An FPGA-based Accelerator for Rapid Simulation of SC Decoding of Polar Codes

Johannes Wüthrich, Alexios Balatsoukas-Stimming and Andreas Burg

Telecommunications Circuits Lab, École Polytechnique Fédérale de Lausanne, Switzerland

Motivation & Contribution

- Polar codes are a new type of **provably capacity achieving** channel codes.
- Polar codes have **no error floor** under successive cancellation (SC) decoding.
- Very attractive for **high speed 5G links** and **storage** applications.

- In general, the **frame error rate** (FER) must be evaluated via lengthy **Monte Carlo simulations**.
- **Polar code construction** can be carried out in the same way.

- SC decoder has complexity $N\log(N)$, but simulating low FER ($\approx 10^{-12}$) is computationally challenging due to data dependencies that lead to **low parallelism**.

- We exploit two observations that allow to fully parallelize the FER simulation of SC decoding **and** polar code construction.
- We present an FPGA-based accelerator $\rightarrow$ **10^8 codewords/s**.
  $\rightarrow$ **10^{12} random bits/s**.

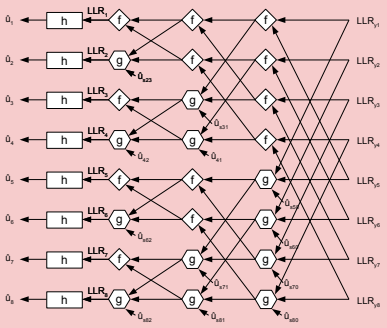
Successive Cancellation Decoding

- The SC decoder decodes bits using LLR values:

$$\mathrm{LLR}_i = \log\left(\frac{W(y_1^N, \hat{u}_1^{i-1}|u_i = 0)}{W(y_1^N, \hat{u}_1^{i-1}|u_i = 1)}\right).$$

- **Butterfly-based** computation graph:
  Input $\rightarrow$ channel LLRs, **Output** $\rightarrow$ decision LLRs
- Decoding of each bit $\hat{u}_i$ requires channel LLRs (y_1^N) and **previously decoded bits** ($\hat{u}_j,\ j < i$).
- Standard SC decoder **can not be fully parallelized** $\rightarrow$ low throughput.

SC decoder computation graph (N = 8)



- **Hardware implementations** employ a min-sum approximation for the calculation of f and g:

$$f(L_1, L_2) = \mathrm{sign}(L_1)\mathrm{sign}(L_2)\min(|L_1|, |L_2|)$$
$$g(L_1, L_2, \hat{u}_s) = (-1)^{\hat{u}_s} L_1 + L_2$$

- $\hat{u}_s$ is a **partial sum** of previously decoded bits.

Fully Parallel Pseudo-SC Decoding

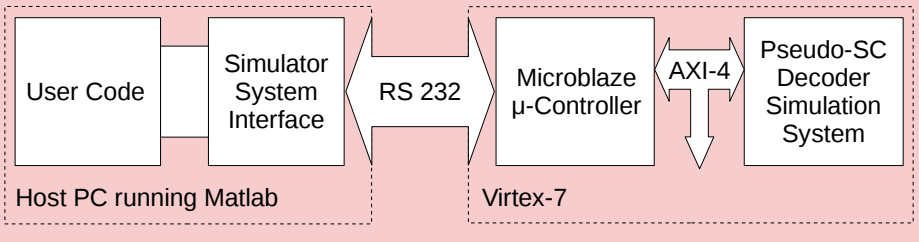
For **rapid FER** simulations, we can exploit two observations:

1. The FER is **independent of the transmitted codeword**.
2. Using the true u_i values or the decision values $\hat{u}_i$ for decision feedback **does not change the simulated FER**.

- **Observation 1** $\rightarrow$ we can assume that the **all-zero codeword** is always transmitted.
- **Observation 2 + all-zero codeword** $\rightarrow$ All partial sums are 0.

- Hence, we can implement a **fully parallelized pseudo-SC decoder** for FER simulations.
- The same decoder can be used to identify the **good synthetic channels** for **polar code construction**.

