Multimed Tools Appl (2016) 75:11291–11310
DOI 10.1007/s11042-015-2854-4

Joint application-architeture design space exploration of multimedia applications on many-core platforms - an experimental analysis

Maryam Moghadas[1] · **Hossein Afshari**[2] · **Mahmoud Reza Hashemi**[1]

Received: 20 October 2014 / Revised: 6 July 2015 / Accepted: 3 August 2015 /
Published online: 13 August 2015

Abstract The design space of mapping multimedia applications on an architectural platform is complex and many parameters are needed to be considered in order to find the optimum mapping. Conventionally, architectural parameters are varied to find different design points and the application side parameters are considered fixed and unchanged, while multimedia applications are equipped with several configurations to control the complexity and quality of the output. In this paper, we experimentally investigate joint application-architecture design space exploration of multimedia applications on many-core platforms. The joint exploration is conducted on two state of the art video coding standards and STHorm many-core platform as the underlying architecture. In the first case study, MPEG4-SP decoder is mapped on STHorm with varying buffer size and variable number of Processing Elements (PEs). In the second case study, HEVC (High Efficiency Video Coding) decoder with variable Quantization Parameter (QP) is mapped on STHorm with variable PE number. The application is characterized with representative parameters on the basis of high level dataflow representation of the application. It is demonstrated that joint exploration of these parameters outperforms that of their separate exploration in terms of well-established combinatory metrics such as space-time product and energy-time product metrics while achieving performance constraints.

✉ Maryam Moghadas
mmoghadas@ut.ac.ir

Hossein Afshari
hossein.afshari@a3.epfl.ch

Mahmoud Reza Hashemi
rhashemi@ut.ac.ir

[1] Multimedia Processing Laboratory, School of Electrical and Computer Engineering College of Engineering, University of Tehran, Tehran, Iran

[2] EPFL Alumni, EPFL University, Lausanne, Switzerland

Keywords Design space exploration · Energy · Execution time · Memory · Video coding · Many-core platform

1 Introduction

Multimedia applications have recently found widespread use on personal computing platforms like tablets and smartphones. HD (High Definition) video games and multi-view coding are examples of new emerging multimedia applications in real life scenarios. Video encoders and decoders have considerably improved to support higher resolutions and more advanced video coding modes. The new video coding standards provide more compression ratio with similar or better quality. The complexity of these applications and huge amount of multimedia data has made this genre of applications so computationally intensive. Hence, they require considerable processing resources to fulfil their operation.

Personal computing platforms have evolved from single core to multicore and many-core platforms offering more computational power for resource hungry applications. Through distributed processing of the application on different cores, the total execution time can be reduced. STHorm is a new many-core co-processing fabric suitable for data-intensive applications [4] (This is the commercial name of the platform which was formerly known and referred to as P2012). This fabric is aimed for low power and area efficiency. This platform is suitable for execution of multimedia applications on portable and mobile devices. Other features of this platform are scalability, programmability and heterogeneity. In this platform, the cores are organized in different clusters with independent controllers and clock domains. Versatile programming models such as OpenCL and OpenMP are supported on this platform.

Mapping an application on to a many-core platform necessitates parallelizing or breaking up the code into functional blocks for parallel execution. To this aim, the application is better represented with a modular model. Dataflow representation is a widely used model for signal processing applications [15] . Graph representation of the application in this model facilitates the mapping process of an application on a many-core hardware platform. RVC (Reconfigurable Video Coding) has been proposed as a new video coding paradigm based on this model. RVC is standardized and customized for video coding applications [13]. In dataflow model representation, each node of the application graph is called an actor. Actors can have different implementations thus leading into different configuration of the application.

In multimedia application implementations on architectural platforms, designers are faced with alternative configurations in the design space that need to be efficiently explored. Different architectural parameters may exist that their changes impact the final design objective function for mapping purposes. The number of processing elements and different memory and interconnect configuration are examples of such architectural parameters. Multimedia applications are equipped with several configurable parameters and coding modes that can be utilised for different rate, distortion and complexity target values. As the applications and the underlying architectures are becoming more complex, more parameters can be considered for the mapping process.

The focus of this article is the joint application-architecture design space exploration for energy, time and memory efficient execution of a multimedia application on a many-core

platform. Two real case studies are conducted. The first is mapping of MPEG4-SP decoder on the STHorm platform. Buffer size is considered as the application variable parameter. The number of PEs is considered as the architectural variable parameter. For this case study the memory-time product is the metric of interest. The second case study is mapping of HEVC decoder with QP as the variable application parameter and the number of PEs as its architectural variable parameter. Energy-Delay Product (EDP) is the hybrid objective function and the rate, distortion and execution times are considered as constraints. We show that by joint exploration of the application and architecture parameters in the design space more efficient mappings can be found in terms of design objectives.

The structure of the paper is as follows. Section 2 reviews the state-of-the-art techniques used for design space exploration. Section 3 explains the joint design space exploration technique. The simulation tool-chain and environment is elaborated in Section 4. Design space exploration of the MPEG4-SP decoder and HEVC decoder with related simulation results are provided in Sections 5 and 6, respectively. In Section 7 the paper is concluded and future works are outlined.

2 State of the art

Joint exploration of different dimensions of design space is not a new concept. In most research works the architecture dimension is explored for different design points. This dimension can be extended to lower level circuit properties as well. The combination of the circuit level and higher level architecture characteristics is explored in [2]. Joint exploration of other dimensions has been the point of interest for researchers as well.

The algorithm and architecture co-exploration paradigm for video coding applications has been discussed in [12]. In this work the co-exploration paradigm is investigated and its future trends are outlined with no experimental result. Joint exploration of application and architecture for the design of the motion estimation module of H.264/AVC encoder is investigated in [8]. In this work, a specific VLSI implementation of motion estimation with application-architecture co-design focus is presented rather than joint exploration of high level parameters of application and architecture. Profiling based approaches have been proposed for the joint exploration of hardware and software in [10], designing optimal memory hierarchy via computational information and data transfer knowledge of the application. In this work, joint exploration is conducted with memory and processor instruction set parameters. Processor instruction set is the effective parameter that affects the software part of the system. No parameter tuning from the application side is considered in this work.

