Efficient Implementation of DCT-based MIMO Channel Estimation on FPGA

Nima Safari, Richard Maiden
Intel Corporations
High Wycombe, UK.
nima.safari@intel.com

Feng Zhou, Sunny Zhang
Intel Corporations
Beijing, China
feng.z.zhou@intel.com

***Abstract*—Channel estimation is a key processing module in next generation of mobile wireless communications. As the radio channel needs to be estimated for each combination of transmit and receive antennas in a multiple-input multiple-output system, the latency and cost-effectivity of the implementation has significant influence on the performance and overall cost of the system. Pilot-based least square channel estimation with cosine domain noise suppression algorithm is described in this paper. Efficient HW implementation of the algorithm is presented with resource utilization and timing performance on FPGA. A multiplexing technique is proposed for time-sharing a single module among multiple antennas reducing resource usages and overall implementation cost. Finally, the accuracy of the module is validated by comparing against MATLAB simulation.**

Keywords— Channel Estimation; FPGA; Discrete Cosine Transform; Fast Fourier Transform; MIMO; resource sharing; Antenna Multiplexing; 5G

I. INTRODUCTION

The fifth Generation (5G) of mobile communication networks promises higher capacity and data rates everywhere anytime. One of the key technologies to deliver 5G capacity promises is the Massive Multiple-Input Multiple-Output (Massive MIMO) antenna schemes [1]. In such system, tens or hundreds of antennas will be deployed at base station (BS) to enable significant improvement of cell spectral efficiency through the multiple user spatial multiplexing. Many efforts are presented to study on the massive MIMO algorithm, implementation and deployment strategies in practical usage. In [2], the high throughput such as 10Gbps Massive MIMO system is studied from theory to practice. Moreover, [3] and [4] give testbed of massive MIMO system which gave solid evidence of the advantages and feasibilities to deploy massive MIMO in future 5G network.

One of the key issues in realization and well exploiting the full benefit of massive MIMO technology is to get the accurate channel state information. A practical way to acquire instantaneous Channel State Information (CSI) of the MIMO channel is training-based channel estimation, with transmitting known sequence, and receiving and estimating it at the receiver. For massive MIMO Time Division Duplexing (TDD) system, with channel reciprocity, Downlink (DL) channel state information can be obtained from Uplink (UL) at Base Stations (BSs). In UL, the instantaneous channel state information for all transmit and receive antennas pairs are required to be estimated at the BS. When the antenna number is large such as 64, 128 and even higher, the complexity of channel estimation implementation has a significant impact on the overall system cost and complexity [5].

After reviewing different channel estimation algorithms, we chose Least Square estimation followed by Discrete-cosine Transform (DCT)- based noise suppression targeting feasibility and performance. This paper provides efficient FPGA implementation of the channel estimation algorithm for multiple antennas for practical massive MIMO system which can be deployed in remote or distributed radio units. Optimization techniques for design multiplexing to reduce resource utilization as well as improving processing latency are presented and discussed.

The paper will be organized as follows: in section II, we describe the massive MIMO system model. Section III summarizes the channel estimation algorithms and describes the DCT based channel estimation algorithm and procedure. Section IV describes the detail of HW implementation. And section V gives the implementation results. Finally, section VI draws conclusions.

II. MASSIVE MIMO SYSTEM MODEL

A simple block diagram of Massive MIMO system based on Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme, with N_R receiving antennas and N_T transmitting antennas is shown in Fig 1.

