



Integrated Performance of Next Generation High Data Rate Receiver and AR4JA LDPC Codec For Space Communications

Michael K. Cheng, Mark Lyubarev, Michael A. Nakashima,
Kenneth S. Andrews, and Dennis Lee

Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA, USA

SpaceOps 2008,
Heidelberg, Germany



Introduction



- NASA has begun efforts to upgrade both the S-band (nominal data rate) and K-band (high data rate) receivers in the Space and Deep Space Networks to support, for example, Constellation and James Webb Space Telescope.
- These modernization efforts provide an opportunity to infuse modern error forward error correcting (FEC) codes.
- The Jet Propulsion Laboratory (JPL) has designed a family of capacity approaching low-density parity-check (LDPC) codes that are similar in structure and therefore, leads to a single decoder implementation.
- The Accumulate-Repeat-by-4-Jagged-Accumulate (AR4JA) code design offers a family of codes with rates $1/2$, $2/3$, $4/5$, and lengths 1024, 4096, 16384 information bits to meet the constraints of bandwidth or power efficient transmissions.



Background



- The Tracking and Delay Satellite System (TDRSS) K-Band Return Upgrade Augmentation Project (TKUP-A) has completed (March, 2008) a successful demonstration of LDPC codes in an integrated receiver decoder unit on a space link [Y. Wong, SpaceOps 2008].
- Constellation tested (February, 2007) an AR4JA decoder at the Electronics Systems Test Lab (ESTL) at Johnson Space Center, Houston Texas with a copy of the Integrated Receiver (IR) used at the White Sands Complex (WSC). Improvements needed to achieve a robust performance.
- Here, we take a different approach and integrate two commercial-off-the-shelf (COTS) components: an AR4JA LDPC decoder built on an FPGA and a modern high data rate receiver.
- This integration work (May-September, 2007) resolved all issues observed in ESTL-I.
- Constellation performed ESTL-II testing (November, 2007) and measured a combined receiver-decoder implementation loss as low as one dB.



The AR4JA LDPC Code Family



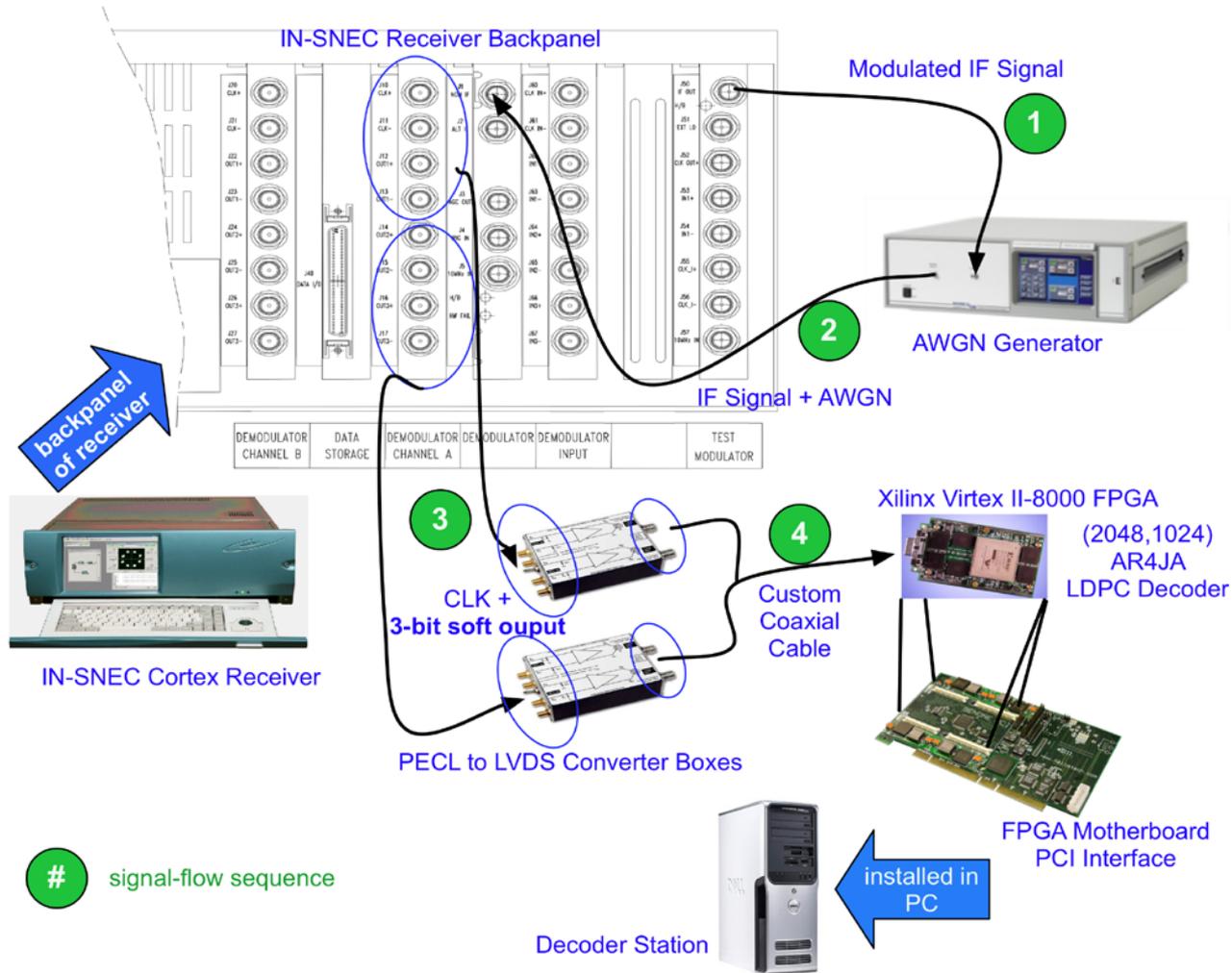
- LDPC codes were first proposed by Robert Gallager of MIT in 1960s ahead of its time.
- All things being equal, normalizing code rate and length, LDPC codes provide an additional 1.5 dB coding gain over legacy concatenated Reed-Solomon and Convolutional codes.
- The Consultative Committee for Space Data Systems (CCSDS) has an on-going effort to standardize LDPC codes, which the AR4JA codes are members of, for Space Communications [Ken Andrews, SpaceOps 2008].
- The AR4JA LDPC code family is built on protographs and this construction facilitates parallel hardware implementation and enables a high speed decoder realization.
- A 80-Mbps decoder has been implemented on a Xilinx Virtex II FPGA by Ken Andrews.



