AFF3CT & Hardware in the Loop

MADE BY:

IAN FISCHER SCHILLING

SUPERVISED BY:

CHRISTOPHE JÉGO

CAMILLE LEROUX

Summary

1. Context
2. Block Schematics
3. Simulations with py-AFF3CT
4. FPGA Implementation

Summary

1. Context
2. Block Schematics
3. Simulations with py-AFF3CT
4. FPGA Implementation

ANR EVASION

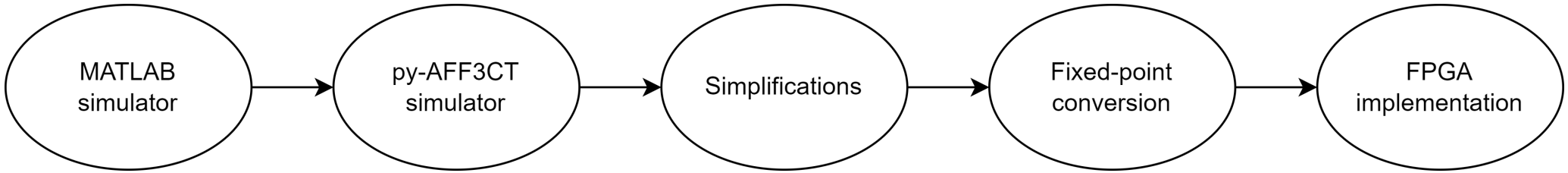
Efficient implementation and optimization of ad**V**anced mess**A**ge pa**S**sing-based rece**I**vers f**O**r future wireless **N**etworks

WP3: Architectural studies and hardware implementation

Objective

Design, implementation and prototyping on FPGA of flexible digital receiver architectures based on Expectation Propagation

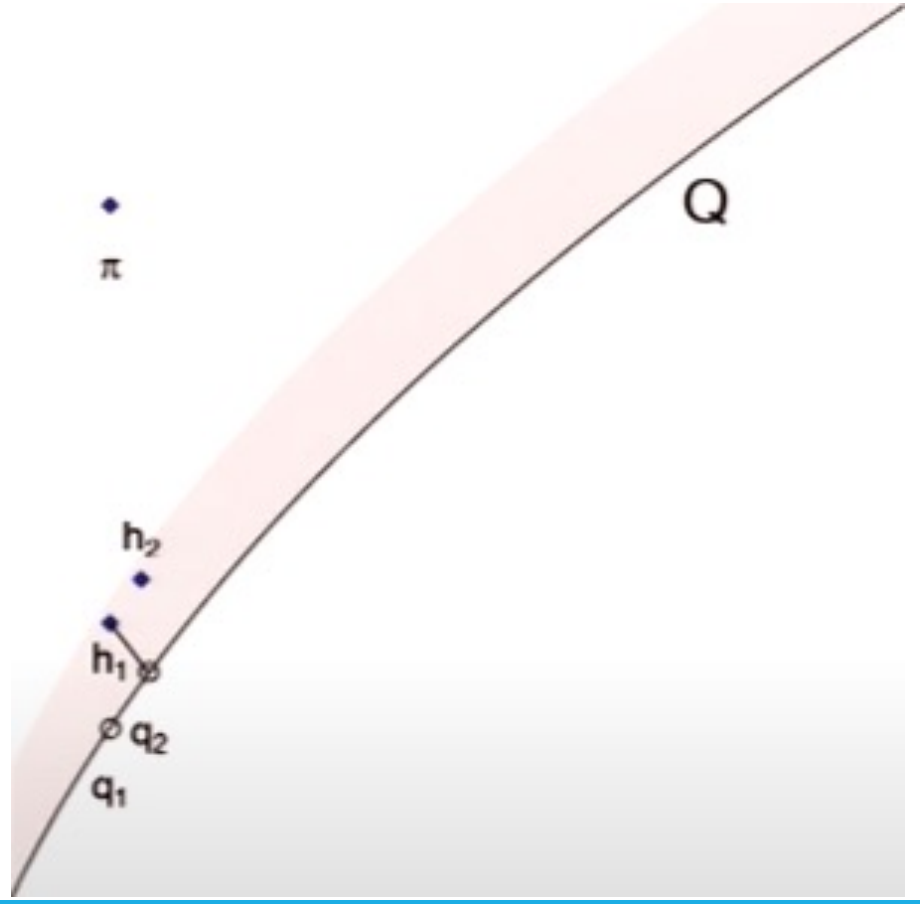
- Use of AFF3CT:
  - Validate and simulate the initial communication chain
  - Test simplifications
  - Fixed-point conversion
  - Co-simulation with the FPGA



Expectation Propagation

Iterative approach to find approximations to a probability distribution

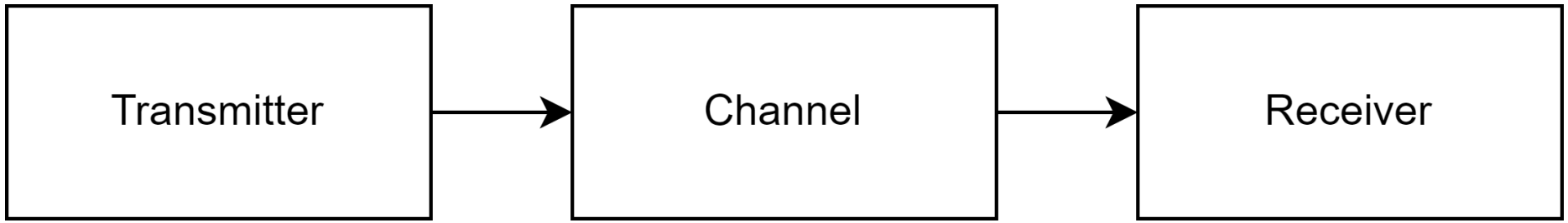
- Q is the space of Gaussians
- π is the target distribution
- We want to project π in Q
- There is a margin near Q that we can use
- Project π in Q to obtain q1
- Introduce hybrid h1
- Project back to q2
- Introduce second hybrid h2
- Repeat



