SUPPLEMENTARY MATERIALS TO THE PAPER "A 3.0 GB/S THROUGHPUT HARDWARE-EFFICIENT DECODER FOR CYCLICALLY-COUPLED QC-LDPC CODES"

BY QING LU, JIANFENG FAN, CHIU-WING SHAM, WAI M. TAM, AND FRANCIS C. M. LAU

A. Descriptions of LDPCCC and tail-biting LDPCCC

By spatially-coupling consecutive LDPC block codes, LDPCCCs are formed. The parity-check matrix of a time-varying LDPCCC can be written as

$$\mathbf{H}_{[0,\infty]} = \begin{bmatrix} \mathbf{U}_0(0) & & & & & & \\ \mathbf{U}_1(1) & \mathbf{U}_0(1) & & & & & \\ \vdots & \vdots & \ddots & & & & \\ \mathbf{U}_{m_s}(m_s) & \mathbf{U}_{m_s-1}(m_s) & \cdots & \mathbf{U}_0(m_s) & & & \\ & \mathbf{U}_{m_s}(m_s+1) & \mathbf{U}_{m_s-1}(m_s+1) & \cdots & \mathbf{U}_0(m_s+1) & & \\ & & \ddots & & & \ddots & \\ & & \mathbf{U}_{m_s}(t) & \mathbf{U}_{m_s-1}(t) & \cdots & \mathbf{U}_0(t) & \\ & & \ddots & \ddots & & & \ddots \end{bmatrix}, \tag{12}$$

where m_s is referred to as the syndrome former memory of the parity-check matrix, and the sub-matrices $\mathbf{U}_i(t), i = 0, 1, \ldots, m_s$, are binary $(c-b) \times c$ matrices. The code rate $R_{[0,\infty]}$ is therefore given by b/c. LDPCCCs can be constructed from LDPC block codes. For example, time-varying LDPCCCs can be formed by unwrapping randomly constructed LDPC block codes [6] or QC-LDPC block codes [7], [32]. Fig. 0 illustrates the construction of a QC-LDPCCC based on a QC-LDPC code with 4 block rows and 32 block columns. However, LDPCCCs are considered to be of variable codelength. Yet, block codes are more preferable in many practical communication systems because they have a fixed block length.

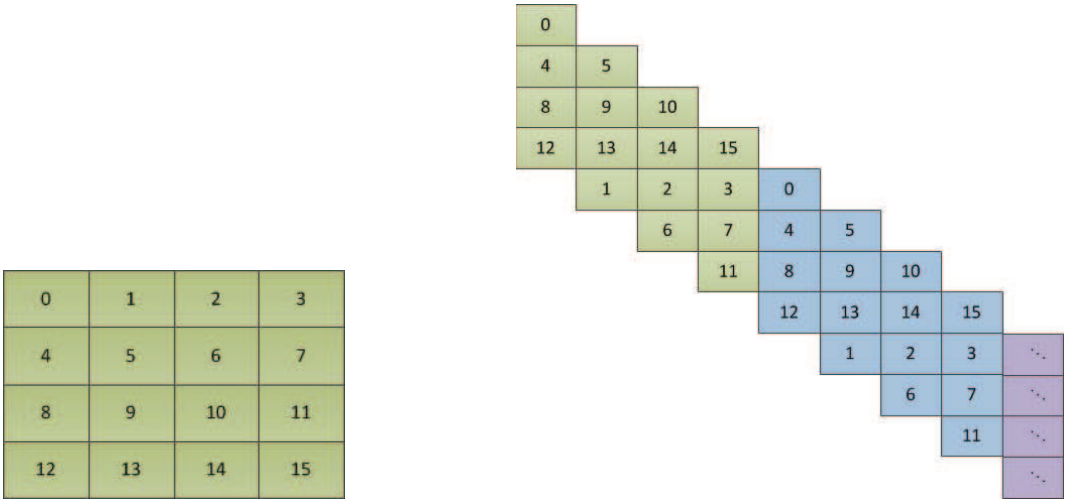


Fig. 0. Constructing a QC-LDPCCC from a QC-LDPC block code. LHS: Dividing a 4×32 QC-LDPC matrix into 16 sub-blocks of equal size. Each sub-block contains 8 circulant permutation matrices. RHS: Cut, paste and repeat to obtain the LDPCCC.

