

Implementation of Individual and Combined In-loop Filters for HEVC and AVC Video Coding Standards on ASIC Platform Using 32 nm Technology Library

Siu Hong Loh^{1*}, Aw Yong Pik Yee¹, Jia Jia Sim² and Kim Ho Yeap¹

¹Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

²Faculty of Business and Finance, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

*Corresponding author: lohsh@utar.edu.my

Submitted 23 April 2025; Revised 26 June 2025; Accepted 30 June 2025; Available online 03 July 2025.

Copyright © 2025 The Authors.

Abstract: High Efficiency Video Coding (HEVC) and Advanced Video Coding (AVC), the developed international standards for video compression, offer significantly better video compression efficiency. However, there are some similarities in the in-loop filtering algorithms which can be combined and implemented in a design. By evaluating the similarities in in-loop filtering algorithms of HEVC and AVC, a combined in-loop filter is designed to evaluate its implementation cost and performance with respect to individual AVC and HEVC in-loop filters. The in-loop filters of three designs are used to process the pixels in a stream of 64×64 blocks. The proposed combined design is implemented in Verilog. The Verilog RTL code is verified in application specific integrated circuit (ASIC) platform using 32 nm technology library and Synopsys EDA tools. Experimental results illustrate that the proposed combined in-loop filters reduce the total resources utilization of individual in-loop filters by 20% in terms of total cell area. In comparison to the individual HEVC in-loop filter design, which is greater than the individual AVC in-loop filter design, the unified design consumed just 0.38 times power consumption overhead.

Keywords: Advanced video coding; High efficiency video coding; In-loop filters; Video compression.

1. INTRODUCTION

For a variety of devices, including computers, smartphones, and tablets, to play videos, video coding is required. The two parts that make it up are the encoder and the decoder. Video content is compressed for data transmission using an encoder, and then it is decompressed and converted back to its original form using a decoder. Using an encoder to compress video content into lower file sizes while keeping quality through a decoder, video coding helps us transmit video content over the internet more easily. With 5G, the mobile ecosystem will grow into new real objects thanks to its high speeds, greater reliability, and zero latency [1]. Concurrently, developments in 5G mobile networks will open new possibilities for video-enabled services like augmented reality, mixed reality, and virtual reality. In order to deliver higher-quality videos, these services are anticipated to rely on ongoing video development up to 8K resolutions and new, stringent standards including low end-to-end latency and fast throughput [1]. Higher video compression efficiency and an appropriate performance level for new media devices over 5G networks will be provided with the aid of video coding standards. By providing higher-quality video, it also makes for improved user experience.

High Efficiency Video Coding (HEVC) is a widely used standard in video compression applications [2]. HEVC standard decoder block diagram with loop filtering is shown in Figure 1 below. Sample adaptive offset filter (SAO) and deblocking filter (DBF) are the two in-loop filtering modules of HEVC shown in Figure 1. To eliminate the blocking artifacts from the encoded image, application of deblocking filter (DBF) is required. The bit rate at which video coding must be reduced is determined by applying the SAO to the output. Thus, there will be unique in-loop filtering modules for every video coding scheme.

Due to the increased demand for video quality, the growing volume of video content, and the need for improved reliability, there is a requirement for in-loop filtering modules that can cater to multiple video coding standards. However, most existing in-loop filtering modules are designed to work only with a specific video coding standard, and their performance is primarily compared across different architectures within the same coding standard [3], [4]. When multiple video coding standards are utilized, it becomes necessary to process video frames or images multiple times for each standard [5], [6]. Therefore, it is essential to develop a unified in-loop filter that can accommodate several video coding standards by identifying operational similarities among them.

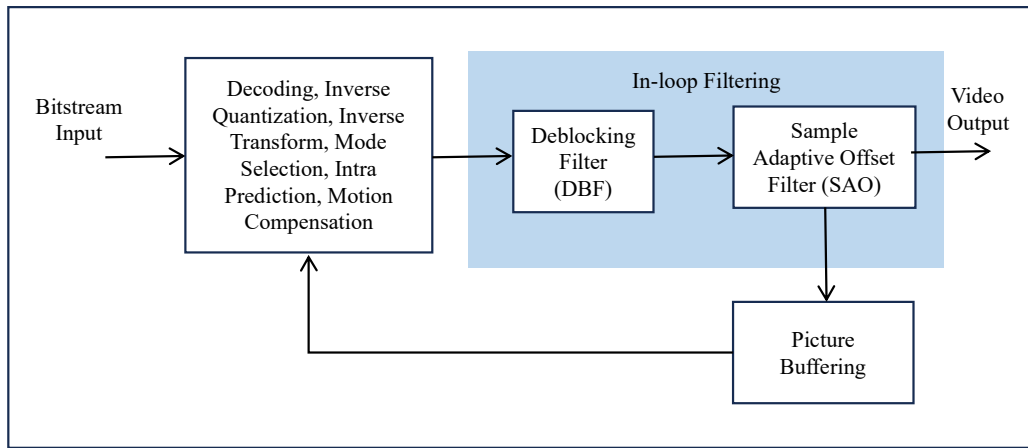


Figure 1. Simplified HEVC decoder block diagram. Sample adaptive offset filter (SAO) and deblocking filter (DBF) are the two in-loop filtering modules of HEVC.

The incorporation of video coding in various applications like cell phones, laptops, televisions, and tablets poses a challenge as it increases the overall product cost. The expenses related to hardware and system efficiency are evaluated for different video coding standards. Consequently, the utilization rate is more efficient when the in-loop filtering module is designed exclusively for a specific video coding standard [4]. Additionally, while the latest video coding standards offer improved quality and performance for video content, the coding tools associated with them are considerably more expensive compared to older video coding standards [6]. This issue can be addressed by implementing combined in-loop filters capable of accommodating multiple video coding standards. By doing so, the utilization rate of individual in-loop filtering modules can be optimized, leading to reduced implementation costs.

The aim of this project is to apply simplification techniques to synthesize and implement individual and combined in-loop filter algorithms for HEVC and Advanced Video Coding (AVC) using 32 nm technology library. To achieve the aim, the following objectives have been identified:

- To develop a unified AVC and HEVC in-loop filters for maximization of resource use (reduce chip area).
- To evaluate the performance of individual and unified AVC and HEVC in-loop filters.