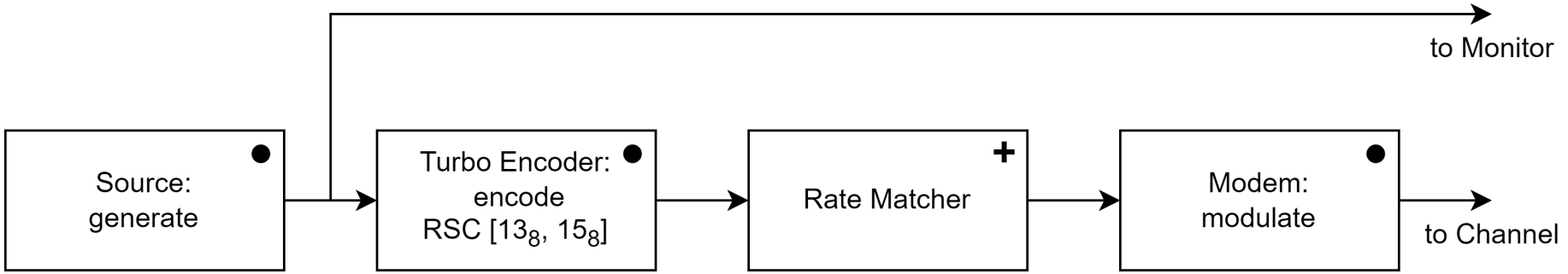
Summary

1. Context
2. Block Schematics
3. Simulations with py-AFF3CT
4. FPGA Implementation

AFF3CT's Architecture



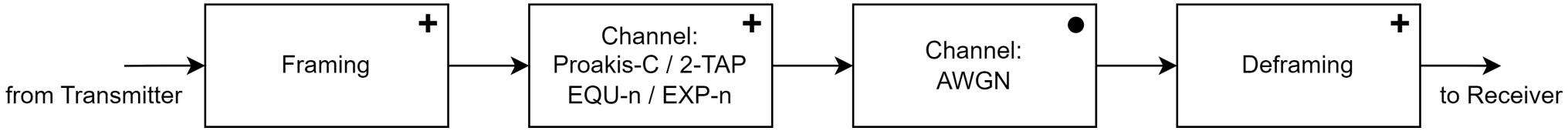
Transmitter



● AFF3CT

✚ Added

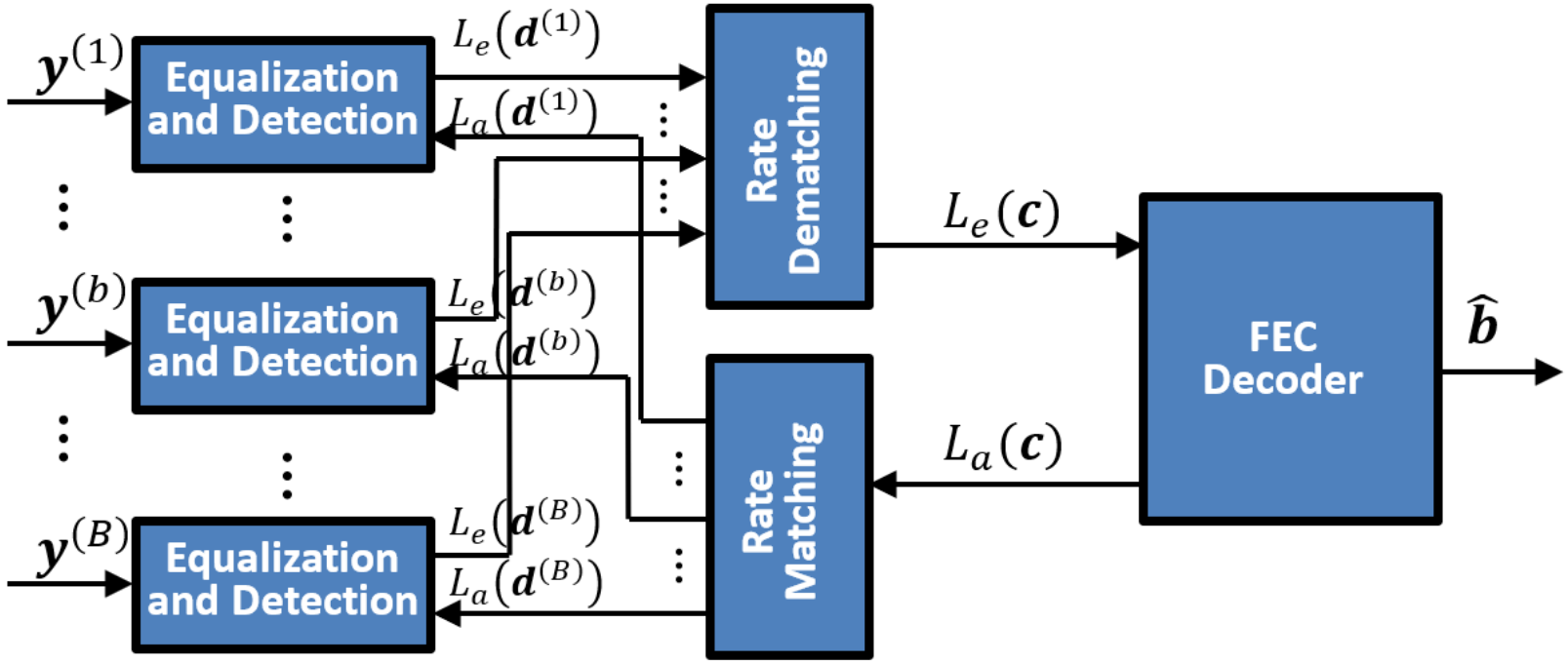
Channel



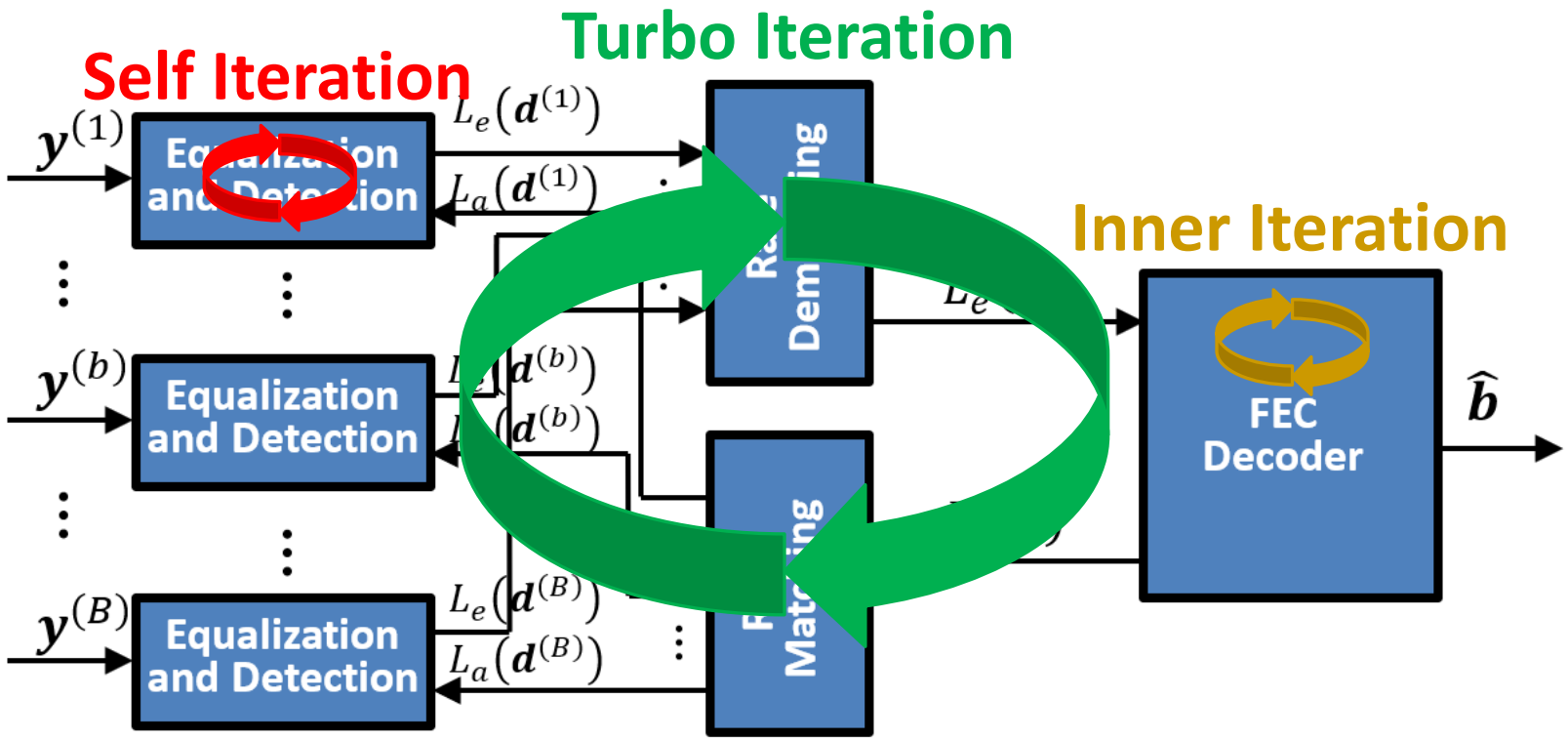
● AFF3CT

\+ Added

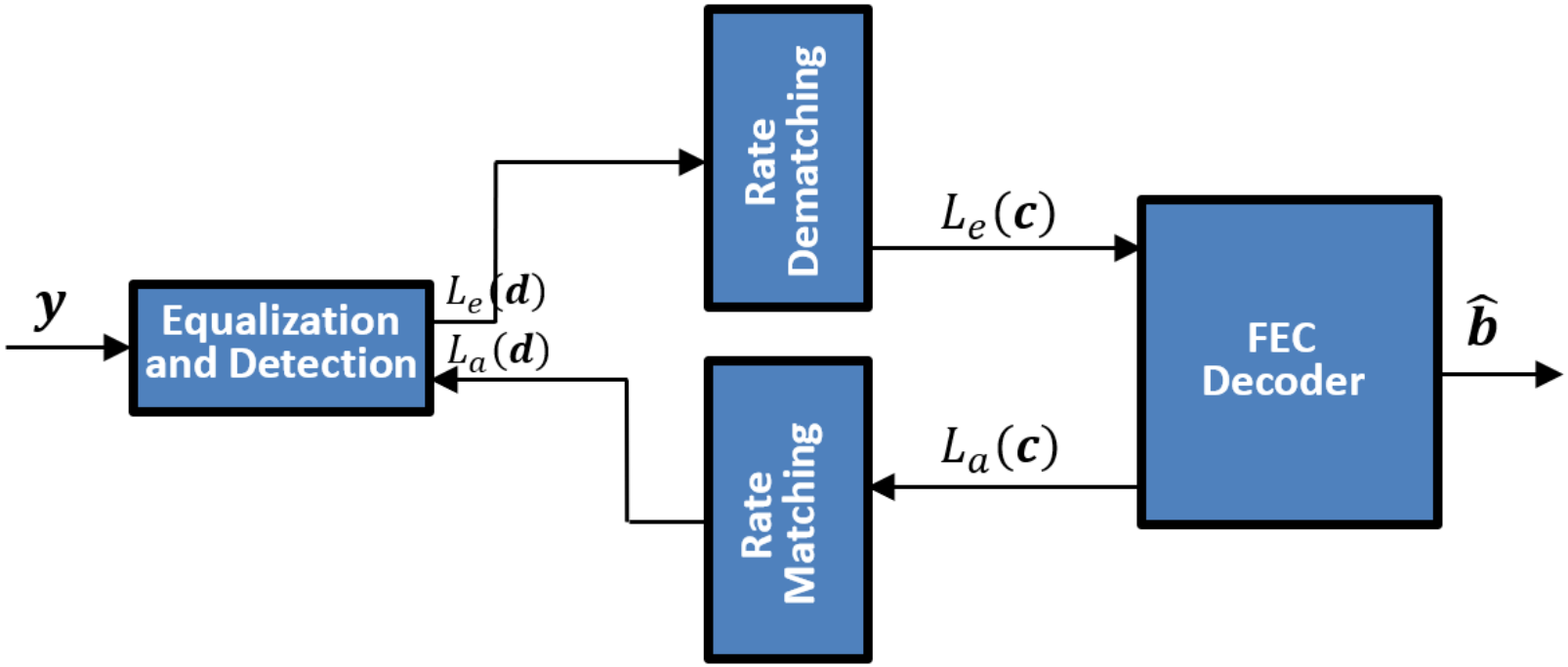
Block Schematic for Turbo Detection



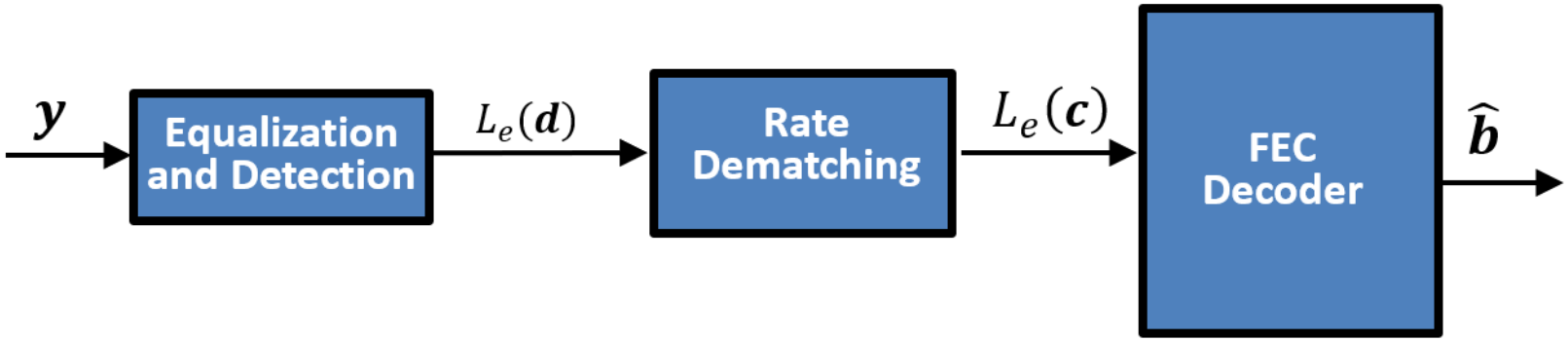
Block Schematic for Turbo Detection



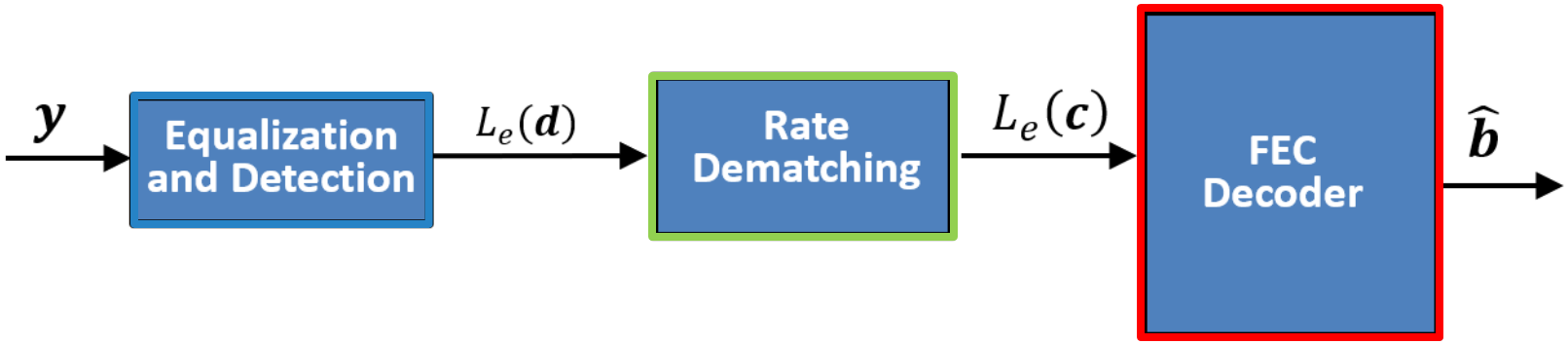
Block Schematic with one block