The joint exploration of the design space is not limited to the application and architecture. The design space is enhanced by different mapping and partitioning configuration of the application as in [11]. A joint optimization of network on chip (NoC) configuration and the application mapping by using evolutionary algorithm is provided in [11]. In this work the mapping of the application actors is jointly encoded with a NoC configuration on a chromosome structure. Another variable aspect in the design space is different network configuration among embedded devices. In [5] the design space is extended to different configurations of networks in addition to the architectural design space dimensions for network of embedded systems.

To sum up, we have investigated research works that have focused on the joint exploration of the design space from different aspects. These dimensions are shown in Fig. 1. Each dimension itself comprises of different variable parameters.

figure 2 shows the different levels of the classical design space. Co-design of application and architecture which is discussed in the literature [8] has been shown as well.

In this work, we have concentrated in the domain of multimedia applications which are equipped with high level user configurable parameters. According to Fig. 2 our focus is on the peak of the cone in the specification part. Our approach in exploration is shown in Fig. 3 which reflects the difference compared to Fig. 2 as well. All combinations of application and architecture parameters are jointly explored to find the best design point. Moreover, the considered parameters at the specification level are configurable by the user. Hence, there is no need for a fundamental change by changing the parameters.

Overall, in this paper, a new look and direction for joint design space exploration based on high level application and architecture parameters for mapping of multimedia applications on a many-core platform is presented. The feasibility of the proposed direction has been shown with two real case studies. The effect of joint architecture and application exploration on energy consumption, memory usage and the execution time are investigated. It has been shown that joint exploration outperforms the separate one in terms of defined combinatory metrics.

3 Joint optimization model and formulation

This section is devoted to joint optimization model and formulation. The application and architecture models are introduced in subsections 3.1 and 3.2. The extended design space is elaborated in subsection 3.3. In subsection 3.4, the joint optimization problem is formulated. The methodology for solving the joint optimization problem is presented in subsection 3.5.

3.1 Application model

RVC is used to model an application at the highest abstract level. RVC-CAL is a standardized language used for modular representation of multimedia applications [13]. Native Programming Model (NPM) is used on top of the architecture for developing parallel applications.

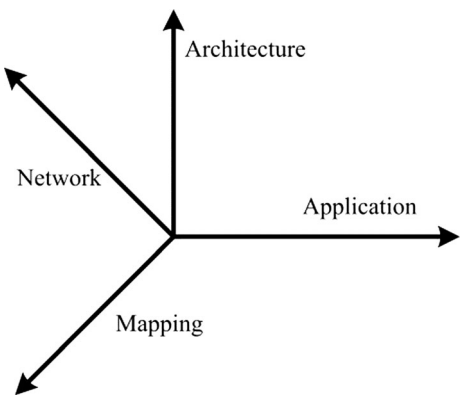

Fig. 1 Design space dimensions

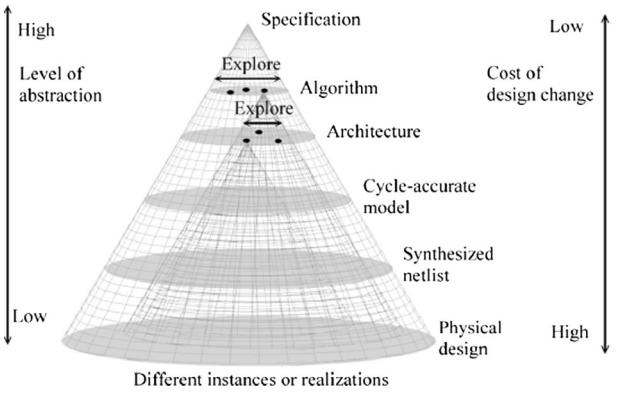

Fig. 2 Classical design space levels [8]

Figure 4 shows the overall structure of CAL dataflow diagram.

In the CAL dataflow, the actors are connected by FIFO channels. Each actor consists of actions that are controlled and executed by a finite-state machine (FSM) controller. This modular representation enables different implementation and configuration for each actor without affecting other actors. In the presented model, each actor is characterized by a set of representative parameters. An application is a combination of actors. Each actor is represented by its set of algorithmic and structural parameters. The nodes in the dataflow graph of the application are represented by the A vector (1).

$$\begin{aligned} A &: (\text{structural param.},\ \text{application specific param.}) \\ A_{ai} &= \qquad\qquad (a_s,\ a_{as}) \end{aligned} \tag{1}$$

Hence, the whole application is characterized by the union of all of its constituent actors' parameters in (2).

$$A_{tot} = \underset{i}{\cup} A_{ai} \tag{2}$$

Critical path, number of actors, FIFO size and computational load are examples of structural parameters. For the case of a video coding application, parameters such as search range, number of reference frames and quantization parameter are sample algorithmic parameters.