In this project, individual and combined in-loop filters are designed and implemented for several video coding standards to be executed on Application Specific Integrated Circuit (ASIC) platform using 32nm technology library. The individual in-loop filters for a video coding standard are used to compare with the performance of combined in-loop filters applicable to at least two video coding standards. To achieve the objectives in the previous section, combined in-loop filtering modules are implemented by combining at least two video coding standards using a multiplexer. All the values/tasks are verified and identified through Synopsys EDA tools such as VCS®, Design Compiler®, and IC Compiler®.

The video coding standards used in this project are AVC and HEVC. The in-loop filtering modules are made up of a few components which are DBF and SAO filter depending on the applied video coding standards. The code used for implementing individual and combined in-loop filters is Verilog code. Data such as total cell area, power consumption, clock cycles per execution, and resources utilization rate on ASIC are collected and tabulated by evaluating the timing report from the Synopsys EDA tools.

2. RELATED WORK

Much research works on in-loop filtering modules for several video coding standards such as HEVC standard, AVC standard and Versatile Video Coding (VVC) have been recently proposed in the literature. The in-loop filtering modules for several video coding standards can be synthesized as individual and combined way. Horizontal directional scalable hardware architecture is proposed by Baldev et al. for the HEVC deblocking filter [3]. Using TSMC 180 nm technology and the Verilog HDL programming language, the design has been implemented in both ASIC and Virtex-6 Field Programmable Gate arrays (FPGA). For control unit (CU) blocks of different sizes, this approach looks into efficient pipeline and parallel processing phases. The processing clock cycles for CU 64x64, 32x32, 16x16, and 8x8 are found to be 96, 128, 176, and 512, respectively, based on the recommended methodology. They have equivalent gate counts of 254.44K, 179.68K, 176.88K, and 489.02K, respectively. The proposed architecture may be used to process Ultra High Definition (UHD) films at 125 MHz and 50 frames per second, making it appropriate for use in broadcast and gaming on the upcoming generation of consumer virtual reality (VR) devices.

Palau et al. proposed a real-time hardware architecture for HEVC DBF seeking to dissipate minimal power while maintaining maximum throughput [7]. Up to 64 samples per clock cycle were handled by the architecture when it was presented. Furthermore, two parallel data pathways were to be used for processing the luminance and chrominance components in the proposed hardware architecture. The designed DBF hardware was validated using the ModelSim® program. The RTL Compiler tool from Cadence was used to synthesize the suggested hardware for a typical cell of TSMC 40 nm library. The architecture achieved the lowest power dissipation of 4.73 mW when running 8K UHD videos in comparison to competing designs. The architecture also reached real-time processing for Full HD videos at 60 frames per second running at 7.73 MHz and 8K UHD videos at 60 frames per second running at 124.4 MHz.

Li et al. proposed an area-efficient hardware architecture for a general DBF that supports both AVC and HEVC video coding standards [8]. The architecture optimizes the reuse of similar filter components to minimize the increased area cost associated with supporting multiple standards, resulting in a 37% reduction in gate count. By dynamically switching the connection form of filter modules between series and parallel, a favorable balance is achieved between throughput and area utilization. The architecture achieves a throughput of 8K@60fps for H.264/AVC and 8K@160fps for HEVC. Experimental results demonstrate that the multi-standard architecture consumes 61.3k gates at 526 MHz, and the design has been successfully integrated into a multi-standard video codec.

Saravanan et al. introduced a novel dual standard deblocking filter that supports both AVC and HEVC [9]. The proposed design utilizes a unique filtering order that allows four edges to be processed simultaneously for both AVC and HEVC. For AVC, the design takes 18 clock cycles to process a 16 x 16 macroblock, while for HEVC, it takes 12 clock cycles to process a 16 x 16 coding transform unit (CTU). Additionally, the proposed design incorporates the concept of parallel processing in HEVC, further improving its execution speed. The synthesis and mapping of the design are performed using Xilinx ISE 14.7 and Virtex-5 FPGA, respectively. This results in an increased of execution speed and a reduction in the number of required clock cycles compared to existing architectures. Finally, the physical implementation of the proposed architecture is carried out using the Semiconductor Lab (SCL) 180 nm process node.

The previously proposed projects focused solely on the architecture of the deblocking filter in HEVC and does not include the SAO filter [3], [7]. In these projects, due to the inadequacy of the operation's implementation, there is less contrast of findings rendered between all the in-loop filters for various video coding standards. Therefore, in our project, combined in-loop filters (DBF and SAO) of HEVC and AVC are implemented. The second research gap is the technology library used to implement the in-loop filters. Many of the researchers are utilizing 90nm technology and 180nm technology library. Therefore, the 32nm technology was employed to implement the in-loop filters as part of this research endeavor.

The main contribution of this work is that the combined HEVC and AVC in-loop filters use a unified DBF for both HEVC and AVC and a SAO filter only applied for HEVC. The rationale for implementing a single DBF for both HEVC and AVC is that the design area can be reduced, and similar resources can be exchanged. As a result, the total utilization rate in combined in-loop filters for HEVC and AVC is lower than in individual in-loop filters.

3. METHODOLOGY

This project started with the implementation of the individual in-loop filtering modules for several video coding standards in Verilog language and the design was synthesized to ASIC platform. ASIC synthesis is performed using Synopsys EDA tools for 32 nm technology. The ASIC interface is used because, in comparison to FPGAs, ASICs' power consumption can be finely controlled and optimized. There are two video coding standards discussed in this project which are HEVC and AVC. Each video coding standard's algorithm was used to execute the individual in-loop filtering modules. For example, DBF and SAO were adopted in HEVC standard and DBF was applied in AVC standard.

