International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

Embedded Intra only Encoding Chain of H.264/AVC HD on MPSoC Technology

Nidhameddine Belhadj, Mohamed Ali Ben Ayed, Nouri Mamsoudi
Electronics and Information Technology Laboratory,
University of Sfax, ENIS Sfax, Tunisia

Zied Marrakchi, Habib Meherz
Laboratory of Computer Sciences LIP6
University of Pierre and Marie Curie
Paris, France

Abstract— **Multiprocessor System on Chip (MPSoC) is an interesting technology to minimize the encoding time required by recent video encoders like H.264/Advanced Video Coding. In this paper, a parallel algorithm for the High Definition (HD) H.264/AVC intra only encoding chain is proposed. This partitioning is implemented on MPSoC architecture in order to accelerate the encoding time of this part of encoder. The proposed parallel execution is based on a mixed partitioning. It combines slice level parallelism and the macro blocks line level parallelism. The proposed MPSoC architecture is developed using SoCLib platform. To discuss the performance evaluation of proposed parallelism, three MIPS32 processors are used. Based on 9 processors, the experimental results show a speed-up of about 86,2% in processing time. Moreover, the proposed solution requires a relatively low memory size which positively affects the final System on Chip (SoC) surface.**

Keywords— H.264/AVC; SoCLib; MPSoC; Intra prediction; Slice

I. INTRODUCTION

Today, the different areas of use are covered by embedded systems, such as, automobile, mobile telecommunication, multimedia, military, and health. The evolution of microelectronic technology makes these systems more powerful, robust, and efficient in order to make easy the quotidian life. Moreover, customer is more exigent and requires more solutions to resolve complex daily problems. These complex requirements are followed by an algorithmic complication and raise the processing time. To deal with this constraint, the new technology innovation allows designing complex and efficient chips for embedded systems which consist of multiple microprocessors (CPUs), communication interfaces (bus, NoC, Crossbar), and memories on the same chip like MPSoC solution [1-2]. MPSoC technology is usually used to execute complex applications which require an intensive processing performance implemented in a small ship with low power consumption. The H.264/AVC video encoder is a complex application which allows a better coding efficiency in terms of visual quality and rate compression compared to related norms [3-4]. This efficiency is accompanied by an algorithmic complexity. Consequently, this norm requires powerful platforms to obtain a real time processing. Several anterior works have shown the efficiency of using multiprocessor technology to accelerate the encoding time [5-6]. Therefore, various parallelism algorithms have been proposed in literature. Two kinds of parallelism approaches are distinguished. The first approach is Functional Level Parallelism (FLP) which consists of performing the different functions which form the encoder in on separated processors. The second approach is Data level parallelism (DLP) which consists in processing different data level on distinct processors where the same program is implemented in each one. To perform a parallel execution for the H.264 encoder, some approaches require respecting data dependencies among the different modules which form this video norm. Different studies have demonstrated the efficiency of using MPSoC architectures for the H.264/AVC [7]. This technique can deal with the constraints required for an embedded system solution such as real time execution, small circuit area, and low power consumption [1]. In this manuscript, a parallel algorithm is proposed for the intra prediction encoding chain of H.264/AVC for High Definition (HD) video resolution. The proposed partitioning consists in mixing two parallel approaches: slice level and macro blocks line parallelisms. The proposed approach is based on Macro Blocks Line data loading to reduce the encoding latency and the size of used memory. Moreover, the proposed solution profits from slice level parallelism to overcome data dependencies among the different modules of intra prediction encoding chain. The proposed parallel algorithm is validated with MPSoC architecture, using 720P HD resolution (1280x720 pixels/frame) as video test sequences. Consequently, this format requires more execution time than smaller formats to achieve real time processing. Therefore, the HD video requires more processing efficiency comparing to smaller images to solve its complexity. The proposed architecture is designed using SoCLib platform which is an open and free platform for virtual prototyping of MPSoC solutions. The MPSoC architecture requires maximum optimization technique for the size of used memory. The experimental results show an interesting time saving which requires the smallest size of used memory comparing to other parallelism approaches. Therefore, it respects the size of used memory constraint, which affects directly the area of final circuit.