Subsequently, a tail-biting convolutional LDPC code has been proposed [11]. It is constructed based on an original block code and it has the same rate of the original code. First, the parity-check matrix $\mathbf{H}_{QC}$ of the original block code is divided into a strictly upper-triangular portion $\mathbf{H}_{QC}^{(U)}$ and a lower-triangular portion $\mathbf{H}_{QC}^{(L)}$. Then a tail-biting convolutional LDPC code constructed with λ upper-triangular portions $\mathbf{H}_{QC}^{(U)}$ (denoted by $\mathbf{H}_{QC}^{(U,i)}, i = 1, 2, \ldots, \lambda$) and λ lower-triangular portions $\mathbf{H}_{QC}^{(L)}$ (denoted by $\mathbf{H}_{QC}^{(L,i)}, i = 1, 2, \ldots, \lambda$) can be written as

$$\mathbf{H}_{TB}^{(\lambda)} = \begin{bmatrix} \mathbf{H}_{QC}^{(L,1)} & & & & \mathbf{H}_{QC}^{(U,\lambda)} \\ \mathbf{H}_{QC}^{(U,1)} & \mathbf{H}_{QC}^{(L,2)} & & & \\ & \mathbf{H}_{QC}^{(U,2)} & \mathbf{H}_{QC}^{(L,3)} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{H}_{QC}^{(U,\lambda-1)} & \mathbf{H}_{QC}^{(L,\lambda)} \end{bmatrix} \tag{13}$$

where λ denotes the "unwrapping factor", i.e., the number of times the sliding convolutional structure is repeated. Note that the overall length of the tail-biting convolutional LDPC code increases linearly with λ and so is the memory requirement at

the decoder. Also, if the decoder hardware is to be fully utilized, multiple codewords have to be decoded at the same time to fill the “pipeline holes” [12]. It implies much more memory will be needed at the decoder. In summary, LDPC convolutional code and its modifications require decoders to have a very high complexity, particularly memory storage, so as to achieve a high throughput.

