CREATING THE LIVING NETWORK™

First Open Air North America Workshop
5G LDPC Implementation on a GPU
June 25-27, 2019

Software implementation of an NR LDPC on a GPU intended to replace an FPGA implementation

INTERDIGITAL®

Agenda

- Describe LDPC and why it was chosen for NR
- Challenges of implementing LDPC decoder on a GPU
- SNR vs BLER Results for GPU implementation
- 28GHz PAA NR Platform Overview

Introduction to LDPC codes

- The 3rd Generation Partnership Project (3GPP) has standardized Low Density Parity Check (LDPC) coding as the solution to satisfy the channel coding demands of 5G NR
- LDPC codes (developed by Gallagher at MIT in 1963 and recently rediscovered in 1996) is recognized as a significant breakthrough in channel capacity
- Performance data has shown that LDPC codes can offer higher coding gains, lower error floors and throughput that increases as the code rate increases in comparison to turbo codes
- LDPC codes advantages
  - High gain, good immunity to burst errors
- LDPC codes disadvantages
  - Decoding complexity
  - Latency

Linear Block Codes

- LDPC codes are an extension of linear block codes such as Hamming codes
- In the below example, c_1,c_2,c_3 and c_4 are information bits and c_5,c_6, and c_7 are parity bits. The 3 parity equations are:

$$c_1 \oplus c_2 \oplus c_3 \oplus c_5 = 0$$
$$c_1 \oplus c_2 \oplus c_4 \oplus c_6 = 0$$
$$c_1 \oplus c_3 \oplus c_4 \oplus c_7 = 0.$$

  - The $\oplus$ symbol is an exclusive or operation. The parity Matrix would be:

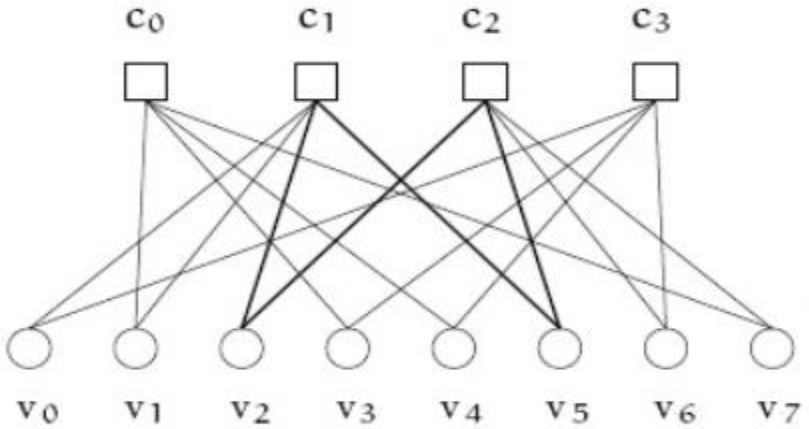
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

  - Length of the code n=7, r=3 equations or parity check bits and k information bits so k=n-r

Tanner Graphs

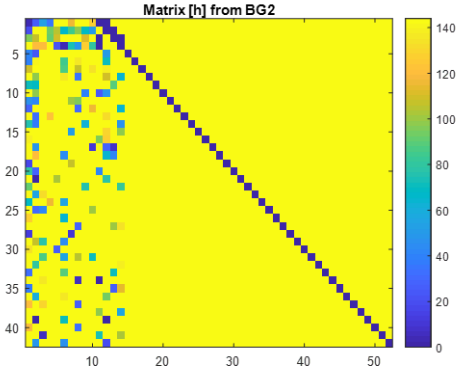
- Bi-partite graph representing parity-check equations.
- The variable nodes represent the codeword bits and the check nodes represent the parity-check equations
- The dependency and messages sent during decoding iterations are shown as lines between nodes
- The messages also represent probability distribution on the associated bit

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$



Enumeration of non-zero entries of a parity-check matrix H and its relation to the corresponding Tanner Graph

LDPC Code for 5G



Example of matrix [h] for BG2, Zc=144
82 compressed [h] matrices occupy only 9kB of local memory.