This paper is outlined as follows; section II presents H.264/AVC video encoder norm, especially the intra prediction encoding chain of H.264/AVC. Section III discusses the parallel algorithms for H.264 proposed in anterior works. Section IV emphasizes the proposed parallel

processing for the intra prediction encoding chain. The outcomes of our MPSOC architecture is validated using various HD video sequences and comparing them to other parallel approach. This section will be enclosed by integrating more processor to the proposed system. Finally, a conclusion and perspectives are proposed in section V.

II. INTRA PREDICTOR ENCODING CHAIN OF H.264/AVC

H.264/AVC video norm is the outcome of cooperation between ITU-T Video Coding Experts Group (VCEG) and ISO/IEC JTC1 Moving Picture Experts Group (MPEG). This standard demonstrates interesting results in terms of coding efficiency and flexibility for various communication networks and different areas of uses. H.264/AVC is a video encoding standard which consists of spatial and temporal predictions [3-4]. Three profiles types are attributed for this norm which are baseline, main, and high profiles. This paper focuses in baseline profile which is characterized by using two frame types, I frames and P frames: I frames, where just the intra prediction process is performed and P frames where inter and intra predictions are applied. In this article, Just the intra predictor encoding chain is discussed which is an important part of H.264 norm. Therefore, only the I frames are used in this study. As illustrated in Figure 1, the intra prediction encoding chain process consists in splitting each frame which belongs to the input video into Macro Blocks (MB) where the size of each one is 16*16 pixels. 13 prediction modes are used for each MB luma and 4 modes for each MB chroma [8]. The best prediction mode is selected based on the comparison result between the source MB and predicted MB, this criterion is called minimum rate distortion. The frequently distortion metric used in almost all encoder is the Sum of Absolute Difference (SAD), due to thinking to its efficiency and simplicity [9]. Then, the source MB is subtracted to predicted MB to build the residual MB. This latter is encoded through Discrete Cosine Transform (DCT), quantization and entropy coding. The encoded MB is decoded through an inverse transform and quantization to form the decoded video. Each decoded macro blocks line (MBL) is filtered over the De-blocking Filter module to obtain the reconstructed video [10].

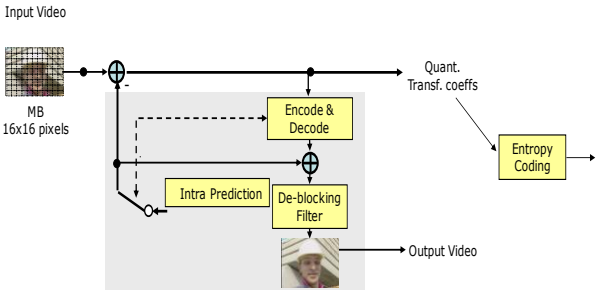

Fig. 1. Intra prediction encoding chain

In intra predictor encoding chain process, two data dependencies are distinguished [11]:

- Data dependency at the intra prediction block:
  - ✓ Each MB which forms the frame requires the last 16 pixels from neighboring MB on top to perform the intra prediction 16*16.
  - ✓ Each MB requires 20 pixels for intra prediction 4*4; the last 16 pixels from neighboring MB on top and the first 4 pixels of last 16 pixels from MB on top right.
  - ✓ Each MB requires the last 16 pixels from neighboring left MB for intra prediction 4x4 and 16x16 (neighborhood LEFT).
- Dependency at the de-blocking filter process:
  - ✓ Each MB processing requires the last 4 lines of MB pixels from top MB, and the last four columns of pixels from the left MB.