Receiver-Decoder Integration



- We integrate the IN-SNEC Cortex high data rate receiver under consideration by the Deep Space Network (DSN) with an FPGA (2048,1024) AR4JA LDPC decoder.
- For Quadrature Phased Shift Keying (QPSK), this receiver is capable of up to 640 Mbps.





Receiver-Decoder Integration



-
- Frame synchronization
 - Soft-Symbol Scaling
 - Avoiding Soft-Symbol Scaling
 - Pseudorandom Number (PN) De-randomization



Frame Synchronization



- LDPC codes are block codes. Decoding of block codes requires synchronization of the codeword boundaries.
- Here, each (2048,1024) AR4JA LDPC codeword is prefixed by a 64-bit CCSDS Attached Synchronization Marker.
- Since the theoretical decoding threshold of of this LDPC code is about $E_b/N_0=2\text{dB}$. A robust algorithm is needed for rapid frame synchronization at this low Signal-to-Noise Ratio (SNR).
- We apply a variation of the Massey optimum frame synchronization algorithm.



Massey Frame Synchronization



- Perfect coherent reception with no phase ambiguity for BPSK
- Observation samples at the output of matched filter

$$y_k = Aa_k + n_k; \quad a_k = \pm 1; \quad n_k = N(0, \sigma^2)$$

- Massey's metric

$$M(\mu) = \sum_{i=0}^{L-1} (s_i y_{i+\mu} - |y_{i+\mu}|)$$

- L is the ASM length, and N is the frame size or ASM+codeword length
- The start of a codeword frame is given by

$$\hat{\mu} = \arg \max_{0 \leq \mu < N} M(\mu)$$

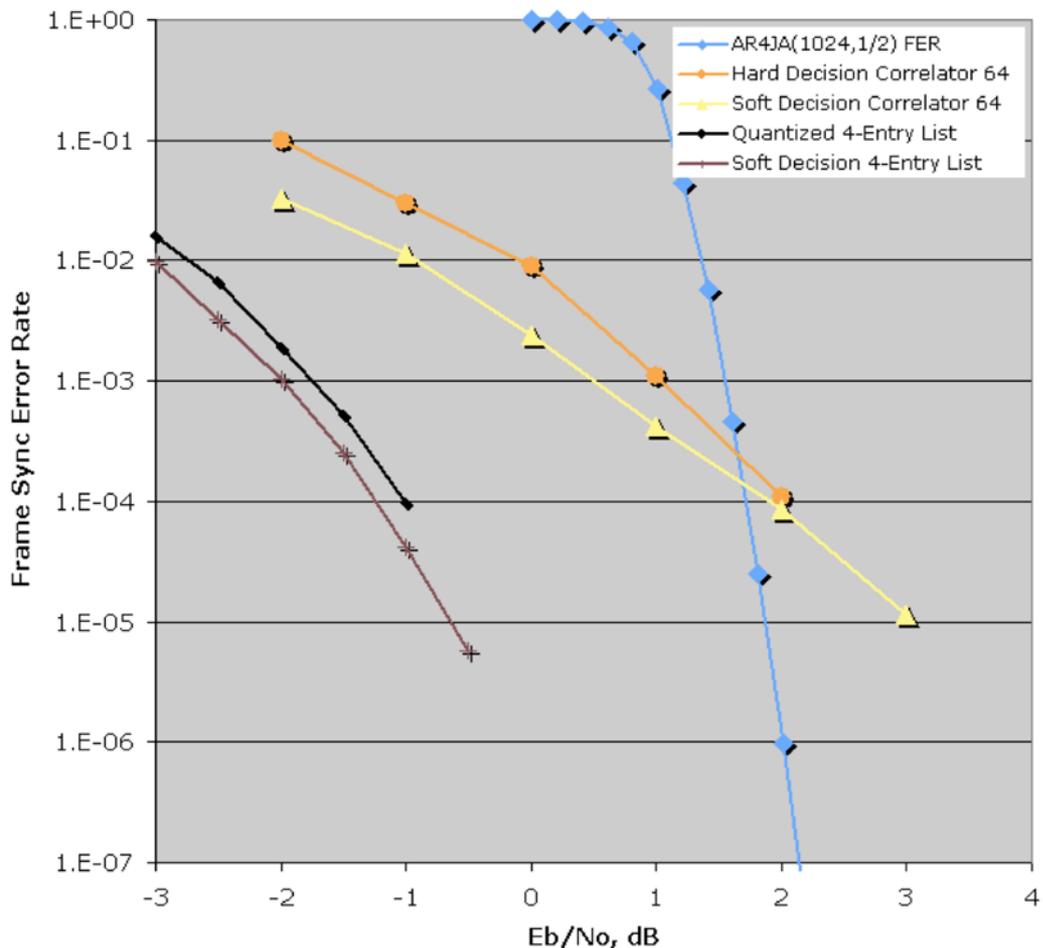
- This computation can be averaged over multiple frames.