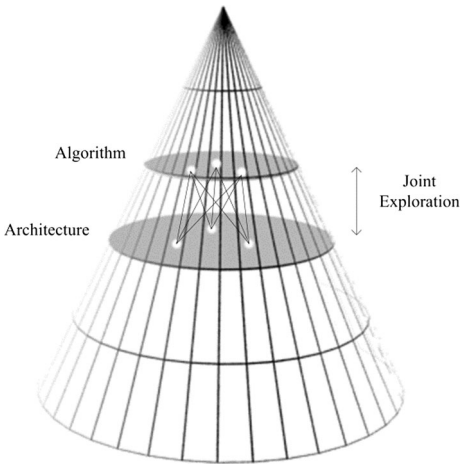

Fig. 3 Proposed joint exploration

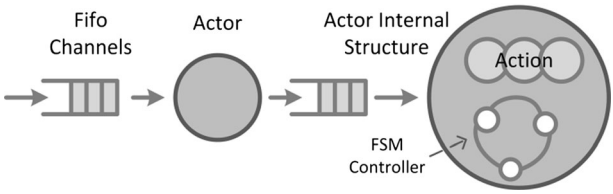

Fig. 4 CAL dataflow network

3.2 Architecture model

The underlying architecture of interest is a many-core platform. The many-core platform is modelled as a two level hierarchy of processing elements organized in clusters of cores. Each cluster controller and the clusters are connected asynchronously via NoC. The architecture is characterized with a set of parameters for the processing cores, memories and interconnects. Number of cores, their operating frequency and heterogeneity are examples of architectural parameters. This representation is shown as in (3).

$$P_{tot} = \underset{j}{\cup} P_j \tag{3}$$

Figure 5 shows the two level hierarchy of a many-core platform.

As can be noticed in Fig. 5, clusters are connected via NoC routers at inter-cluster level. Within each cluster, cores and memories reside. Memories at the first level are specific to each core (L1). At the second level (L2), the memory is shared that can be accessed by all the cores in the cluster. The cluster is connected to NoC router via Network Interface (NI).

Depending on the implementation of the platform, different number of processing elements may reside in a cluster that can be GPP (General Purpose Processer), DSP, ASIC or a hardware block (HWPU) in general. In this way, the cluster is heterogeneous. When the processing elements are identical in the cluster, a homogeneous cluster is formed. Many-core platforms can be designed just with one cluster.

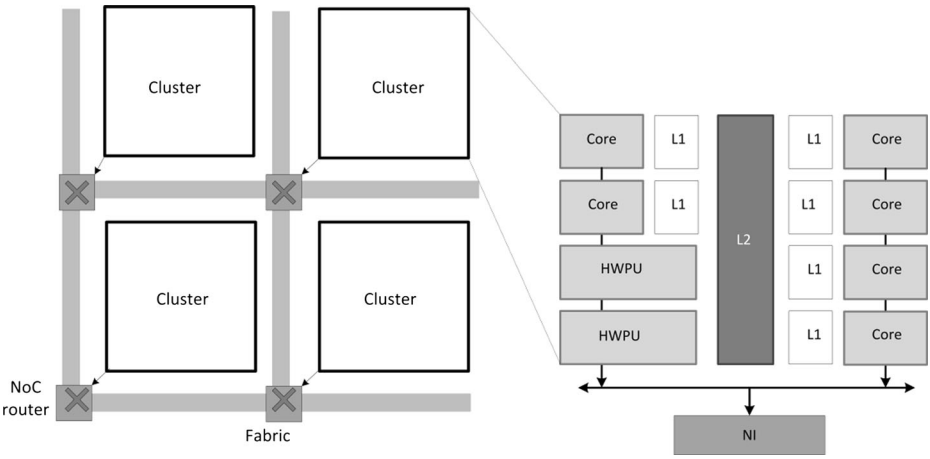

Fig. 5 High level inter-cluster and intra-cluster model of a many-core platform [3]

3.3 Extended design space

In our model, each design point is represented as $D_m=(P_{tot}, A_{tot})$. m is the cardinality of the design space. Corresponding to each design point is a vector of constraints and objectives which form the utility space shown by C_n. n is the cardinality of the utility space. The relation between C_n and D_m is written as $C_n=(f_1(D_m), \ldots, f_n(D_m))$ $f_i(.)$ is a generic function for evaluating the design points and can be a closed form analytical model or obtained by simulations. By adding the application dimension to design space, new introduced points have the potential of being better candidates for mapping. This fact can be observed intuitively since application and architecture domains are correlated domains. Hence, we expect that their joint exploration of them would result into better design points. If we show the conventional utility space as C_n^{orig} and the extended one as C_n^{ext}, we can deduce that $C_n^{ext} \geq C_n^{orig}$. However, the conventional space is a subset of the new space and in the worst case the results from conventional space are retained. The penalty is the evaluation of more points in the extended design space.

3.4 Joint optimization formulation

The aim is to utilize the application side parameters alongside their architectural counterparts to minimize the design metrics while achieving design constraints.

The customary design flow consists of two separate and consecutive optimizations steps, one for application and the latter for architecture. The application parameters are first configured for a given design characteristics. Obtained parameters from the first optimization step are given to the designer for an energy and performance efficient hardware mapping at the second stage of optimization. In this step, architecture parameters are the only variable parameters. In the separate optimization case, the optimization procedure is formulated as (4) for the first step (i.e. the application part). $f(.)$ is a generic objective that is the function of application parameters. $\overrightarrow{x}$ is a vector of constraints.

$$\begin{cases} Min\ f\ (.) \\ \overrightarrow{x} < \overrightarrow{x}_{target} \\ (A_{tot}) = argmin(f) \end{cases} \tag{4}$$

The architecture side optimization is written as in (5). $g(.)$ is a generic architecture objective which is a function of the resulting application parameter from the previous step as well. $\overrightarrow{y}$ is the vector of constraints on the architecture side.

$$\begin{cases} Min\ g(., A_{tot}) \\ \overrightarrow{y} < \overrightarrow{y}_{target} \\ (P_{tot}) = argmin(g) \end{cases} \tag{5}$$

In our proposed approach, the two separate steps are joined and merged in one optimization problem. This optimization is formulated as in (6). The final objective is a function of both application side and architecture side parameters. This function is generically shown by h. $\overrightarrow{z}$ is the superposition of the two vectors $\overrightarrow{x}$ and $\overrightarrow{y}$.

$$\begin{cases} Min\ h(.,\ A,\ P) \\ \overrightarrow{z} < \overrightarrow{z}_{target} \\ (A_{tot},\ P_{tot}) = argmin(h) \end{cases} \tag{6}$$

It is shown via simulation that the joint optimization outperforms the two separate optimization case.