III. RELTATED WORKS FOR PARARLLEL PROCESSING OF H.264/AVC

Related works studies have demonstrated the efficiency of H.264/AVC performance on multiprocessor architectures, using different parallel approaches and platforms. Two kind of parallel approaches are distinguished, which are; Function Level Parallelism (FLP) and Data Level Parallelism (DLP). Zhibin et al [13] and Zrida et al [12] proposed various parallel algorithms of FLP for the H.264 encoder. The execution time of each module forming the intra prediction encoding chain is not similar from one to another. Consequently, some processors complete their processing before others and remain in a standby state. This blockage is due to decisional dependency among different tasks. In fact, this parallelism accelerates the encoding time, but this acceleration is not optimal. The second kind of partitioning is Data Level Parallelism (DLP). It consists in dividing the inputted data into different data elements which can be MB, slice, frame, and GOP (Group of picture). Thus, each task performs the same instructions but with different data elements. Wang et al [14] proposed a Macro Block Level Parallelism which consists in assigning each MB on separated CPU. However, MBLP requires vertical and horizontal neighboring pixels to respect the data dependencies constraints in intra prediction (16x16 and 4x4) and filter modules. This constraint affects negatively the processing time since the processors have to exchange neighboring pixels. Sankaraiah et al [15] presented a GOP Level Parallelism (GLP) which consists in assigning each GOP on distinct CPU. However, its big data structure raises the data loading which decreases the final execution time. Moreover, this partitioning requires a big size of used memory to host a GOP. Therefore, this solution cannot respect the area of final circuit constraint strongly imposed by MPSoC technology. Belhadj et al [16] proposed a frame level parallelism (FLP). It consists in processing each frame separately on a corresponding CPU. To reduce the data loading latency and the size of used memory constraints, this parallelism is based on Macro Blocks Line (MBL) loading. Rodrigues et al [17] proposed a partitioning based on dividing each frame into separated and independent slices. Consequently, each slice is treated by a distinct processor. However, the neighboring pixels are ignored. Therefore, it leads to a degradation of the decoded video and bit rate.

TABLE I. PARALLELISM ALGORITHMS FEATURES

Parallelism Solution	Complexity	Lossless partitioning	Size of used Memory
MBLP[14]	Complicated	YES	Low
SLP[17]	Simple	NO	Average
FLP[16]	Simple	YES	Low
GLP[15]	Simple	YES	Huge

Table 1 resumes the related works in terms of: algorithmic complexity to respect data dependency among processors, the quality of reconstructed video compared to an execution based on single CPU, and the size of used memory.

IV. THE PROPOSED PARALLEL ALGORITHM ON MPSOC ARCHITECTURE

In this paper a new MPSoC architecture is proposed. It hosts an efficient parallelism for the intra predictor encoding chain. This architecture is designed through SoCLib platform which is an open platform for virtual design of MPSoC architectures; it offers different Intellectual Properties (IPs) components developed in System C: CPUs, coprocessors, memories (Cache, RAM and ROM), MPSOC communication components like bus and Network On Chip (NoC) [19-20]. Based on table 1, our partitioning result's are compared to SLP and FLP. To ensure an efficient and appropriate comparison environment, we implemented SLP and FLP as described by their authors. These results are discussed in terms of gain in processing time, bit rate raise, reconstructed video quality and size of required memory. The resolution of video test sequences is 720p High Definition: 1280x720 pixels/frame. Therefore, each frame consists of 45 MBLs.

A. The proposed MPSoC architecture

The proposed parallelism SLP, and FLP and are implemented in an architecture composed of 2 co-procesors and three processors, as illustrated in Figure 2. The first coprocessor transmits the inputted video to be processed by processors. Therefore, it acts as a video source, like a camera. The second one reconstructs each decoded frame and builds the final bit-stream. Processors send and receive data to co-processors through FIFO. The intra prediction encoding chain was performed with three MIPS32 processors. The proposed MPSoC architecture is composed also by: a RAM memory, Rom memory, TTY terminal for displaying messages. The different components forming the proposed MPSoC architecture are connected directly to a VCI bus [21]. Each processor is linked directly to this bus through its L1 cache memory. Figure 3 shows the proposed data parallelism implemented in MPSoC architecture. The whole intra prediction encoding chain is implemented in each processor. The intra prediction encoding chain is composed of 5 mainly modules which are; intra prediction module, encoding the predicted video, decoding module to form the decoded video, filtering process and finally the entropy coding. This latter consists of two parts: Sequence Parameter Set (SPS) and Picture Parameter Set. SPS is processed before the prediction and whose role is to encode parameters of the video sequence. The PPS part consists in encoding the predicted slice. The proposed approach consists in splitting each frame forming the video into separated and uniformed slices where each one is composed by 15 MBLs. Then, performs each of them by the corresponding processor; the first one is executed in processor1, the second in processor2, and the third in processor3. The SPS is performed once during the whole process of intra prediction, and PPS is processed for each slice [16]. Consequently, the SPS process is hosted by processor1 before the intra prediction process and the PPS is integrated in all processors. The proposed parallelism is based on MBL data loading in order to decrease the size of used memory. Consequently, this partitioning takes advantage of the SLP to manage the data dependency constraint among different modules of the intra prediction encoding chain.

