the synthesis results with previous designs proposed in literature. The SW/HW implementation and experimental validation are reported on in Section 4. Finally, some concluding remarks are drawn in Section 5.

2. IQ/IT algorithms for HEVC decoder

In HEVC, both the IQ/IT are applied to the quantized transform coefficients resulting from the entropy decoding algorithm. The HEVC quantizer is developed not only to maximize the trade-off between the bit-rate and the image quality, but also to reduce the complexity of the inverse quantization process. However, the general formula of inverse quantization [16, 17] is given in equation 1 which is valid for all TU sizes ranging from 4×4 to 32×32.

$$CoeffIQ = ((level * IQstep << (QP/6)) + offset) >> (M - 1 + (B - 8)), \quad (1)$$

where CoeffIQ is the inverse quantized transform coefficient of the level which is the quantized transform coefficient, offset is equal to 1<<(M-2+(B-8)), QP is the quantization parameter with a range of 0 to 51, M is equal to $\log_2(N)$ where N is the TU size (4, 8, 16, or 32) and B is the bit depth. In order to avoid floating point arithmetic operation in the formula of inverse quantization, the divisions are replaced by integer multiplications involving the inverse quantization step (IQstep) size and arithmetic shifts. However, only six IQstep values are computed in Table 1 which are determined according to the value of the QP with an increase of approximately 12%.

Table 1. Computed IQstep values according to QP.

QP%6	0	1	2	3	4	5
IQstep	40	45	51	57	64	72

The next step consists of applying the inverse transform. In fact, HEVC standard specifies the 2D-IDST for intracoded luminance transform blocks of size 4×4 pixels and the 2D-IDCT for all other transform blocks (of sizes 4×4, 8×8, 16×16, and 32×32 pixels). The unified equation defining the behavior of the 2D-IDCT [18] is given as follows:

$$Y = A^T X A, \quad (2)$$

where A^T, X, and Y are the transform block, the de-quantized block, and the result matrix (residual block), respectively, with sizes varying from 4×4 to 32×32 pixels. The first matrix multiplication $A^T X$ corresponds to 1D-IDCT/IDST with the core of the transform being the transposed A matrix which contains only integer coefficients. The second matrix multiplication accomplished the 2D-IDCT/IDST transform using the A matrix. According to the property of separability, the 2D-IDCT/IDST of a matrix of size $N \times N$ pixels can be decomposed into two 1D-IDCT/DST of length N which are performed column-wise and row-wise.

3. IQ/IT hardware architecture

The proposed architecture for IQ/IT, supporting 4×4 to 32×32 TU block sizes, is composed of two parts: the inverse quantization part and the inverse transform (IDCT and IDST) part. Reusability and configurability are the main features of this architecture where the architecture changes dynamically with the TU block size. In addition, the pipelined structure is used to achieve high throughput. In this section, we present the proposed architecture in detail, the synthesized results into Xilinx XC7Z020 FPGA as well as TSMC 180 nm standard-cells [19], and we conclude the section by attempting a comparison of our results with those of previously published works.

3.1. The inverse quantization hardware design

Figure 2 illustrates the proposed hardware architecture for inverse quantization. This architecture contains four processing units "Dequant_i" which are capable of processing 4 quantized pixels (Level0, ..., Level3) in parallel every 2 clock cycles and a control unit which is responsible for managing the processing of all TUs. In other words, the control unit receives input control signals (Reset, CLK, Start_IQ, QP and sel: select the TU size) and generates all internal control signals for each stage and the output flag (Done_IQ) signal to indicate that outputs CoeffIQ are valid.

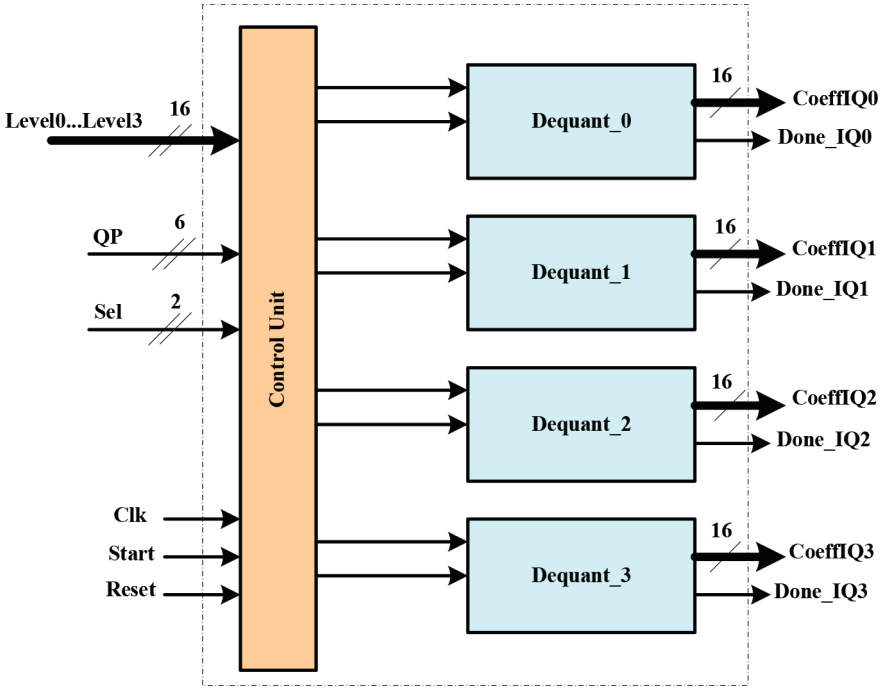

Figure 2. The proposed hardware architecture for inverse quantization.

The block diagram of the processing unit "Dequant_i" is presented in Figure 3. It allows the calculation of one inverse quantized pixel every 2 clock cycles. In order to optimize the computational complexity shown in the inverse quantization formulation (equation 1), the IQstep and QP/6 terms are stored in the ROM (Read-only Memory) blocks which are addressed by QP values. Then, the IQstep from ROM1 is multiplied by Level coefficient and left-shifted by value from ROM2. The last calculation step is to add an offset followed by a right-shift whose values change according to the TU size.