3.5 Methodology

The proposed design flow is shown in Fig. 6. The flow begins with separate modelling of architecture and application. The utility functions are separately modelled as well. From application and architecture models, the parameters that are effective on the utility functions and constraints are extracted. Concerning multimedia applications and specifically video coding applications, in general, investigation of the related standards helps identify and extract the parameters as the first step of parameter extraction process. From the different parameters mentioned by the standard, those parameters that are effective on objective functions and constraints are selected at the second step. Since considering all the effective parameters in conjunction with architecture parameters yields a huge design space, for the purpose of proof of concept, at the final stage of parameter extraction process, we choose a parameter which is simple and easy to configure. After the extraction of the parameters, the feasible range of each parameter is identified. In the next step, all combinations of parameters are considered for evaluation of the objective function. The same constraints are applied for each set of parameters. Hence, the points in the design space are identified. In the final step the design space is explored by an exploration strategy.

4 Simulation setup

In this section, we describe the simulation setup. First we introduce the STHorm platform in subsection 4.1. Simulation toolchain is described in subsection 4.2.

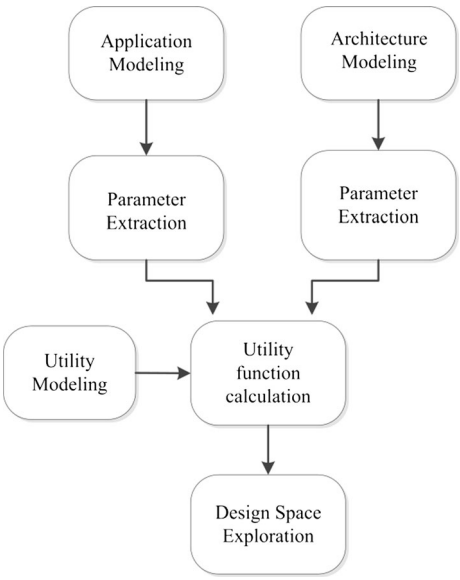

Fig. 6 Design flow

4.1 STHorm platform

For simulation purposes, STHorm many-core platform has been chosen which is a co-processing fabric targeted for visual computing and multimedia applications [4]. It is designed by STMicroelectronics. Figure 7 shows the internal architecture of this platform. The underlying fabric of this platform is organized into clusters or tiles on which PEs reside. Each cluster has a different clock domain controller and is connected asynchronously to other clusters. In each cluster, there is an internal shared memory which all cores can access it. Cluster controller manages and controls all the signaling and communications among the blocks within the cluster. Each cluster contains 16 cores which are internally connected to other blocks via memory-mapped interconnect. All the cores are homogeneous.

There is also an external memory outside the clusters, which is connected to the clusters via NoC routers. This platform needs to be connected to a host such as an external CPU.

In Fig. 7, two clusters are shown which are connected by NoC routers. For the purpose of our experiments we have used just one cluster.

4.2 Simulation toolchain

The tool-chain of the simulations is shown in Fig. 8.

The application is implemented in RVC-CAL which enables parallel implementation on a many-core platform due to its modular structure. The ORCC compiler [1] is used to compile the

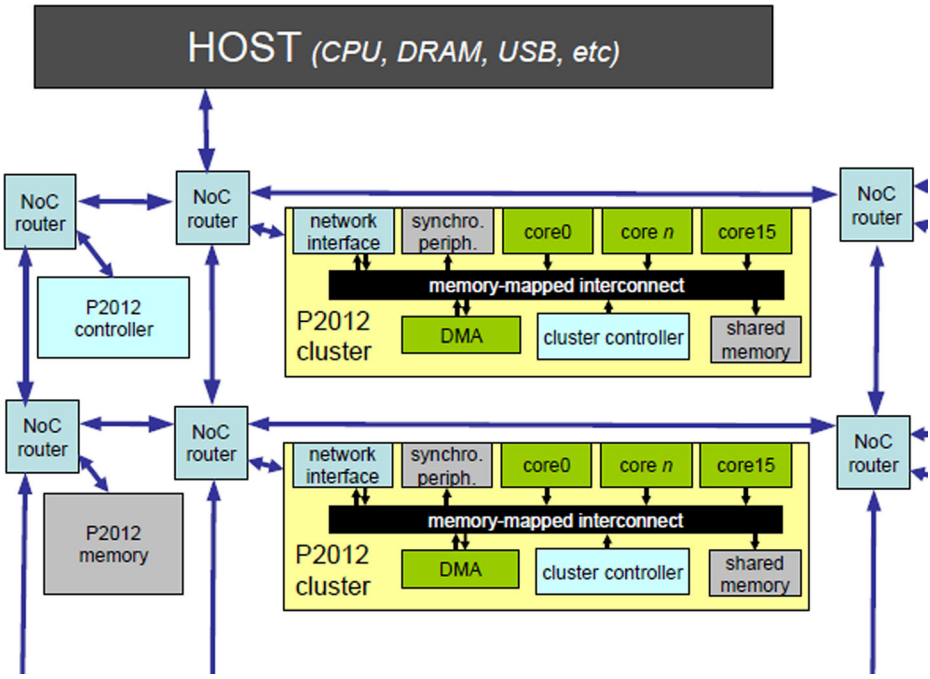

Fig. 7 STHorm Architecture [9]