B. The temporal schedule processing

Figure 4 explains the processing schedule of proposed partitioning which is implemented on architecture based on 3 CPUs and 2 coprocessors. Therefore, the execution is as follow:

* T0: MIPS1 starts the process of SPS, MIPS2, MIPS3 and co-proc2 remain in a standby state. Co-proc1 sends the first MBL of first slice: Luma Y which its size is 1280*16 pixels through FIFO1 (F1) and two MBLs Chroma U and V which its size is 640*8*2 pixels via F2 to the first CPU MIPS1. I t is important to note that using two FIFOs for MBL Y, U and V give a faster transmission comparing to use single one. The use of one FIFO requires to configure it by six packets, 4 for MBL Y, 1 for U and 1 for V. Consequently, a loop is required to combine the MBL Y in CPU. Consequently, this detail increases transmission time.

*T1: Co-proc1 sends the first MBL of the second slice (MBL16) to MIPS2 through F7 and F8. MIPS1 receives MBL1.
*T2: Co-proc1 sends the first MBL of the third slice (MBL31) to MIPS3 through F13 and F14, MIPS2 receives MBL16 and MIPS1 starts the process of intra prediction for the first MB of MBL1.
*T3: Co-proc1 sends the second MBL of the first slice to MIPS1. MIPS2 starts the process of intra prediction and MIPS3 receives MBL31.
*T4 and T5: Co-proc1 sends the second MBL of the second slice to MIPS2 and the second MBL of third slice to MIPS3.
*T6: co-proc1 is in a standby state because F1, F2, F7, F8, F13 and F14 are locked waiting CPUs to receive data. MIPS1, MIPS2 and MIPS2 continue their processing for all MBs of MBL (the rest of 79 MBs).
*T0': After finishing each MBL processing, CPUs filter them using the Filter module. Therefore, MIPS1 starts filtering the first MBL.
*T1': MIPS1 sends the filtered MBL to the co-proc2 through F3 and F4. MIPS2 begins filtering the first MBL of second slice.
*T2': MIPS1 receives MBL2, MIPS2 sends the filtered MBL to co-proc2 through F9 and F10 and MIPS3 starts the MBL31 filtering. Co-proc2 receives the reconstructed MBL from MIPS1.

*T3': F1 and F2 are now unlocked. Thus, co-proc1 sends MBL3 to MIPS1. In the meantime, MIPS2 receives MBL17, Co-proc2 receives the reconstructed MBL from MIPS2.

* T4': Co-proc1 sends MBL18 to MIPS2. This latter starts the MBL17 processing. MIPS3 sends the filtered MBL to co-proc2.

These operations are repeated for all MBLs of each slice. Co-proc2 builds the processed slices to form the reconstructed frame

*T0'': The entropy coding is a Variable Length Coding (VLC). Consequently, a study is done to identify the maximum size of encoded data for each slice using various HD sequence videos. The obtained result is about 98 Kilo bytes which defines the sizes of F6, F12 and F18 for transmitting the encoded data from CPUs to co-proc2. F5, F11 and F17 send the number of encoded data to co-proc2, it varies from slice to another. Meanwhile, MIPS1 starts its PPS building.

*T1'': MIPS1 sends SPS and PPS to co-proc2, MIPS2 begins its PPS.

*T2'': MIPS2 sends its PPS to co-proc2 and this latter receives SPS and PPS from MIPS1. MIPS3 starts its PPS
*T3'' and T4'':MIPS2 and MIPS3 send its PPS to co-proc2. This latter receives respectively from MIPS2 and MIPS3 their PPS.

These processing steps are conducted for the whole video sequence.

