An MPEG-2 to H.264/AVC Intra-frame Transcoder Architecture with Mode Decision in Transform Domain

Milica Orlandić, *Student Member*, IEEE, Kjetil Svarstad, *Member*, IEEE

Abstract — *The computational complexity of the MPEG-2 to H.264/AVC transcoder, in particular the encoder, is technically challenging, but it can be reduced by reusing the information accessible in the decoding process. A low-latency mode decision algorithm performed in transform domain within the MPEG-2 decoder is proposed. The encoder stage contains two mutually exclusive intra prediction algorithms of block sizes 4x4 and 16x16 sharing the hardware logic. The shared intra prediction unit is supported by an on-chip memory organization. The proposed architecture is implemented on FPGA development board. Its implementation supports high throughputs that correspond to a real-time processing of a variety of video resolutions including QFHD 2160p at 30 fps. Furthermore, the minimal required frequency for CIF, SD and HD1080p resolutions are significantly reduced compared to the state of the art*[1].

Index Terms — MPEG-2 to H.264/AVC Intra Transcoder, FPGA Hardware Implementation, DCT domain, Mode Decision.

I. Introduction

Current trends of an increased usage of portable devices and high network bandwidth capabilities indicate that the real-time visual communication over wireless networks has a significant role in various applications. A real-time video streaming with heavy resource demand consumes a large part of battery energy on hand-held devices. Media adaptations, such as transcoding, are important means for constraints introduced by network and device performances. Transcoding is defined as a conversion of coded signal S_1 to newly coded signal S_2 which differs compared to signal S_1 in some of the video characteristics such as bitrate, spatial and temporal resolution or coding standard. Various multimedia systems have been developed and the need for transcoding systems, that provide fast video streaming over various network channels, has increased in the recent years.

Due to the high bandwidth demand for video material, video coding algorithms such as H.264/AVC have been established for faster video transmission, lower bitrate and good quality. Transcoding MPEG-2 to H.264/AVC is performed to adapt the stream to an increasing number of mobile terminals. It converts MPEG-2 bit-stream into a bit-stream with lower bandwidth that meets network requirements and that provides energy optimization in order to extend battery life. A cascaded pixel domain transcoder (CPDT) consists of a MPEG-2 decoder and H.264/AVC encoder. An input MPEG-2 bit-stream is decoded in order to produce a reconstructed frame. The reconstructed image is used as a source for H.264/AVC encoder.

There are several challenges in cascading a decoder and encoder. The main differences between these two standards are listed in TABLE I. These differences make transcoding more complex compared to other transcoding techniques such as MPEG-2 to MPEG-4 transcoding. For example, the H.264/AVC standard uses spatial and temporal predictions to exploit the redundancies in video sequences. In the case of intra frames, H.264/AVC uses intra prediction algorithms and block-wise mode decision for achieving a higher coding efficiency. However, the similarities between H.264/AVC and MPEG-2 are exploited in order to reduce the transcoding complexity. In particular, DCT coefficients statistics in the MPEG-2 can be used for block-wise intra mode decision performed in the H.264/AVC encoder.

TABLE I
Differences between H.264/AVC and MPEG-2 Video Standards

Tools	MPEG-2	H.264/AVC
Transform Type	Real-value DCT	Integer DCT
Transform Block size	8x8	4x4
Spatial redundancy reduction	Transform Coding	Transform Coding, Intra Prediction

The paper is organized as follows. The description of decoding and encoding stages and recent state of the art for hardware architectures are presented in Section II. Low complexity mode decision techniques and experimental analysis on the proposed mode decision algorithm are described in Section III. The proposed hardware architecture of an efficient and low-complexity MPEG-2 to H.264/AVC intra-frame transcoder is presented in Section IV. The results are presented in Section V. The conclusions are summarized in Section VI.

II. MPEG-2 to H.264/AVC Transcoder

The transcoder hardware implementation includes computationally intensive, regular units such as inverse quantization and DCT transform (MPEG-2 decoder), highly data-dependent intra prediction and reconstruction loop

[1] M. Orlandić is with the Norwegian University of Science and Technology, Trondheim, Norway (e-mail: orlandic@iet.ntnu.no).

K. Svarstad is with the Norwegian University of Science and Technology, Trondheim, Norway (e-mail: svarstad@iet.ntnu.no).

(H.264/AVC encoder). The DCT transform is characterized by 8x8 block granularity, whereas the computations in H.264/AVC encoder are performed on 4x4 block level due to 4x4 integer DCT transform. The H.264/AVC standard supports a number of intra prediction algorithms for spatial redundancy reduction. These intra prediction algorithms, based on directional predictions performed in the spatial domain, predict pixels within a macroblock using the reconstructed samples in the left and the upper available neighboring blocks. The block-type intra prediction algorithms proposed by the standard are: Intra 4x4 and Intra 16x16 for luma component, and Intra 8x8 for chroma component. The intra 4x4 prediction defines nine prediction modes: DC mode (mode 2) and eight directional modes, whereas Intra prediction algorithm of size 16x16 defines 4 modes as illustrated in Fig. 1.

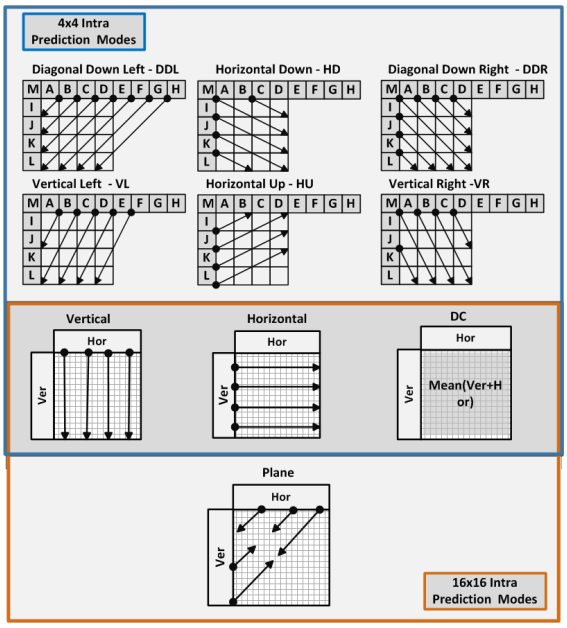

Fig. 1. Prediction modes in two Intra Prediction algorithms defined in H.264/AVC standard