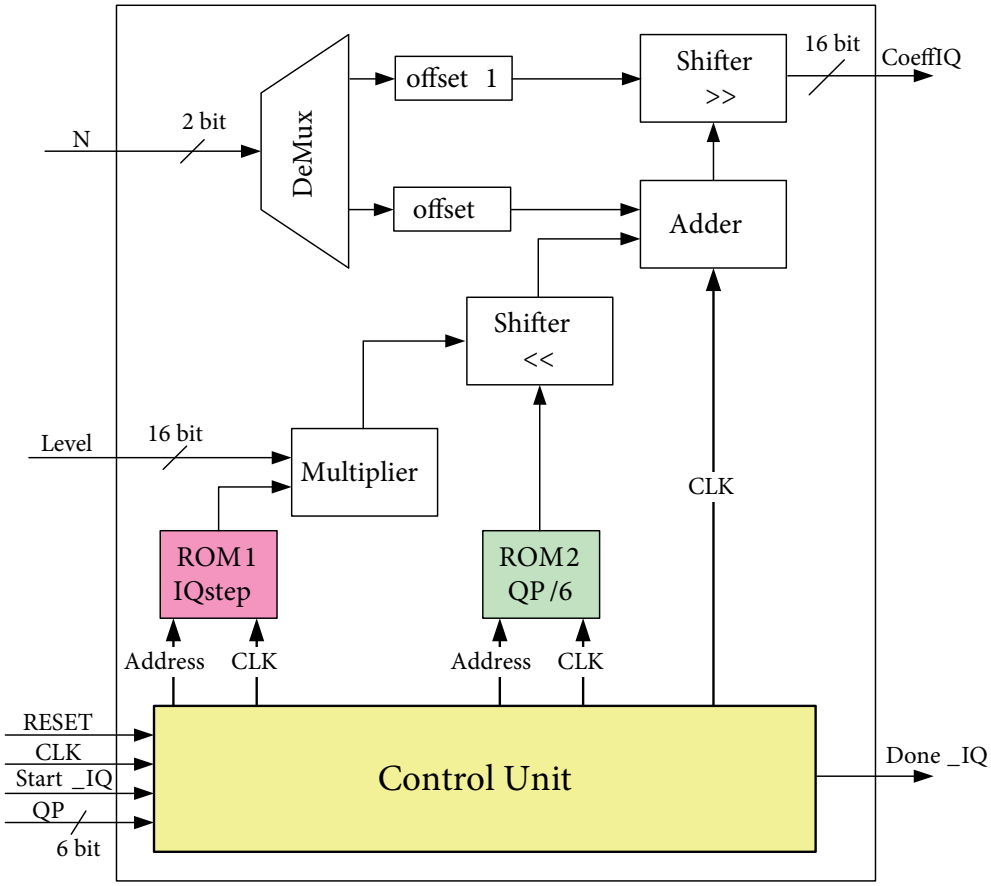

Figure 3. The processing unit "Dequant_i" block diagram.

3.2. The 2D-IDCT/IDST hardware design

The proposed architecture of the 2D-IDCT/IDST is presented in Figure 4. The operation in this architecture is performed in a sequential manner by exploiting the same 1D-IDCT/IDST processing unit to obtain 2D-IDCT/IDST transforms. This proposition enables to attain the best trade-off between time and hardware cost. According to Figure 4, this architecture is divided into three parts: 1D-IDCT/IDST, transpose memory and control unit which receives input control signals (Reset, CLK, Start_IDCT and sel: select the TU size) and generates all internal control signals for each stage and the output flag (Done_IDCT) signal to indicate that outputs coefficients are valid.

The main core of our architecture is the 1D-IDCT/IDST unit. It is used to support all transform matrix sizes (from 4×4 to 32×32 TUs). The IDCT equations defined in the HEVC test model decoder reference software version 10 (HM10) [20] present many multiplications by constants, which is very costly in terms of area and power consumption [21]. To reduce the computational complexity and increase the performance of the proposed architecture, these multiplications are replaced by shift and addition operations. Table 2 shows how to implement the most occurrent coefficients (2, 4, 9, 18, 36, 64, and 90) presented in all TU sizes using shift and addition operations. The calculation of these coefficients is performed inside the component XCoeff (Figure 5). On the other hand, the less redundant coefficients in Table 2 are calculated based on Xcoeff block by using

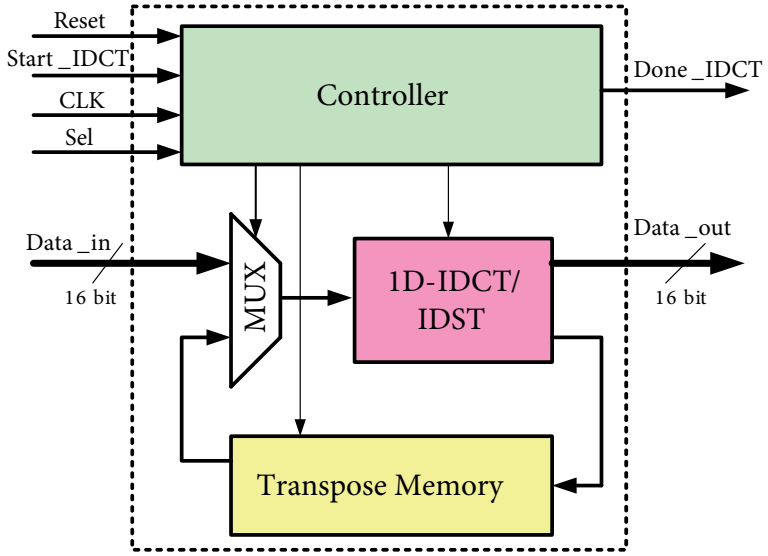

Figure 4. Architecture of the 2D-IDCT/IDST.

adders/subtractors operations in terms of the refinement blocks (R0 to R3 in Figure 5). Therefore, as shown in Table 3, the number of adder and shift operations in the proposed architecture is reduced by nearly 10% and 20% compared to the works presented in [22] and [23], respectively. In [23], a multiple constant multiplication (MCM) circuit was used to implement the multiplication coefficients in DCT algorithm using shifters and adders instead of regular multipliers.

On the other hand, it can be noticed from Figure 5 that the decomposition method based on an even–odd algorithm for all inverse transform matrices is used in order to achieve hardware sharing across transform sizes. Thus, the butterfly of 1D-IDCT4 is the even part of the 1D-IDCT8, itself is in turn the even part of the 1D-IDCT16 and this is repeated to build the 1D-IDCT32. These features allow us to better optimize hardware resources and facilitate the implementation stage. However, the implementation of the 1D-IDST4 is conducted separately from the 1D-IDCT4 using a multiplexer as shown in Figure 6. The 1D-IDST equations are hence kept the same as provided in the HEVC decoder (HM10) with the exception of the multiplications which are replaced by shift and addition operations. Our architecture processes both the 1D-IDCT4 and the 1D-IDCT8 in 3 clock cycles. However, 5 clock cycles are required for the 1D-IDCT16 and the 1D-IDCT32.