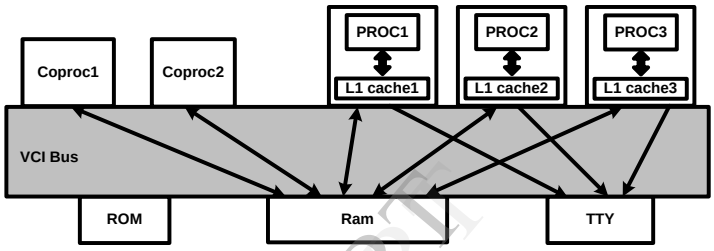

Fig. 2. The proposed MPSoC architecture

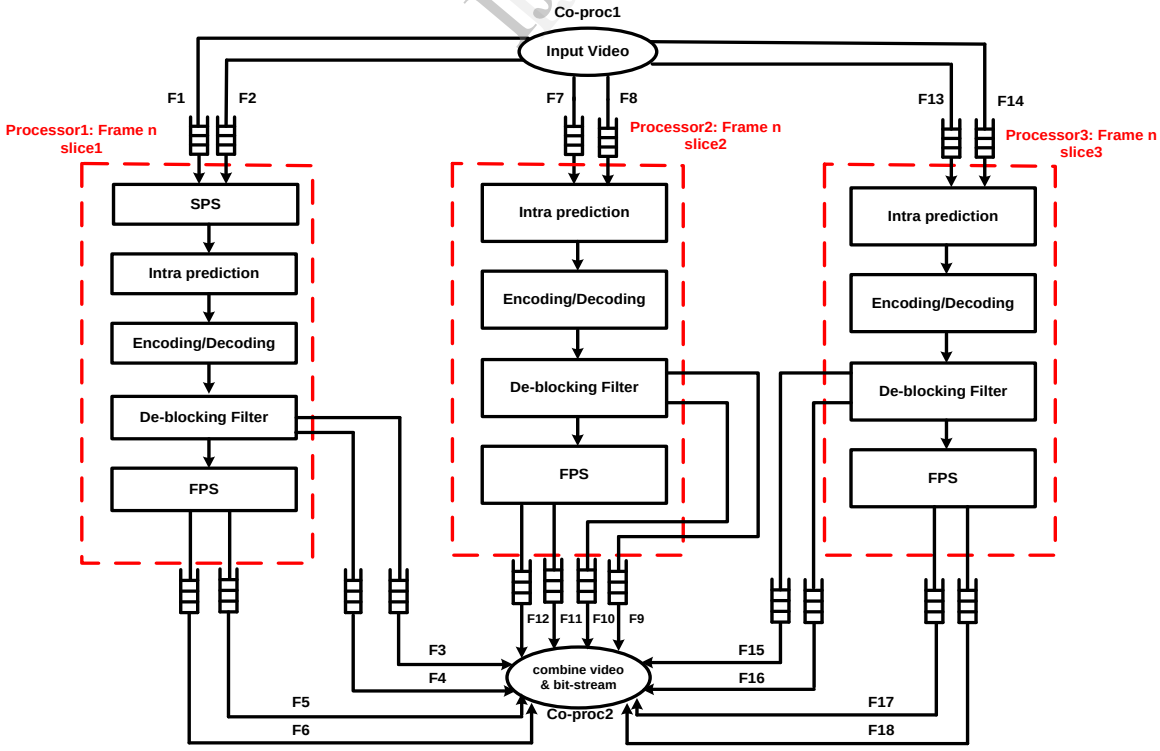

Fig. 3. The proposed parallel algorithm on MPSoC

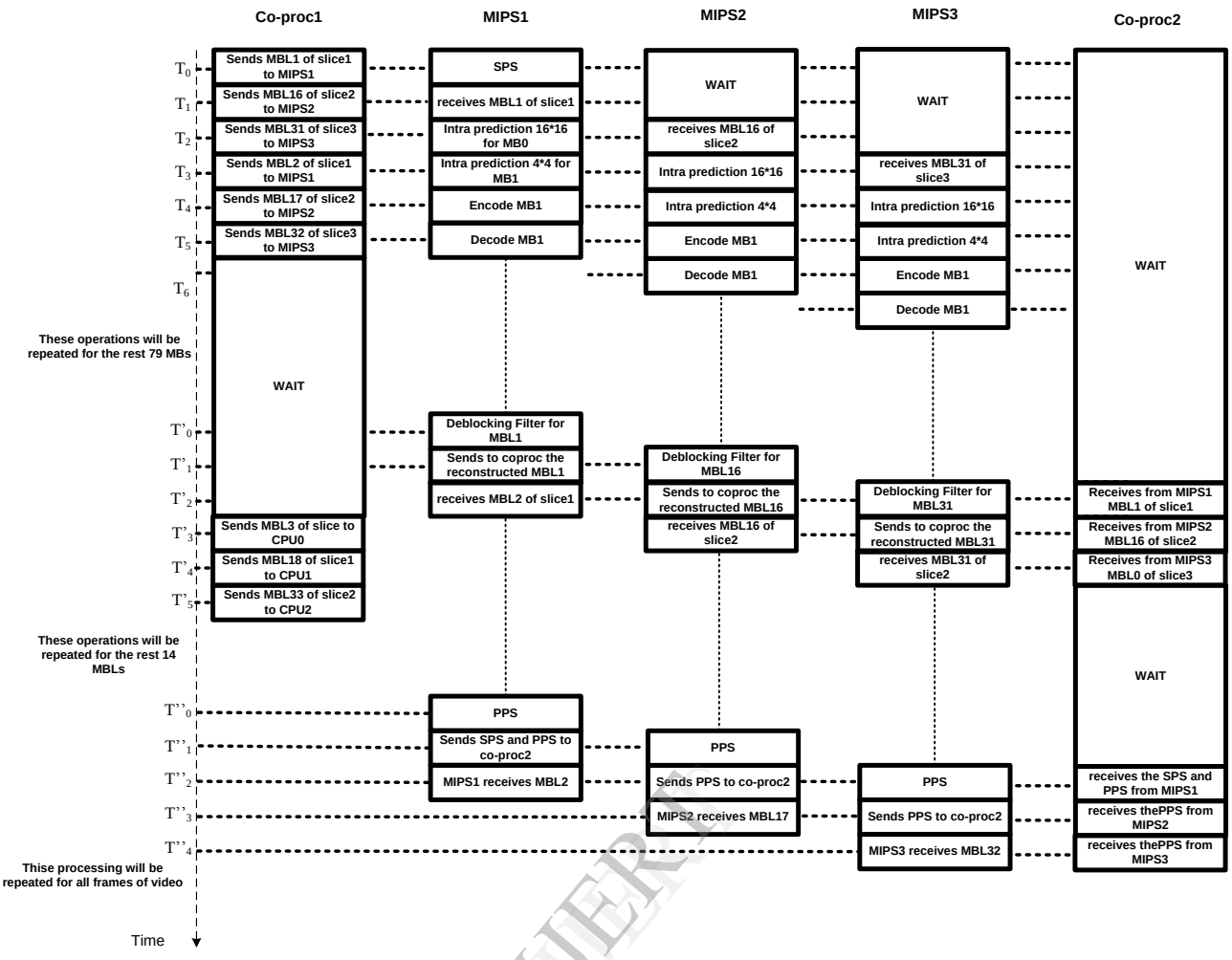

Fig. 4. temporal scheduling diagram

C. Results and discussion

The results of our partitioning, SLP and FLP are shown in table 2 using various video test sequences, and are compared to an execution based on single CPU in terms of:

- Gain in processing time between an architecture based on single CPU and the proposed MPSoC. It is expressed in %.
- ΔPSNR=video quality with 1 CPU - video quality with 3 CPU.
- Δbit rate = bit rate with 1 CPU − bit rate with 2 CPU.
- The size of required memory used for each partitioning.

As illustrated in table 2, SLP implementation is featured by a slight degradation of decoded video in terms of PSNR and a bit rate raise. These degradations are due to the neighboring pixels belonging to adjacent slices are ignored. Furthermore, this approach requires a large size of used memory due to the big size of slice.

FLP gave a faster execution time than SLP which is about 65,3%. This parallelism does not affect the quality of reconstructed video and bit rate. It keeps the same results obtained with sequential execution. FLP is based on MBL data loading. Consequently, this partitioning used a small size of required memory comparing to SLP.