Performance of Frame Synchronizers



Frame Synchronization Algorithms



- Frame synchronizer also resolves phase ambiguity.
- Conventional correlator makes a decision on one frame while the Massey averages over 2 frames.



Soft-Symbol Scaling



- Binary LDPC decoding iteratively refines the a posteriori probability that a bit in the transmitted codeword equals to 1 given the received data.
- The bit reliability at the input of the decoder is initialized to a log likelihood ratio (LLR) given by

$$LLR(c_i) = 2(A/\sigma^2)y_i = 2\alpha y_i$$

- Where c_i is the i th codeword symbol, A is the signal amplitude, and σ^2 is the noise power.
- The combining ratio α needs to be estimated by the decoder.



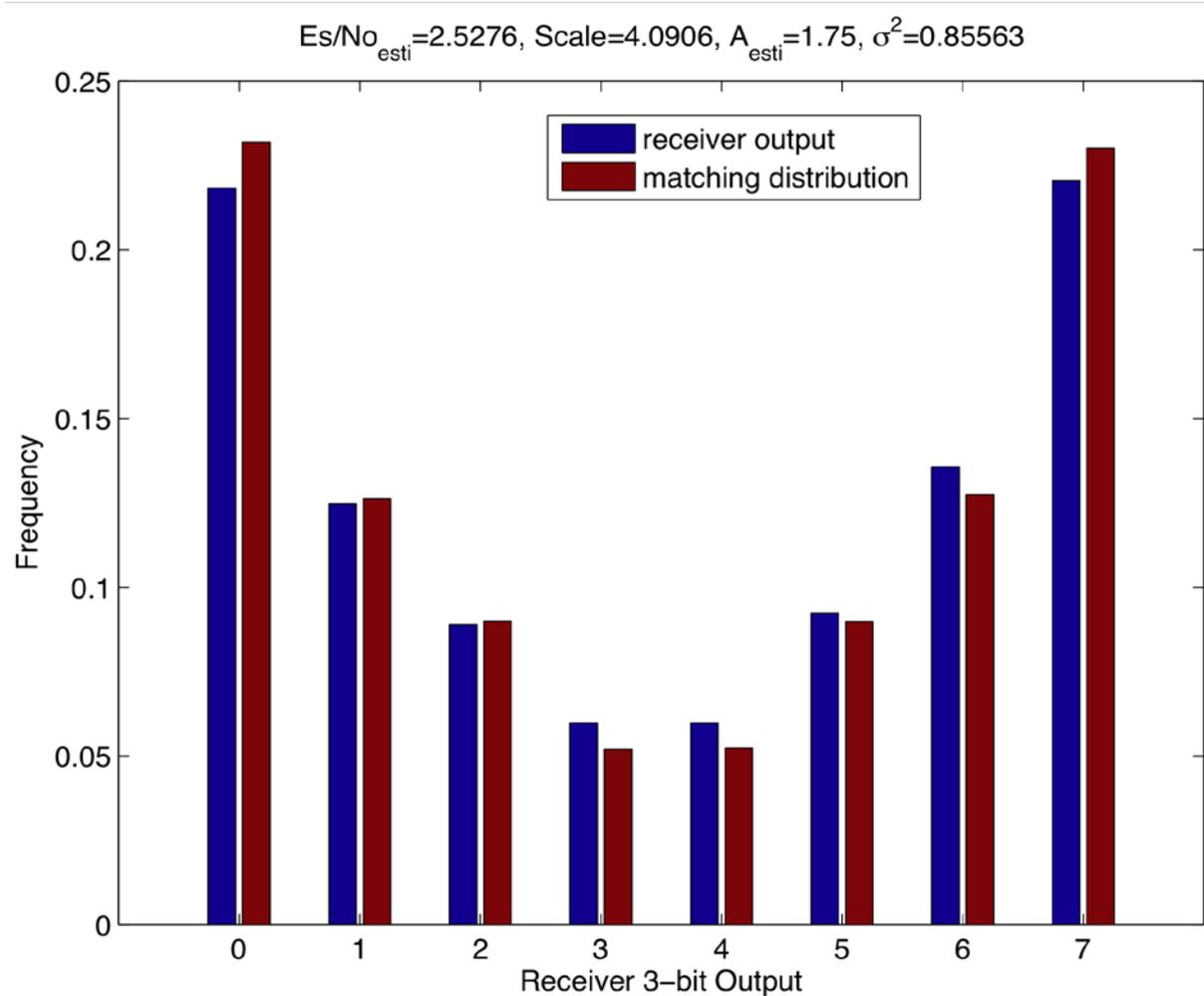
Estimating α Through Distribution Matching