- Section 5.3.2 of 38.212 specifies tables for two different base graphs. Base graph is chosen as a function of the desired transport block size and code rate.
- The parity check matrix *H* is generated by replacing each element of the compact matrix [h] with a Zc x Zc matrix
- Each blank element of the [h] matrix is replaced by an all-zero matrix of size Zc x Zc;
- Each non-blank element of the [h] matrix is replaced by a circular permutation of an identity matrix of size Zc x Zc. The size of the parity check matrix *H* generated this way is (46Zc) x (68Zc) for BG1, and (42Zc) x (52Zc) for BG2, respectively.
- Impractical to store full matrix

With compression, we can store in a 9KB block of local memory a matrix that would have otherwise taken over 300KB.

LPDC Decoder Implementations

- We have two implementations: FPGA and GPU
- FPGA gives low latency, but GPU allows faster development time
- Many processes in the decoder are parallelized, so GPU with hundreds CPU core should fit better for this implementation
- Comparison of some KPI presented below

Implementation	Development Time	HW Cost	Decoder Latency
FPGA	Long	High	Very Low
GPU	Faster	Lower	Longer

Analysis of the results

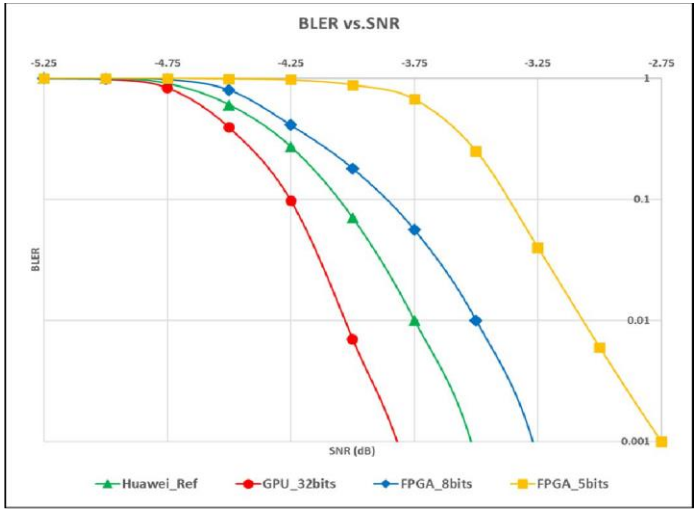
The average power consumption for the GPU was calculated as the difference between the power consumption during idle mode and normal operation of the decoder at full speed using a single multi-processor

Parameters (Zc=384)	GPU decoder	Target	FPGA decoder
BG1 (R=1/3)	520us/iteration	178us/iteration **	4.25us/iteration
BG2 (R=1/5)	330us/iteration	178us/iteration **	3.14us/iteration
Average Power Consumption	~5W* 1 multiprocessor +1/68 of power for memory and clock		7.5W

**For 120KHz subcarrier spacing, u=3, category 1 service

GPU has longer latency in this intermediate step, the path to smaller latency will be presented in "Next steps" section

Performance



BG2, QPSK, Coding Rate R=1/5 (50 Iterations)

- Test results for
  - 50 iterations max
  - BER vs SNR
  - BG2
- GPU LDPC decoder
- Huawei reference decoder
- 8bit FPGA LDPC decoder
- 5bit FPGA LDPC decoder

GPU LDPC and FPGA LDPC decoders use Sum – Product algorithm

Huawei reference uses Flooding Belief Propagation algorithm

Next Steps

- We plan to implement the following enhancements for latency reduction:
  - We expect to see about two-fold latency reduction by switching to the latest tools and paralleling our implementation to 2048 threads (in Turing)
  - Reduce the amount of lines being processed in matrix H when operating with large R, e.g. with R=1/2 thus, the amount of operations and latency will be reduced
  - Introducing fixed point (8-bit operation) in our implementation. We expect to have at least 4 times processing time reduction vs. 32-bit operation using Turing tensor cores INT8.
  - Combining two or more lines with low correlation from matrix H when we form variable node messages will reduce the latency by at least 2 times
- We expect to see a total latency reduction that is greater than 32 times improved after successful implementation of the steps mentioned above