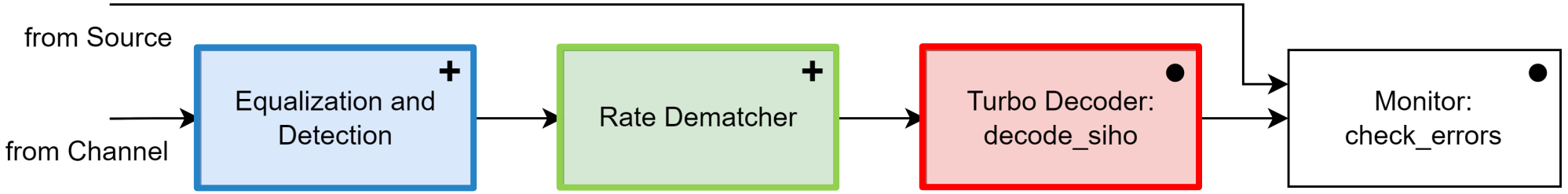
Block Schematic without Turbo Iterations



Block Schematic without Turbo Iterations



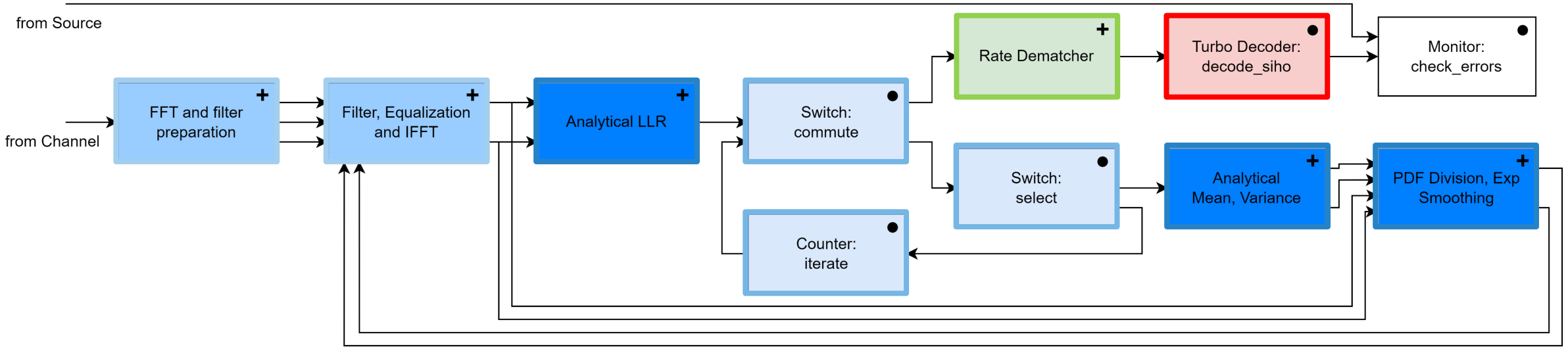
Receiver



● AFF3CT

\+ Added

Receiver subdivided in blocks



● AFF3CT

✚ Added

Summary

1. Context
2. Block Schematics
3. Simulations with py-AFF3CT
4. FPGA Implementation

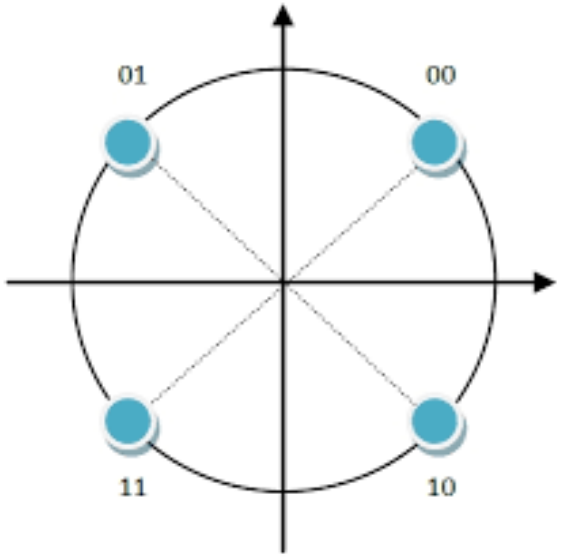
Simplification: QPSK Analytical LLR

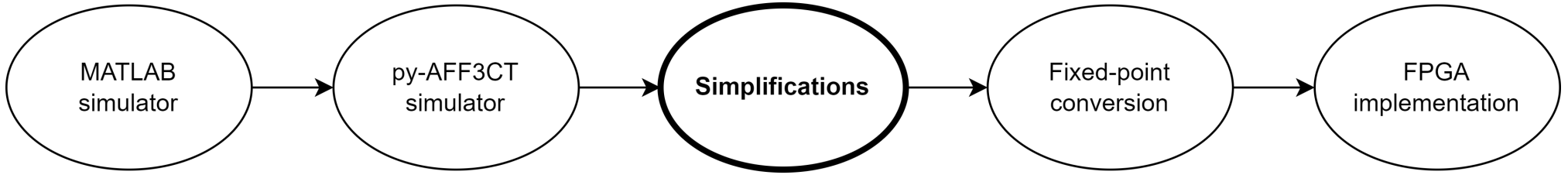
$$\mathcal{D}'^{(s)}_{k,b}(\alpha) = \exp\left(-\left|x^{e(s)}_{k,b} - \alpha\right|^2 / \left(k_{\text{PAM}} v^{e(s)}_{x,b}\right)\right)$$