Hardware solutions for H.264/AVC encoder are characterized by high power consumption, high latency and area requirements due both to data-dependency bottlenecks and to intensive communication with memory for storing intermediate data from both algorithms. Efficient H.264/AVC intra-encoder architectures [1]-[7] have been reported. The ongoing technical advances are achieved in FPGA solutions to support robustness, reconfiguration, lower costs and real-time processing. Hamzaoglu *et al.* [8] propose a low-cost intra prediction implementation capable of processing 35 CIF frames per second. Diniz *et al.* [9] propose FPGA architecture which lowers the critical path of the encoder by avoiding re-processing of 16x16 best mode intra predictions by the use of BRAMs for all prediction modes. The intra-encoder is able to process HD 1080p resolution at frame rate of 60 fps. Previous work [10] on a fast intra prediction architecture suggests the use of wavefront 4x4 intra prediction in order to process ultra HD sequences. Hsia *et al.* [11] present a VLSI architecture of intra 16x16 prediction which provides central based computation for the plane mode. Power efficient architecture proposed by Xu *et al.* [12] focuses on data reuse and decomposition, whereas architecture presented by Nadeem *et al.* [13] provides the modified equations for plane mode parameters and achieves partial data-path overlap of intra 4x4 and 16x16 prediction algorithms.

A. Intra 4x4 Prediction

Intra 4x4 prediction is performed on textured areas and is characterized by high data dependency. Vertical and horizontal mode predictions are generated by copying the reconstructed pixels from upper or left 4x4 blocks respectively, whereas DC mode is the mean value of the upper and/or left 4x4 neighbor blocks. The modes from diagonal group (DDL, DDR, VL, VR, HU, HD) are composed of 2 or 3-tap filters. In fact, most of the results of the 2-tap filter equations are reusable for calculating the 3-tap filter equations. The cost function Sum of Absolute Differences (SAD) is used for comparison of computed predictions and original pixels. The prediction mode with the minimum SAD value is selected as the best mode for block-wise prediction.

B. Intra 16x16 Prediction

Intra 16x16 prediction algorithm is introduced for prediction of smooth image areas. It is performed over complete macroblock based on the reconstructed samples of the neighboring macroblocks. The vertical mode extrapolates samples from the reconstructed last row of the upper neighbor macroblock, whereas the horizontal one extrapolates samples from the reconstructed column of the left neighbor. Furthermore, the DC mode calculates the mean of the upper and/or left neighbor. The plane mode performs linear spatial interpolation and its intermediate parameters H, V, a, b and c are computed in the following manner:

$$\begin{aligned} H &= \sum_{x=0}^{7}(x+1)\times(P(x+8,-1)-P(6-x,-1)),\\ V &= \sum_{y=0}^{7}(y+1)\times(P(-1,y+8)-P(-1,6-y)),\\ a &= 16\times(P(-1,15)+P(15,-1)),\\ b &= (5\times H)>>6,\\ c &= (5\times V)>>6. \end{aligned} \quad (1)$$

Finally, the plane mode is computed by:

$$Plane(x,y) = \mathrm{Clip}_1(a+b(x-7)+c(y-7)>>5) \quad (2)$$

where round-off function *Clip_1* limits the result to the range [*0*, *255*].

C. Reconstruction Loop

The reconstruction loop contains 4 separate modules: integer DCT, quantization, inverse quantization and inverse DCT transform. The last two units are part of the reconstruction path for generating reference samples required for intra prediction. The latency of the path is critical for achieving high performance. The dataflow of the loop starts with transformation and quantization of the residual samples with

minimal SAD. Quantized residual samples are sent for entropy encoding and back to the reconstruction path. Reconstructed residual samples are then added to the best mode prediction coefficients and the results are stored in the memory to be used as reference samples for incoming blocks.

III. MODE DECISION

Intra prediction H.264/AVC standard suggests two methods for selecting the best block-type: Lagrangian Rate-Distortion Optimization method (RDO) as high-complexity mode decision and the low-complexity approach based on SAD computation. The RDO method finds the mode that minimizes the cost $J = R + \lambda D$ where the rate R is the bitrate after the entropy coding and the distortion D represents the differences between the original pixels and the prediction pixels. The tradeoff between rate R and distortion D is controlled by the Lagrange multiplier λ that depends on Quantization Parameter (*QP)*. In order to determine the best RDO mode, 592 different RDO calculations are needed. Therefore, the computational demand for encoding stage is extremely high, especially in the case of real-time compression of larger video resolutions. Compared to RDO approach, SAD is the commonly used cost function for the optimal block size determination of the intra prediction in hardware architectures [1]-[3] due to its lower complexity. However, since it performs both the intra prediction algorithms simultaneously, it requires complex control and large memory capacity for their synchronization. In general, the block size of intra prediction is correlated with the content in the block and the QP value. An alternative mode decision based on decision tree data mining technique, proposed by Jillani *et al.* [14], exploits statistical properties of MBs in a video frame. In the case of transcoding between two compression standards, it is computationally less expensive to conduct the block classification in decoder's transform domain rather than in the spatial domain. In this way, the computational complexity is not only reduced by excluding RDO or SAD mode decision, but also by performing only one intra prediction algorithm at once instead of two or three (depending on the profile).

The core of MPEG-2 video compression algorithms is a two-dimensional 8x8 DCT. Each DCT coefficient is a linear combination of the pixel values within one 8x8 block and captures local image features. The upper left coefficient in DCT block, DC component, represents the average luminance of the block, whereas the other coefficients are defined as AC coefficients.

A. Low complexity mode decision algorithms

A number of alternative low-complexity algorithms for mode decision in the transform domain have been proposed to reduce the computational complexity. These techniques are based on texture classification and edge detection algorithms due to consistency of directional correlation with edge direction within a block. An edge detection model based on non-normalized Haar transform (NHT) [15] is used for mode selection of the block-wise intra prediction algorithms in H.264/AVC, whereas an efficient edge histogram algorithm (EHDiD) [16] in DCT domain is generated by the use of edge detectors based on a ratio between two AC (AC_{01}, AC_{10}) coefficients of the DCT block. Mode decision can be turned into pattern classification problem based on support vector machines (SVMs) in down-sized video sequence [17]. An approach based on intra prediction decision matrices is proposed by Yu *et. al* [18].