B. Full-size version of the figures in the paper and Fig. 9

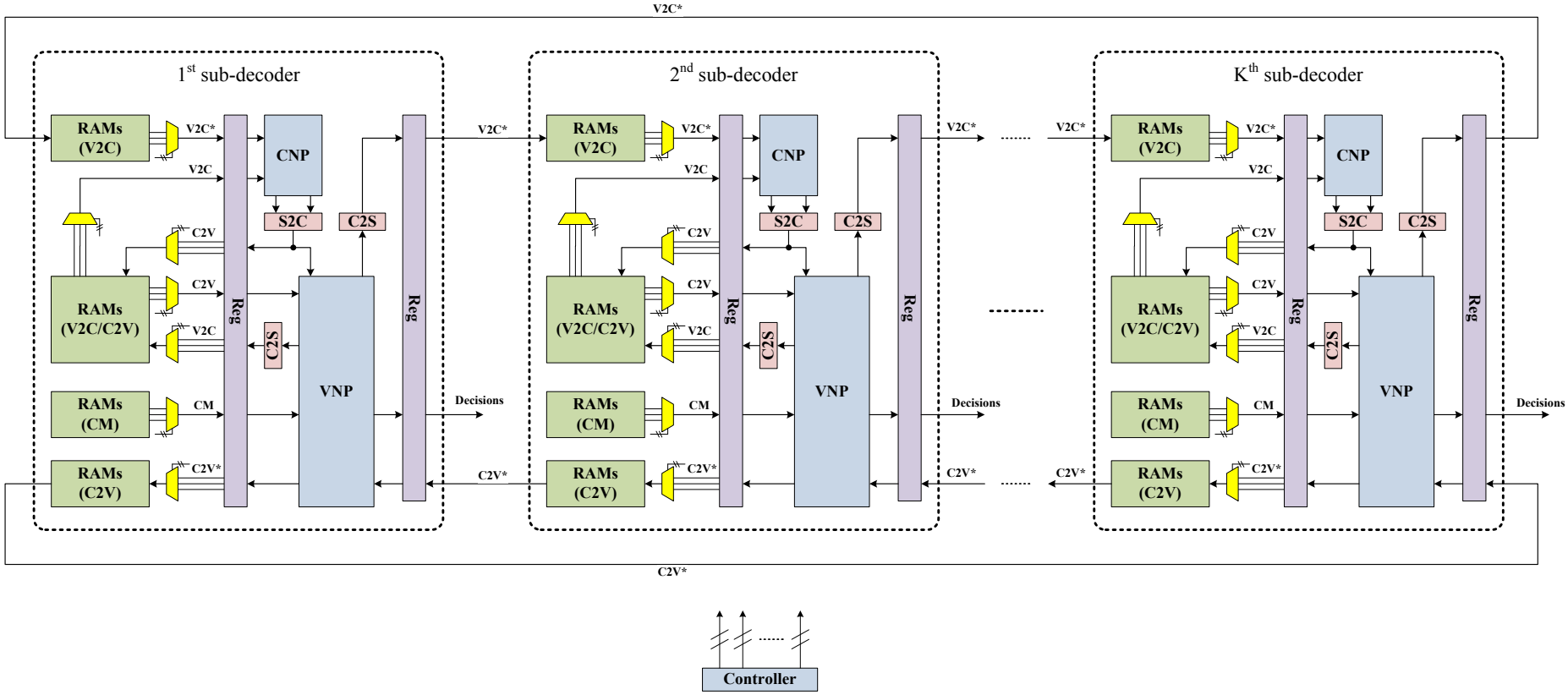


Fig. 1. The overall architecture of the CC-QC-LDPC decoder. The messages passing between adjacent sub-decoders are marked by asterisks.