The proposed parallelism approach gave the best speedup comparing to other partitioning. Furthermore it requires the less size of required memory thanks to the data loading is based on MBL. Moreover, in this case the PPS supports a slice not a frame like FLP. The results of the proposed approach seem to be very interesting thanks to the fact that the data dependency constraint is overcome using the Slice Level Parallelism, and decrease the data loading latency using the MBL load. However, a slight degradation is noted which is equal to SLP but without affecting subjectively the quality of the reconstructed video. Furthermore, the bit rate is noted by a slight raise which do not affects the data transmission price.

TABLE II. THE RESULTS OF IMPLMENTED PARALLELISM APPROACHES

Video Sequences HD	Parallelism approach	Gain in processing time	Δ PSNR			Δ Bit rate	Size of required Memory (Mb)
			Y	*U*	*V*		
720P_ParkJoy	FLP[10]	65,3	0	0	0	0	1,185
	SLP[11]	54	-0,02	-0,03	-0,04	0,26	3,5
	Our	65,7	-0,02	-0,03	-0,04	0,26	0,79
720P_Sunflower	FLP[10]	65,3	0	0	0	0	1,185
	SLP[11]	54	-0,02	-0,01	-0,03	0,3	3,5
	Our	65,7	-0,02	-0,01	-0,03	0,3	0,79
720P_OldTownCross	FLP[10]	65,3	0	0	0	0	1,185
	SLP[11]	54	-0,01	-0,03	-0,04	0,18	3,5
	Our	65,7	-0,01	-0,03	-0,04	0,18	0,79
720P_CrowdRun	FLP[10]	65,3	0	0	0	0	1,185
	SLP[11]	54	-0,03	-0,02	-0,03	0,88	3,5
	Our	65,7	-0,03	-0,02	-0,03	0,88	0,79

D. Multiple CPU implementations

More CPUs are integrated in the proposed MPSoC architecture to accelerate more and more the encoding process of the intra prediction chain. The proposed partitioning is validated through 3, 5, and 9 processors to respect the burden balance among CPUs. The table below highlights the resulting gain in processing time and the size of required memory for each architecture. The gain in time increases when more processors are integrated. However, this acceleration slows down due to the data loading latency which rises when more processors are added. The proposed partitioning produces a gain in the processing time for about 86,2% with 9 processors. Therefore, this number of CPUs allows processing 9 parallel slices. On the other hand, table 3 shows the size of required memory with respect to the number of CPUs. It is clear that the increase is linear with a constant rate.

TABLE III. PARALLELISM APPROACHES FEATURES

CPUs number	Speed-up (%)	Size of used memory (Mb)
1	-	0.27
3	65.7	0.79
5	78,4	1.34
9	86,2	2.51

V. CONCLUSION

In this paper, a new MPSoC architecture for the H.264/AVC intra prediction encoder was proposed. The results of the proposed parallel approach are compared to related works in terms of gain in processing time, quality of decoded video, bit rate raise, and size of required memory. Experimental results shows a speed-up at the order of 86,2% with a little size of used memory enabling a reduced area of SoC. These architectures are designed and explored through SoCLib platform. As future work we propose to add the inter prediction module and deal with its data dependency constraint. This study can be used for the latest High Efficiency Video Coding (HEVC) encoder since it adopts GOP, frame, and slice data structures are exactly the same as in AVC.

REFERENCES

[1] Niveditha. K, and Kalpana. K, "Reducing Communication Overhead for Streaming Application by Memory-Aware Scheduling on Mpsoc," International Journal of Engineering Research & Technology (IJERT), vol. 3, (3), March 2014, pp.301-303.

[2] Gayathri, V., and Arun Kumar.: 'MPSOC Design Approach of FPGA Based Controller For Induction Motor Drive', International Journal of Engineering Research & Technology (IJERT), vol. 3, (3), March 2014, pp. 1909–1914.

[3] Gayathri, V., and Arun Kumar.: 'MPSOC Design Approach of FPGA Based Controller For Induction Motor Drive', International Journal of Engineering Research & Technology (IJERT), vol. 3, (3), March 2014, pp. 1909–1914.

[4] Ammari, A. C., Jemai, A.: 'Multiprocessor platform-based design for multimedia', IET Computers and Digital Techniques., 2009, 3, (1), pp. 52–61.