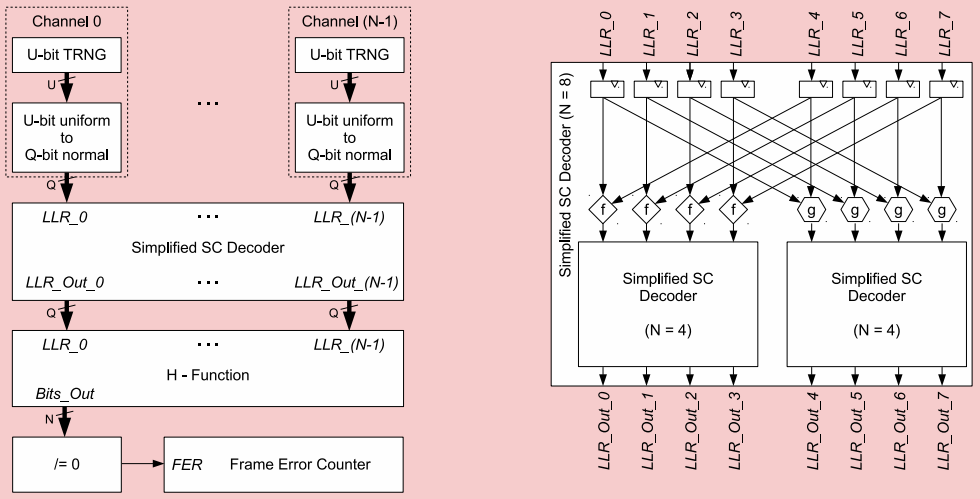
Simulation System Setup



- Different simulation setups are defined by a MATLAB script.
- A Microblaze μ-Controller controls the simulation system.
- Results are fed back to the **MATLAB script for post processing**.
- Channel SNR, polar code definition and number of simulated frames can be **configured on the fly**.

Hardware Implementation

Top-Level Decoder



- Heavily pipelined implementation of fully parallel pseudo-SC decoder.
- Outputs **one decoded frame per cycle**.
- Recursive and generic VHDL description.
- **Fixed-point parameters** and **blocklength** are **configurable** at compile time.

Additive White Gaussian Noise (AWGN) Generation

- Accurate AWGN generation is **challenging** due to Gaussian distribution tails.
- To overcome this, we **directly generate quantized LLRs**.

- U uniform random bits are generated for each LLR.
- The U bits are used as an address for a BRAM-based LUT that performs **translation to the desired LLR distribution**.

- True randomness is generated using **free running XOR oscillators**.

Implementation Results

- System configured with blocklength $N = 1024$ and $U = 10$ bits of randomness per channel LLR.
- Resulting operating frequency of 100 MHz, **10^8 codewords/s**.
- C implementation of pseudo-SC decoder decodes **10^4 codewords/s**.

- Virtex-7 XC7VX485T device utilisation:

	Available	Total Utilization	SC & TRNG
Slice LUTs	303,600	113,018 (37.2%)	111,277 (36.6%)
Slice Registers	607,200	75,756 (12.5%)	74,223 (12.2)%
BRAMs	1,030	273 (26.5%)	257 (24.9%)

- FPGA resources would allow for a $N = 2048$ decoder, but P+R was not able to route the design.

Simulation Results

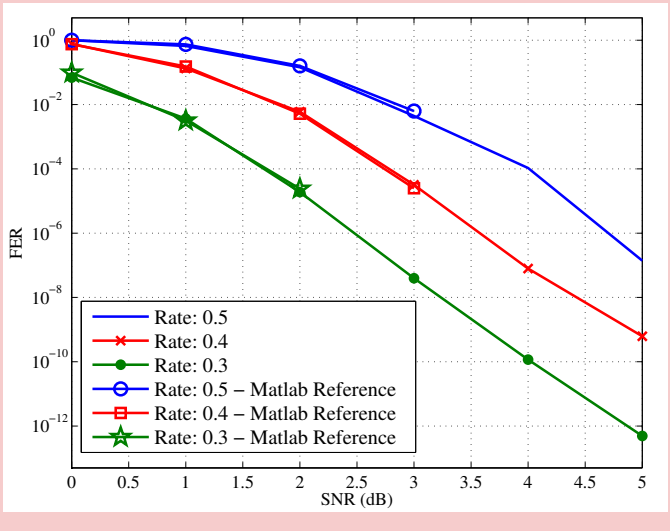


Figure: FER of a polar code with blocklength $N = 512$ under SC decoding.

- FPGA accelerated values **agree** with values calculated via a MATLAB reference implementation.
- Total simulation time @ FER $= 10^{-12}$ is **approximately 4 days**.
- C simulation on our 200-core cluster would require **more than 50 days**.