In our proposed architecture, after achieving the 1D-IDCT/IDST column process, the calculated intermediate coefficients are stored in a transpose memory, then used as inputs for the 1D-IDCT/IDST row process. Each 8 intermediate coefficients are concatenated in 128-bit (8×16-bit) to minimize the memory access. As shown in Figure 7, the transpose memory is implemented based on 32 FIFO (First In First Out) memory blocks to support all TU sizes. In fact, the 1D-IDCT4 requires 4 FIFOs to store all coefficients of a 4×4 TU while the 1D-IDCT8 requires 8 FIFOs to store all coefficients of an 8×8 TU. In the case of the 1D-IDCT16 and the 1D-IDCT32, they require 16 and 32 FIFOs, respectively. Each FIFO is designed to read/write four 128-bit values in 4 clock cycles. Moreover, the main goals of the MUX and DEMUX in Figure 7 are to decide between the 1D and 2D transforms and fix the size of matrix (4, 8, 16, or 32) that will be treated by the hardware design. In fact, the results of the 1D-IDCT column are generated column by column with each column being stored in one FIFO. Therefore, 1 clock cycle is needed to store one column in the transpose memory for 4×4

Table 2. Transformation of the multiplication constants to the shift and addition operations.

Less redundant coefficients	Most redundant coefficients (+ used coefficients)							Operation
	X_2	X_4	X_9	X_{18}	X_{36}	X_{64}	X_{90}	
	$<<1$	$<<2$	$<<3+1$	$<<4+X_2$	$X_4+<<5$	$<<6$	$X_{64}+$ $X_{18}+<<3$	
X_{89}							+	X_{90}-1
X_{88}	+						+	$X_{90}-X_2$
X_{87}		+					+	$X_{90}-X_4+1$
X_{85}		+					+	$X_{90}-X_4-1$
X_{83}				+		+		$X_{64}+X_{18}+1$
X_{82}				+		+		$X_{64}+X_{18}$
X_{80}						+		$X_{64}+<<4$
X_{78}		+		+		+		$X_{64}+X_{18}-X_4$
X_{75}	+		+			+		$X_{64}+X_9+X_2$
X_{73}			+			+		$X_{64}+X_9$
X_{70}	+	+				+		$X_{64}+X_4+X_2$
X_{67}		+				+		$X_{64}+X_2+1$
X_{61}		+				+		$X_{64}-X_2-1$
X_{57}	+		+			+		$X_{64}-X_9+X_2$
X_{54}				+	+			$X_{36}+X_{18}$
X_{50}				+				$X_{18}+<<5$
X_{46}				+		+		$X_{64}-X_{18}$
X_{43}	+		+		+			$X_{36}+X_9-X_2$
X_{38}	+				+			$X_{36}+X_2$
X_{31}		+	+	+				$X_{18}+X_9+X_4$
X_{25}			+					$X_9+<<4$
X_{22}		+		+				$X_{18}+X_4$
X_{13}		+	+					X_9+X_4

Table 3. Comparison of the operations numbers.

Architectures/Operations	Additions	Shifts	Multiplications
HM Original	402	32	344
Proposed	500	234	–
[22]	548	249	–
[23]	682	278	–

and 8×8 TU sizes. Two and 4 clock cycles are needed for 16×16 and 32×32 TU sizes, respectively. Once the operation of the 1D inverse transform is achieved, the memorized values resulting from 1D-IDCT/IDST are de-concatenated and processed row by row by the 2D inverse transform. This mechanism is summarized in Figure 8 where the 2D-IDCT/IDST hardware architecture finalizes IDCT operations for 4×4, 8×8, 16×16, and 32×32 TU sizes in 28, 54, 165, and 327 clock cycles, respectively.