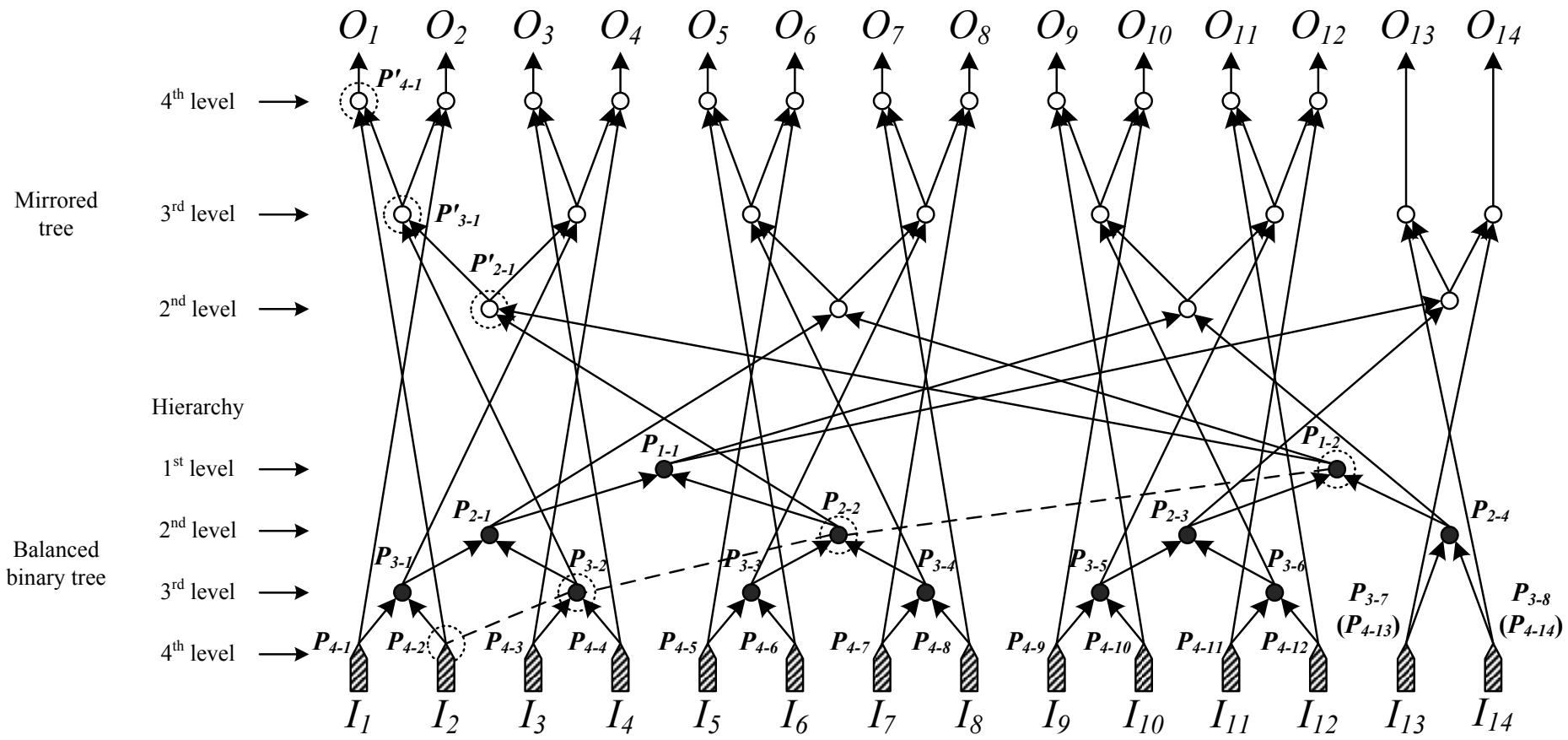


Fig. 2. The parallel structure of LUTs in the CNP with $d_c = 14$. $I_1 \sim I_{14}$ and $O_1 \sim O_{14}$ are the V2C inputs and C2V outputs, respectively. Solid/hollow nodes represent the LUTs and the arrows represent the connections among them. P_{i-j} denotes the partial product of V2C messages, and that with a prime denotes the partial product of a mirror node. Dashed circles and lines are, respectively, the LUTs and valid connections related to the computation of O_1.