Next, both the HEVC and AVC individual in-loop filtering modules were merged to form a combined in-loop filtering module, and the resources used by both standards were combined and reused to minimize usage rates. The combined in-loop filtering module was then executed on the ASIC platform. Subsequently, performance such as total cell area, power consumption, clock cycles per execution, and resources utilization were recorded and tabulated by evaluating the timing reports generated. The tabulated data can be used to compare with the individual in-loop filtering modules' tabulated data. The functional correctness of the two individual and combined design were verified. To demonstrate that the resource utilization rate of combined AVC and HEVC in-loop filtering modules is decreased, the cumulative resource utilization of individual AVC and HEVC in-loop filtering modules is gathered. All the tasks mentioned are summarized in Figure 2.

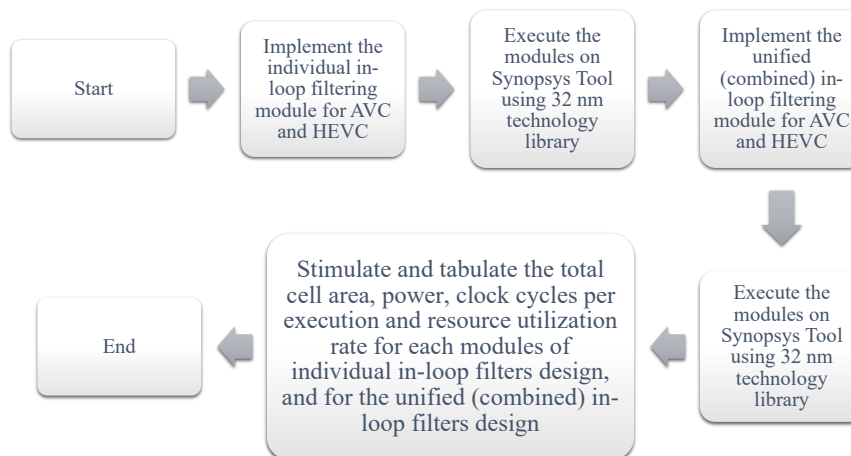


Figure 2. Methodology overview.

3.1 In-loop Filtering Modules of Individual HEVC

The individual in-loop filtering module in HEVC mainly consists of two main components, namely DBF and SAO. For DBF, it involves several steps inclusive of boundary strength judgement, filtering decision, and filtering operation shown in Figure 3(a). For SAO, it involves several steps inclusive of control unit, filtering decision (edge offset (EO)/band offset (BO)) and filtering operation shown in Figure 3(b).

3.2 In-loop Filtering Modules of Individual AVC

AVC's individual in-loop filtering modules are made up of just one part, the DBF. As seen in Figure 4, the DBF consists of several steps which are boundary strength calculation, filter decision, and filtering operation.

3.3 In-loop Filtering Modules of Combined HEVC and AVC

The unified in-loop filtering combined HEVC and AVC is made up of three parts: horizontal boundary DBF, vertical boundary DBF, and the SAO filter. Since the vertical boundary DBF is used and similar in both individual HEVC and AVC, the vertical boundary DBF is unified. The user is given an input signal called `usr_choice` to choose between HEVC and AVC. When `usr_choice` equals to 1, AVC is selected. When `usr_choice` equals to 0, HEVC is selected. Figure 5 shows the overview of combined in-loop filters. The horizontal boundary DBF works similarly to the horizontal boundary DBF in individual AVC, while the SAO filter works similarly to the SAO filter in individual HEVC.

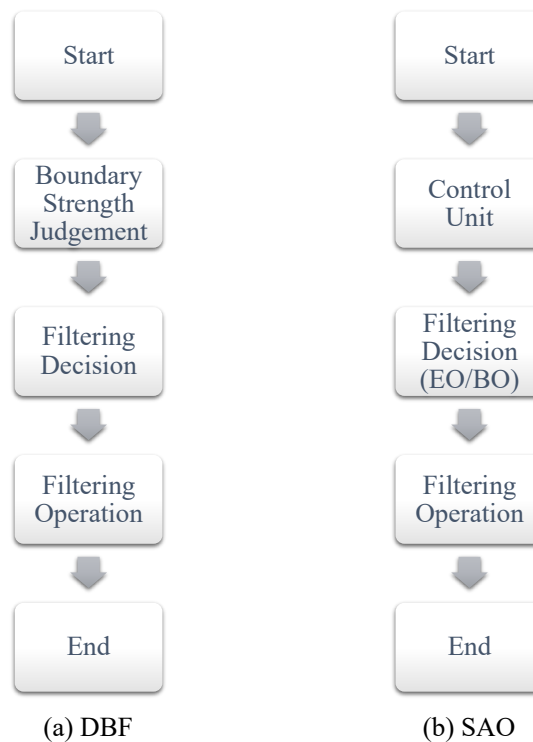


Figure 3. Components of individual in-loop filtering modules in HEVC. The individual in-loop filtering module in HEVC mainly consists of two main components, namely DBF and SAO.



Figure 4. Components of individual in-loop filtering modules in AVC.

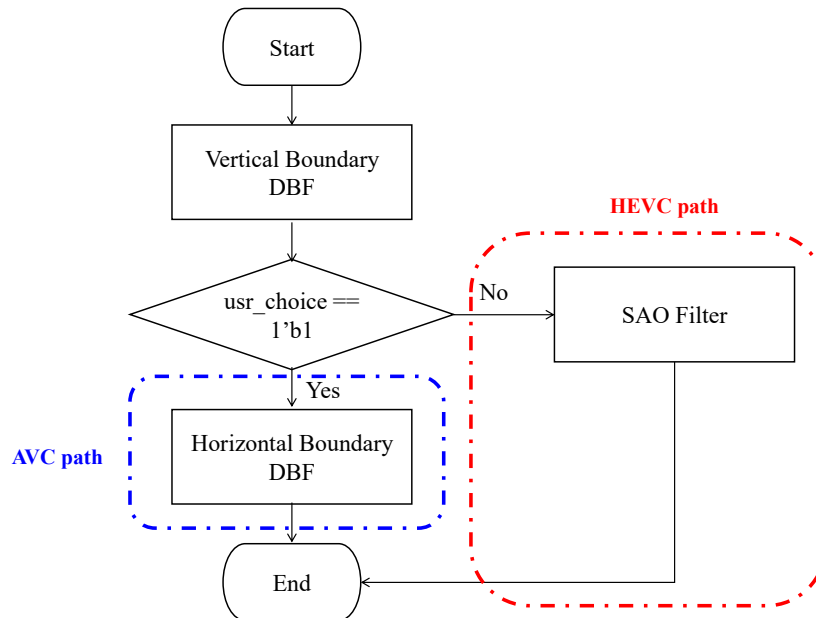


Figure 5. Components of combined in-loop filtering modules in HEVC and AVC. When `usr_choice` equals to 1, the AVC path is selected. In this case, the SAO filter is not involved. When `usr_choice` equals to 0, the HEVC path is selected, and the SAO filter is incorporated. The horizontal boundary DBF works similarly to the horizontal boundary DBF in individual AVC, while the SAO filter works similarly to the SAO filter in individual HEVC.