-
- We collect a histogram of the soft output from the IN-SNEC receiver.
 - We attempt to match the collected histogram to a series of randomly generated Gaussian distributions with varying first two moments A and σ^2 .
 - The receiver also clips the Gaussian tails to a maximum value f . So this clipping point also needs to be estimated.
 - We empirically find these three parameters A , σ^2 , and f by searching through various combinations.
 - The combining ratio is determined by the parameters of the best matched distribution.



Estimating α Through Distribution Matching (cont.)





Avoiding Soft-Symbol Scaling



- Decoding of LDPC codes iteratively refines the LLRs of the received symbols through a so called min* function.

$$\min^* (\lambda_1, \lambda_2) = \text{sign} (\lambda_1 \lambda_2) \min (|\lambda_1|, |\lambda_2|) + s (|\lambda_1|, |\lambda_2|)$$

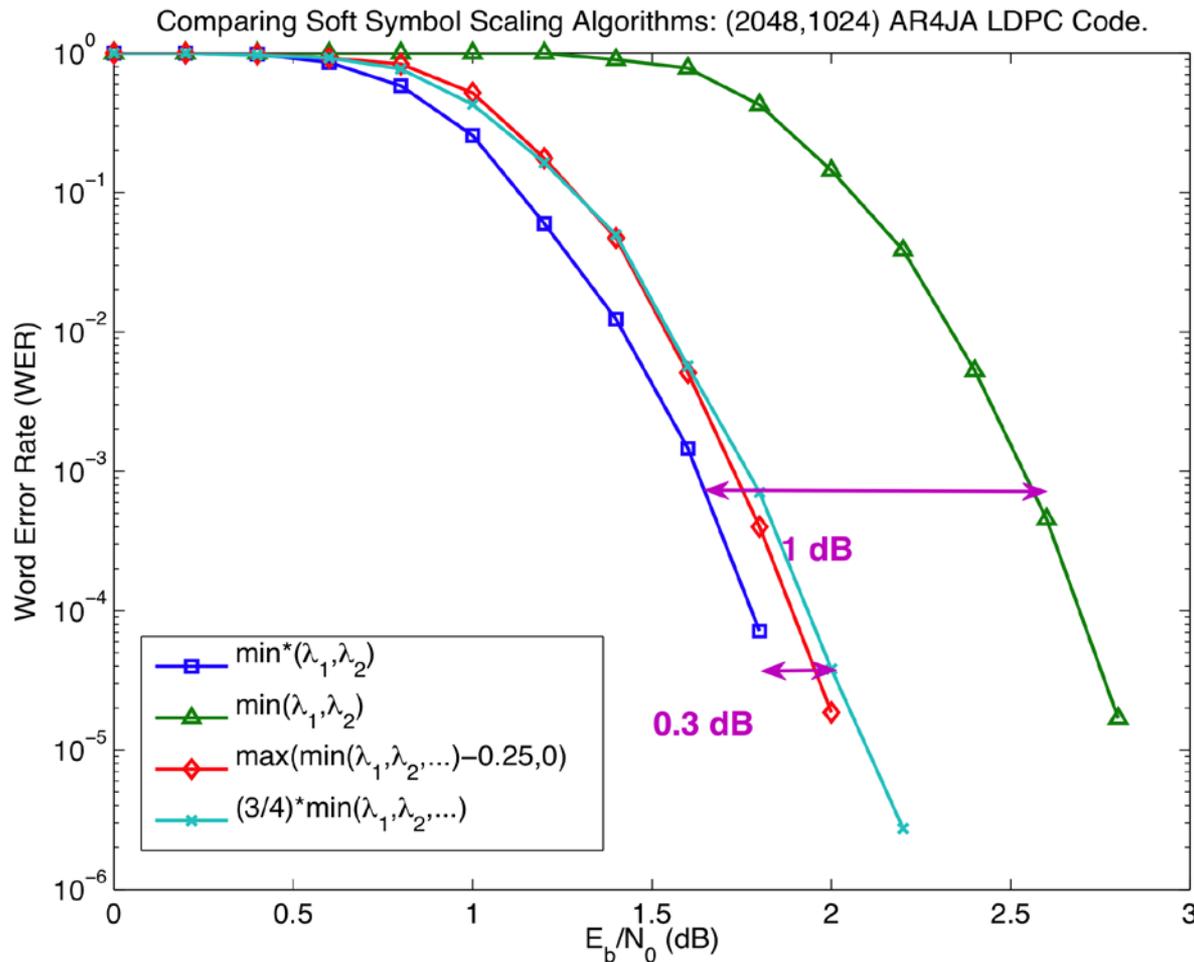
- Where the adjustment term is:

$$s (\lambda_1, \lambda_2) = \log (1 + e^{-|\lambda_1 + \lambda_2|}) - \log (1 + e^{-|\lambda_1 - \lambda_2|})$$

- The positive portion is small and can be ignored with little impact on performance.
- The entire adjustment term can be ignored and min* reduces to min.
- A constant offset in $[0, \log(2)]$ can be used – offset min.
- A scaling factor can be used in conjunction with min.