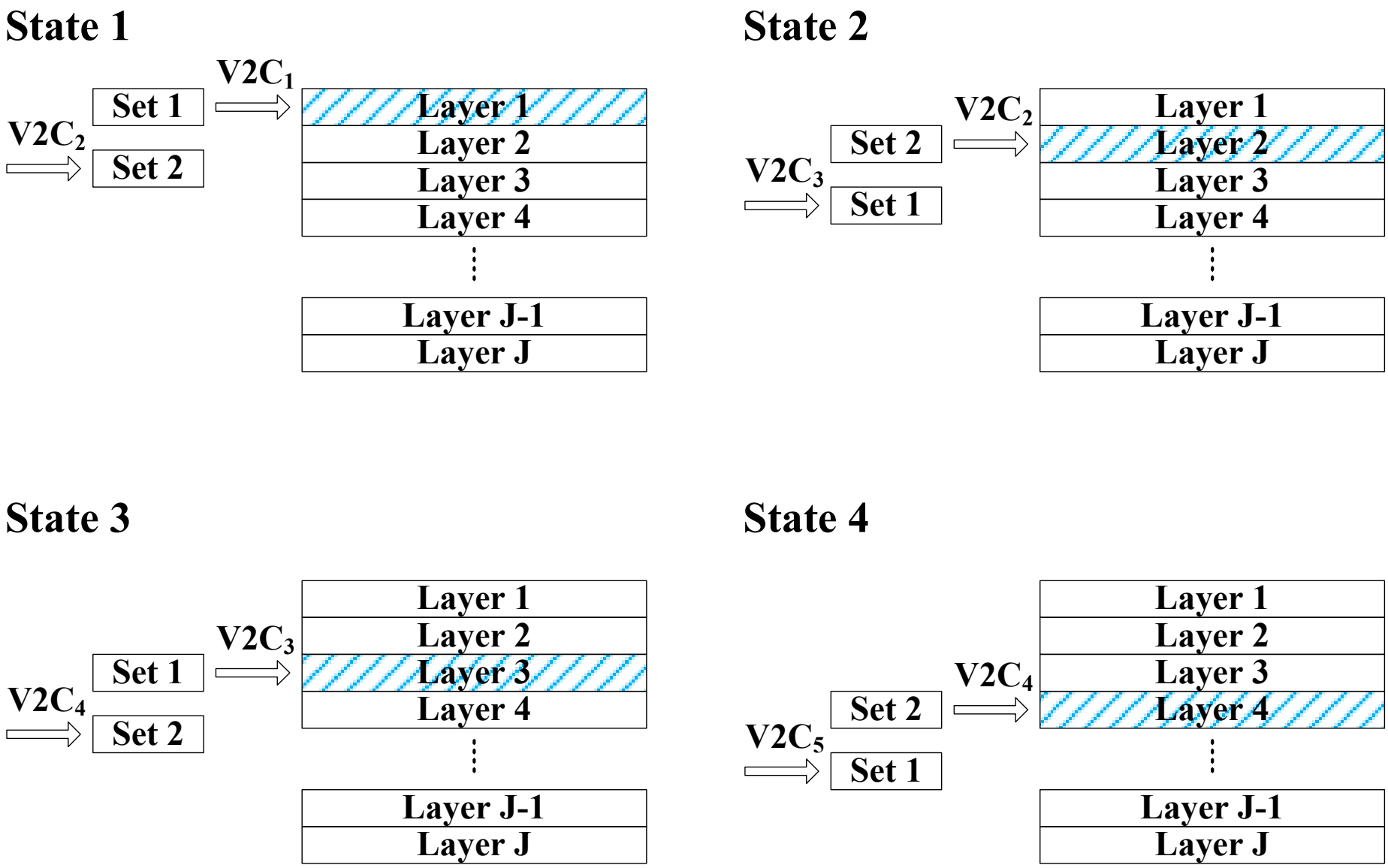
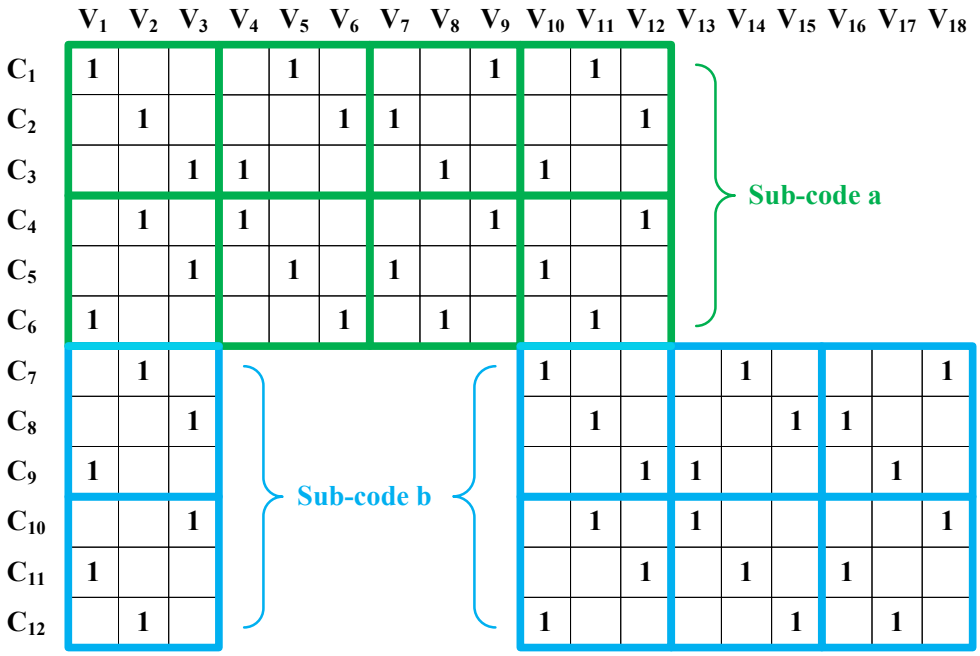