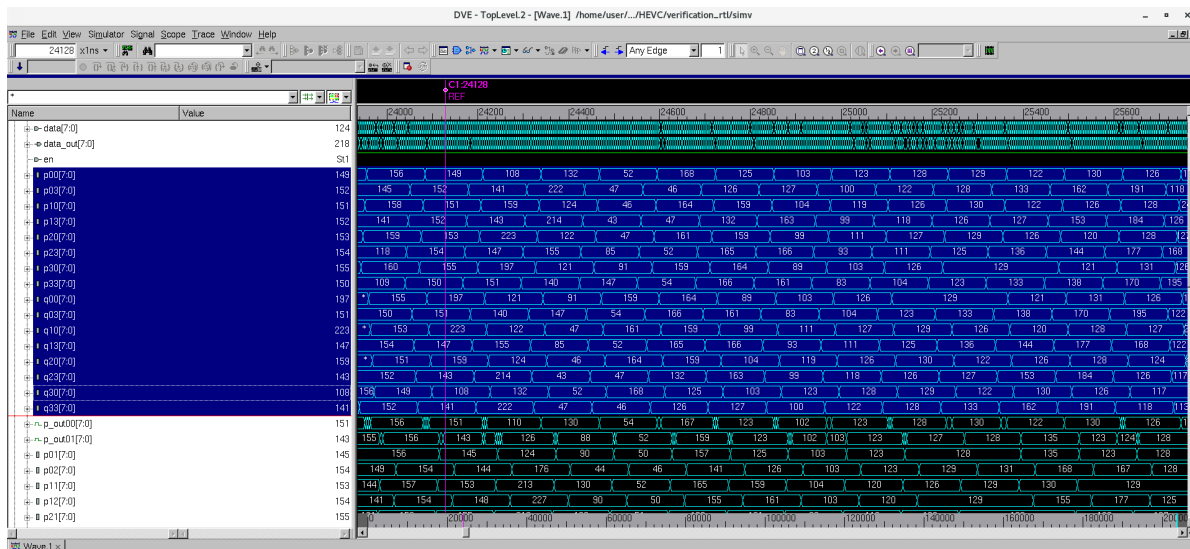


Figure 6. Fragment of result waveforms from the HEVC design.

4. RESULTS AND DISCUSSION

4.1 Logic Synthesis

Functional verifications have been performed for in-loop filtering modules of individual HEVC, individual AVC and the combined HEVC and AVC, according to the methodologies as stated in the previous projects [10], [11], [12]. Figure 6 shows the fragment of waveforms used to verify the conditions.

To achieve the goal of the project, the designs are implemented on ASIC platform. Therefore, front-end design, logic synthesis is required. In this work, the Design Compiler® from Synopsys EDA software tool is used to perform logic synthesis process. The RTL codes in Verilog format are converted into a logic representation format known as the gate-level netlist during the logic synthesis stage. To guarantee that the timing, area, power, and other requirements are satisfied, logic optimization are performed. All the steps in logic synthesis are the same for three different designs in this project as detailed in previous studies [13], [14]. Figure 7 shows the logic synthesis steps used to generate gate-level netlist.

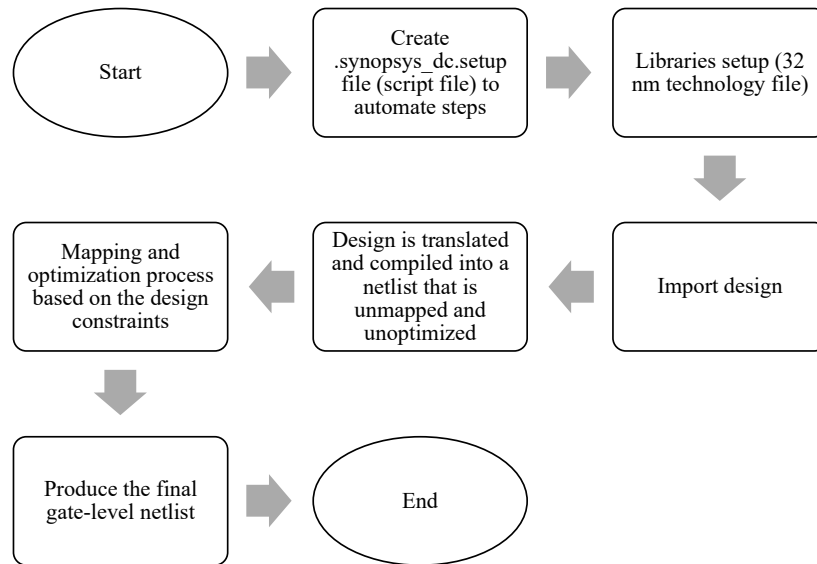


Figure 7. Logic synthesis steps used to generate gate-level netlist.

clock clk (rise edge)	8.00	8.00
clock network delay (ideal)	0.20	8.20
clock uncertainty	-0.01	8.19
filter_vertical/data_out_reg_7/_CLK (DFFARX1_LVT)	0.00	8.19 r
library setup time	-0.04	8.15
data required time		8.15

data required time		8.15
data arrival time		-6.68

slack (MET)		1.47

Figure 8. Fragment of timing report generated using Design Compiler® (without violations).