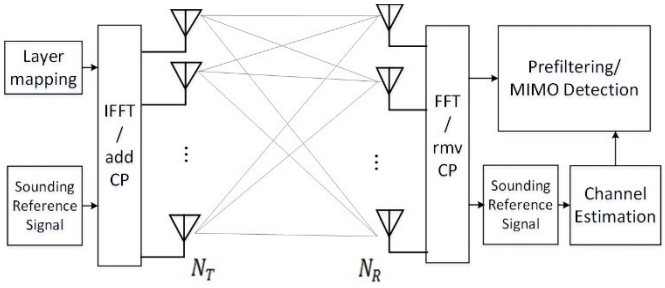

Fig. 1. Massive MIMO-OFDM system model

Assuming x_k is the transmitted signal from kth transmitting antenna, the transmit vector is then denoted as $X = [x_1, x_2, \dots, x_{N_T}]^T$. Denoting $h_{i,k}$ as the channel response in frequency

domain between the k-th transmitting antenna and the i-th receiving antenna, the channel frequency response matrix between transmit and receive antennas is then given by $H = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{N_T}]$, where $\mathbf{h}_k = [h_{1,k}, h_{2,k}, h_{3,k}, \dots h_{N_R,k}]^T$ gives the channel response vector from kth transmitting antenna to all the receiving antennas. The received signal in frequency domain, Y, can be denoted as:

$$Y = H \cdot X + N \tag{1}$$

where N is the additive noise vector in frequency domain.

As shown in Fig. 1, our proposed system insertes Reference Signal (RS) in the transmit radio frame along with the data before IFFT-based OFDM modulation. Sounding Reference Signal (SRS) is used to estimate the Uplink radio channel response [13]. A base sequence of Zadoff-Chu sequence is selected for the SRS realization. To support Multi-user MIMO, many users deploy the same subcarriers for sending their reference signals. Cyclic shifted Zadoff-Chu sequence can be used for different User Equipments (UEs) sharing the same subcarriers to achieve orthogonality, and hence interference-free channel estimation.

Estimating the radio channel for the entire transmit-receive antenna combinations, the CSI information is used for Prefiltering for antenna combination and final MIMO detection. In this paper, we mainly focus on the design and implementation of the channel estimation module, and other MIMO processing units are outside of the scope of this paper.

III. Channel Estimation Algorithm

For MIMO-OFDM system, Least-square (LS), and Minimum Mean Square Error (MMSE) are all well-known algorithms used for channel estimation. In LS estimation, there's no consideration on the noise effect, so the performance will be degraded specially in low Signal to Noise Ratio (SNR) environment. In MMSE based channel estimator, the statistical characteristics of the channel including the channel autocorrelation matrix and SNR needs to be obtained or assumed in advance. DCT- based channel estimation [6-11] is introduced to provide an acceptable compromise on the performance and complexity compared to MMSE estimation. Similar to LS-based estimation, DCT-based approach doesn't need the statistical information of the channel. DCT-based channel estimation shows significantly better performance in low SNR as compared to LS estimators. Getting a good compromise of performance and cost-effectivity, we choose the DCT based solution for our channel estimation implementation.

The DCT based channel estimation is illustrated in Fig 2.

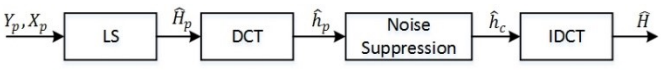

Fig. 2. DCT-based Channel Estimation

The received pilot signal, Y_p, can be described as:

$$Y_p = H_p X_p + N \tag{2}$$

Where X_p is the known transmitted reference signal, H_p, is the channel gain affecting the pilot signal, and N is the independent additive white Gaussian noise. Upon receiving the reference signal, DCT-based channel estimation is performed as follows:

- First, least-square estimator (LS) is used to estimate the channel frequency response at all pilot subcarriers. As X_p is known is the receiver, the LS estimation of the channel gain, $\hat{H}_p$, is obtained by dividing the received pilot by the reference signal:

$$\hat{H}_p = \frac{Y_p}{X_p} = H_p + N/X_p \tag{3}$$

- Second, DCT operation is performed for uncorrelated noise suppression. DCT linear transformation expresses data points in terms of a sum of cosine functions oscillating at different frequencies. For correlated data points, DCT has a compression property that can express the input data by using fewer coefficients in cosine domain. DCT operation is performed on the LS estimator output as follows:

$$\hat{h}_p(m) = w(m) \sum_{k=0}^{M-1} \hat{H}_p(k) cos \frac{(2k+1)\pi m}{2M}, \tag{4}$$

$$m = 0,1, \dots, M-1$$

$$w(m) = \begin{cases} \frac{1}{\sqrt{M}}, & m = 0; \\ \sqrt{\frac{2}{M}}, & m \neq 0; \end{cases}$$

Here M is the number of DCT points which is the same as the number of subcarriers used in the reference signal.