Fig. 3. Two sets of V2C-dedicated RAMs alternate to serve the odd or even layers of a sub-code. V2C_j is the messages from common (with the preceding previous sub-code assumed) variable nodes to the check nodes of Layer j.



RAM$_a$1	RAM$_a$3	RAM$_a$5	RAM$_a$7	RAM$_a$9	RAM$_a$11	RAM$_a$13
V$_1$2C$_1$	C$_1$2V$_1$	V$_5$-C$_1$	V$_9$-C$_1$	V$_{11}$-C$_1$	CM$_4$	CM$_{10}$
V$_2$2C$_2$	C$_2$2V$_2$	V$_6$-C$_2$	V$_7$-C$_2$	V$_{12}$-C$_2$	CM$_5$	CM$_{11}$
V$_3$2C$_3$	C$_3$2V$_3$	V$_4$-C$_3$	V$_8$-C$_3$	V$_{10}$-C$_3$	CM$_6$	CM$_{12}$

RAM$_a$2	RAM$_a$4	RAM$_a$6	RAM$_a$8	RAM$_a$10	RAM$_a$12
V$_2$2C$_4$	C$_4$2V$_2$	V$_4$-C$_4$	V$_9$-C$_4$	V$_{12}$-C$_4$	CM$_7$
V$_3$2C$_5$	C$_5$2V$_3$	V$_5$-C$_5$	V$_7$-C$_5$	V$_{10}$-C$_5$	CM$_8$
V$_1$2C$_6$	C$_6$2V$_1$	V$_6$-C$_6$	V$_8$-C$_6$	V$_{11}$-C$_6$	CM$_9$

RAM$_b$1	RAM$_b$3	RAM$_b$5	RAM$_b$7	RAM$_b$9	RAM$_b$11	RAM$_b$13
V$_{10}$2C$_7$	C$_7$2V$_{10}$	V$_{14}$-C$_7$	V$_{18}$-C$_7$	V$_2$-C$_7$	CM$_{13}$	CM$_1$
V$_{11}$2C$_8$	C$_8$2V$_{11}$	V$_{15}$-C$_8$	V$_{16}$-C$_8$	V$_3$-C$_8$	CM$_{14}$	CM$_2$
V$_{12}$2C$_9$	C$_9$2V$_{12}$	V$_{13}$-C$_9$	V$_{17}$-C$_9$	V$_1$-C$_9$	CM$_{15}$	CM$_3$

RAM$_b$2	RAM$_b$4	RAM$_b$6	RAM$_b$8	RAM$_b$10	RAM$_b$12
V$_{11}$2C$_{10}$	C$_{10}$2V$_{11}$	V$_{13}$-C$_{10}$	V$_{18}$-C$_{10}$	V$_3$-C$_{10}$	CM$_{16}$
V$_{12}$2C$_{11}$	C$_{11}$2V$_{12}$	V$_{14}$-C$_{11}$	V$_{16}$-C$_{11}$	V$_1$-C$_{11}$	CM$_{17}$
V$_{10}$2C$_{12}$	C$_{12}$2V$_{10}$	V$_{15}$-C$_{12}$	V$_{17}$-C$_{12}$	V$_2$-C$_{12}$	CM$_{18}$

Fig. 4. A CC-QC-LDPC code with $K = 2$, $L = 4$, $J = 2$, $W = 1$ and $z = G = 3$.