In particular case of MPEG-2 to H.264/AVC transcoding, several software-based alternative mode decision techniques are reported in literature [16], [19]-[21]. Variance in both spatial and frequency domain is often used for determining the texture i.e. weather the block contains edges. The variance in the transform domain is presented by:

$$\sigma^2 = \frac{1}{N^2}\sum_{u=0}^{7}\sum_{v=0}^{7}(X_{dct}(u,v))^2, \ (u,v) \neq (0,0). \tag{3}$$

It represents the energy level within 8x8 block and the equation for computing variance that requires less computation is given by:

$$\sigma^2 \cong E_{8,8} = \sum_{u=0}^{7}\sum_{v=0}^{7}\left|X_{dct}(u,v)\right|, \ (u,v) \neq (0,0). \tag{4}$$

Total energy of a macroblock is sum of four $E_{8,8}$ values computed for each 8x8 block within a macroblock. Approach based on estimation of directional features proposed by Kalva *et al.* [19] uses edge angle, mean and variance obtained in DCT domain for determining intra prediction modes. Furthermore, it also develops a method for deriving 4x4 and 16x16 DCT from 8x8 DCT transform in order to improve accuracy. Wang *et al.* [20] defines two requirements for the low-complexity prediction block-size. The DC-based requirement relies on the difference between maximal and minimal DC values in a macroblock, whereas the AC-based condition computes the energy of a block. The smoothness decision proposed by Liu *et al.* [21] can be used as a cost function for block-wise decision among intra prediction algorithms as presented:

$$C_{DC} = 0.5\left(\sum_{n=0}^{3}\left|E_{8,8}^{n}(u,v) - \bar{E}\right|\right) < \gamma. \tag{5}$$

where $\bar{E}$ is the average energy across the complete macroblock and γ is the experimentally determined threshold.

B. Proposed Mode Decision Analysis

The cost functions proposed in current state of the art [19]-[21] can be used under certain modifications to achieve fast real time hardware implementation of transcoding system. Higher indexed AC coefficients are commonly close or equal to zero, whereas lower indexed coefficients preserve the information and determine texture features of the block. A block containing details has larger variance, whereas luminance change within a macroblock is determined by DC coefficients of all four 8x8 blocks. Thus, highly active block areas (with edges and textures) are characterized by high luminance coefficients variation among 8x8 blocks and high

levels of energy in the lower AC coefficients. These conditions are presented as follows:

$$C_{DC} = \sum_{n=0}^{3} \left| X_{dct}^{n}(0,0) - \bar{X}(0,0) \right| < \alpha$$
$$C_{AC} = E_{aprox} \cong \sum_{n=0}^{3} \sum_{i,j=0}^{3} X_{dct}^{n}(i,j) < \beta, \ (i,j) \neq (0,0) \quad (6)$$

where $\bar{X}(0,0)$ is an average DC value within a macroblock, and α and β are experimentally determined thresholds. Coefficients used in cost function are presented in Fig. 2. The low luminance variation within a macroblock is determined by the criterion C_{DC}, whereas the presence of edges is given by criterion C_{AC}. The simplified variance C_{AC} contains less computation than the original variance/energy equations and is suitable for low-complexity hardware implementation of the mode decision module which does not introduce additional delays. Taking into consideration the results of rate-distortion analysis from the literature which the proposed mode decision algorithm relies on [19]-[21], the overall impact on rate and distortion of the algorithm with introduced modifications is minor.

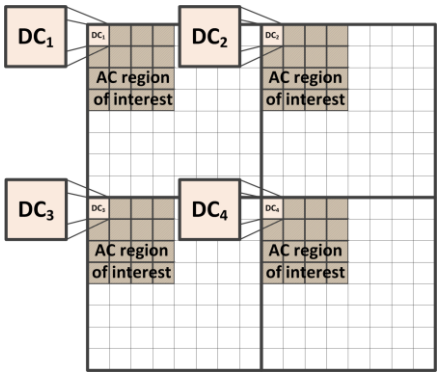

Fig. 2. DCT coefficients of interest for mode decision optimization technique

The thresholds for the two presented cost functions are experimentally determined. Their values also change depending on the quantization level used in the encoding process since it affects the block-size selection process. Higher is the value of the QP, fewer are details present in the frame. The number of intra 16x16 predicted blocks increases with the quantization. Kalva *et al.* [19] present mode mismatches minimization which determines the thresholds for complete range of the QP for CIF sequences. The statistics in CIF and HD sequences can differ significantly, thus an experiment on the mode decision in HD sequences is performed in order to determine the thresholds for the cost functions. The simulation environment is presented in TABLE II. The HD sequences of resolutions 1080p (Riverbed, Pedestrian Area and Rush hour) are used. The proposed cost functions perform the mode decision using DCT transform coefficients. The thresholds are selected to minimize mode mismatches of macroblocks. Fig. 3 presents the block-type mode decision accuracy with respect to the QP range. The thresholds are experimentally selected and further used in the hardware implementation. The average threshold dependency function over QP range in listed video sequences is presented in Fig. 4. The experiment for defining the threshold includes comparison with results in JM model [22] for all defined values of the QP in the range (0-51). The RD optimization is disabled in H.264/AVC stage and the low complexity mode decision based on SAD is set in JM model.