$$\mathcal{D}^{(s)}_{k,b}(\alpha) = \mathcal{D}'^{(s)}_{k,b}(\alpha) / \sum_{\alpha' \in \mathcal{X}} \mathcal{D}'^{(s)}_{k,b}(\alpha')$$

$$L\left(d^{(b)}_{k,q}\right) = \ln \sum_{\alpha \in \mathcal{X}^0_q} \mathcal{D}^{(s)}_{k,b}(\alpha) - \ln \sum_{\alpha' \in \mathcal{X}^1_q} \mathcal{D}^{(s)}_{k,b}(\alpha')$$

$$L(d_1) = 2\sqrt{2}\,\Re(x^e)/v^e_x$$
$$L(d_2) = 2\sqrt{2}\,\Im(x^e)/v^e_x$$





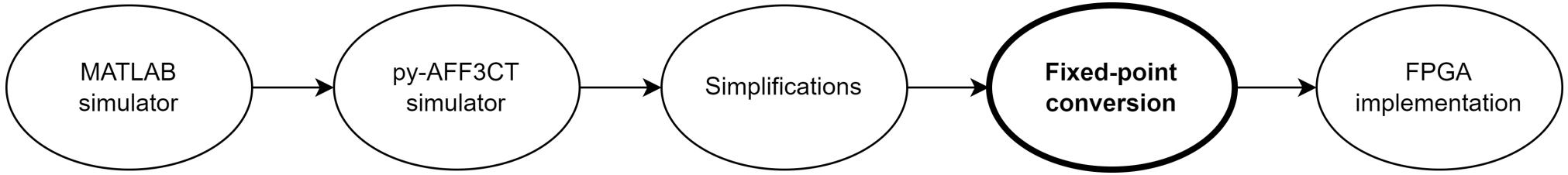
Fixed-Point Conversion

Fxp Math (https://github.com/francof2a/fxpmath)