Before using the Design Compiler®, the `synopsys_dc.setup` file (script file) is created in order to automate the library setup, import design, compilation and design translation processes. When the Design Compiler® is initiated, the script file's instructions are executed by the compiler. The loading design comes after the libraries have been set up. The design (written in Verilog code) is automatically imported into the compiler after using the script file. The design is then translated and compiled into a netlist that is unmapped and unoptimized. The tool is then used to do the mapping and optimization process depending on the design restrictions. The tool uses logic gates based on the target technology libraries to execute the mapping process, resulting in a gate-level implementation of the design that fulfils design requirements. During the optimization process, the tool decreases the number of cells, area, power, and nets while maintaining the design's functionality. The tools also correct timing problems such as setup and hold violations.

After the mapping and optimization process, several analyses such as timing analysis and area analysis, must be completed. If the design does not fulfil the criteria, the process of re-constraint is necessary, followed by the mapping and optimization processes. To satisfy the criteria, these processes must essentially go through multiple iterations. Figure 8 shows the timing report without any violations with a clock period set at 8 ns or 125 MHz. Positive slack shows that the design can operate at a higher frequency. Although greater positive slack is suggested, area constraints and timing constraints are prioritized in all the designs. Additionally, the chip's area can be calculated during chip area analysis. The area optimization effort is configured to be high during the mapping and optimization procedure. As a result, the tool conducts area optimization in a few rounds while maintaining the chip's functionality and meeting the design restrictions. Finally, the design is saved as a netlist file for the physical design when it passes all analysis and fulfils all design requirements.

4.2 Physical Design Synthesis

The physical design comes next after the optimal netlist has been produced. The optimized gate-level netlist is implemented into the layout at this step. In this work, the IC Compiler® from Synopsys EDA software tool is used to perform physical design process. Physical design synthesis includes six steps which are data setup, design planning, placement, clock tree synthesis, routing, and GDSII output as shown in Figure 9.

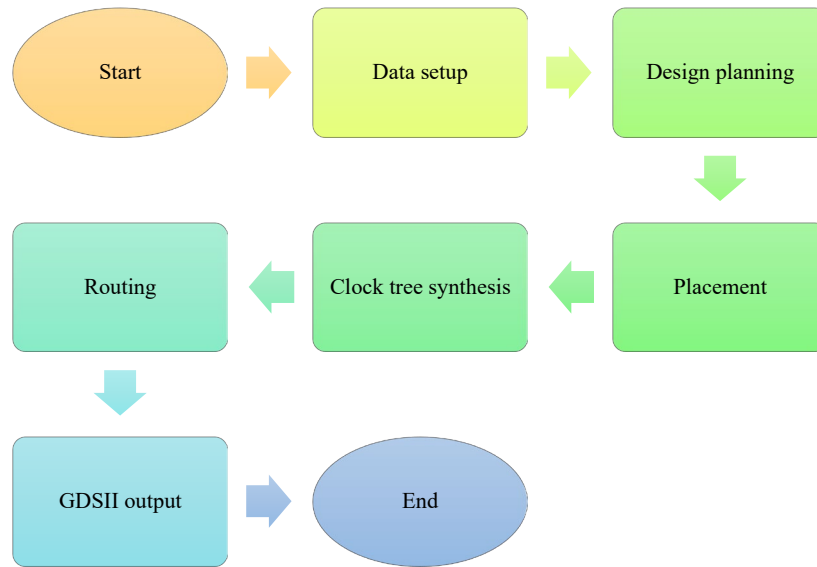


Figure 9. Physical design synthesis steps. The six steps are data setup, design planning, placement, clock tree synthesis, routing, and GDSII output.

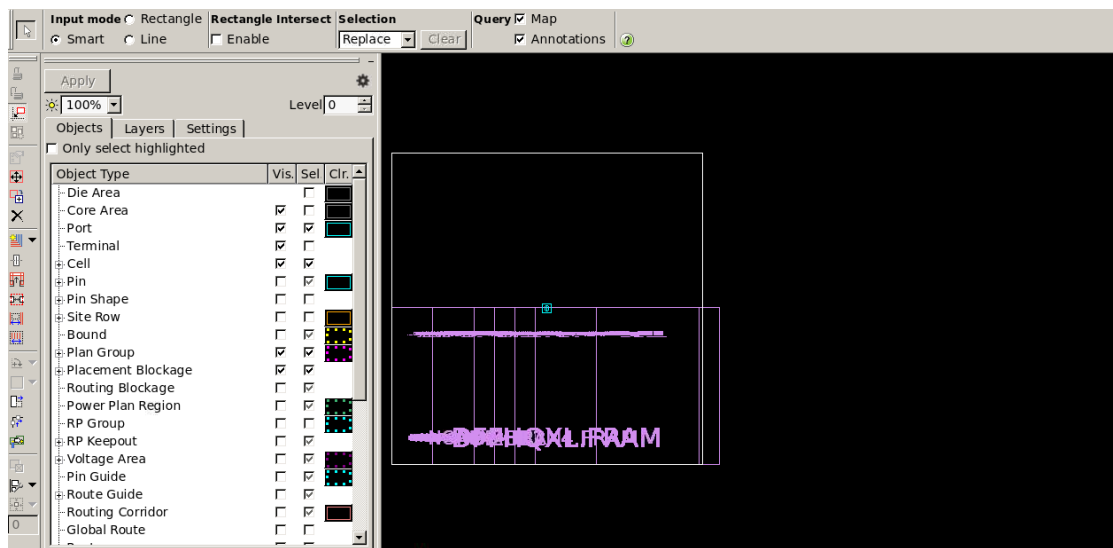


Figure 10. Initial top cell view of the design. It can be seen that all cells are overlapping together. The design is ready for floor planning stage.

4.2.1 Data Setup

Before starting to perform physical design, library setup is required. Synopsys' start-up file - .synopsys dc.setup, is used to configure technology libraries and library locations. After setting up the technology library, the next step is to create the design library (milkyway library in .mw format). The design is then imported into the library, and the physical design process is begun from this point.