- Third, we select and truncate the proper length of the channel impulse response in DCT domain. The length L is typically related to the length of cyclic prefix (CP) which can cover the maximum path delay of the channel characteristics. The noise suppressed channel response is Cosine domain, $\hat{h}_c$, is then given by:

$$\hat{h}_c(m) = \begin{cases} \hat{h}_p(m), & 0 \leq m \leq L-1; \\ 0, & L \leq m \leq M-1; \end{cases} \tag{5}$$

- Finally, IDCT is performed on noise-suppressed channel values to get the estimated channel frequency response.

$$\hat{H}(k) = \sum_{m=0}^{M-1} w(m) \hat{h}_c(m) cos \frac{(2k+1)\pi m}{2M}$$

$$k = 0, 1, \dots, M-1$$

where $w(m)$ is the same as given in DCT operation in equation (4).

IV. HW Implementation

This section goes through the HW-optimized implementation of the channel estimation algorithm described in previous

section. The design is generally optimized for lower latency, higher FPGA performance, and lower resource utilization.
In our proof of concept Massive MIMO setup, we assumed 64 physical antennas in the base station (BS) side, and 16 UE streams in the downlink. Each two receive antennas are combined digitally in the FPGA to form a total of 32 Virtual Antennas (VAs), i.e. $N_T = 16$, $N_R = 32$ in Fig 1.
OFDM modulation is performed for each UE stream through dedicated 2048-point IFFT modules with 1200 subcarriers, and subcarrier spacing of 75 KHz, resulting in 100 MHz RF bandwidth after pulse-shaping. Each radio is assumed to handle uplink digital down conversion and channel estimation for two VAs, so a total of 16 radios (Arria10 SoC FPGAs) are used to cover BS. Considering the channel reciprocity in TDD system, we use sounding reference signal to get the channel response which can be for both the uplink and downlink processing. To support up to 16 streams for downlink, total 16 SRS are required for channel estimation. Allocating 4 users reference signals in each radio frame of 2 ms, we can estimate the entire number of channels in 4 radio frames i.e. 8 ms. Each radio will sound 4 users x 2 VAs reference signals, and hence it needs to estimate the channel for 8 SRS in every radio frame of 2 ms.
Intel's high level HW synthesis tool, DSP Builder Advanced (DSPBA) is used to implement the channel estimation. DSPBA is a model-based design tool that converts block design into efficient RTL code targeting Intel's FPGA devices.
As described in section II, the channel between each UE and BS antenna is estimated by using received SRS and locally generated Reference Signal (RS) in the Uplink radio. The channel estimation module is intended to run at 491.52 MHz clock frequency for optimal latency and resource sharing. The input is 16 bit fixed-point complex data, and the entire channel estimation module is implemented in fixed-point data format. The implementation of the DCT-based CE is described in four stages as given below:

1) Least Square Estimation based on Zadoff-Chu Reference Signal:

As the reference signal is known in the Uplink for a UE, the least square estimation of the channel complex gain is obtained by dividing the received SRS signal by the reference sequence. The reference signal for a UE is a Zadoff- Chu sequence with constant amplitude and unique root and user specific cyclic shift. Hence the division is implemented by a multiplication with the complex conjugate of the reference signal as shown in Fig 3. We used RAM blocks to store reference sequences for the all UEs. The channel complex gain after least square estimation is noisy and therefore is further processed in Cosine domain for noise suppression.

2) DCT Implementation:
DCT operation is resource demanding if implemented directly. DCT can be implemented through Discrete Fourier Transform (DFT) operation [12]. As DFT is efficiently implemented using Fast Fourier Transform (FFT), the algorithm is called Fast Cosine Transform (FCT).