Tests to define the size of each variable in fixed-point

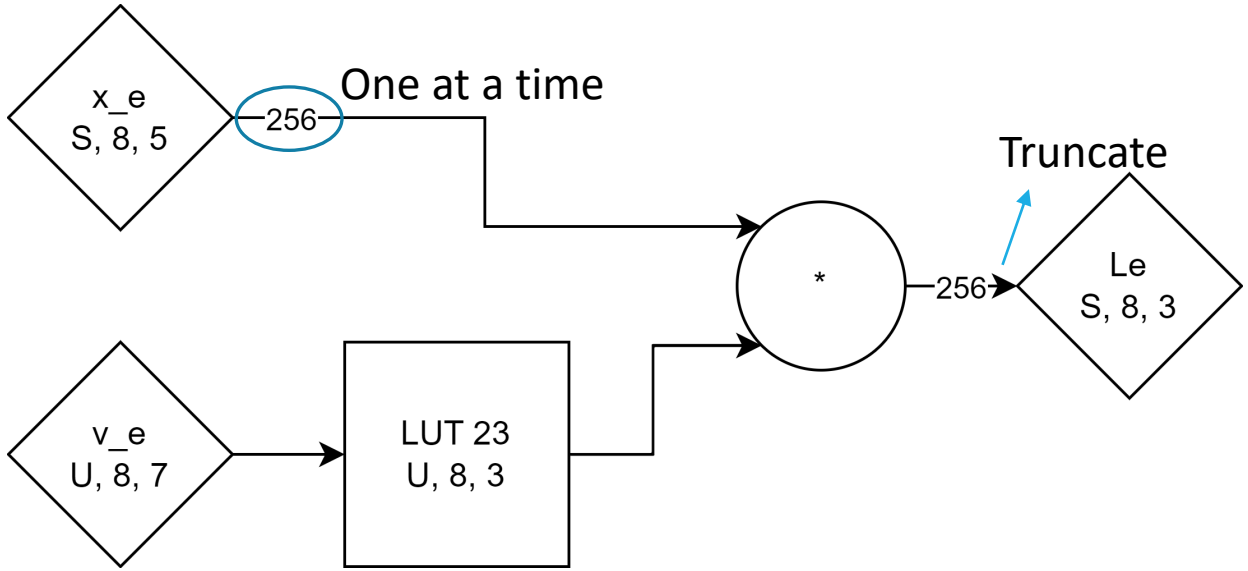
Fixed-point operations not used on simulations for being too slow

- Convert floating-point to fixed-point and then back to floating-point for operations

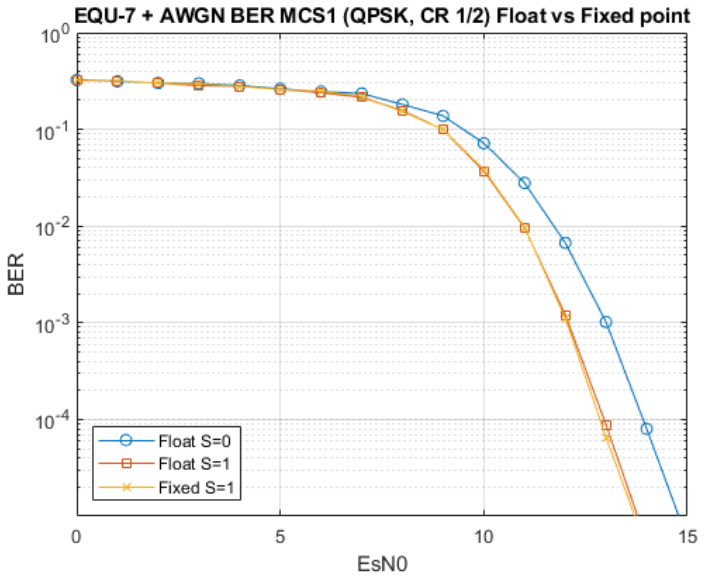


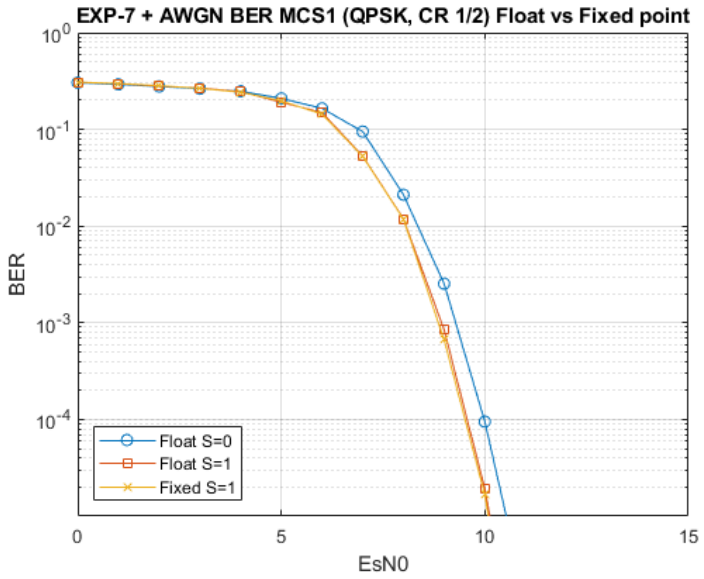
Example: QPSK Analytical LLR

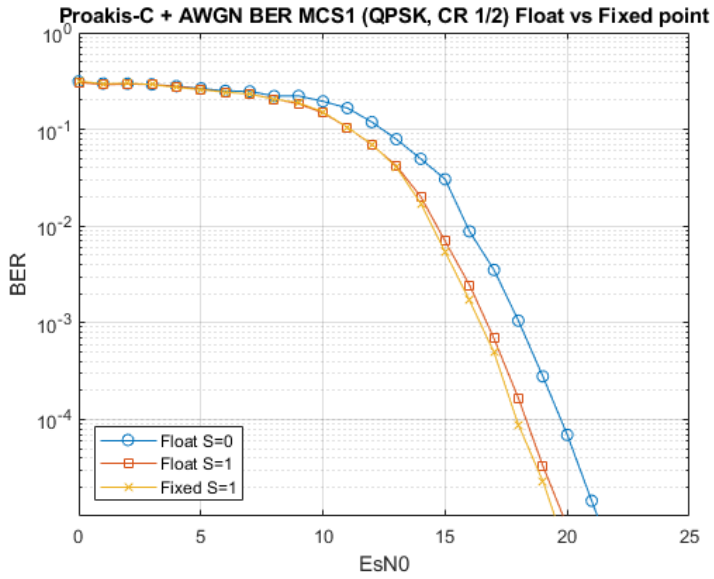
Constellation	Label
QPSK (4-QAM)	$L(d_1) = 2\sqrt{2}\,\Re(x^e)/v_x^e$ $L(d_2) = 2\sqrt{2}\,\Im(x^e)/v_x^e$