4.2.2 Design Planning

After the libraries have been set up, the design floor plan may begin. The tool creates the first top cell view of the design once it is imported into the IC Compiler®. Based on the physical design libraries, the design in .ddc format is converted into the standard cell view. Figure 10 shows the initial top cell view of the design. In this figure, it can be seen that all cells are overlapping together. The design is then ready for floor planning stage. The targeted core utilization ratio is around 50%. The area for standard cells, macros, and blocking placement is defined by the core utilization ratio. The space available for regular cells, macros, and blockage placement is 50%. The remaining 50% of the space is used for routing. The tool will determine the most suitable design area. The tool will allocate 50% of the ideal area for placement and 50% of the ideal area for routing.

After floorplan initialization, virtual placement is performed to verify the created floorplan in terms of routing congestion and timing violations. The virtual placement report generated with IC Compiler® is shown in Figure 11. The total number of cells that violate the core region in this scenario is 0. This means that any cell can be placed in the core region. After the virtual placement, the chip summary is shown in Figure 12. The utilization rate of the core is 50.01%. Previously, the core utilization

value was set at 50%. The tool will compute the best core utilization ratio, but the result will be near to the figure entered by the user. As a result, the tool's estimated core utilization ratio is the best number that comes closest to the user-defined number. Furthermore, the chip size is enlarged due to the addition of gaps on four sides between the core region and the I/O ports. It is important to note that the size of the area is not completed at this stage. To connect the designs with power and ground, the rectilinear rings are created around the edges. After that, power straps are created to get additional power network robustness.

4.2.3 Placement

The design is then ready for placement. The tool is configured to perform the placement with high effort, including steps such as area optimization, routing optimization, and time optimization. Figure 13 shows the status of the placement. The placement is done with 148 Design Rule Check (DRC) violations. In the routing stage, the violations will be corrected.

4.2.4 Clock Tree Synthesis (CTS)

In CTS stage, the designs are clock tree synthesized (perform connection of clock signal paths to all clocking elements). In this case, the global routing is used by the tool. Figure 14 depicts the result of the global routing, indicating that the routing of the clock networks has been completed successfully. If there is any violation at this stage, the violation will be fixed in the routing stage.

4.2.5 Routing

The next step after CTS is routing. Routing is the process of performing interconnection between the signal pins of logic gates (cells). For initial routing, the tool uses a global routing technique. Figure 15 depicts a summary of the initial route. In this example, the total number of DRC violations is zero. If there are some DRC violations, detail routing and ECO routing are re-run to solve the DRC violations.

After ensuring all the DRC are fixed, the Layout versus Schematic (LVS) checking is done by using the tool as shown in Figure 16. The layout's electrical interconnections are identical to those in the gate-level netlist created via logic synthesis. When the "Total Must Joint Error" net value is 0, all wire nets declared in the netlist are appropriately routed. There are a few minor LVS violations. Floating port violations, shorted nets violations, and open-nets violations are the minor violations. Some of the ports in the design do not have any wire nets attached to them. As a result, the ports are classified as floating ports. These violations can be overlooked since they have no impact on the layout's operation.

```
*****
Report      : Virtual Flat Placement
Design      : top
Version     : R-2020.09-SP5
Date        : Thu Nov 23 14:14:36 2023
*****
Total wirelength: 36975355.76
Number of 100x100 tracks cell density regions: 22801
Number of low (< 10%) cell density regions: 0 (0.000%)
Number of high (> 200%) cell density regions: 0 (0.000%)
Maximum cell density: 58.60% (at 2049 489 2065 504)
Checking hard macro to hard macro overlaps...
Number of hard macro to hard macro overlaps: 0
Checking hard macro to std cell overlaps...
Number of hard macro to std cell overlaps: 0
Checking plan group to plan group overlaps...
Number of plan group to plan group overlaps: 0
Number of TL cells overlapping PG: 0
Number of cells violating core area: 0
Total number of cells violating plan group or core area: 0
Transferring Data to Milkyway ...
*** global placement done.
```

Figure 11. Fragment of virtual placement report. The total number of cells that violate the core region in this scenario is 0. This means that any cell can be placed in the core region.

```
Placement Optimization Complete
-----
Design  WNS: 0.00  TNS: 0.00  Number of Violating Paths: 0

Nets with DRC Violations: 148
Total moveable cell area: 1625695.4
Total fixed cell area: 0.0
Total physical cell area: 1625695.4
Core area: (20000 20000 1751432 1750520)
```

Figure 13. Status of placement report. The placement is done with 148 Design Rule Check (DRC) violations. In the routing stage, the violations will be corrected.

```
*****
Report      : Chip Summary
Design      : top
Version     : R-2020.09-SP5
Date        : Thu Nov 23 14:14:48 2023
*****
Std cell utilization: 50.01% (10432319/(20860173-0))
(Non-fixed + Fixed)
Std cell utilization: 50.01% (10432319/(20860173-0))
(Non-fixed only)
Chip area:      20860173 sites, bbox (20.00 20.00 2322.65 2322.34) um
Std cell area:  10432319 sites, (non-fixed:10432319 fixed:0)
                694115 cells, (non-fixed:694115 fixed:0)
Macro cell area: 0 sites
                0 cells
Placement blockages: 0 sites, (excluding fixed std cells)
                    0 sites, (include fixed std cells & chimney area)
                    0 sites, (complete p/g net blockages)
Routing blockages:  0 sites, (partial p/g net blockages)
                    0 sites, (routing blockages and signal pre-route)
Lib cell count:    65
Avg. std cell width: 1.99 um
Site array:        unit (width: 0.15 um, height: 1.67 um, rows: 1377)
Physical DB scale: 1000 db_unit = 1 um
```

Figure 12. Fragment of chip summary. The utilization rate of the core is 50.01%. The tool will compute the best core utilization ratio, but the result will be near to the figure entered by the user.

```
Total number of nets = 729894
0 open nets, of which 0 are frozen
Total number of excluded ports = 0 ports of 0 unplaced cells connected to 0 nets
                                0 ports without pins of 0 cells connected to 0 nets
                                0 ports of 0 cover cells connected to 0 non-pg nets

Total number of DRCs = 0
Total number of antenna violations = antenna checking not active
Total number of voltage-area violations = no voltage-areas defined
Topology ECO iteration 1 ended with 0 qualifying violations.
Updating the database ...
Information: RC extraction has been freed. (PSYN-503)
Router separate process finished successfully.
1
save_mv_cel -as clock_opt
```

Figure 14. Results of clock net routing (Global Routing) after CTS process, indicating that the routing of the clock networks has been completed successfully.