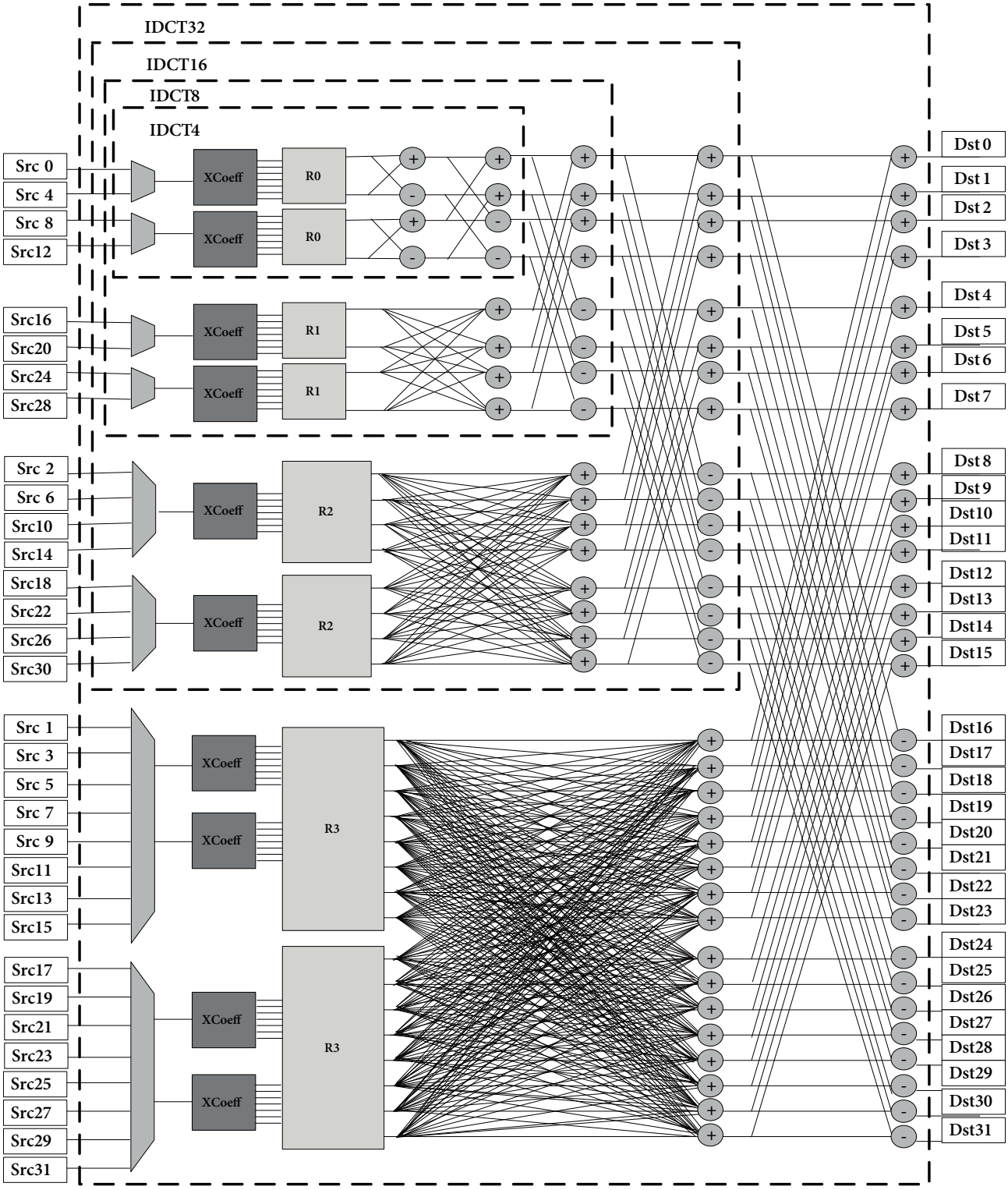

Figure 5. The 1D-IDCT block diagram

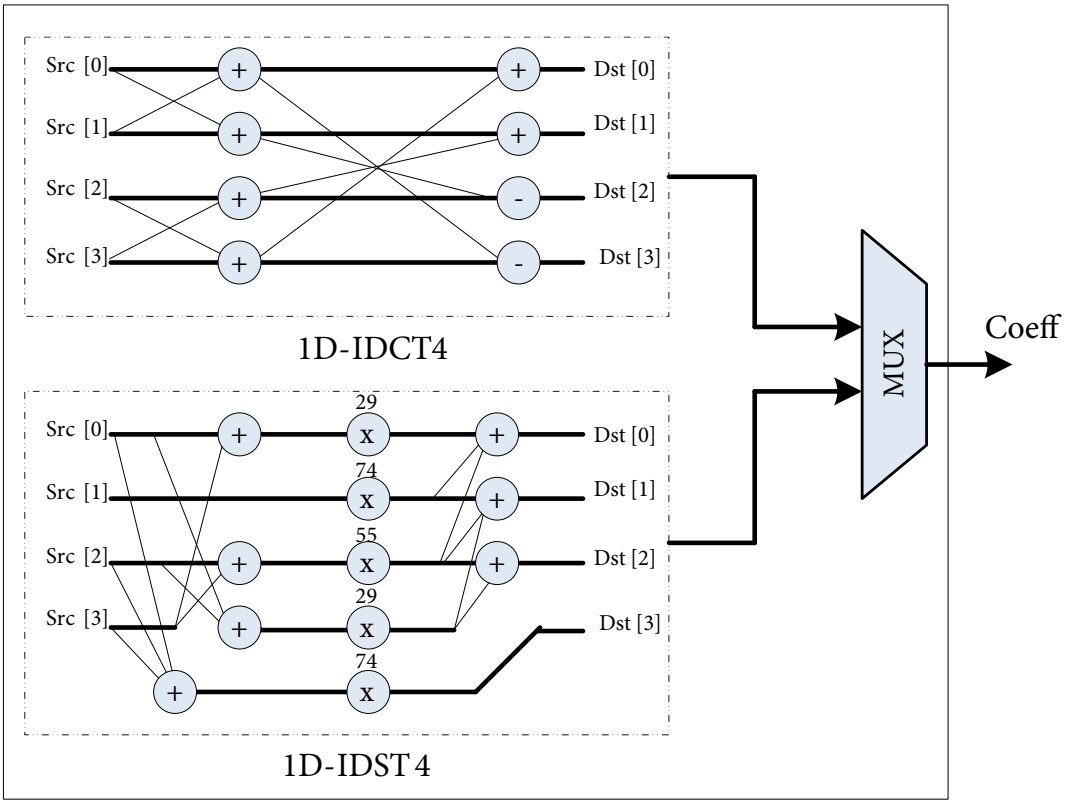

Figure 6. The 4×4 1D-IDCT/IDST datapath

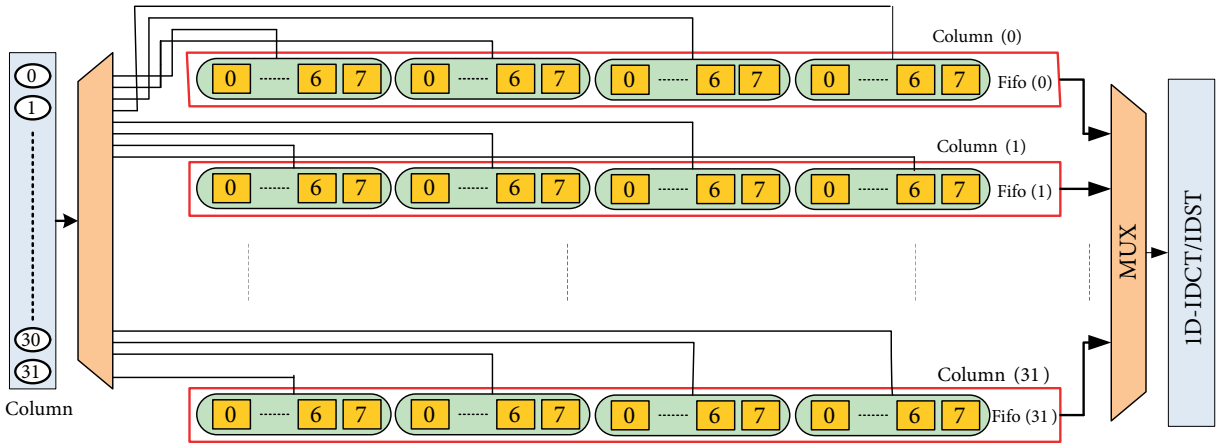

Figure 7. Transpose memory for the 2D-IDCT/IDST.

3.3. Hardware architecture of IQ/IT

Figure 9 describes the proposed architecture for the IQ/IT component of the HEVC decoder. This architecture contains the inverse quantization and transform blocks as well as the control unit which is used to synchronize both the access and data processing of the whole design. Our proposed architecture is thus designed to include

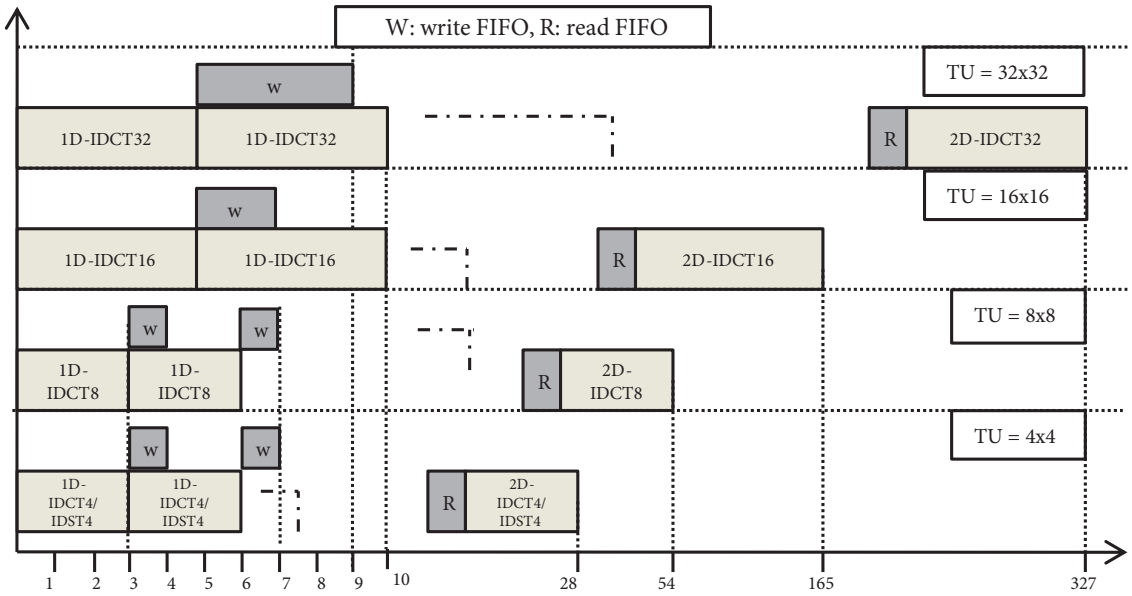

Figure 8. The 2D-IDCT clock cycles needed for 4×4, 8×8, 16×16, and 32×32 TU sizes.