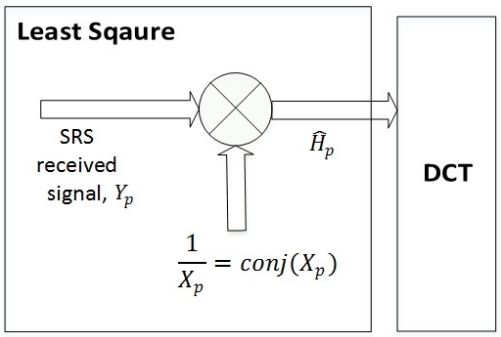

Fig. 3. Least Square Estimation of the channel using reference signal

Using FCT the computation complexity drops from $O(M^2)$ to $O(MlogM)$.
Fig 4 shows the block diagram of the FCT implementation in HW. FCT is implemented through three stages i.e. input reordering, FFT operation, and vector rotation.

Input Reordering:
The noisy LS-estimated input needs to be reordered first before the FFT. The reordering is implemented using a dual memory block. A compile-time configurable RAM block with the depth equal to the input size is deployed to implement the reordering.

$$v(n) = \begin{cases} \hat{H}_p(2n) & 0 \le n \le M/2 \\ \hat{H}_p(2M - 2n - 1) & \frac{M}{2} < n \le M - 1 \end{cases}$$

FFT operation:
A full radix-2 streaming implementation is used for the FFT operation. The block can be used for fixed-point or floating-point data. Here we used 16-bit fixed-point implementation with natural order input and bit-reversed output.

$$V_k = FFT\,[\,v(n)]$$

The FFT block is compile-time reconfigurable to adapt the input vector length. If the length of the input vector is non-radix-2, the input vector is zero-padded to reach the closest radix-2 number, before feeding the input to the FFT block.

Vector Rotation:
The FFT output needs to be phase rotated to obtain the DCT values, as given below:

$$\hat{h}_p(k) = \begin{cases} V_0 & k = 0 \\ V_k \cdot e^{-\frac{i\pi k}{2M}} + V_{N-k} \cdot e^{\frac{i\pi k}{2M}} & 0 < k \le M - 1 \end{cases}$$

As the FFT output is in bit-reversed order, a configurable RAM block is used to reorder the output to natural order before phase rotation. A single RAM block is used to combine bit-reverse and reordering operations to save resources and latency. As shown in Fig 4, reading the complex phasor from a separate RAM block, we phase-rotate the FFT samples to generate the DCT values.