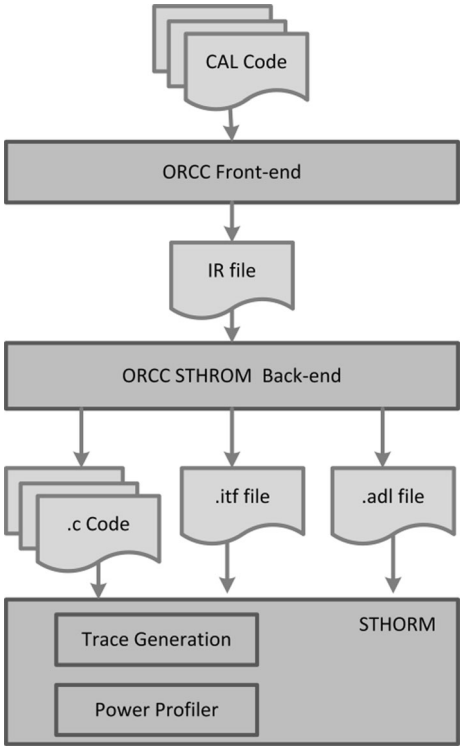

Fig. 8 Tool-chain of simulation

CAL code into different target languages. The ORCC is an open source compiler. The compilation is done in two automated steps. The CAL code is first converted into an Intermediate Representation (IR) [15]. The IR files are then compiled into the target language. For the purpose of simulations, the target language is STHorm specific C code which is compiled and executed on this platform. Two other files are also generated. One is the architecture description language (.adl) file that contains the mapping of the actors onto the cores. This file is manually editable by the designer. The other is interface (.itf) file which contains the interfacing description among the components. Timing traces are generated during execution of the application code. The timing traces are used for measuring the execution time of the different actors in the application. The power profiler of the STHorm simulator is used to measure the power consumption of the cores. The SDK version used for the simulations is STHorm SDK 160.

5 MPEG4-SP decoder case study

In this section, a joint exploration of application and architectural parameter is conducted for the MPEG4-SP decoder. The considered objective functions are execution time and memory consumption. In order to find a suitable application parameter, RVC standard has been investigated. From different variable parameters, FIFO buffer size is one parameter which is effective on memory consumption and execution time. It can be easily configured by user via compiler interface when compiling the CAL code into lower level code as well. Hence, we

have selected FIFO size as the structural application parameter. Concerning the architecture side, number of PEs is a well-known parameter which is extensively investigated in literature. Moreover, it can be easily configured via configuration file in our case. Hence, it has been chosen as the architectural parameter.

5.1 Application model

Figure 9 shows the CAL dataflow network of MPEG4-SP decoder. This is a parallel implementation of the decoder for which the Y, U and V channels are separately implemented. The actors are connected through FIFO channels. According to the MPEG4-SP standard, this decoder is mainly comprised of parser and three parallel paths for decoding the luminance (Y) and the chrominance (U, V) part of the video. Each path itself consists of texture decoding and motion compensation. At the end, the decoded data is merged into one single bitstream. The information among the actors is transferred via tokens.

5.2 Joint optimization definition

In separate optimization, the considered objective function from the application side is the total memory consumption shown with M. The constraint is execution time shown with T. Equation (7) represents the optimization problem. Solving (7) leads to M_{f1}.

$$\begin{cases} \min\{M\} \\ s.t.\ T < T_{target} \end{cases} \tag{7}$$

From the architecture side, the total execution time is considered as the objective function with energy constraint. Optimization formulation is shown in (8). T represents execution time and E represents energy. Solving (8) leads to T_{f1}.

$$\begin{cases} \min\{T\} \\ s.t.\ E < E_{target} \end{cases} \tag{8}$$

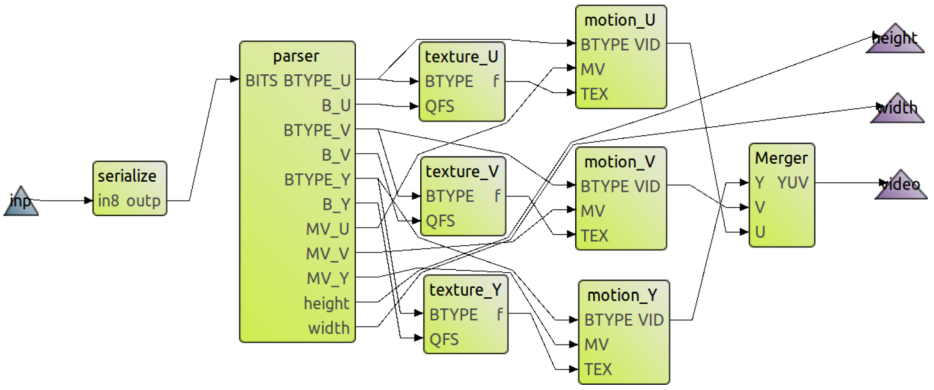

Fig. 9 Parallel MPEG4-SP decoder

Table 1 Parameter range

Parameter	Range
Buffer size	{minimum, 128, 256, 512, 1024}
PE No.	{1, 2, 4, 8, 12, 16}

In joint optimization, the aim is to explore both execution time and memory simultaneously as is written in (9). Result of solving (9) is (M_{f2}, T_{f2}).

$$\begin{cases} min\ \{T,\ M\} \\ s.t.\ E < E_{target} \\ T\ <\ T_{target} \end{cases} \tag{9}$$

In the multi-objective space of time and memory, different metrics can be defined to compare the two points. One meaningful and beneficiary metric is memory-time product or space-time product. This is one of the embedded design metrics that has been previously used in literature [14]. Hence, the comparison between the two points in time-memory space is written as in (10).

$$M_{f2}\ T_{f2} <\ M_{f1}\ T_{f1} \tag{10}$$

5.3 Results

The ranges of values for the FIFO buffer size and the number of PEs that have been experimented are reported in Table 1. The size of the buffers is in terms of bytes.

"Minimum" in Table 1 refers to the minimum possible value for the buffers that prevents deadlock in the system based on the analysis in [6].

Three test sequences have been used for our simulations. Their properties are shown in Table 2. In this analysis, the time constraint has been considered as 200 ms for 50 fps target. The energy constraint is considered as the mean values of the profiled energy consumptions.