both algorithms in the same hardware and used to decode the residual coefficients for 4×4, 8×8, 16×16, and 32×32 TU sizes. Accordingly, the inverse quantization is applied and followed by 1D-IDCT/IDST for each column and row of TUs by exploiting the pipeline technique existing between both blocks. This enables our architecture to receive 4×16-bit quantized pixels simultaneously every 2 clock cycles and to process them simultaneously by means of the de-quantization block. Then, each inverse quantized column of an $N \times N$ sized TU is processed by the 1D-IDCT/IDST. To accelerate data treatment, the pipeline technique is used between the inverse quantization and the 1D-IDCT/IDST hardware blocks so that the inverse quantization of the second column starts at the same time as the 1D-IDCT/IDST of the first column. The intermediate pixels processed by the 1D-IDCT/IDST are stored into transpose memory to be used as inputs for the 2D inverse transform process. Indeed, the 1D-IDCT/IDST is applied on the coefficients of the matrix row. Finally, the reconstructed residual block is obtained through "data_out" signal in N clock cycles which is dependent on the matrix size (4, 8, 16, and 32).

The operation of the IQ/IT architecture for a matrix of size 4×4 is conducted using the pipeline process as shown in Figure 10. Two clock cycles are enough to complete the inverse quantization of the first column followed by a memorization cycle. Then, each obtained column is processed by 1D-IDCT4/IDST4 and stored in a transpose memory block to be used for the 2D inverse transform. All in all, 34 clock cycles are enough to achieve the IQ/IT for a 4×4 sized matrix. Further, the IQ/IT for an 8×8 sized TU is carried out in the same way as for the 4×4 sized TU. At this stage, 6 clock cycles are required to de-quantify and store the first column and 63 clock cycles to operate all 8×8 TUs. For the rest of the matrix sizes, the calculation of the 1D-IDCT16 and the 1D-IDCT32 must begin after 12 and 24 clock cycles, respectively. Then, taking advantage of the pipeline process between the inverse quantization and the 1D-IDCT hardware blocks, the operation of the 2D-IDCT16 and the 2D-IDCT32 is finished after 218 and 684 clock cycles, respectively.

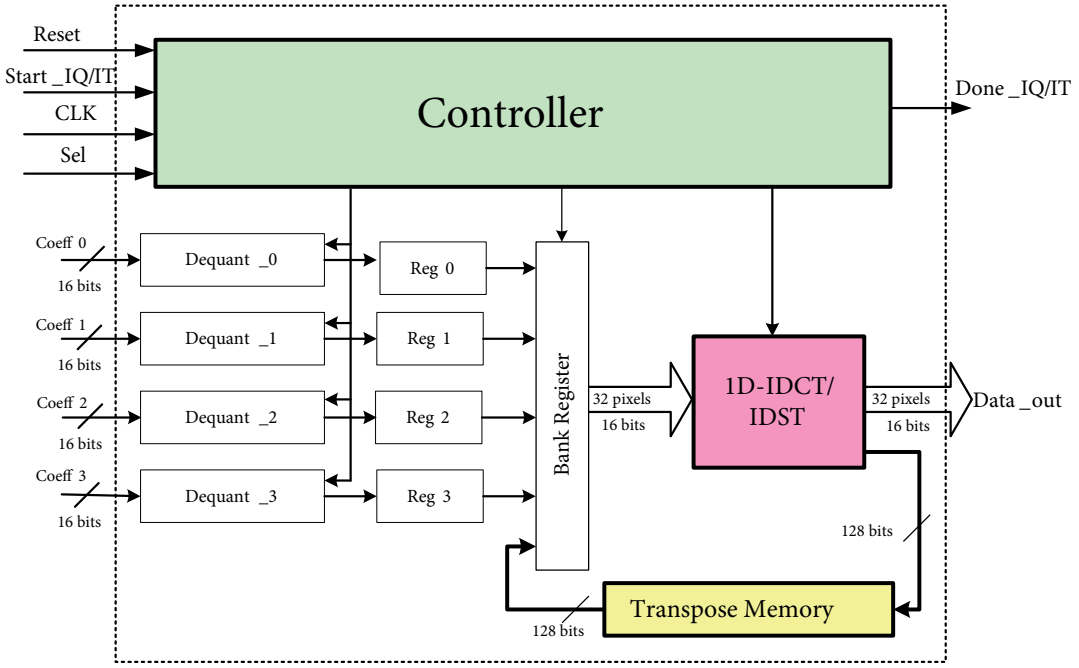

Figure 9. Hardware architecture for the IQ/IT component.

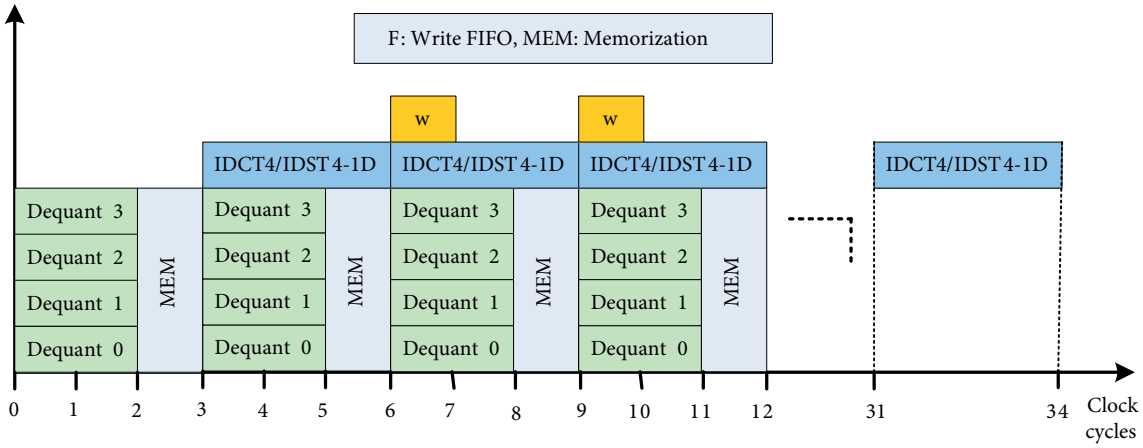

Figure 10. IQ/IT hardware operation for the 4×4 TU size.