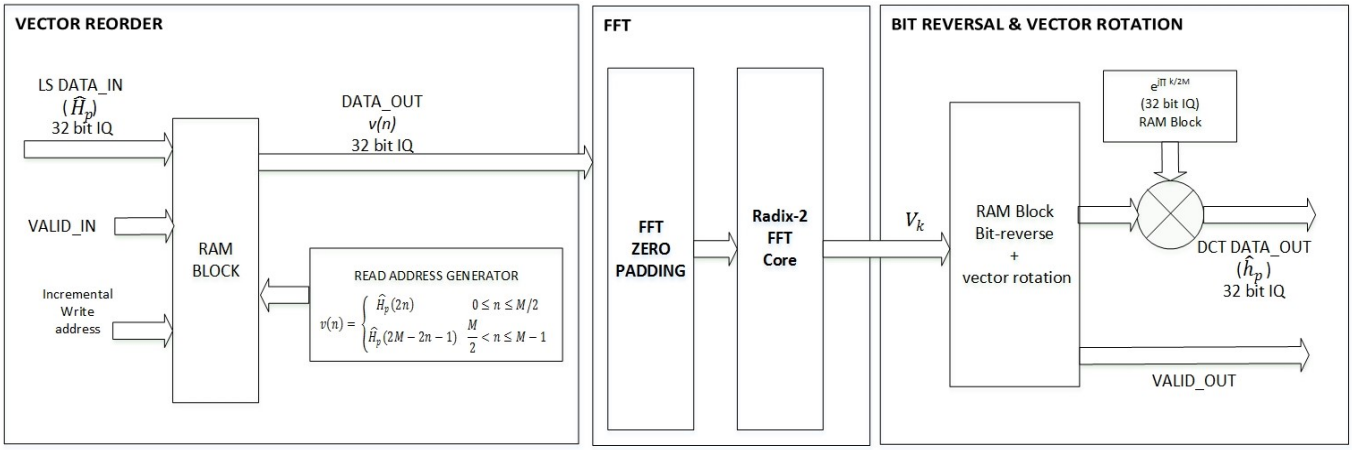

Fig. 4. Discrete Cosine Tranfrom (DCT) HW implementation

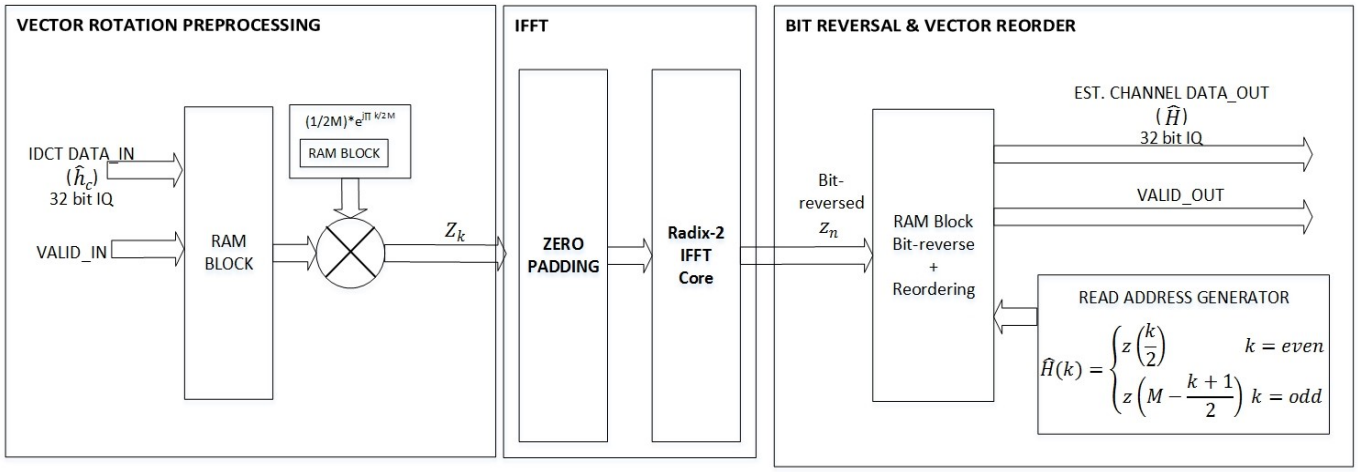

Fig. 5. Inverse Discrete Cosine Tranfrom (IDCT) HW implementation

3) Noise Suppression:

Selecting the most significant coefficients of the channel gain in Cosine domain, the module provides smoother and less noisy channel estimation. The channel gain vector is truncated according to the maximum length of the channel impulse response. Here we used the Cyclic Prefix (CP) length to select the most significant coefficients of the channel estimated values.

4) IDCT Implementation

The channel estimated values after noise suppression are transformed back to frequency domain by the inverse DCT operation. Similar to DCT, IDCT can be efficiently implemented using Inverse Fast Fourier Transform (IFFT), and through reverse stages as compared to the DCT i.e. *vector pre-processing*, *IFFT operation* and *data reordering*:

Vector Pre-Processing:

The input vector needs to be pre-processed and phase-rotated first before the IFFT operation. Assuming $\hat{h}_c(k)$ is the input vector, the output of reordering module, Z_k is given by:

$$Z_k = \begin{cases} \hat{h}_c(k) & k = 0, M/2 \\ e^{\frac{i\pi k}{2M}} \cdot \hat{h}_c(k) & 1 \le \mathrm{k} < \mathrm{M}/2 \\ e^{-\frac{i\pi(M-k)}{2M}} \cdot \hat{h}_c(M-k) & \frac{N}{2} + 1 \le k \le M-1 \end{cases}$$

IFFT operation:

A full radix-2 streaming implementation is used for the IFFT operation.

$$z_n = IFFT\,[\,Z_k]$$

If the length of the input vector is non-radix-2, the input vector is zero-padded to reach the closest radix-2 number, before feeding the input to the IFFT block.