Avoiding Soft-Symbol Scaling (cont.)

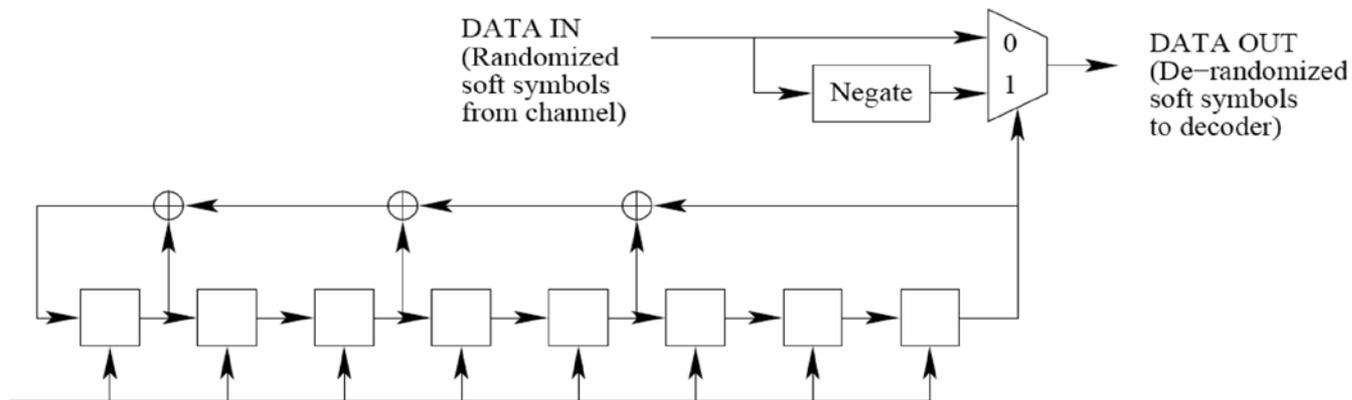




PN De-randomization



- A long run of constant 1's or 0's are often avoided in transmission in order to promote accurate symbol timing recovery.
- To do so, sender data is bit-randomized by XORing with a pseudo-random number (PN) sequence before transmission.
- The sequence is undone at the receiver by
 - XORing received data with the same PN sequence in hard-decision decoding
 - Modulating the signs of the soft-symbols in soft-decision decoding





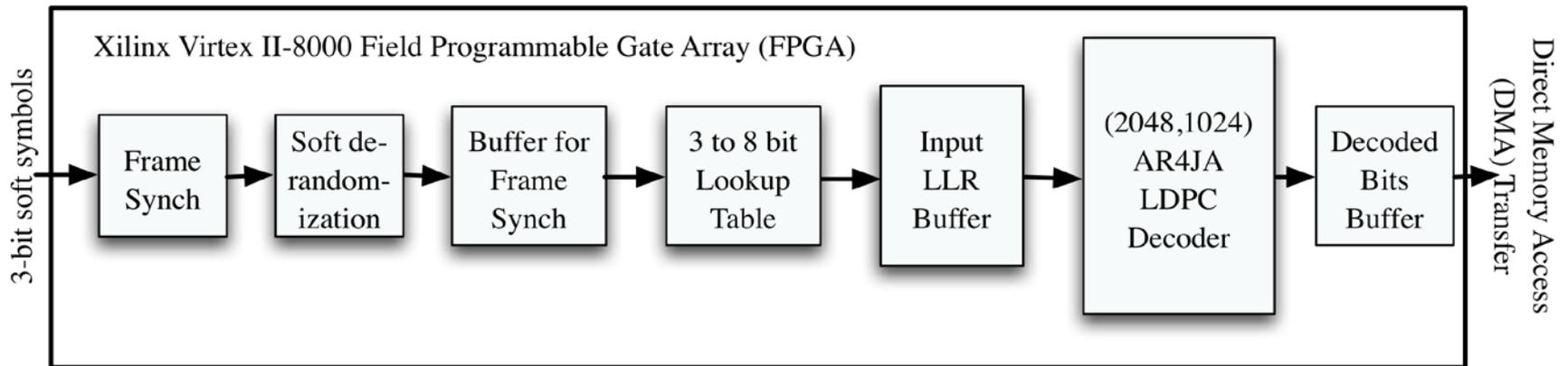
Performance Results



-
- Stand-Alone FPGA Decoder
 - Integrated Performance with a Conventional Frame Synchronizer
 - Integrated Performance with a Massey Synchronizer
 - Integrated Performance with min* and scale-min
 - Comparison to legacy
 - FPGA resource utilization