TABLE II
SIMULATION ENVIRONMENT

Sequences	Resolution
Pedestrian area	1920x1080, 4:2:0, 25 Hz
Riverbed	1920x1080, 4:2:0, 25 Hz
Rush Hour	1920x1080, 4:2:0, 25 Hz

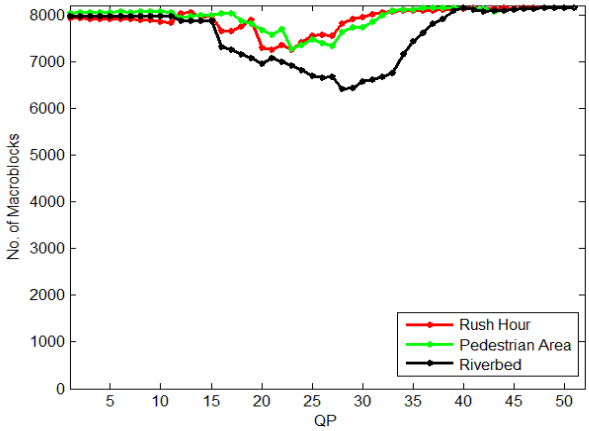

Fig. 3. Mode estimation accuracy of proposed mode decision algorithm

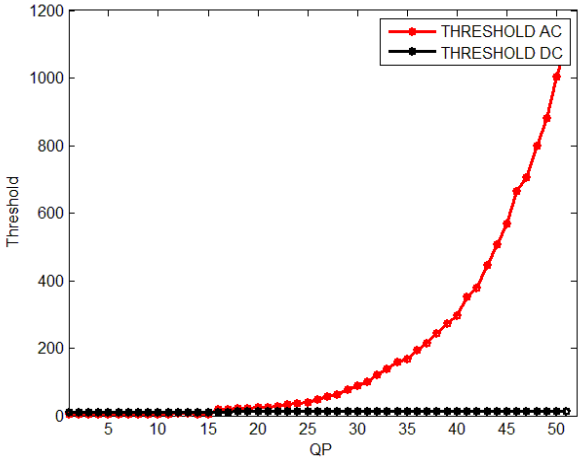

Fig. 4. Threshold dependency over QP parameter in HD sequences for proposed mode decision cost function

IV. PROPOSED ARCHITECTURE

An MPEG-2 to H.264/AVC intra-frame transcoder architecture with a low-complexity block-type decision algorithm in transform domain and high-throughput intra prediction algorithms is presented. The block-type mode decision uses properties of DCT coefficients gathered from the MPEG-2 decoding process. The approach avoids the use of

RDO and SAD mode decision algorithms and the decision is made prior to the prediction process.

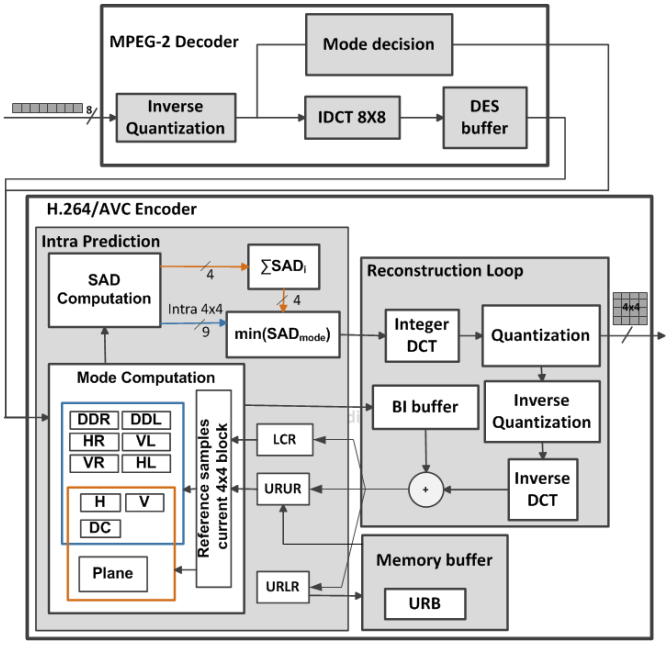

Fig. 5. Block diagram of the proposed transcoder

The top level hardware organization of the transcoder is depicted in Fig. 5. It consists of the inverse 8x8 DCT, the inverse quantization, the Decoder-Encoder Synchronization (DES) buffer and the mode decision as parts of the MPEG2 decoder, the intra prediction unit and the reconstruction loop in the H.264/AVC encoder. The prediction process within the encoder is orchestrated by a 4x4 block level pipeline and the processing orders of 4x4 blocks for both intra prediction algorithms are presented in Fig. 6. The mode computation module is partially shared by the two algorithms in the case of common modes. The rest of the encoder datapath including SAD computation, reconstruction loop and the memory communication are adapted to be shared by both algorithms.

A. Intra 16x16 Prediction

The 16x16 intra prediction process for one macroblock takes 48 clock cycles, including the reconstruction loop. The dataflow of the complete process is presented in Fig. 7. The process starts by fetching the reconstructed neighbor samples from the local registers (LCR, URUR). These samples are used for the computation of the prediction modes in the first 8 clock cycles. Vertical, horizontal and DC mode pixels are calculated in the first clock cycle and are placed in the local registers in order to wait for sixteen original 4x4 blocks. The fourth prediction mode, the plane mode, performs the computation of the intermediate parameters in order to produce the prediction seed sample for the complete 16x16 block i.e.

$$P(0,0) = a - 7b - 7c\,. \tag{7}$$

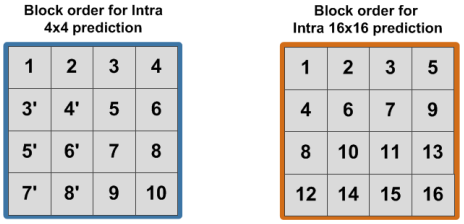

Fig. 6. Macroblock scanning order for two intra algorithms (block-types 4x4 and 16x16)

The computation of the intermediate parameters requires a period of initial 8 clock cycles. Based on the seed value, other prediction samples for the plane mode within the first 4x4 block are computed before the arrival of original pixels by adding parameters *b* or *c*, depending on the pixel position. The plane mode computation within 4x4 block and the dependency between the two 4x4 blocks are described in Fig. 8 a).