NR SDR Platform – Live OTA demonstration

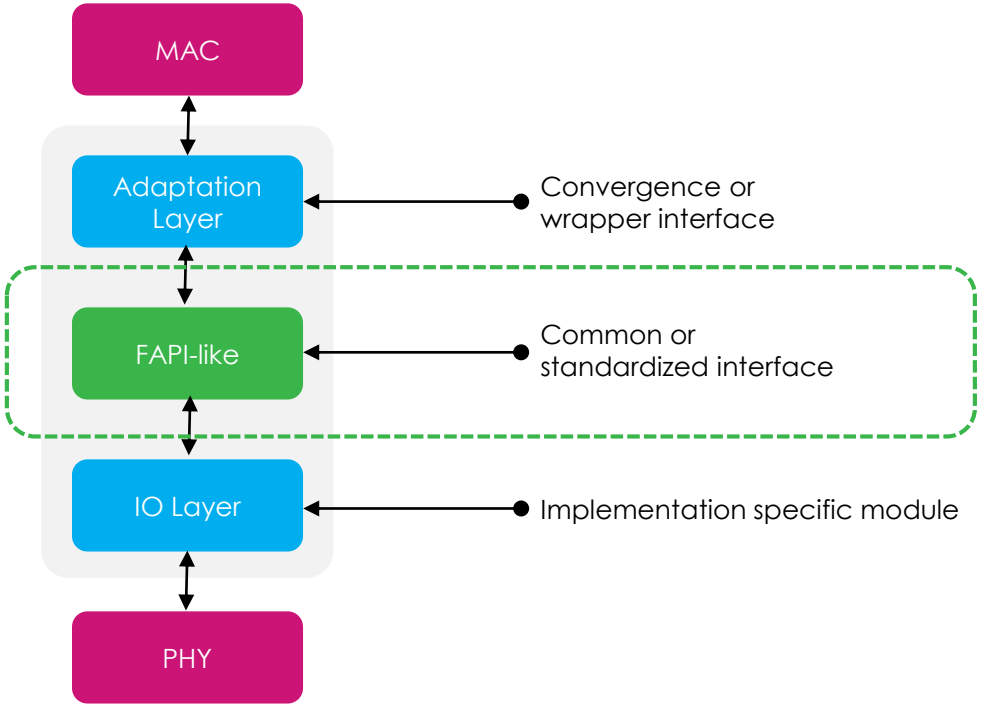
SDR Platform for NR Research and Development

- Low cost SDR platform with beamforming capability
- 5G NR Rel15 PHY
- Up to +40 dBm RMS @ 200MHz BW
- Capable of HOM up to 256 QAM
- MHU outdoor ready, up to 100 Ft between units
- Inertial Measurement Unit (IMU) provides assistance for customer install and service monitoring
- Low Power <50 W @24VDC
- Small form factor

Open MAC-PHY Interface

Motivation

- Enable the integration of 3rd party PHY with OAI L23 stack
- Leverage OAI NR stack and CI framework as a common baseline for testing and validation
- Provide an open reference implementation
- Provide new capabilities and collaboration opportunities in development, research and testbeds



Summary

- We discussed our enhanced 5G NR LDPC decoder
- We compared the enhanced 5G NR LDPC decoder on a FPGA & GPU
  - Presented our BLER vs SNR results from the FPGA & GPU with a 3GPP design and achieved good results
  - Decoder latency on FPGA was significantly better… at this point!
- GPU based implementation may allow for an all software based implementation with real-time performance for > 6 GHz NR @120KHz SCS.
  - No Specific FPGA HW platform and RTL design knowledge is required for modifications.
  - Pure CPU implementation would cause a huge performance decrease.
  - Runs from a high end desktop with NVIDIA graphics card in standard PCIe slot.
- Plan to make source code available to OAI 5G NR Gitlab in later this year!