Table 3 reports the objective function evaluation in two cases of separate and joint optimization. The memory consumption is reported in terms of Kilo Bytes. The execution time is reported in terms of millisecond. We can notice that joint exploration outperforms separate exploration in terms of Time-Memory product. For the case of Container sequence, there is no feasible answer in the first step of optimization that satisfies the time constraint.The resultant parameters are reported in Table 4.

From Table 4, we notice that both buffer size and number of PEs can be reduced which yields lower Time-Memory product. Though reduction of number of PEs increases the execution time, but at the same time, it reduces the memory consumption of the system.

Table 2 Test sequence properties

Test sequences	Value
Name	Foreman, Akio, Container
Resolution	176 × 144
Number of frames	10

Table 3 Objective Evaluation

Sequence	Optimization	Mem. (KB)	Time (ms)	Time-Mem Product
Foreman	Separate	73.00	60.16	4391.42
	Joint	12.69	90.72	1151.26
Akio	Separate	40.5	53.25	2156.43
	Joint	90.83	12.69	1152.64
Container	Separate	—	—	—
	Joint	79.90	16.69	1333.59

Reducing the buffer size increases the execution time, but reduces the total memory. The augmentation of execution time is negligible compared to reduction of memory consumption provided by buffer size and number of PEs reduction.

From the results in Table 3 and Table 4 we conclude that joint exploration of parameters leads us to design points that were not explored in separate optimization. Comparison of the acquired resulting parameters shows that by using less resources (i.e. less number of PEs, less memory) better points are obtained in terms of the defined design metric.

Figures 10 and 11 show the memory-time design space and the related Pareto frontier for two sequences Container and Foreman. In Fig. 10, when the memory consumption metric is more important, we need to reduce the buffer size of the FIFOs and reduce the number of PEs. On the other side, when the execution time is more important, number of PEs and the buffer size should increase. However, the design points residing on the Pareto frontier show that this is not the straightforward rule as can be noticed in the vertical part of the frontier. Hence, we cannot determine the optimal point in terms of memory and execution time beforehand without joint exploration of the parameters. The same behavior can be noticed in Fig. 11.

6 HEVC decoder case study

In this case study, an algorithmic application parameter is explored in conjunction with an architectural parameter. Energy and execution time are the objective functions with rate and distortion constraints. QP is the considered algorithmic application parameter. Tests are conducted to analyze its effectiveness on the objective functions and constraints. Moreover, it can be simply configured via the configuration file. It is also a well-studied parameter in the

Table 4 Resultant Parameters

Sequence	Optimization	Buffer size (B)	PE No.
Foreman	Separate	1024	8
	Joint	128	4
Akio	Separate	512	8
	Joint	128	4
Container	Separate	—	—
	Joint	128	8

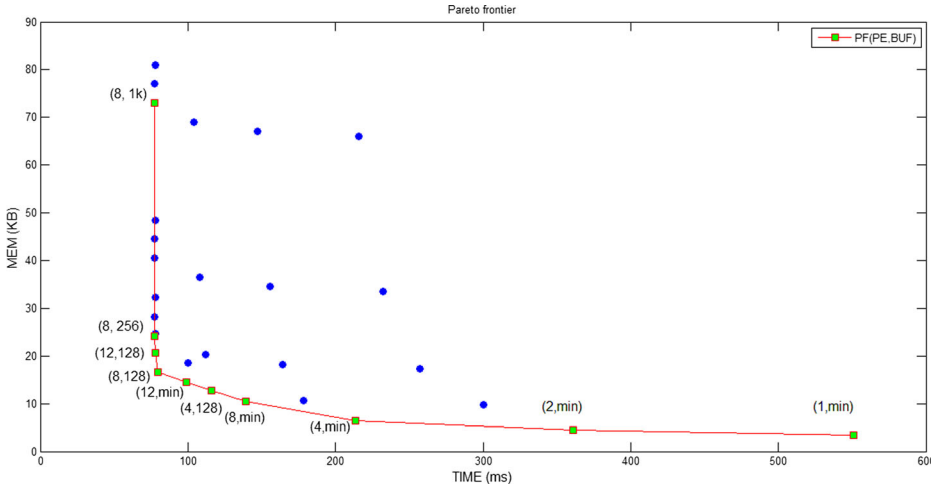

Fig. 10 Container sequence design space and Pareto frontier

literature [17] concerning its effect of rate, distortion and execution time. Like in the case of MPEG4-SP decoder, number of PEs is considered as the architectural parameter. For HEVC decoder, though the decoder cannot be configured directly, we assume that the tuning and configuration can be controlled from the encoder side.

6.1 Application model

The block diagram of HEVC decoder is shown in Fig. 12. This version is compliant with HM 10.0 test model. We have chosen HEVC decoder as the state of the art decoder. It is the newest