3.4. Results of the IQ/IT implementation

The IQ/IT hardware architecture is designed by dint of the VHDL language, simulated with the Mentor Graphics ModelSim simulation tool, and synthesized with two different technologies: the Loenardo Spectrum tool using TSMC 180nm standard cells and the Xilinx XC7Z020 FPGA with speed grade 3 using Xilinx vivado 2015.2. The comparisons in terms of technology, frequency, area, and frame rate of the proposed architecture and previous works under ASIC technology are summarized in Table 4.

According to the findings presented in Table 4, the 2D-IDCT/IDST architecture proposed in this paper yields an area optimization of about 39% and 16% with less performance compared to [5] and [6], respectively,

Table 4. Implementation results of the 2D-IDCT/IDST and IQ/IT blocks under ASIC.

Design	Martuza et al. [4]	Chiang el al. [5]	Liang et al. [6]	Goebel et al. [7]	Proposed 2D-IDCT/IDST	Proposed IQ/IT
Technologies (nm)	180	90	65	45	180	180
Total gates (Kgates)	12.3	133.8	183.6	97.3	112	123
Memory (Kbits)	-	-	32.7	-	16	16
Frequency (MHz)	211.4	270	435	50	110	110
fps	1080 p at 67	4K at 30	8K at 30	1080 p at 30	1080 p at 30 4K at 8	1080 p at 25 4K at 6
Transform size	8×8	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32
Algorithms	1D-DCT	2D-IDCT	2D-IDCT /IDST	1D-DCT	2D-IDCT/IDST	2D-IQ&IDCT /IDST

knowing that [5] and [6] use recent technology and high frequency. On the other hand, the designs [4] and [7] propose the implementation only of the 1D-DCT which can achieve 1080 p at 67 fps and 1080 p at 30 fps, respectively. However, our proposed designs for 2D-IDCT/IDST and IQ/IT can reach 1080 p at 30 fps and 1080 p at 25 fps in worst case, respectively. Hence, our designs have better performances than [4] and [7] in terms of frame rate and use lower gate count than [5] and [6].

Table 5. Implementation results of the 2D-IDCT/IDST and the IQ/IT blocks under FPGA.

Design	Chen et al. [8]	Ercan et al.[9]	Mohamed et al. [10]	Proposed 2D-IDCT /IDST	Kammoun et al. [3]	Proposed IQ/IT
FPGA	Xilinx Zynq	Virtex 6	Xilinx Zynq	XC7Z020	Xilinx Zynq	XC7Z020
LUTs	5.8K	38.7K	5.6K	24.5K	22.6K	25K
FF	-	11.7K	4.7K	13.2K	18.8K	13.7K
RAM/FIFO (Kbits)	-	1152	10	16	24	16
Frequency (MHz)	222	150	-	146	100	146
DSP Blocks	128	-	108	-	44	4
fps	4K at 54	4K at 48	1080 p at 30	1080 p at 40	1080 p at 9	1080 p at 33 4K at 8
Algorithms	2D-IDCT /IDST	2D-IDCT /IDST	2D-DCT	2D-IDCT /IDST	2D-IQ&IDCT /IDST	2D-IQ&IDCT /IDST
HLS	no	no	yes	no	yes	no
Transform sizes	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32	4×4 to 32×32

Table 5 presents the complete synthesis results of the 2D-IDCT/IDST and the IQ/IT architectures under the Zynq XC7Z020 FPGA and the comparison with related works. According to these results, the percentage of

LUTs used by the 2D-IDCT/IDST is decreased by nearly 37% compared to [9]. Furthermore, our architecture does not use any DSP blocks as in [8] and [10]. The frame rate of the proposed 2D-IDCT/IDST represents enhancements of 25% compared to [10]. On the other hand, not only does our IQ/IT hardware architecture use about 47% of the LUTs, but it also employs only 2% and 1% of the available DSPs and block RAMs, respectively. In worst case, the proposed IQ/IT architecture allows the decoding of 1080 p at 33 fps at 146 MHz which presents an improvement of about 72% relative to [3]. By and large, the architecture that we have designed here is very appropriate to the HEVC decoder aiming for full HD real-time processing.

Table 6 presents the estimated throughput according to average numbers of TUs over various video sequences [24]. From this table, it is obvious that the throughput of the proposed 2D-IDCT/IDST and 2D-IQ&IDCT/IDST architectures can reach up to 4K at 35 fps at and 4K at 21 fps at 146 MHz, respectively.

Table 6. Estimated frame rate according to average numbers of TUs.

HEVC decoder		2D-IDCT/IDST		2D-IQ&IDCT/IDST	
TUs Size	% of used TUs [24]	clock cycles	fps	clock cycles	fps
32×32	40%	327	4K at 35 1080 p at 141	684	4K at 21 1080 p at 84
16×16	30%	165		218	
8×8	18%	54		63	
4×4	12%	28		34	

4. SW/HW performance evaluation

Figure 11 illustrates the SW/HW implementation of the IQ/IT architecture on the ZC702 based platform [25]. The software can be carried out using the dual core ARM Cortex A9 with an operating frequency of up to 700 MHz and compiled with a standalone application using a software development toolkit in the Windows development environment. However, the HW blocks are created and customized using the Xilinx platform studio.

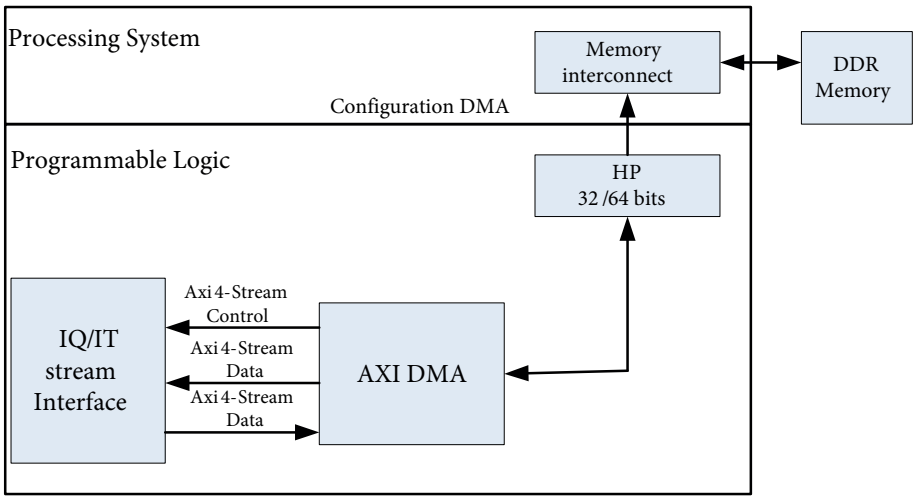

Figure 11. Heterogeneous SW/HW system

From Figure 11, we can see that the communication between the ARM Cortex A9 processor and the IQ/IT hardware block is carried out using the AXI4-stream interface [21, 26] which is specially designed for maximum bandwidth access to the on-chip memory and DDR memory of the PS. This mode of transfer supports unlimited data burst sizes and provides point-to-point streaming data without using any addresses. However, it is necessary to fix a starting address to begin a transaction between the processor and HW block. In our case, the AXI DMA (Direct Memory Access) peripheral is introduced between the IQ/IT hardware block and SW parts as presented in Figure 11 in order to provide a high bandwidth for data transfer.