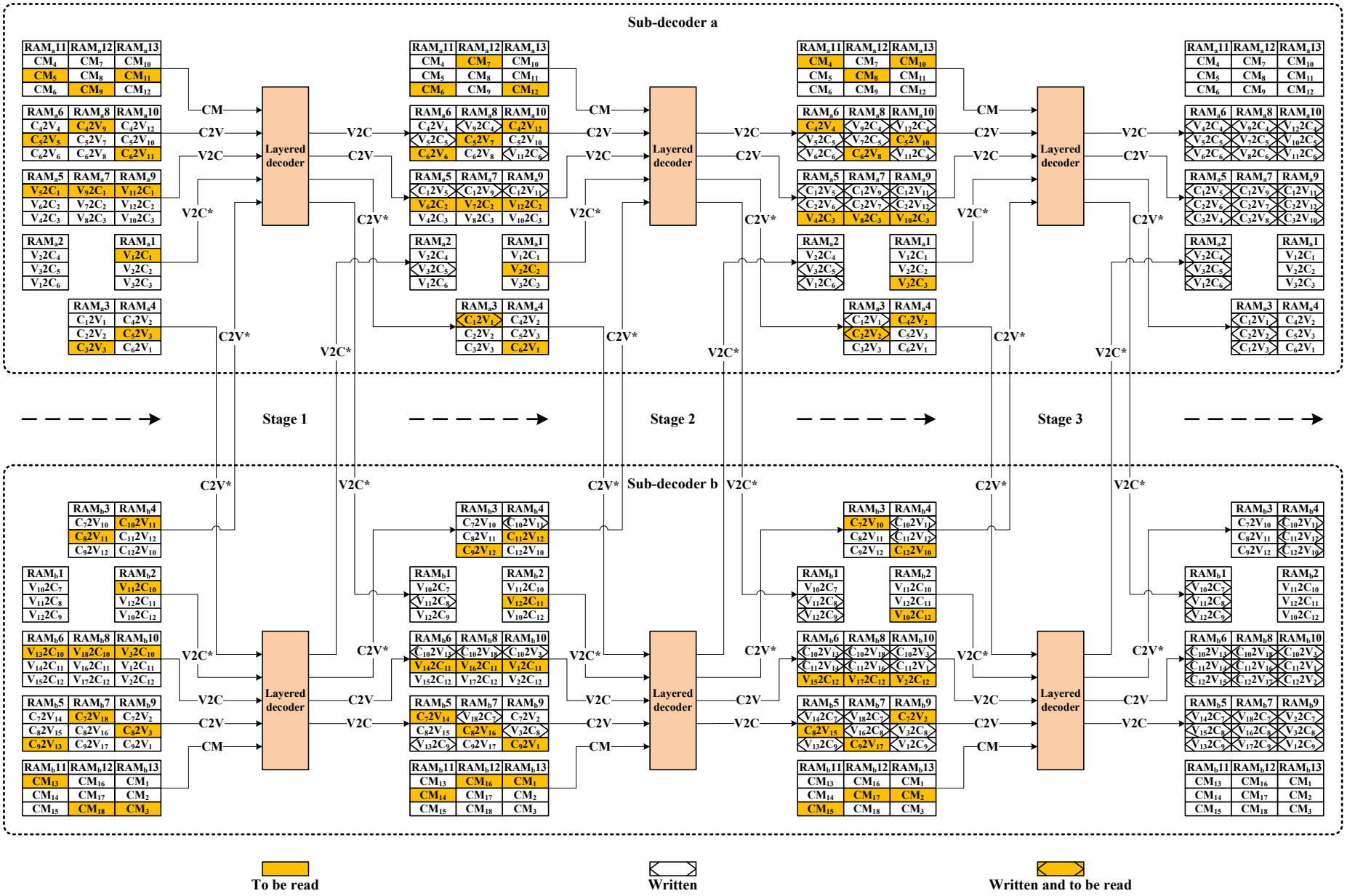


Fig. 5. State 1 of the decoding procedure in the example.