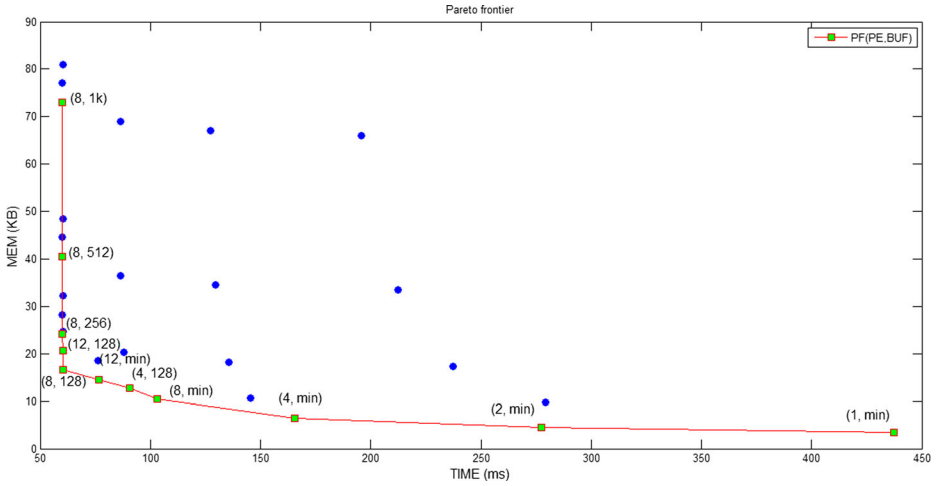

Fig. 11 Foreman sequence design space and Pareto frontier

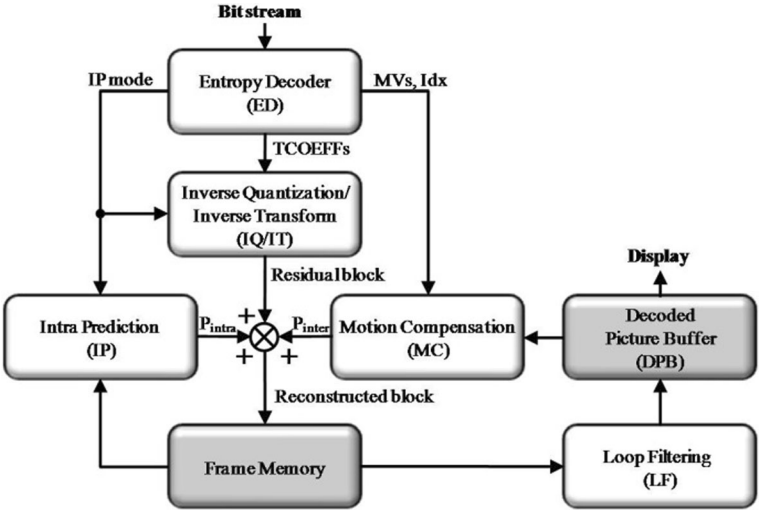

Fig. 12 Block diagram of HEVC decoder [18]

video coding standard aimed for doubling the compression efficiency while providing the same level of quality [16].

6.2 Joint optimization definition

In the first step of separate optimization, the application side objective function is optimized. Total execution time is the objective function with rate and distortion as constraints. These metrics are shown with T, R and D respectively. Optimization is formulated in (11). Solving (11) leads to T_{f1}.

$$\begin{cases} \min\{T\} \\ s.t.\ R < R_{target} \\ D < D_{target} \end{cases} \tag{11}$$

In the second step, the architecture side is optimized. For the architecture, energy consumption is the objective function with time constraint. This step is written as in (12). Solving (12) leads to E_{f1}.

$$\begin{cases} \min\{E\} \\ s.t.\ T < T_{target} \end{cases} \tag{12}$$

Table 5 HEVC test sequence properties

Test sequences	Value
Name	RaceHorses, BasketballPass, BQSquare
Resolution	416 × 240
Number of frames	8

Table 6 Parameter range

Parameter	Range
QP	{20, 22, 24, 26, 28, 30, 32, 34, 36, 38}
PE No.	{3, 4, 6, 9,14}

In joint optimization, the aim is to explore the execution time and energy consumption simultaneously by varying the application and architecture parameters. This joint optimization is written as in (13). Solving (13) leads to (E_{f2}, T_{f2}).

$$\begin{cases} min\ \{T,\ E\} \\ s.t.\ R < R_{target} \\ D < D_{target} \\ T < T_{target} \end{cases} \tag{13}$$

When energy and time are of equal importance in optimization, energy delay product (EDP) is used as a design metric which is a widely used one in literature [7]. It shows the area under the curve of energy and execution time and the point with minimum EDP is a suitable one in both terms of execution time and energy. Hence, we compare the two obtained design points from separate and joint optimization in terms of this metric (14).

$$E_{f2}\ T_{f2} <\ E_{f1}\ T_{f1} \tag{14}$$

6.3 Results

The experiments are conducted on standard test sequences of HEVC with properties that have been reported in Table 5.

The range of QP parameter setting is based on the standard test conditions of HM test model. The range of PE number is selected to change from serial execution of the application on single-core to maximum number of actors, i.e. one PE for one actor. These values are reported in Table 6.

The design space is explored manually to show the proof of concept. However, when the number of parameters increases, efficient search strategies should be deployed to explore the design space.

Table 7 shows the objective evaluation for considered test sequences in two cases of separate and joint optimization. Energy consumption is reported in terms of Joules and execution time is

Table 7 Objective evaluation

Sequence	Optimization	Energy (J)	Exe. Time (ms)	Normalized Energy-Delay Product
RaceHorses	Separate	0.17	3.24	0.16
	Joint	0.24	2.24	0.08
BasketballPass	Separate	0.12	2.35	0.15
	Joint	0.17	1.56	0.04
BQSquare	Separate	0.15	2.90	0.12
	Joint	0.21	2.00	0.064

Table 8 Resultant parameters

Sequence	Optimization	QP	PE No.
RaceHorses	Separate	32	6
	Joint	32	14
BasketballPass	Separate	34	6
	Joint	34	14
BQSquare	Separate	30	6
	Joint	30	14

reported in terms of seconds. From these two metrics, normalized EDP values have been calculated. It is shown that joint optimization of the parameters will result in smaller EDP value in comparison to separate exploration. We have managed to reduce the EDP value by 2.5 times on average via joint exploration. These results are obtained since we can reduce execution time more by simultaneous consideration of the objectives. Due to different video content and motion, the amount of coded data varies and hence different energy consumption and execution times are obtained. Consequently, various improvements are achieved for different sequences.

Table 8 shows the resultant parameters for the two cases of separate and joint optimization for HEVC decoder case study. It is shown that in joint exploration, we have the degree of freedom to increase the PE No. and hence decrease the delay of the decoder. On the other hand, the energy consumption is increased, but the reduction of execution time is more pronounced in final production of these two metrics. The reported execution time and energy values in Table 7 verify this fact. As a result, EDP is decreased.

Since we were working with the initial version of the application, the real measurements were far from real-time executions.