4.2.6 GDSII Output

In physical design, GDSII output is the final, completely verified, industry-standard file format that contains the full layout of a design. Several verifications are done before the GDSII file is streamed out. The final timing report is shown in Figure 17. Based on the report, the timing slack is improved compared to the previous report (Figure 8). The CTS helps to improve the timing slack. The design will not suffer from any timing issues. Lastly, the GDSII file is streamed out by using the tool. The status of the stream out process is shown in Figure 18. The GDSII file is then used for chip fabrication.

4.3 Synthesis Result and Throughput Estimation

To achieve the second objective of the project which is to evaluate the implementation cost and the performance of the individual and unified AVC and HEVC in-loop filters, the synthesis results such as total cell area, power consumption, and clock cycles are collected and discussed. In the implementation's hardware, the three circuits are described in the Verilog HDL. Furthermore, 32 nm technology library is used to construct the final implementations of the three designs. The designs all have an optimum frequency of 125 MHz, but the individual design can obtain higher frequency. The implementation results of the three designs employing the 32 nm technology library are shown in Table 1.

According to Table 1, the total cell area of Individual HEVC is 1.26 times more than the total cell area of Individual AVC. Individual AVC solely consists of DBF, whereas Individual HEVC consists of DBF plus SAO. The same reasoning applies to overall power. Individual HEVC consumes 1.26 times more power than Individual AVC consumes. This is owing to the fact that a smaller area consumes less power. However, the clock cycles to execute the Individual AVC is 1.26 times slower than the Individual HEVC. This is because Individual AVC consists of two types of DBF in sequential order. Since the two types of DBF are executed in sequential order, therefore Individual AVC needs 1.26 times more clock cycles to run. The clock cycles for each design in Table 1 are used to estimate the throughput for a 1920×1080 resolution.

```
Total number of nets = 729894
0 open nets, of which 0 are frozen
Total number of excluded ports = 0 ports of 0 unplaced cells connected to 0 nets
                                0 ports without pins of 0 cells connected to 0 nets
                                0 ports of 0 cover cells connected to 0 non-pg nets

Total number of DRCs = 0
Total number of antenna violations = antenna checking not active
Total number of voltage-area violations = no voltage-areas defined
Topology ECO iteration 1 ended with 0 qualifying violations.
Updating the database ...
Information: RC extraction has been freed. (PSYN-503)
Router separate process finished successfully.
ROPT: Initial Route Done Thu Nov 23 17:49:23 2023
```

Figure 15. Summary of the initial route. In this example, the total number of DRC violations is zero.

```
-- LVS START : --
Total area error in layer 0 is 0. Elapsed = 0:00:15, CPU = 0:00:08
Total area error in layer 1 is 0. Elapsed = 0:00:36, CPU = 0:00:29
Total area error in layer 2 is 0. Elapsed = 0:00:58, CPU = 0:00:49
Total area error in layer 3 is 0. Elapsed = 0:01:09, CPU = 0:00:59
Total area error in layer 4 is 0. Elapsed = 0:01:13, CPU = 0:01:02
Total area error in layer 5 is 0. Elapsed = 0:01:14, CPU = 0:01:04
Total area error in layer 6 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 7 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 8 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 9 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 10 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 11 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 12 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 13 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 14 is 0. Elapsed = 0:01:15, CPU = 0:01:04
Total area error in layer 15 is 0. Elapsed = 0:01:15, CPU = 0:01:04
** Total SHORT Nets are 0.
** Total OPEN Nets are 0.
** Total Electrical Equivalent Error are 0.
** Total Must Joint Error are 0.
-- LVS END : --
```

Figure 16. Results of LVS. The layout's electrical interconnections are identical to those in the gate-level netlist created via logic synthesis. When the “Total Must Joint Error” net value is 0, all wire nets declared in the netlist are appropriately routed.

clock clk (rise edge)	8.00	8.00
clock network delay (propagated)	0.49	8.49
vertical_boundary_DF/data_out_reg_7/_CLK (DFFARX1_LVT)		
library setup time	0.00	8.49
data required time	-0.04	8.45

data required time		8.45
data arrival time		-5.82

slack (MET)		2.63

Figure 17. Final timing report. The timing slack is improved compared to the previous report (1.47 ns previously, versus 2.63 ns now). Larger positive slack value shows that the design can operate at a higher frequency.

```
====> TOTAL CELLS OUTPUT: 1 <====
Outputting Contact $SVIA125Q_C
Outputting Contact $SVIA12BAR_C
Outputting Contact $SVIA12BAR
Outputting Contact $SVIA235Q_C
Outputting Contact $SVIA23BAR_C
Outputting Contact $SVIA23BAR
Outputting Contact $SVIA345Q_C
Outputting Contact $SVIA34BAR_C
Outputting Contact $SVIA455Q_C
Outputting Contact $SVIA45LG_C
Outputting Contact $SVIA565Q_C
Outputting Contact $SVIA675Q_C
Outputting Contact $SVIA785Q_C
Outputting Contact $SVIA89_C
Outputting Contact $SVIA125Q_120_120_1_2
Outputting Contact $SVIA235Q_120_120_1_2
Outputting Contact $SVIA125Q_C_120_120_1_2
Outputting Contact $SVIA235Q_120_120_2_1
Outputting Contact $SVIA785Q_C_120_120_2_1
Outputting Contact $SVIA785Q_120_120_2_1
Outputting Contact $SVIA125Q_C_120_120_2_1
Outputting Contact $SVIA125Q_120_120_2_1
write_gds completed successfully!
1
```

Figure 18. Status of stream out process. The report indicates a successful GDSII stream-out.