[5] Krichene, H., Ammari, A.C., Jemai, A., Abid, M.: 'Performance/Complexity Analysis of a H264 Video Encoder', in Del Ser Lorente J.(E.D.): 'Recent Advances on Video Coding' (InTech, 2011, 1st edn.), pp.27-48.

[6] A.S Tushar, "Reducing the encoding time for H.264 baseline profile using parallel programming techniques". PHD Thesis, the University of Texas at Arlington, 2012.

[7] Yan, L., Renfa, L., Cheng, X., Fei, Y.: 'HW-SW Framework for Multimedia Applications on MPSoC: Practice and Experience', Journal of computers, 4, (3), 2009, pp.238-244.

[8] Bharathi.S.H, Nagabhushana Raju, K.: 'Verilog realization of Diagonal-Down-Left intra prediction for H.264 Video Encoder', International Journal of Engineering Research & Technology (IJERT), July 2012, 1, (5), pp. 1–13.

[9] Kim, J.H., Kim, B.G.: 'Efficient intra-mode decision algorithm for inter-frames in H.264/AVC video coding', IET Image Processing, 2011, 5, (3), pp. 286–295.

[10] M. Yang and N. Bourbakis, "A H.264/AVC intra-only coding (iAVC) and neural network based fast prediction mode decision," 22nd International Conference on Tools with Artificial Intelligence, 2010, pp. 57-60.

[11] N. Bahri, I. Werda, T. Grandpierre, M.A Ben Ayed, N. Masmoudi and M. Akil, "Optimizations for real-time implementation of H.264/AVC video encoder on DSP processor," International Review on Computers and Software, 2013, vol. 8, NO. 9, 2013, pp. 2026-2035.

[12] H. K. Zrida, A. C. Ammari, A. Jemai and M. Abid, "High Level Optimized Parallel Specification of a H.264/AVC Video Encoder," International Journal of Computing & Information Sciences, 2011, vol. 9, NO. 1, 2011, pp. 34–46.

[13] X. Zhibin, L. Stephen and B. Bevan, "A Fine-grained Parallel Implementation of a H.264/AVC Encoder on a 167-processor Computational Platform," Proc. Signals, Systems and Computers (ASILOMAR), 2011, pp. 2067-2071.

[14] Z. Wang, L. Liang, G. Yang, X. Zhang, J. Sun, D. Zhao and W. Gao, "A Novel Macro-Block Group Based AVS Coding Scheme for Many-Core Processor," Journal of Signal Processing Systems, October 2011, vol. 65, NO. 1, 2011, pp. 129-145.

[15] S. Sankaraiah, H.S. Lam, C. Eswaran and J. Abdullah, "GOP Level Parallelism on H.264 Video Encoder for Multicore Architecture," Proc. International conference on circuits, system and simulation IPCIST, 2011, pp. 127 - 132.

[16] N. Belhadj, N. Bahri, Z. Marrakchi, M. Ben Ayed and H. Mehrez, "Data Level parallelism for H.264/AVC baseline intra prediction chain on MPSoC," .Proc. 10th multi-conference on systems, signals and devices SSD2013, 2013.

[17] A.M. Rodrigues, "Software implementation of an MPEG-4 Part 10 (H.264/AVC) video encoder for embedded systems," Master, Technical University of Lisbon, 2010.

[18] Javaid, H., Shafique, M., Parameswaran, S., Henkel, J.: 'LowPower Adaptive Pipelined MPSoCs for Multimedia: An H.264 Video Encoder Case Study'. Proc. Design Automation Conference 2011, San Diego, California, USA, June 2011, pp. 1032-1037.

[19] J. Ye, Q. Tan, T. Li, B. Wu and Y. Meng, "Feature-Oriented Refactoring Proposal for Transaction Level Models in SoCLib," .Proc. IET. 2010 Forum on Specification & Design Languages (FDL 2010), 2010, pp. 22–27.SoClib Platform http://www.soclib.fr/trac/

[20] Q.L Zhang, M.Y. Yu, J.X. Wang, Y.Z. Zheng and F.C. Lai, "The Design of AMBA AHBNCI Wrapper," Proc. 5th International Conference on ASIC, 2003, pp. 438-442.