Overall, by adding the application parameter dimension to the design space exploration of multimedia applications, we have provided the designer with more points that can be more efficient in terms of the design objective functions. The trade-off is the evaluation of larger number of points and proposing efficient search strategies for exploring such a huge space.

7 Concolusion & future works

In this paper, a new direction on joint exploration approach for mapping of multimedia applications on many-core platforms is proposed. This approach is tested experimentally via high level configurable application and architecture parameters. The feasibility of this approach is investigated and validated through mapping MPEG4-SP decoder and HEVC decoder on STHorm many-core platform with variable application and architecture parameters. By the achieved results we have shown that joint exploration outperforms the separate one in terms of the objectives we defined.

Future works will focus on expanding our experiments with more parameters from both the application and the architecture side. Another direction of the work is to consider different mapping strategies to explore the design space.

Acknowledgment Part of this research has been conducted in EPFL SCI-STI-MM lab under supervision of Dr. Marco Mattavelli. The authors are grateful for the STHorm platform simulation environment provided by the lab and the application models. Useful discussions, help and support from all the team members are also appreciated heartily by authors.

References

1. "OPEN RVC-CAL Compiler." [Online]. Available: http://orcc.sourceforge.net/
2. Azizi O, Mahesri A, Stevenson JP, Patel SJ, and Horowitz M (2010)"An integrated framework for joint design space exploration of microarchitecture and circuits," 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), pp. 250–255
3. Benini L (2010) Programming heterogeneous many-core platforms in nanometer technology : the P2012 experience. ARTIST DESIGN Summer school Autrans, France
4. Benini L, Flamand E, Fuin D, and Melpignano D (2012) "P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator," 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 983–987
5. Bombieri N, Fummi F, Quaglia D (2010) System/network design-space exploration based on TLM for networked embedded systems. ACM Trans Embed Comput Syst 9(4):1–32
6. Brunet SC, Mattavelli M and Janneck JW (2013) "Buffer optimization based on critical path analysis of a dataflow program design," 2013 I.E. International Symposium on Circuits and Systems (ISCAS2013), pp. 1384–1387
7. Cho S, Melhem RG (2010) On the interplay of parallelization, program performance, and energy consumption. IEEE Trans Parallel Distrib Syst 21(3):342–353
8. Giun G, Lee C, Wang M, Lin H, Su DW, Lin B (2007) Algorithm / architecture Co-design of 3-D spatio – temporal motion estimation for video coding. IEEE Trans Multimedia 9(3):455–465
9. Helmstetter C (2012) "Guided Tour of the P2012 TLM Model : Simulation Speed and Timing Accuracy," 19th Open International Workshop on Synchronous Programming, SYNCHRON
10. Hübert H, Stabernack B (2009) Profiling-based hardware / software Co-exploration for the design of video coding architectures. IEEE Trans Circuits Syst Video Technol 19(11):1680–1691
11. Le Beux S, Nicolescu G, Bois G, Bouchebaba Y, Langevin M. , and P. Paulin, "Optimizing Configuration and Application Mapping for MPSoC Architectures," 2009 NASA/ESA Conference on Adaptive Hardware and Systems, pp. 474–481, Jul. 2009
12. Lee GG, Chen Y, Mattavelli M, Jang ES (2009) Algorithm / architecture Co-exploration of visual computing on emergent platforms : overview and future prospects. IEEE Trans Circuits Syst Video Technol 19(11):1576–1587
13. Mattavelli M, Amer I, Raulet M (2010) The reconfigurable video coding standard. IEEE Signal Process Mag 27(3):159–165. doi:10.1109/MSP.2010.936032
14. Nicol DM, Simha R, Towsley D (1996) Static assignment of stochastic tasks using majorization. IEEE Trans Comput 45(6):730–740
15. Roquier G, Bezati E, Mattavelli M (2012) Hardware and software synthesis of heterogeneous systems from dataflow programs. Can J Electr Comput Eng 2012:1–11
16. Sullivan GJ, Ohm J-R, Han W-J, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
17. Vanne J, Viitanen M, Hämäläinen TD (2012) Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs. IEEE Trans Circuits Syst Video Technol 22(12):1885–1898
18. Viitanen M, Vanne J, Hämäläinen TD, Gabbouj M and Lainema J (2013) "Complexity Analysis of Next-Generation HEVC Decoder," 2012 I.E. International Symposium on Circuits and Systems (ISCAS), pp. 20–23

Maryam Moghadas received her B.Sc. degree in Computer Hardware Engineering from Ferdowsi University of Mashad, Mashad, Iran, in 2004. She received her M.Sc. degree in Computer Architecture Engineering from Sharif University of Technology, Tehran, Iran, in 2007. She is currently working towards her Ph.D. degree in Computer Architecture Engineering at the University of Tehran, Tehran, Iran. She was a visiting researcher at EPFL University, Lausanne, Switzerland for one year. Her research interests include Reconfigurable Multimedia Architectures.

Hossein Afshari received his B.Sc., and M.Sc. in Electrical Engineering from the University of Tehran. He pursued his Ph.D. at the EPFL University in Lausanne, Switzerland. He is currently a software development engineer in Neurobat AG Company. His research interests include hardware and software design and implementation of embedded systems.

Mahmoud Reza Hashemi received his B.Sc., and M.Sc. in Electrical Engineering from the University of Tehran. He pursued his Ph.D. at the University of Ottawa, Canada. He is currently an Assistant Professor at the School of Electrical and Computer Engineering of the University of Tehran. Dr. Hashemi is the co-founder and Director of the Multimedia Processing Laboratory (MPL). His research interests include reconfigurable hardware architectures for multimedia processing, receiver aware video encoding and adaptation, scalable multi view video coding, 3D video, and recently cloud gaming. Dr. Hashemi has chaired or served on the organizing and program committees of a number of international conferences in multimedia, and information technology.