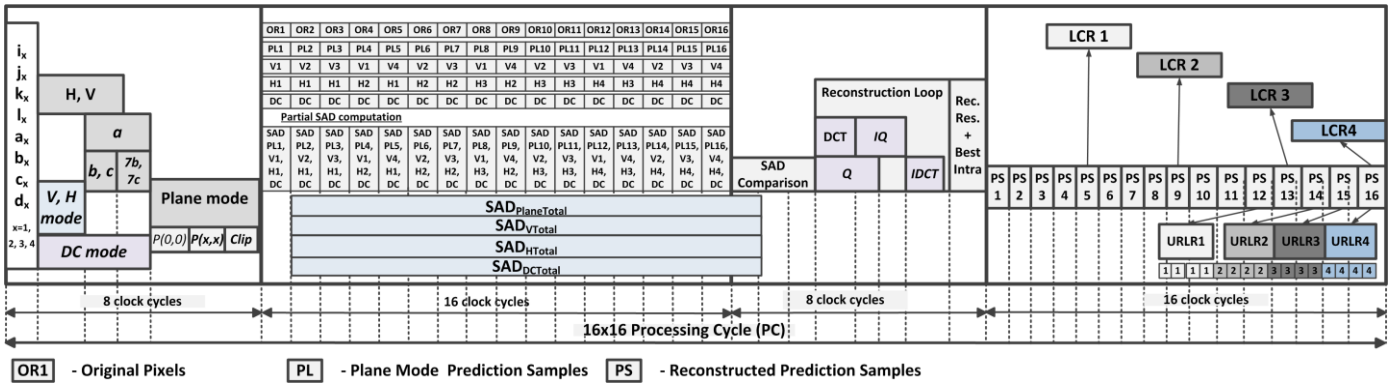

Fig. 7. Intra 16x16 prediction dataflow for one macroblock

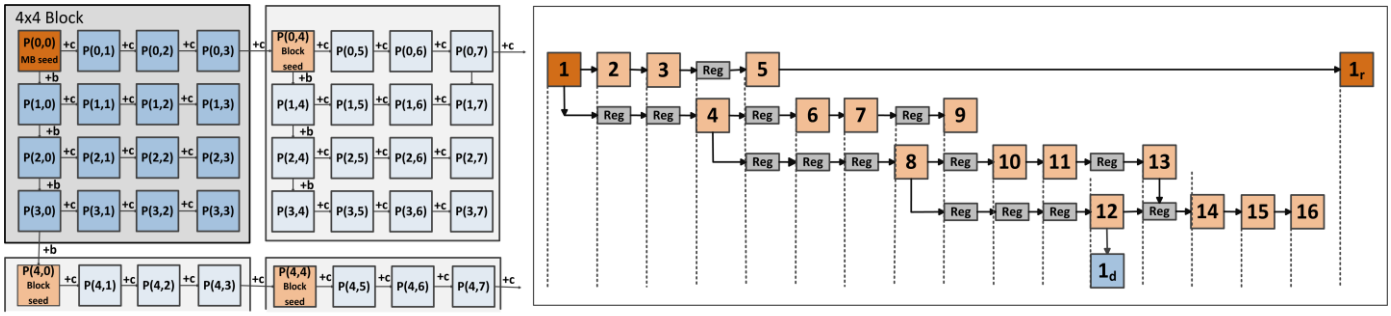

Fig. 8. a) Plane mode calculation for one 4x4 block, b) Memory organization for storing seed values within one macroblock

Likewise, the seeds for the neighboring 4x4 blocks (lower and right) are computed in the current clock cycle and further used for computing plane prediction samples for the 4x4 block that is determined by the block order for 16x16 prediction algorithm. The computations for the other 4x4 blocks are done one clock cycle before original data arrive in order to save logic and decrease latency. Once the seed values for all 4x4 blocks are computed, they are used for the computation of other plane prediction samples or stored in registers while waiting for processing as presented in Fig. 8.b).

After the initial phase, the original 4x4 blocks arrive per clock cycle in the previously defined scanning order. This process takes 16 clock cycles. The plane mode prediction samples for the current 4x4 block are simultaneously computed based on their seed value computed in one of the previous cycles. The SAD cost function for the 4x4 block between the original and the prediction samples is computed for each mode. SAD values for the 4x4 blocks within a macroblock are summed up. In the third phase, SAD_{mode} values on the macroblock level for four prediction modes are compared. The residual samples that correspond to the mode with minimal cost function are further processed by the reconstruction loop.

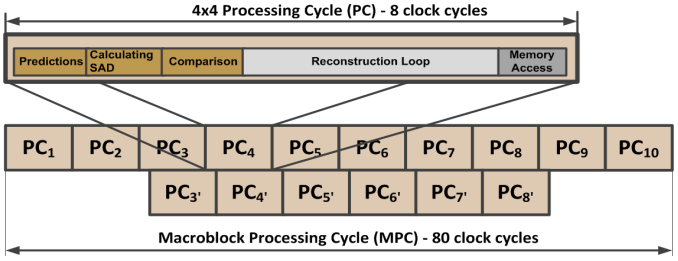

Fig. 9. Intra 4x4 prediction dataflow per macroblock

The last phase of the macroblock prediction processing cycle is the communication with memory for storing prediction data samples. Relevant data required for further prediction process, last row and column of the reconstructed macroblock, are stored in the corresponding registers.

B. Intra 4x4 Prediction

Data dependency between 4x4 blocks in intra 4x4 prediction algorithm is found to be the performance bottleneck of the whole transcoding system. However, the data-independent 4x4 blocks within the same macroblock can be processed in pipeline and, in this way, decrease the total latency. In the proposed architecture, the processing cycle for the 4x4 block takes 8 clock cycles as presented in Fig. 9. It consists of prediction mode computation, computation of SADs for all the prediction modes and their comparison. The best mode residual is sent further to the reconstruction loop. Reconstruction loop takes 4 clock cycles to produce reconstructed residuals [23]. Finally, reconstructed residuals from the last column and row of the 4x4 block are stored in the local registers. This pipelined structure provides a complete intra prediction process for one macroblock to be performed in 80 clock cycles, resulting in 37% of saving in latency compared to the same architecture without use of the pipeline.

C. Mode Decision Implementation

The MD module in MPEG-2 decoders determines intra prediction block-type algorithm for the current macroblock. Two thresholds are set based on the experimental results performed on a number of video sequences. The value of the thresholds changes according to the QP value and experimentally determined threshold values for different QP are stored in RAM memory. Relevant data in the transform domain are used for computing the mode decision cost functions that are compared with the threshold values. The sum of differences between the DC values of the four 8x8 blocks and the average DC value across the complete macroblock represents the DC cost function. The sum of DC coefficients is updated in each clock cycle when the first column of 8x8 block arrives to the module.