Results of the Fixed-Point Conversion







Summary

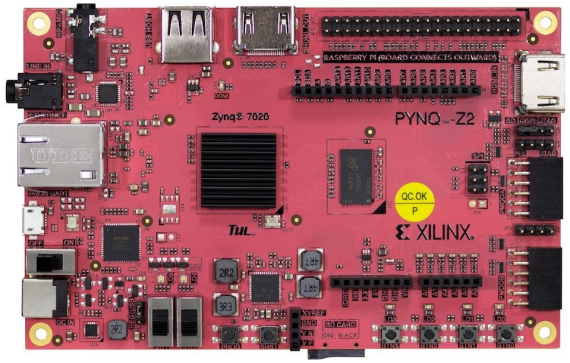
1. Context
2. Block Schematics
3. Simulations with py-AFF3CT
4. FPGA Implementation

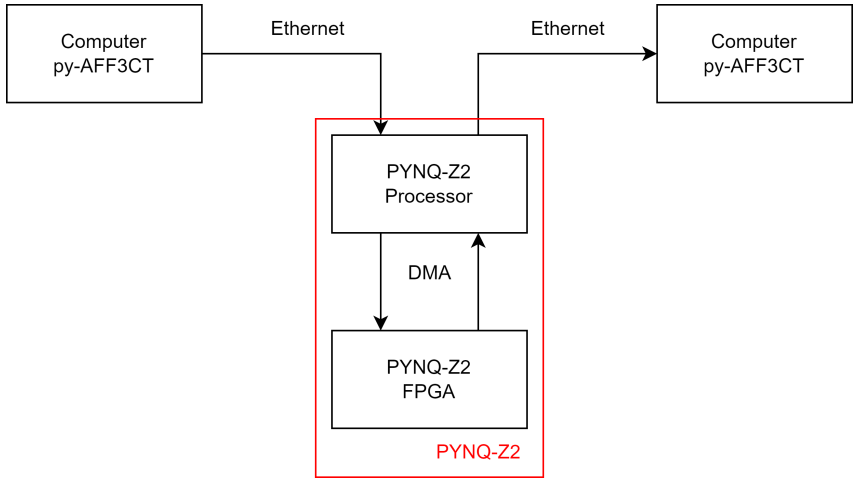
PYNQ-Z2 Board

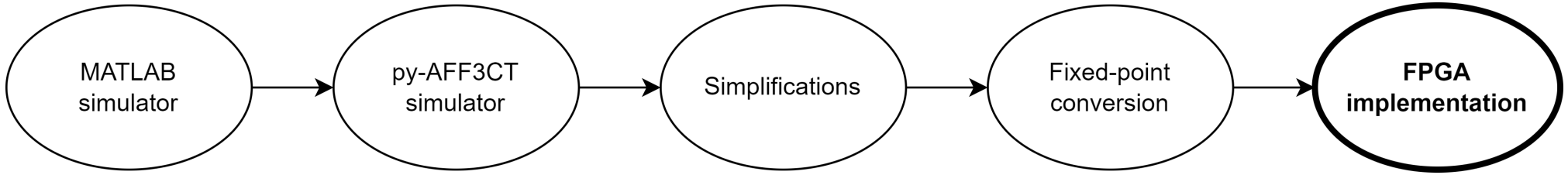
Zynq SoC

- combines dual-core Arm Cortex-A9 processors with Xilinx programmable logic (FPGA), enabling flexible hardware and software co-design

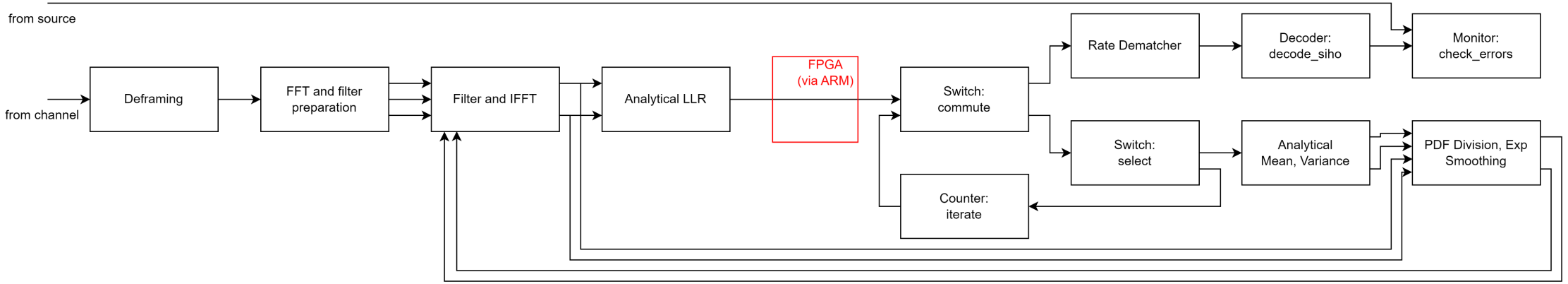
Gigabit Ethernet port used to connect with py-AFF3CT