Vector Reordering:

Finally, the IFFT output needs to be re-ordered to obtain the IDCT values, as given below:

$$\hat{H}(k) = \begin{cases} z\left(\frac{k}{2}\right) & k = even \\ z\left(M - \frac{k+1}{2}\right) & k = odd \end{cases}$$

Fig 5 shows the block diagram of the IDCT HW implementation. The HW resources to implement each individual stage are also similar to those in DCT implementation. Again bit-reverse operation after the IFFT and the data reordering are combined in a single dual-port RAM block to save resources as well as latency.

For N_T users and N_R base station (BS) receive antennas, we need to estimate $N_T \times N_R$ channel gains over the entire received OFDM subcarriers. The estimated channel gains are then sent to the Server together with the received data for MIMO processing.

V. Implementation and Validation Results

A. FPGA resource usage and performance

The implemented channel estimation is then compiled in Intel's Quartus II synthesis tool. Table 1 shows the resource usage of a single instance of the DCT-based channel estimation module in both Arria 10 and Stratix 10 Intel FPGA devices.

Table 1 Resource count of the DCT-based channel estimator

Device / Resource	Logics (ALMs)	Memory (RAM)	DSP Blocks
Arria 10	5140 (2%)	528 kbits	44
Stratix 10	6373 (< 1%)	528 kbits	44

As we required the module to run at 491.52 MHz clock rate, the entire design is required to close timing at *Fmax* of higher than the clock rate. *Fmax*, is defined as the maximum frequency that the entire module can run without timing violation when targeting a system clock rate. Table 2 shows the *Fmax* values when the design is synthesized targeting Arria 10 and Stratix 10 Intel FPGA devices and system clock rate of 491.52 MHz.

Table 2 Fmax report of the channel estimation module.

Intel Devices / Performance	Arria 10 FPGA	Stratix 10 FPGA
Fmax (MHz)	549.45	581.17

B. Latency:

5G has a strict requirement for communication latency. Therefore using FPGA and its concurrent processing power, we can achieve the 5G latency requirement. As the number of Antennas tends to increase for 5G such as in massive MIMO systems, the number of concurrent channel estimation module grows as well. Hence an efficient channel estimation algorithm as well as implementation has a direct influence on the overall communication latency.

The HW-friendly proposed implementation of the DCT-based channel estimation can run at a clock rate of more than 0.5 GHz, minimizing the processing latency. The overall latency of the implemented module for 1200 subcarriers at 491.52 MHz clock rate is measured as *23.84 μs.* Table 3 presents the latency for the sub-modules for comparison.

Table 3 Processing latency of the DCT-based channel estimation

Submodule / Latency	Least Square	DCT	Noise Suppression	IDCT	Total
clock cycles	5	5438	5	6274	11722
Time (μsec)	0.01	11.06	0.01	12.76	23.84

C. Resource sharing and antenna multiplexing

In MIMO systems, many channels are required to be estimated each time a SRS symbol arrives. Instantiating a single channel estimation module for each radio channel, we end up utilizing massive logics and HW resources which is inefficient and costly. As each radio device in our setup handles 8 radio channels, we required to operate 8 channel estimations for every radio frame of 2 ms. Here we propose a multi-channel implementation of the DCT-based channel estimation to pipe-line sequential sub-modules to share resources for a cost-effective implementation. Fig 6 shows timing diagram of the sub-module pipe-lined implementation of the DCT-based channel estimation. The channel gain after least square estimation are stored in RAM blocks, and Ready signals control the interface of the submodules and passing the data from different channel RAMs into the sub-modules sequentially. As shown the entire latency of 8 channel estimations for 1200 subcarriers (M = 1200) using a single instantiation of the DCT-based channel estimation module is measured as *58.44* μs. We should point out that the proposed multi-channel DCT design is flexible to support more antennas.

