

---

# Fall 2021 OpenAirInterface Workshop

LDPC GPU acceleration presentation

---



# LDPC GPU Acceleration

- OAI organisation
  - A. Same system for variable HW like RF boards or HW accelerators
  - B. Shared lib system
    - The .so file is linked at runtime (dlopen()call)
    - Fixed function names by OAI convention
    - Use the configuration to know what .so file to load
- Available CUDA based LDPC decoder
  - openair1/PHY/CODING/nrLDPC\_encoder/ldpc\_encoder\_optim8segmulti.c
- Compilation
  - Cd cmake\_targets/ran\_build/build
  - Ninja/make ldpc\_cuda ldpc\_test
  - Creates libldpc\_cuda.so

# Test and results

- Unitary tests are performed today
  - Global benchmark to do with CPU, T1 Xilinx
- Ldpctest
  - No option
    - cpu regular version (loads libldpc.so)
  - Ldpctest -v \_cuda
    - Load libldpc\_cuda.so
- Result
  - GPU implementation is today not faster, but we have a lot of cores
  - Reach Shannon limit is mandatory
    - Limit the number of iterations makes it fat
    - If we want to reach Shannon limit we need more than 10 iterations
      - 1) Whatever CPU, GPU, ... 20 iterations cost 10x 2 iterations
      - 2) Fast CRC check is also important to early detect the decoding is done
      - 3) In lab conditions allow a bit more power, to make better than optional SNR can reach high throughput