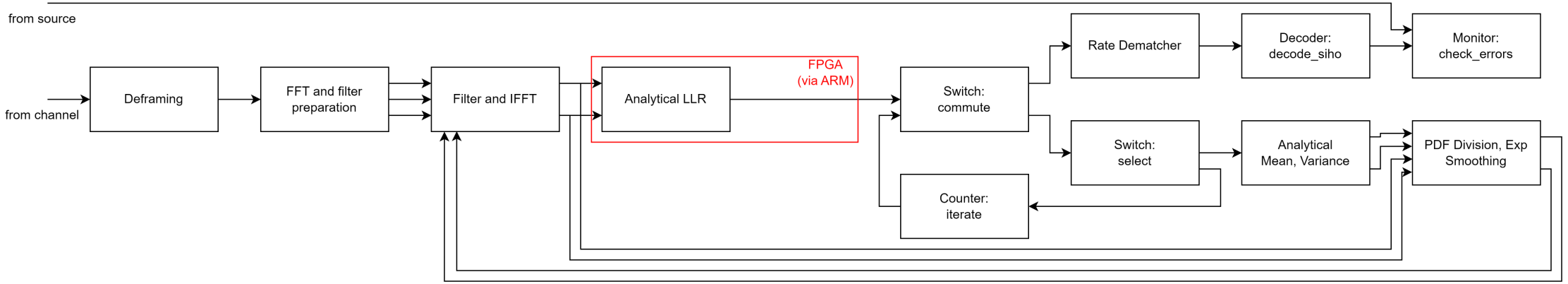




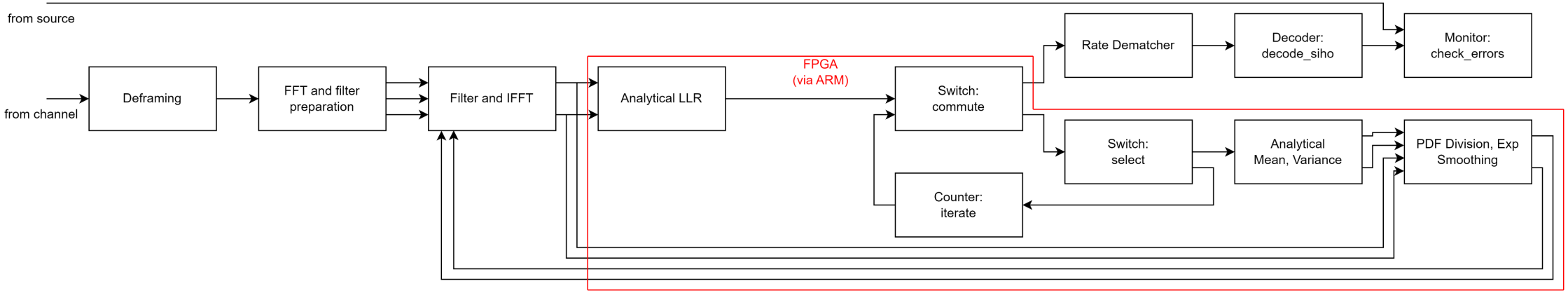
Step 1: Passthrough the FPGA



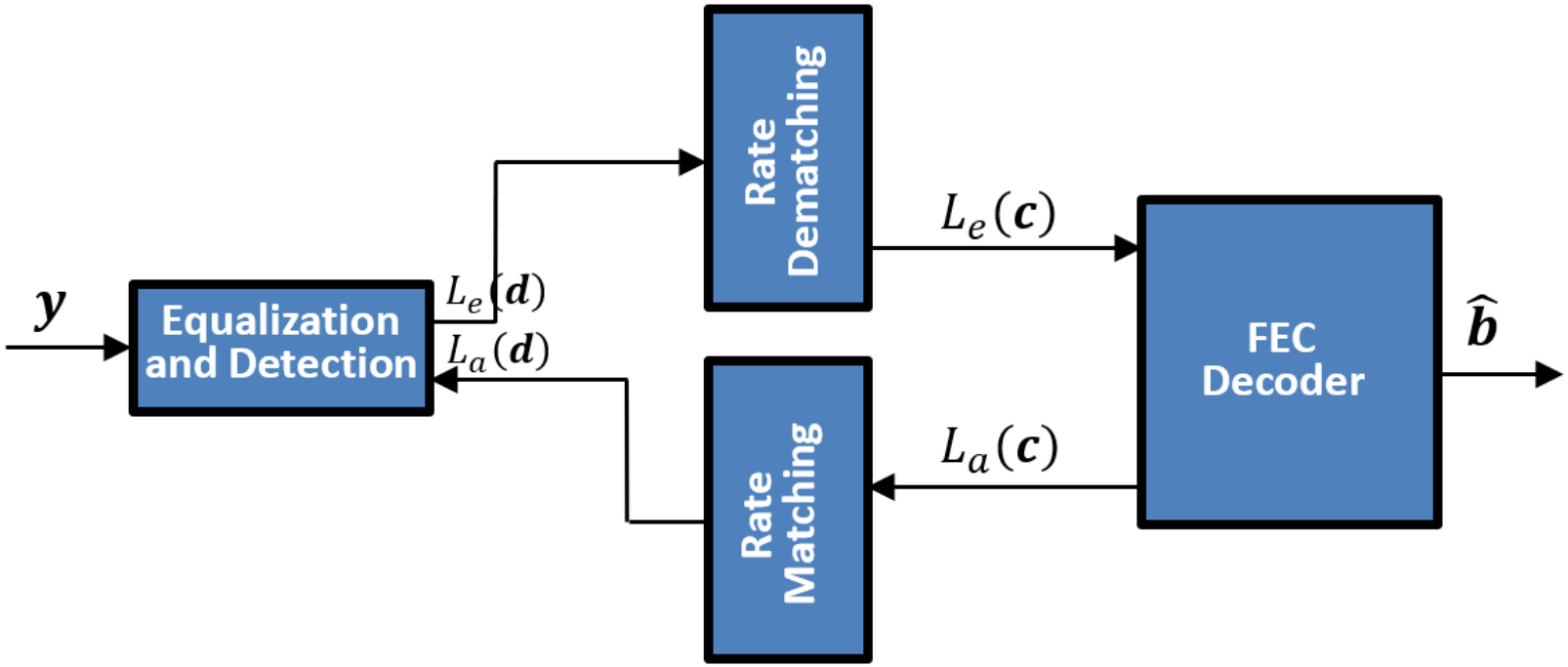
Step 2: Analytical LLR



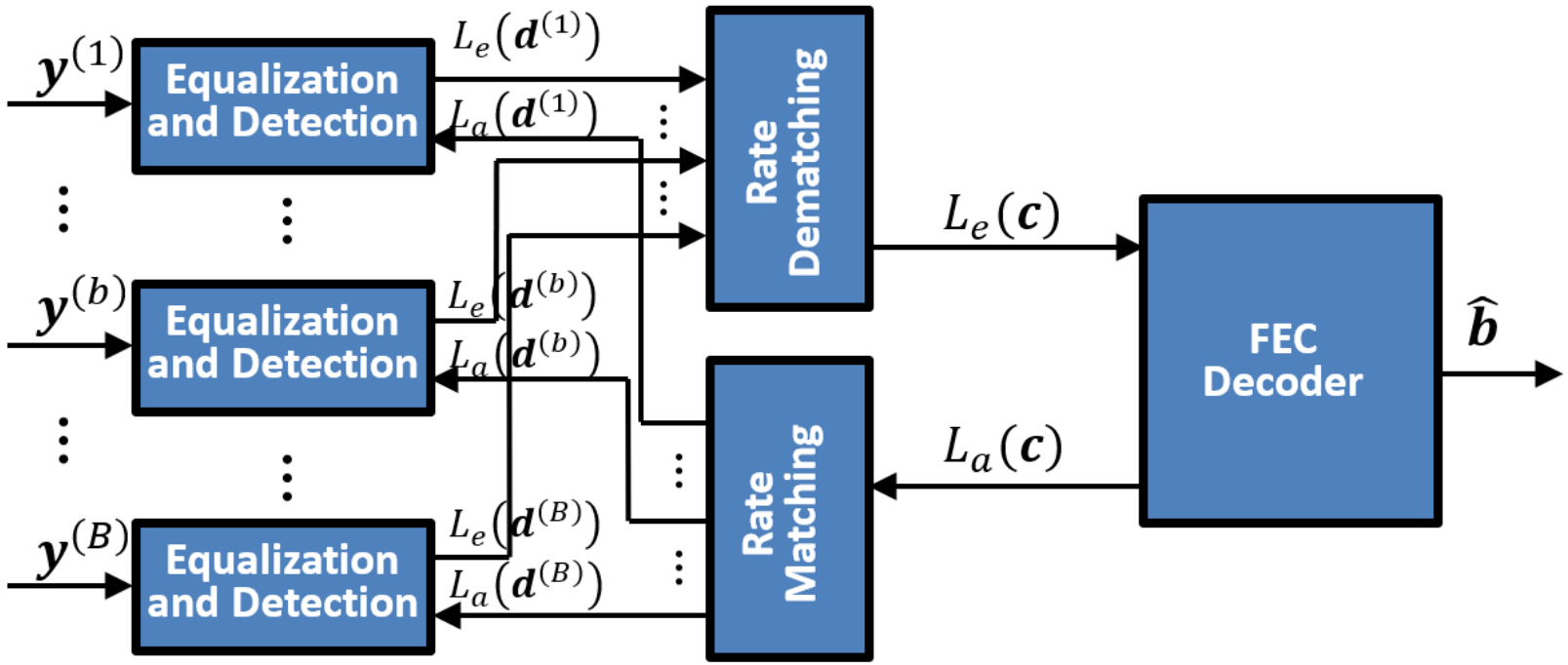
Step 3: Soft Demapper and Mapper



Step 4: Use Turbo Iterations



Step 5: Use Multiple Blocks



References

EVASION, ANR-20-CE25-0008-01, “D1.2: State of the art of receivers, channels and system specifications”, 08/11/2021.