The experimental validation is limited to the 16×16 sized TU because the total space of the XC7Z020 FPGA cannot support the generated interface for the 32×32 sized TU. On the other hand, the SW/HW implementation of the IQ/IT hardware block in the XC7Z020 FPGA is realized through Vivado 2015.2 tool for the 4×4, 8×8, and 16×16 sized TUs. As a result, this system requires 17.5k (33%) LUTs, 22 (16%) RAM blocks and 8 (4%) DSP blocks. The measurement is carried out for the execution time and the power consumption in the SW/HW context of the IQ/IT block in order to estimate the performance compared to the SW implementation. In fact, the processor timer is used to determine the execution time. The power consumption measurement is done by using Texas Instruments fusion digital power designer software. The SW implementation is realized based on the HEVC reference software decoder (HM10.0). For verification, The 100 TUs blocks are selected from the 1080p full high definition (FHD) sample encoded video with different sized TUs (4×4 to 16×16) and QP values equal to 22, 27, 32 and 37. Hence, Figure 12 presents execution time and power consumption comparison between the SW and the SW/HW implementations of the IQ/IT.

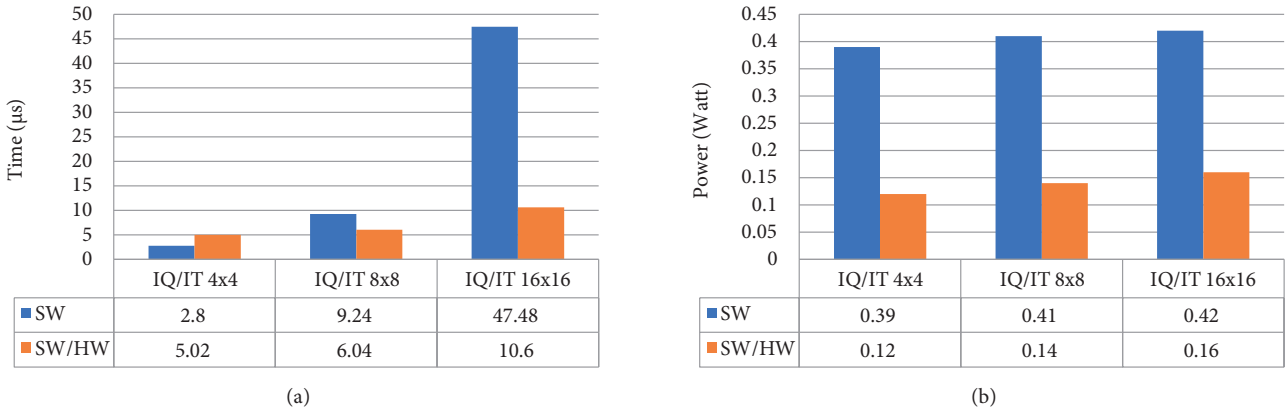

Figure 12. Execution time (a) and power consumption (b) comparison between the SW and the SW/HW implementations of the IQ/IT

From Figure 12a, we can notice that the execution time depends on the size of the block. Indeed, we can achieve a gain of about 70% when adopting the SW/HW solution with TU of size 16×16 compared to the SW solution. Otherwise, in the case of 4×4 TU, the SW implementation seems better than SW/HW case. In this case, the variation of the execution time is absolutely influenced by the reset time of DMA, design complexity, and also the amount of data communicated between processor and hardware block. Therefore, acceleration efficiency will be highlighted when increasing the complexity of the hardware design. Additionally, the power consumption measurement shows that about 60% is saved with the SW/HW solution compared to the SW solution as illustrated in Figure 12b. This gain can be explained by the fact that the operating frequency of the SW is about 700 MHz while the SW/HW design is influenced by the frequency of the bus and hardware part

which are equal to 100 MHz. These findings point out that the SW/HW solution allows a compromise between the SW flexibility and the HW performance.

Table 7 presents a comparison between our SW/HW architecture and other realizations. The execution time and the power consumption of our architecture are the average of the obtained values after executing the SW/HW HEVC decoder using classes B, C, and D video sequences with QP values equal to 22, 27, 32 and 37. From Table 7, we can conclude that our design has better performance than those in [27–29]. On the other hand, our design allows a high decrease in power consumption equal to 99% with almost the same performance compared to [30].

Table 7. Performance comparisons of the video decoder.

Ref	fps	Power consumption (Watt)	Decoder video	Specification
[27]	CIF at 0.0625	-	H.264/AVC (INTER)	PowerPC 440 at 400 MHz (Linux/Xenomai)
[28]	1080 p at 1.45	-	HEVC (INTRA)	ARM Cortex-A9 processor at 1.2 GHz
[29]	1080 p at 0.25	1.609 Watt	HEVC (INTER)	ARM Cortex-A9 processor at 700 MHz + interpolation filter accelerator at 100 MHz
[30]	1080 p at 9.1	15 Watt	HEVC (INTER)	Tilera at 1.3 GHz (TILE-Gx8036)
Our Design	240 p at 15.5 480 p at 4 1080 p at 1.5	0.107 Watt	HEVC (INTRA)	ARM Cortex-A9 processor at 700 MHz + IQ/IT accelerator at 100 MHz

5. Conclusion

High performance hardware architecture of the IQ/IT is proposed in this paper that is used for the HEVC decoder. The proposed architecture is designed to support all HEVC Transform Unit (TU) sizes: 4×4, 8×8, 16×16, and 32×32. The pipeline technique is used between the inverse quantization and the 1D-IDCT/IDST to achieve a higher frame rate. The decoding speeds of 1080 p at 33 fps at 146 MHz and and 1080 p at 25 fps at 110 MHz were achieved when mapped onto Xilinx XC7Z020 FPGA and to TSMC 180 nm standard-cell, respectively. Finally, the IQ/IT architecture is connected as coprocessor to the ARM Cortex A9 processor using the AXI4-stream interface. Experimental results demonstrate that SW/HW accelerations are more than 70% improved in terms of the run-time speed relative to the SW solution. Besides, estimated power consumption of our designed systems is 60% less than the SW part which runs at 700 MHz.

Acknowledgment

The authors would like to express their deepest gratitude to Prof. Ali M. Amri, ENET'Com Sfax, for his painstaking editing of the paper.