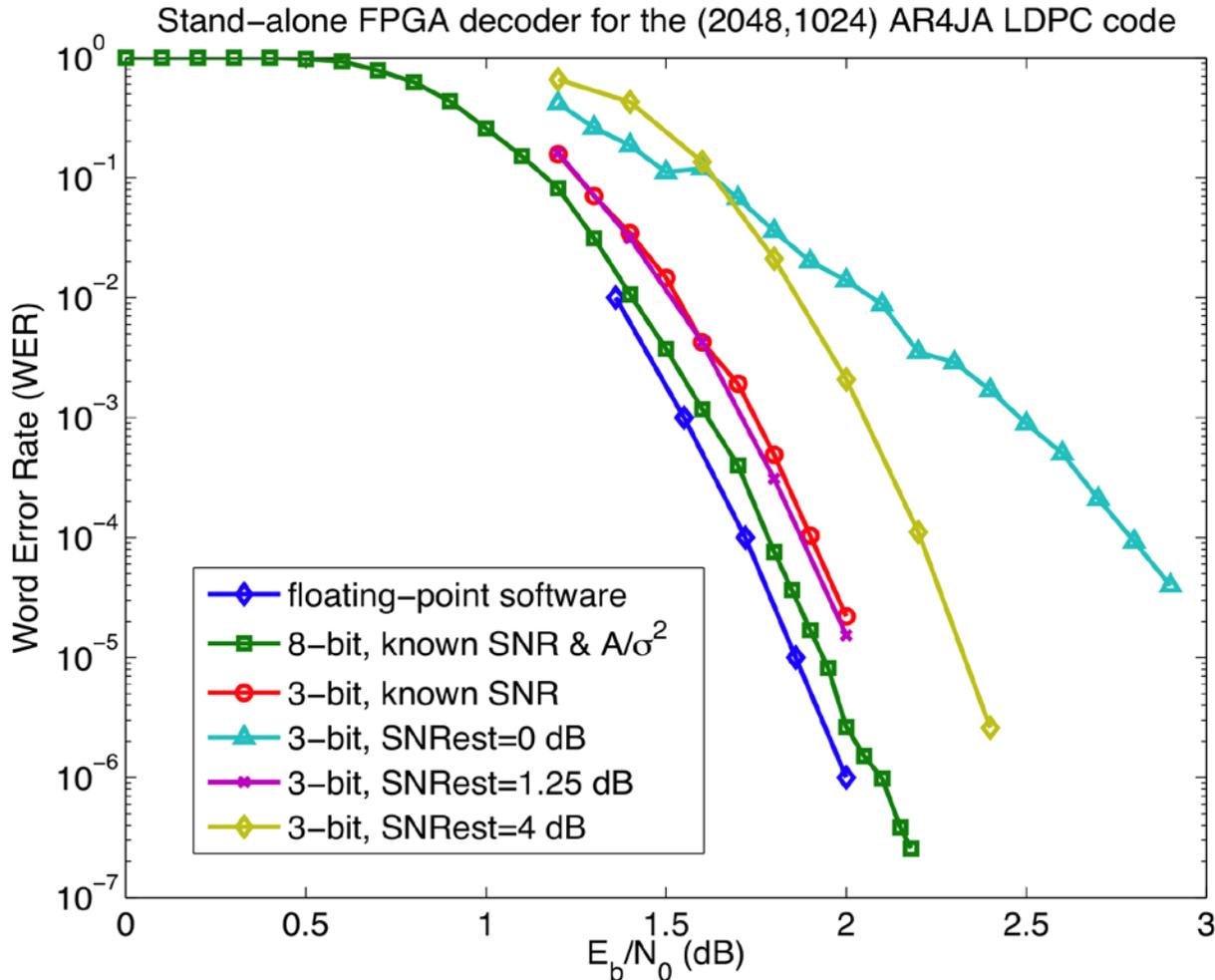


Decoder Data Path





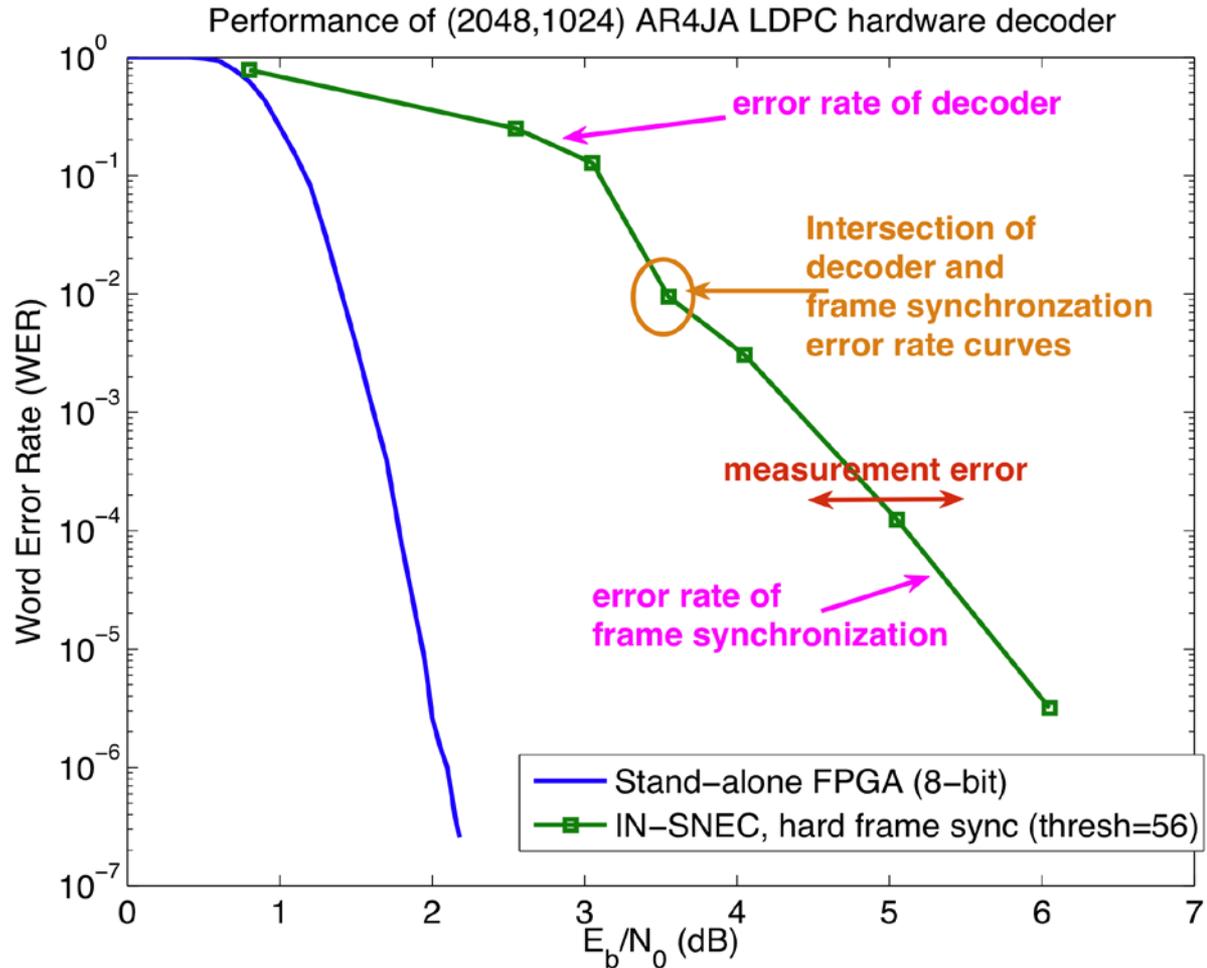
Performance Result: Stand-alone FPGA Decoder