EVASION, ANR-20-CE25-0008-01, “D1.3: Report on Simplification and State Evolution”, 05/07/2023.

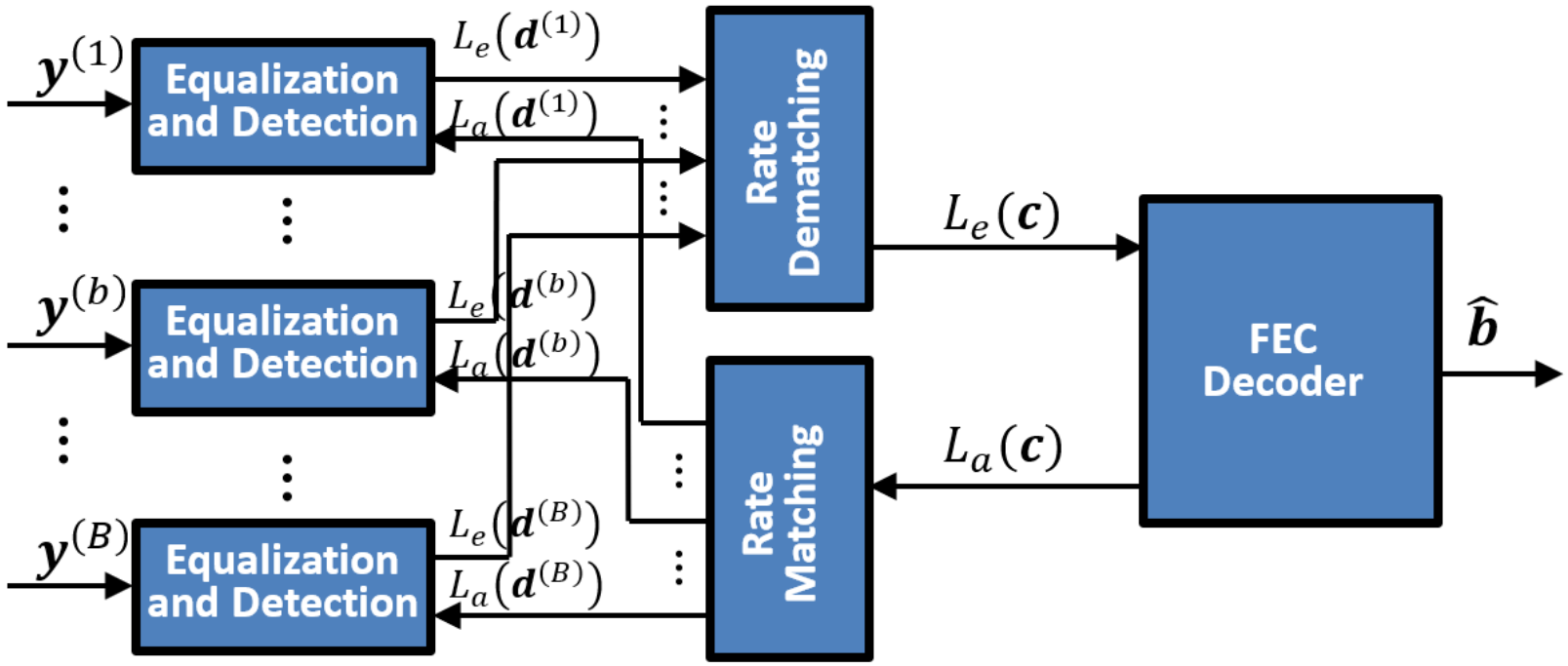
EVASION, ANR-20-CE25-0008-01, “D3.1: Impact Study of the Fixed-Point Conversion for the Receiver Algorithms”, 30/09/2023.

EVASION, ANR-20-CE25-0008-01, “D3.3: Fixed-point version of the simplified version (architectural-aware) of the receiver”, 21/09/2023.

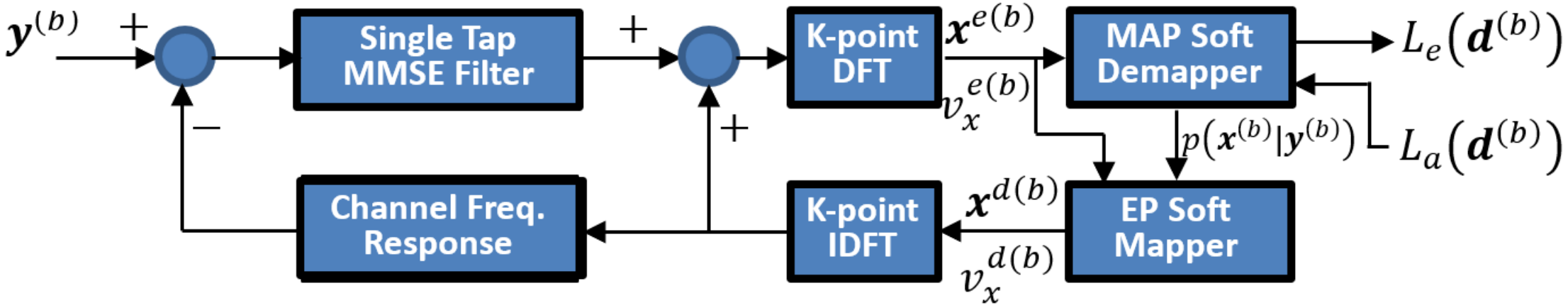
EVASION, ANR-20-CE25-0008-01, “D3.4: Study of the Architectural Exploration and Hardware Architecture Design Schemes”, 21/11/2023.

Thank you!
Questions?

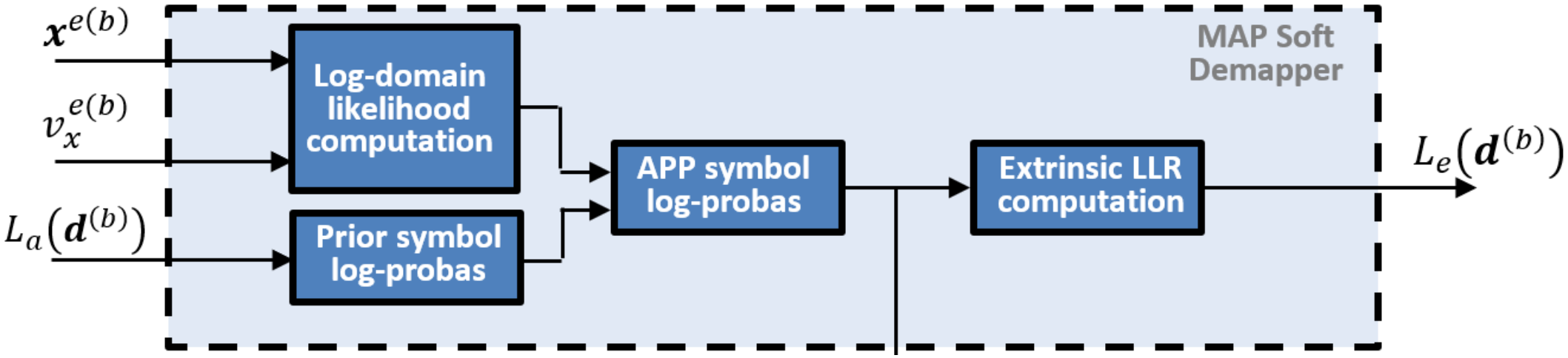
Block Schematic for Turbo Detection



Equalization and Detection



MAP Soft Demapper



EP Soft Mapper