The presence of the edges in the block is explored by the second cost function estimated as the sum of the highest AC coefficients in the four 8x8 blocks. Furthermore, by performing computation over the data of the first four columns of each 8x8 block, the mode decision delay is minimized and the cost function computation can be completed in parallel with IDCT computation in the decoder stage. If one among the cost functions is higher than the set thresholds for the current macroblock, the activity of the macroblock is high and intra prediction of block size 4x4 is chosen. The MD module implementation for both DC and AC cost functions is presented in Fig. 10.

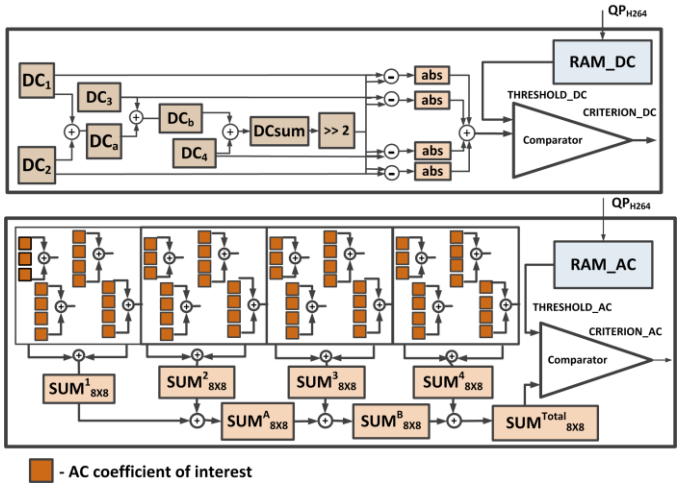

Fig. 10. Mode decision implementation that consists of DC and AC cost functions

D. Memory Organization

Video compression systems are characterized by complex structure and, in that sense, a significant amount of intermediate results of various parallel operations have to be fetched in the requested time. This grows the need for a memory organization that consists of a large number of buffers and registers that are orchestrated by complex modules such as intra prediction. In addition, the communication with memory can introduce significant delays and power costs in case off-chip memory is used.

The proposed architecture includes only on-chip memory resources in order to speed up the complete system. In particular, the decoding and encoding stages require different degrees of parallelism and an intermediate reordering buffer is required to adapt the incoming data for encoding stage. The intra prediction shared Upper Row Buffer (URB) stores one frame line that corresponds to the last row of the active macroblock line. The URB and three registers, Upper-Row-Update Register (URUR), Upper-Row-Local Register (URLR) and Left-Column Register (LCR) build a complex prediction memory unit that orchestrates prediction samples of the current active 4x4 block/macroblock depending on which intra prediction algorithm is chosen by mode. The general organization of the communication with URB is presented in Fig. 11.

The communication with the memory is adapted to switch between intra algorithms. In the case of 16x16 prediction, the data from the upper macroblock neighbor are stored in the URUR during the complete processing cycle i.e. 48 clock cycles. More complex data-dependency in the case of 4x4 intra prediction process requires partial update of data after each 4x4 block processing depending on the position of the 4x4 block within the macroblock as presented in Fig. 12. The local memory system for storing upper macroblock neighbor consists of registers URLR and URUR. The need for two registers arises due to the critical operations in clock cycle *t*+80 where two simultaneous operations take place. The first one, which fetches reference samples from the URB memory buffer, is performed in order to update data for the macroblock whose prediction takes place in the incoming processing cycle. The second operation, which temporary store the relevant reference samples from the current macroblock, is required for communication towards the URB. In this scenario, the URLR register performs the writing to the URB, while, in the exact same moment, data are being read from URB to URUR.

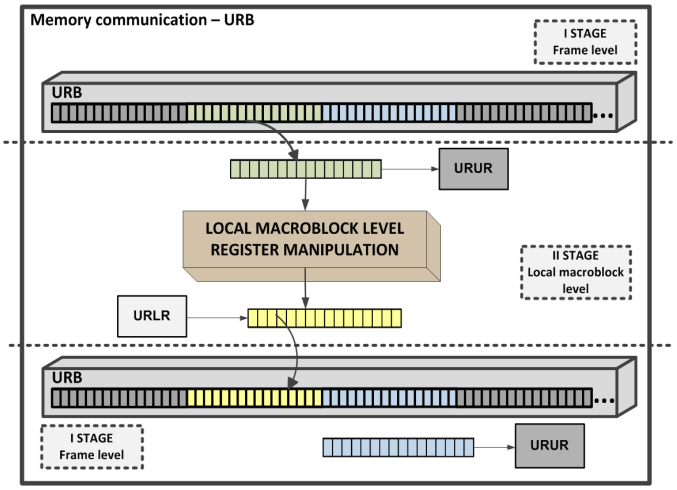

Fig. 11. Hierarchical organization of the shared memory for intra prediction algorithms

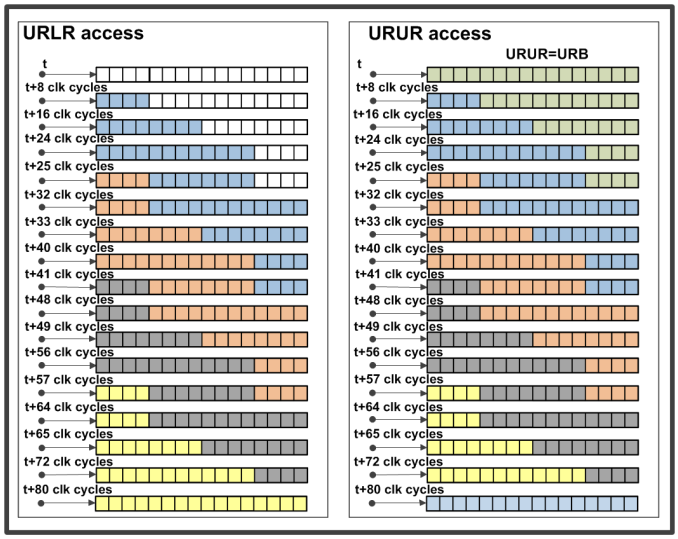

Fig. 12. Data-flow in local registers for storing intermediate predicted samples for intra 4x4 prediction

V. RESULTS

The proposed MPEG-2 to H.264/AVC transcoder architecture is designed in VHDL. The reference model with input vectors is used for verification of each transcoder stage. The performance results are compared with JM18.3 reference software results [22]. The FPGA board on which architecture is realized consists of 203,800 LUTS and 4000 Kb of distributed RAM. TABLE III shows area usage of the FPGA board (33% of LUTS used).