- Perfect codeword synchronization
- Software channel

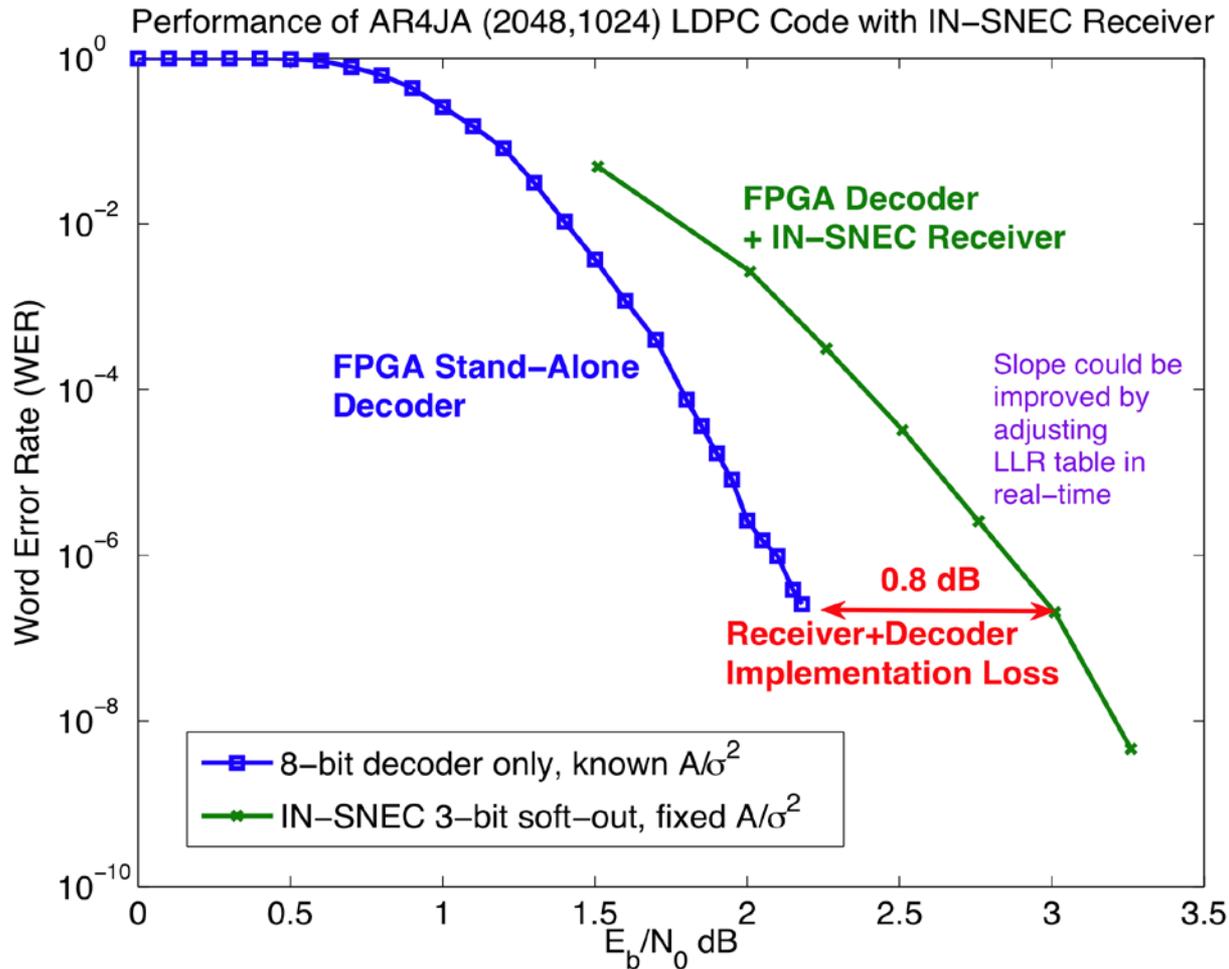


Performance Result: IN-SNEC – LDPC Conventional Frame Synchron



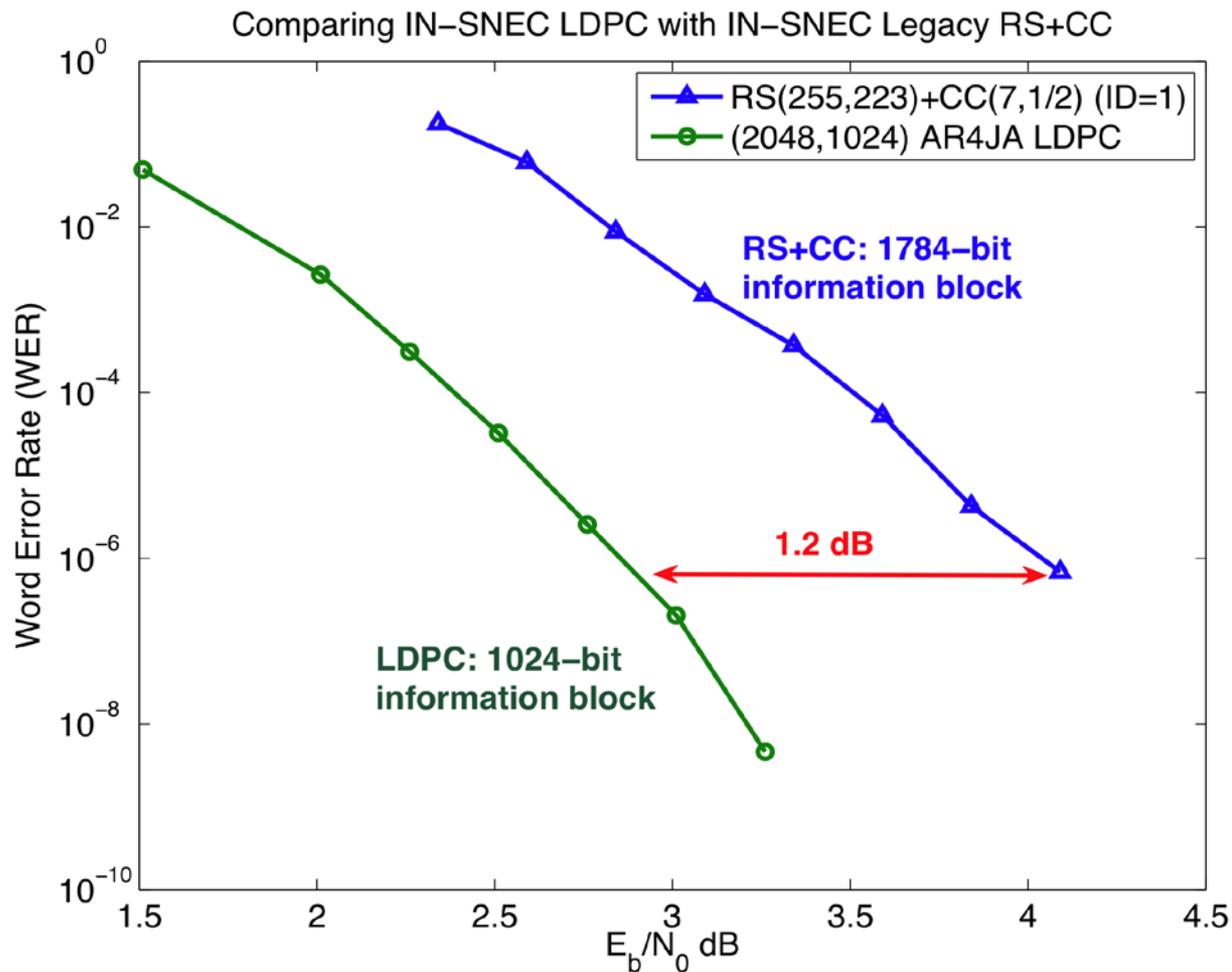


Performance Result: IN-SNEC – LDPC Massey Frame Synchron





Comparison with Legacy Codes





Decoder Specification



- The (2048,1024) AR4JA LDPC decoder can support a clock speed of 60MHz and performs a single decoder iteration in 73 clock cycles.
- At 25 Mbps information rate, the K-band (2048,1024) AR4JA LDPC code throughput requirement, the decoder can run on average 35 iterations.
- The average number of iterations in the threshold region is about 14.
- The prototype decoder built on a COTS FPGA can already meet mission requirements.