References

[1] High Efficiency Video Coding, ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, 2013.

[2] Pourazad M T, Doutre C, Azimi M, Nasiopoulos P. HEVC: The new gold standard for video compression. IEEE Consumer Electronics Magazine 2012; 1 (3): 36-46. doi: 10.1109/MCE.2012.2192754

[3] Kammoun M, Ben Atitallah A, Ali KMA, Ben Atitallah R. Case study of an HEVC decoder application using high-level synthesis: intra prediction, dequantization, and inverse transform blocks. Journal of Electronic Imaging 2019; 28 (03): 1-20. doi: 10.1117/1.JEI.28.3.033010

[4] Martuza MA, Wahid K. A cost effective implementation of 8×8 transform of HEVC from H.264/AVC. In: IEEE Canadian Conference on Electrical and Computer Engineering (CCECE); Montreal, QC, Canada; 2012. pp. 1-20.

[5] Pai-Tse C, Tian Sheuan C. A Reconfigurable inverse transform architecture design for HEVC Decoder. In: IEEE International Symposium on Circuit and System (ISCAS); Beijing, China; 2013. pp. 1006-1009.

[6] Hong L, He W, He G, Mao Z. Area-efficient HEVC IDCT/IDST architecture for 8Kx4K video decoding. IEICE Electronics Express 2016; 13(6): 1-20. doi: 10.1587/elex.13.20160019

[7] Goebel J, Paim G, Agostini L. An HEVC multi-size DCT hardware with constant throughput and supporting heterogeneous CUs. In: IEEE International Symposium on Circuits and Systems (ISCAS); Montreal, QC, Canada; 2016. pp. 2202-2205.

[8] Chen M, Zhang Y, Chao L. Efficient architecture of variable size HEVC 2D-DCT for FPGA platforms. International Journal of Electronics and Communications (AE) 2017; 73: 1-8. doi: 10.1016/j.aeue.2016.12.024

[9] Kalali E, Ozcan E, Yalcinkaya O, Hamzaoglu I. A Low Energy HEVC Inverse Transform Hardware. IEEE Transactions on Consumer Electronics 2014; 60 (4): 754-761.

[10] Mohamed B, Elsayed A, Amin O, Khafagy E, Kennelly J et al. High-Level Synthesis Hardware Implementation and Verification of HEVC DCT on SoC-FPGA. In: 13th International Computer Engineering Conference (ICENCO); Cairo, Egypt; 2017. pp.1-20.

[11] Kthiri M, Loukil H, Ben Atitallah A, Kadionik P. FPGA architecture of the LDPS motion estimation for H.264/AVC video coding. Journal of Signal Processing Systems 2012; 68 (2): 273-285. doi: 10.1007/s11265-011-0614-x

[12] Ben Atitallah A, Loukil H, Kadionik P, Masmoudi N. Advanced design of TQ/IQT component for H.264/AVC based on SoPC validation. WSEAS Transactions on Circuits and Systems 2012; 11(7): 211-223.

[13] Xilinx. Inc. Zynq®-7000 family is based on the Xilinx SoC architecture. Zynq-7000 SoC Data Sheet: Overview; DS190 (v1.11.1); 2018.

[14] Ben Atitallah A, Kadionik P, Masmoudi N, Levi H. FPGA implementation of a HW/SW platform for multimedia embedded systems. Design Automation for Embedded Systems 2008; 12(4): 293-311.

[15] Ben Atitallah A, Kadionik P, Ghozzi F, Nouel P. An FPGA implementation of HW/SW codesign architecture for H. 263 video coding. AEU-International Journal of Electronics and Communications 2007; 61(9): 605–620.

[16] Gweon R, Lee Y. N-Level Quantization in HEVC. In: IEEE International Symposium on Broadband Multimedia Systems and Broadcasting; Seoul, South Korea; 2012. pp.1-20.

[17] Budagavi M, Fuldseth A, Bjontegaard G. HEVC transform and quantization. High Efficiency Video Coding (HEVC) 2014; 1: 141-169.

[18] Sullivan Gary J, Ohm JR, Han WJ, Wiegand T. Overview of the high efficiency video coding (HEVC) standard. IEEE Transition Circuits System Video Technology 2012; 22(12): 1648-1667. doi: 10.1109/TCSVT.2012.2221191

[19] Artisan Components. TSMC 0.18um 1.8-Volt SAGE-XTM Standard CellLibrary Databook, 2001.

[20] Bross B, Han W-J, Ohm J-R, Sullivan Gary J. High Efficiency Video Coding (HEVC) text specification draft 10. JCT-VC, Doc. JCTVC-L1003 Geneva, Switzerland, 2013.

[21] Kammoun M, Ben Atitallah A, Ben Atitallah R, Masmoudi N. Design exploration of efficient implementation on SoC heterogeneous platform: HEVC intra prediction application. Wiley & Sons, International Journal of Circuit Theory and Applications 2017; 45 (12): 2243-2259. doi: 10.1002/cta.2308

[22] Ahmed A, Muhammad Usman S, Ata ur R. N Point DCT VLSI Architecture for Emerging HEVC Standard. VLSI Design 2012, 2012(2): 1-13. doi: 10.1155/2012/752024

[23] Pramod K M, Sang Y P, Basant K M, Khoon S L. Efficient Integer DCT Architectures for HEVC. IEEE Transactions on circuits and systems for video technology 2014, 24(1):168-178.

[24] Stankowski J, Grajek T, Karwowski D, Klimaszewski K, Stankiewicz O et al. Analysis of frame partitioning in HEVC. International Conference on Computer Vision and Graphics (ICCVG); Warsaw, Polandpp; 2014, pp.602-609.

[25] ZC702 Evaluation Board for the Zynq-7000 XC7Z020 SoC User Guide, UG850 (v1.7), March 27, 2019.

[26] Xilinx AXI reference guide, UG761 (v13.1), March 7, 2011.

[27] Kthiri M, Kadionik P, Le gal B, Lévi H. Performances analysis and evaluation of Xenomai with a H.264/AVC decoder. In: IEEE ICM; Hammamet, Tunisia; 2011. pp. 1-20.

[28] Chi CC, Alvarez-Mesa M, Bross B, Juurlink B, Schierl T. SIMD acceleration for HEVC decoding. IEEE Transactions on Circuits and Systems for Video Technology 2015; 25(5):1-20.

[29] Ayadi LA, Loukil H, Ben Ayed MA, Masmoudi N. Efficient implementation of HEVC decoder on Zynq SoC platform. In: IEEE ATSIP; Sousse, Tunisia; 2018. pp. 21-24.

[30] Chi CC, Alvarez-Mesa M, Lucas J, Juurlink B, Schierl T. Parallel HEVC Decoding on Multi-and Many-core Architectures. Journal of Signal Processing Systems 2013; 71:247-260. doi: 10.1007/s11265-012-0714-2