Based on the exploration of data-dependency, mutually-independent 4x4 blocks in intra prediction algorithms are

processed in a pipelined manner. The design supports an intra-prediction transform-domain mode decision performed in MPEG-2 decoder, two block-type prediction algorithms of sizes 4x4 and 16x16, a 4x4 integer DCT transforms in encoder, a 8x8 DCT in decoder stage and an on-chip memory system for storing intermediate results. The intra 16x16 prediction implementation supports all 4 prediction modes defined by the standard. In the case of 4x4 prediction algorithm, a new scanning order which enables pipeline processing is proposed. The pipelined structure achieves 37% lower latency than strictly sequential version of the proposed implementation.

TABLE III
RESOURCE UTILIZATION

Modules	FPGA Slices	FF Util.	LUTS	RAM (kbits)	Multipliers
MPEG-2 IQ	512	448	1206	-	16
MPEG-2 Mode decision	126	206	349	1	-
MPEG-2 DCT 8x8	2022	1421	6441	-	-
H.264/AVC Intra Prediction (4x4, 16x16)	19149	26833	68861	28.5	-

The design contains a complex on-chip memory organization in order to speed up the complete transcoding process. The intra prediction data dependency and the shared logic used for both the intra prediction algorithms require memory support for storing and fetching relevant data after the reconstruction is performed. The system contains a number of registers such as URUR, URLR and LCR and a memory buffer URB that provide secure and accurate communication within the intra prediction module. Furthermore, the encoder stage contains a control for switching functionality of the shared logic in case of different intra algorithms. The FPGA implementation with the same level of parallelism [9] reports increased memory use compared to the proposed implementation due to extensive use of BRAMs in order to perform computational reduction. In the proposed implementation the hardware resources are extensively used for the dynamic adaptations of intra prediction datapaths which enable the design to function with full independence and provide a necessary speed-up for achieving higher throughput.

State of the art on MPEG-2 to H.264/AVC transcoding systems and on mode decision algorithms in transform domain is limited to software optimized solutions. Thus, TABLE IV presents a comparison to the recent related works [3-5, 8-9, 11] on VLSI and FPGA implementations of H.264/AVC intra-frame encoder. The full performance of the 16x16 intra module processes one macroblock within 48 clock cycles, whereas, in the case of 4x4 intra prediction, the macroblock processing is achieved in 80 clock cycles. The maximum throughput is determined by the slowest path of the design, i.e. 4x4 intra prediction. Thus, the maximum throughput in the proposed design is 349 Mpixels/s, whereas the minimal frequency for 1080p resolution at 30 fps is 19.4 MHz. Compared to the state of the art of H.264/AVC intra-frame encoders, the proposed architecture requires significantly less clock cycles while keeping the operating frequency high and thus fulfills throughput requirements for HD video resolutions. Furthermore, lower minimal frequencies required for processing CIF, SD and HD sequences are reported. Finally, the proposed design can support throughputs that correspond to the real-time processing of video sequences of QFHD 2160p resolution at 30 fps with the low-input parallelism of 8 pixels/cycle.

TABLE IV
COMPARISON TO RECENT WORKS ON INTRA PREDICTION ARCHITECTURES

Design	[3]	[4]	[5]	[8]	[9]	[11]	Proposed Design
Technology	ASIC 90 nm	ASIC 90nm	ASIC 130 nm	FPGA	FPGA	ASIC 180 nm	FPGA
Supported Intra Prediction	4x4 16x16	4x4 16x16	4x4 16x16	4x4 16x16	4x4 16x16	16x16	4x4 16x16
Max Throughput (Mpixels/s)	-	151	62	3.5	113	201	349
Input Parallelism	4-pixel	16/8-pixel	32-pixel	1-pixel	16-pixel	4/1-pixel	8/16 - pixel
Max. Supported Resolution	1080 30 fps	1080p-480p+CIF	1080p 30fps	CIF 35 fps	1080p 60 fps	1080 60 fps	2160 30fps
Cycles/Macrobl.	N/A	<454	464	5151	<300	267	80
Frequency	180	135	114	71	168	210	109
Min. Frequency 2160@30fps	-	-	-	-	-	-	77.8
Min. Frequency 1080@30fps	152	110	114	-	73	65	19.4

VI. CONCLUSION

This paper presents an efficient, high-throughput architecture for MPEG-2 to H.264/AVC intra-frame transcoding system. The complexity reduction is achieved through a mode decision algorithm performed in the transform domain. In this way, only one of the intra prediction algorithms is active in the current macroblock processing cycle. This mutual exclusive prediction system that switches between two algorithms provides that the intra prediction algorithms share the logic and that additional storage of intermediate data and power consumption are lowered. Furthermore, highly pipeline implementation increases substantially the throughput despite high data dependency.

In summary, compared to the state of the art, the proposed design achieves a better performance in terms of throughput of real-time processing of HD 1080p video sequences, which is the maximum resolution processed by the state of the art implementations. The proposed implementation is area-efficient and, as such, it leaves space for the implementation of additional coding options within the MPEG-2 and H.264/AVC transcoding system e.g. an inter prediction module.

REFERENCES

[1] Y. W. Huang, B. Y. Hsieh, T. C. Chen, and L. G. Chen, "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 378-401, Mar. 2005.

[2] K. Suh, S. Park, and H. Cho, "An efficient hardware architecture of intra prediction and TQ/IQIT module for H. 264 encoder," *ETRI journal*, vol. 27, no. 5, pp. 511-511, Oct. 2005.

[3] J.-W. Chen, H.-C. Chang, J.-S. Wang, and J.-I. Guo, "A dynamic quality-adjustable H.264 intra coder," *IEEE Trans. Consumer Electron.*, vol. 57, no. 3, pp. 1203-1211, Aug. 2011.