As in Table 1, the combined in-loop filter design achieves a 20% lower of the total cell area, indicating that the total resources utilized by the two individual in-loop filters is reduced in combined in-loop filters. Furthermore, the combined in-loop filter's total power consumption is 212 mW, which is 23% less than the total power consumed by the total of two independent in-loop filters (274 mW). By consuming the power of only 1.38 times higher than the individual HEVC (153 mW), the combined in-loop filters can handle two video coding standards whereas individual HEVC only applies to one video coding standard. This demonstrates that the usage of the proposed combined in-loop filter design provides a gained advantage over the individual HEVC and AVC in terms of handling video coding standards.

Table 1. Comparison of cell area, total power, maximum frequency and clock cycles.

Design	Total Cell Area (μm^2)	Total Power (mW)	Maximum Frequency (MHz)	Clock Cycles for 64×64 pixels	Throughput (Resolution@fps)
Individual HEVC	2076909	153	167	50941	1920x1080@6.5
Individual AVC	1647467	121	200	64489	1920x1080@6.0
Total AVC + HEVC	3724376	274	-	-	-
Combined HEVC & AVC	2915236	212	125	57704	1920x1080@4.30
Total Reduction	22 %	23%	-	-	-

5. CONCLUSION

In this work, the implementation of individual and combined in-loop filters for HEVC and AVC video coding standards on ASIC platform have been successfully developed by exploiting algorithmic similarities between AVC and HEVC in-loop filters. The design flow (from RTL to layout) of the three different designs are successfully implemented. The logic synthesis stage is done without any violations (timing and constraints violations). All architectures can filter the 64×64 block in sequential and are able to operate at a frequency of 125 MHz. The algorithms of individual and combined in-loop filters for HEVC and AVC are all functionally verified. Thus, the first objective has been successfully met.

The second objective in this work is to evaluate the performance of the individual and unified AVC and HEVC in-loop filters. In this work, the 32 nm ASIC process technology and Synopsys EDA tools were utilized. The obtained resource utilization for individual HEVC, individual AVC, and unified AVC and HEVC in-loop filter designs are $2076909 \mu\text{m}^2$, $1647467 \mu\text{m}^2$, and $2915236 \mu\text{m}^2$, respectively. The power consumptions of these designs are 153 mW, 121 mW, and 212 mW, respectively. The unified design exhibited around 20% savings in the area as compared to the cumulative individual AVC and HEVC designs. It also only has a power consumption overhead of 0.38 with respect to the individual HEVC design. In a nutshell, the second objective has also been met. By the proposed shared algorithms of combined in-loop filters for HEVC and AVC, it could save the resource utilization rate to the 20% reduction ratio of all. Further, the power consumption of the combined design is also reduced by 23%.

However, due to constraints in time and resources, there are still opportunities for improvement. The achieved throughput does not support real-time processing at FHD resolution, mainly due to the use of single-port memory, which limits read/write operations to once per clock cycle. Future work should consider dual-port memory to enhance data access efficiency. Additionally, incorporating more than two in-loop filters and extending support to other video coding standards, especially the emerging Versatile Video Coding (VVC), will be a promising direction. The proposed unified architecture lays a scalable foundation that can be extended for integration with future video standards, enabling efficient hardware reuse and adaptability across evolving codec requirements.

ACKNOWLEDGEMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

- [1] M. S. Elbamby, C. Perfecto, M. Bennis and K. Doppler, Toward low-latency and ultra-reliable virtual reality, *IEEE Network*, 32, 2018, 78-84.
- [2] V. Sze, M. Budagavi and G. J. Sullivan, *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Switzerland: Springer International Publishing, 2014.
- [3] S. Baldev, K. K. Anumandla and R. Peesapati, Scalable wavefront parallel streaming deblocking filter hardware for HEVC decoder, *IEEE Transactions on Consumer Electronics*, 66, 2020, 41-50.
- [4] A. Singhadia, Md. Minhazuddin, M. Mamillapalli and I. Chakrabarti, A fast integrated deblocking filter and sample-adaptive-offset parameter estimation architecture for HEVC, *Microprocessors and Microsystems*, 85, 2021, 104317.
- [5] J. Lee and J. Jeong, Performance comparison of weak filtering in HEVC and VVC, *Electronics*, 9, 2020, 960.

- [6] P. Lakshmi Amruthavalli and P. Nalluri, A review on in-loop filters for HEVC and VVC video coding standards, *Proceedings of 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2022, 997-1001.
- [7] R. D. C. N. Palau, J. Goebel, D. Palomino, G. Correa, M. Porto and L. Agostini, Real-time and low-power HEVC deblocking filter architecture targeting 8K UHD @ 60fps Videos, *Journal of Integrated Circuits and Systems*, 15, 2020, 1-9.
- [8] S. Li, L. Huang, X. Xiong, D. Xu, X. Zhu and Y. Fan, An area-efficient deblocking filter architecture for multi-standard video codec, *Proceedings of IEEE 4th International Conference on Circuits and Systems (ICCS)*, Chengdu, China, 2022, 149-154.
- [9] P. Saravanan, B. S. Priyadarshini, P. V. Kanna and P. Vaishnavi, Hardware accelerator for dual standard deblocking filter, *Proceedings of 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID)*, Guwahati, India, 2021, 210-215.
- [10] P. R. Christopher and S. Sathasivam, Five-stage pipelined dual-edge deblocking filter architecture for H.265 video codec, *IEICE Electronics Express*, 16, 2019, 1-6.
- [11] C. M. Diniz, M. Shafique, F. V. Dalcin, S. Bampi and J. Henkel, A deblocking filter hardware architecture for the high efficiency video coding standard, *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, France, 2015, 1509-1514.
- [12] K. Shukla, B. Swamy and P. Rangababu, Area efficient dataflow hardware design of SAO filter for HEVC, *Proceedings of International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, Shillong, India, 2017, 16-21.
- [13] S. H. Loh, I. M. Tan and J. J. Sim, VLSI design course with commercial EDA tools to meet industry demand - from logic synthesis to physical design, *Proceedings of 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, 2021, 55-60.
- [14] S. H. Loh, Y. N. Koay, K. H. Yeap and J. J. Sim, Analysis of Logic synthesis results with various technology nodes for RISC-V design, *Proceedings of IEEE 14th International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, 2024, 100-105.