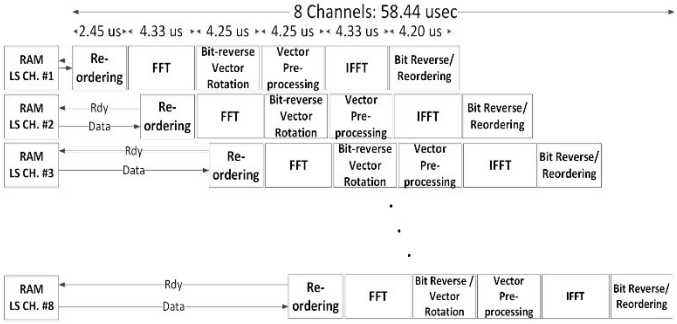

Fig. 6. Multi-channel multiplexing of 8 radio channel estimations using DCT-based channel estimation.

D. HW Validation

The implemented channel estimation is validated by capturing the data at the input and output of the module and comparing the estimated complex channel gains with Matlab simulation for the same input and output data.

Fig. 7 shows the magnitude of the least square estimated channel gain after applying the DCT operation. As shown in the figure, the significant information of the channel gain is compressed and allocated at the beginning of the DCT output vector.

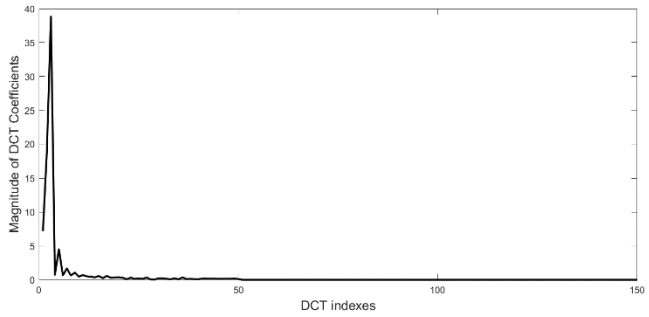

Fig. 7. Least Square estimated channel gain in Cosine domain.

Fig 8 shows the least square channel estimated values captured before DCT-based noise suppression, together with HW DCT-based channel estimated versus Matlab simulation. The noise suppression window length of 128 is applied for this testing. The estimation is obtained using 1200 subcarriers. As seen the LS estimator based on known RS lets the noise going through the estimation results. The estimated channel gains are then captured after DCT-based noise suppression at the output of the iDCT and compared to Matlab simulated output to validate the entire chain of the channel estimation. We should point out that in our Matlab simulation we used Matlab 1200-point DCT function in floating point, whereas in HW, the DCT is implemented through 2048-FFT module in 16-bit fixed point format. As seen in the figure, an accurate match with Normalized Mean Squared Error (NMSE) of around -50 dB is obtained against simulation.