[4] G. L. Li, T. Y. Chen, M. W. Shen, M. H. Wen, and T. S. Chang, "135-MHz 258-K gates VLSI design for all-intra H. 264/AVC scalable video encoder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 636-647, Apr. 2013.

[5] H.-C. Kuo, L.-C. Wu, H.-T. Huang, S.-T. Hsu, and Y.-L. Lin, "A Low-Power High-Performance H.264/AVC Intra-Frame Encoder for 1080pHD Video," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 6, pp. 925-938, Jun. 2011.

[6] Y.-K. Lin, C.-W. Ku, D.-W. Li, and T.-S. Chang, "A 140-MHz 94 K gates HD1080p 30-frames/s intra-only profile H. 264 encoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 432-436, Mar. 2009.

[7] C. Diniz, B. Zatt, C. Thiele, A. Susin, S. Bampi, *et al.*, "A high throughput H. 264/AVC intra-frame encoding loop architecture for HD1080p," in *IEEE Int. Symp. Circuits and Syst.*, Rio De Janeiro, Brazil, pp. 579-582, May 2011.

[8] I. Hamzaoglu, O. Tasdizen, and E. Sahin, "An efficient H.264 intra frame coder system," *IEEE Trans. Consumer Electron.*, vol. 54, no. 4, pp. 1903-1911, Nov. 2008.

[9] C. M. Diniz, A. A. Susin, and S. Bampi, "FPGA design of H. 264/AVC intra-frame prediction architecture for high resolution video encoding," in *Southern Conf. Prog. Logic*, Bento Goncalves, Brazil, pp. 1-6, Mar. 2012.

[10] M. Orlandić and K. Svarstad, "A Low Complexity H.264/AVC 4X4 Intra Prediction Architecture with Macroblock/Block Reordering," in *IEEE Int. Conf. Reconfig. Computing and FPGAs*, Cancun, Mexico, pp. 1-6, Dec. 2013.

[11] S. Hsia, W. Hsu, and Y. Chou, "Fast low-complexity computation and real-time architecture for H.264/AVC intra-prediction," *Real time Image Processing*, Mar. 2012.

[12] K. Xu and C.-S. Choy, "A power-efficient and self-adaptive prediction engine for H. 264/AVC decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 3, pp. 302-313, Mar. 2008.

[13] M. Nadeem, S. Wong, and G. Kuzmanov, "An efficient hardware design for intra-prediction in H. 264/AVC decoder," in *Saudi Int. Electron., Comm. and Photon. Conf.*, Riyadh, Saudi Arabia, pp. 1-6, Apr. 2011.

[14] R. Jillani and H. Kalva, "Low complexity intra MB encoding in AVC/H. 264," *IEEE Trans. Consumer Electron.*, vol. 55, no. 1, pp. 277-285, Feb. 2009.

[15] Z. Wei, H. Li, and K. N. Ngan, "An efficient intra mode selection algorithm for H.264 based on fast edge classification," in *IEEE Int. Symp. Circuits Syst.*, New Orleans, USA, pp. 3630-3633, May 2007.

[16] M. Eom and Y. Choe, "Fast extraction of edge histogram in DCT domain based on MPEG7," in *Proc. World Academy Science, Engineering Tech.*, pp. 209-212, Nov. 2005.

[17] Z.-Y. Lu, K.-B. Jia, and W.-C. Siu, "Low-complexity intra prediction algorithm for video down-sizing transcoder," in *IEEE Visual Comm. Image Proc.*, Tainan, Taiwan, pp. 1-4, Nov. 2011.

[18] S.-J. Yu, J.-S. Myung, D.-G. Sim, and S.-J. Oh, "Enhanced transform domain intra prediction for MPEG-2 to H. 264/AVC transcoding," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 703-708, May 2008.

[19] H. Kalva and B. Petljanski, "Exploiting the directional features in MPEG-2 for H. 264 intra transcoding," *IEEE Trans. Consumer Electron.*, vol. 52, no. 2, pp. 706-711, May 2006.

[20] L. Wang, Q. Wang, Y. Liu, and W. Lu, "A fast intra mode decision algorithm for MPEG-2 to H. 264 video transcoding," in *IEEE Int. Symp. Consumer Electron.*, St. Petersburg, Russia, pp. 1-5, 2006.

[21] X. Liu, K.-Y. Yoo, and S. W. Kim, "Low complexity intra prediction algorithm for MPEG-2 to H. 264/AVC transcoder," *IEEE Trans. Consumer Electron.*, vol. 56, no. 2, pp. 987-994, May 2010.

[22] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, "Joint Model Reference Software," vol. jm18, 2012.

[23] M. Orlandić and K. Svarstad, "An Area Efficient Hardware Architecture Design for H. 264/AVC Intra Prediction Reconstruction Path based on Partial Reconfiguration," in *IEEE Int. Symp. Design Diagn. Electron. Circuits Syst.*, Karlovy Vari, Czech Republic, pp. 86-91, Apr. 2013.

BIOGRAPHIES

M. Orlandić (S'08) was born in Montenegro in 1984. She received the B.S. and M.S. degrees in electrical engineering at University of Montenegro in 2007 and 2009 respectively.

She is currently working toward the Ph.D. degree at the Norwegian University of Science and Technology-NTNU, Trondheim, Norway. Her research interests include video and image compression, digital IP design, FPGAs, run-time reconfigurable systems on FPGA.

K. Svarstad (M'91) received the M.Sc. and Ph.D. degrees from NTH (now NTNU, Norwegian University of Science and Technology) in 1984 and 1989 in VLSI Design and architectures. From 1990 to 2005 he worked at SINTEF and participated in pilot FPGA and ASIC development projects in satellite communication and digital TV broadcasting, amongst others. He has been a board member for a technology startup company and an electronic arts center, and for several years leader of the national VHDL users' group in Norway. From 2006 he has been a professor at the Department of Electronics and Telecommunication at NTNU. His teaching and research are in areas such as system-on-chip design, system level languages and descriptions, run-time reconfigurable systems on FPGA, formal and assertion based verification, and probabilistic analysis of state systems.