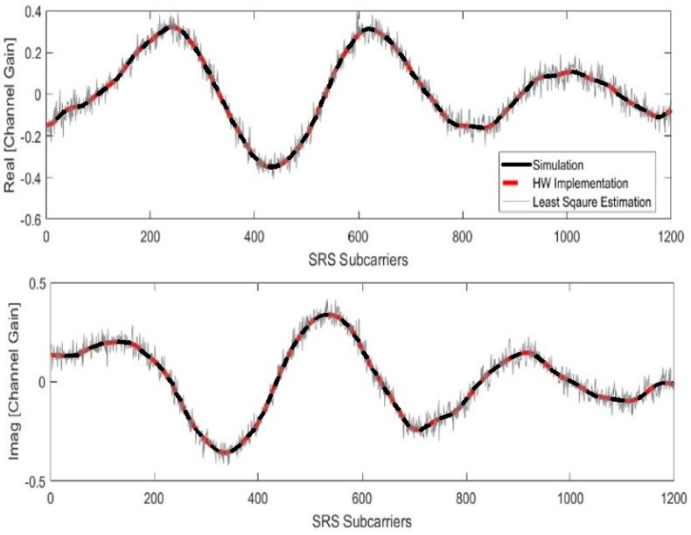

Fig. 8. Comparison of the HW DCT-bassed channel estimated versus Matlab simulation with noise suppression length of 128. NMSE = 1e-5.

VI. CONCLUSIONS

In this paper, we presented an efficient implementation of DCT-based channel estimation for massive MIMO systems. The requirements for the HW implementation is assumed based on a 64 x 16 antenna setup, and 4 user reference signal per radio frame of 2 ms. The design is optimized for lower latency, higher FPGA timing performance, and lower FPGA resource utilization by time-sharing between 8 antennas. 8 antenna channels are estimated in 58.44 us. FPGA resource counts are given after synthesis. Finally, the accuracy of the HW implemented channel estimation is compared with Matlab simulation.

ACKNOWLEDGMENT

The authors would like to thank IntelLab team in China and Intel's Wireless System Solutions Group and DSP builder team in UK for their kind support, guidance and advices.

REFERENCES

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G", *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74-80, Feb. 2014

[2] B.Panzner, W.Zirwas, S.Dierks, M.Lauridsen, P.Mogensen, K.Pajukoski and D.Miao, "Depolyment and implementation Strategies for Massive MIMO in 5G", in IEEE Global Com Workshop on Massive MIMO: from theory to practice, Dec. 2014, pp 346-351.

[3] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I.Wong, V. Owall, O. Edfors, and F. Tufvesson, A flexible 100-antenna testbed for Massive MIMO, in IEEE Global Com Workshop on Massive MIMO: from theory to practice, Dec. 2014, pp. 287-293.

[4] C.Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical many-antenna base stations," in *ACM Intern. Conf. on Mobile Computing and Net. (Mobicom)*, Istanbul, Turkey, Aug. 2012, pp. 53–64.

[5] A. M. K., "A Simple Scheme of Channel Estimation in Large MIMO Systems," 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, 2016, pp. 1-5.

[6] Y. H. Yeh and S. G. Chen, "DCT-based channel estimation for OFDM systems," Communications, 2004 IEEE International Conference, vol. 4, pp.2442-2446, June 2004.

[7] Y. H. Yeh and S. G. Chen, "Efficient channel estimation based on discrete cosine transform," *Proc. IEEE Int'l. Conf. Acoust., Speech, and Signal Processing*, vol. 4, Hong Kong, China, Apr. 2000, pp. 676-679

[8] C. Qi, G. Yue, L. Wu, Y. Haung, A. Nallanthan, Pilot design scheme for sparse channel estimation in OFDM systems, IEEE trans. On Veh. Tech. 64, 2016, pp. 1493-1505.

[9] H. Zhang, H. C. Wu, Robust pilot detection techniques for channel estimation and pilot detection in OFDM systems, IEEE signal process. Lett. 22, 2015, pp. 733-737.

[10] F. Cruz-Rolden, M. E. Dominguez-Jimenex, G. Sansigre-Vidal, D. Luengo, M. Moonen, DCT-based channel estimation for single- and multi-carrier communications, Signal Process. Journal 128, 2016, pp. 332-339.

[11] L. Xiaobai, Y. Ruijuan, Ch. Wei, The truncation DCT channel estimation for OFDM system, IEEE International Conf. on Intelligent System Design and Eng. App., 2012.

[12] J. Makhoul, "A fast cosine transform in one and two dimensions," IEEE Trans. Acoustics. Speech Sig. Proc. 28 (1), 27-34 ,1980.

[13] LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation. 3GPP TS 36.211 version 10.0.0 Release 10