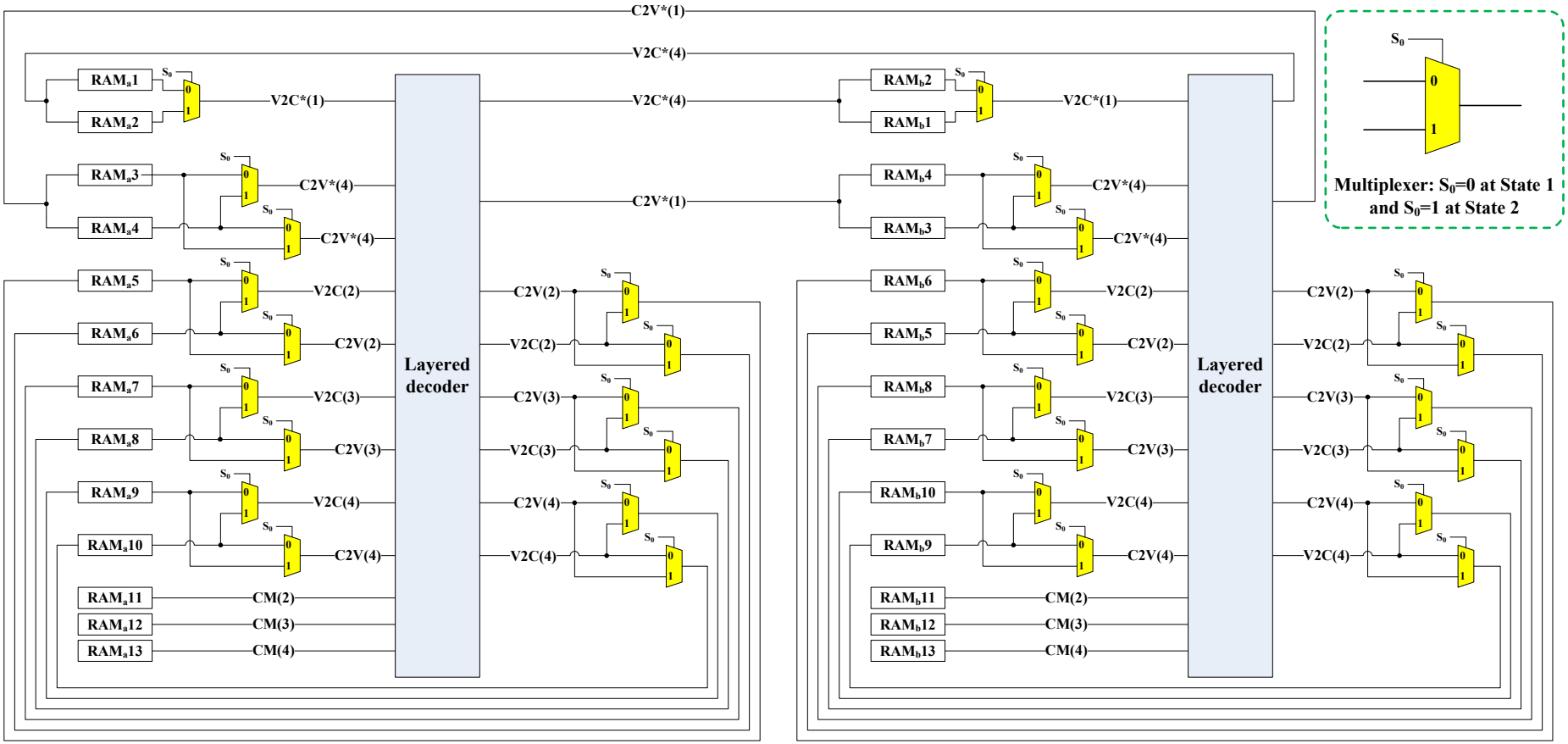


Fig. 6. The switch network for the decoder example. The numbers in parentheses indicate the order of block columns in the sub-code to which the messages correspond.

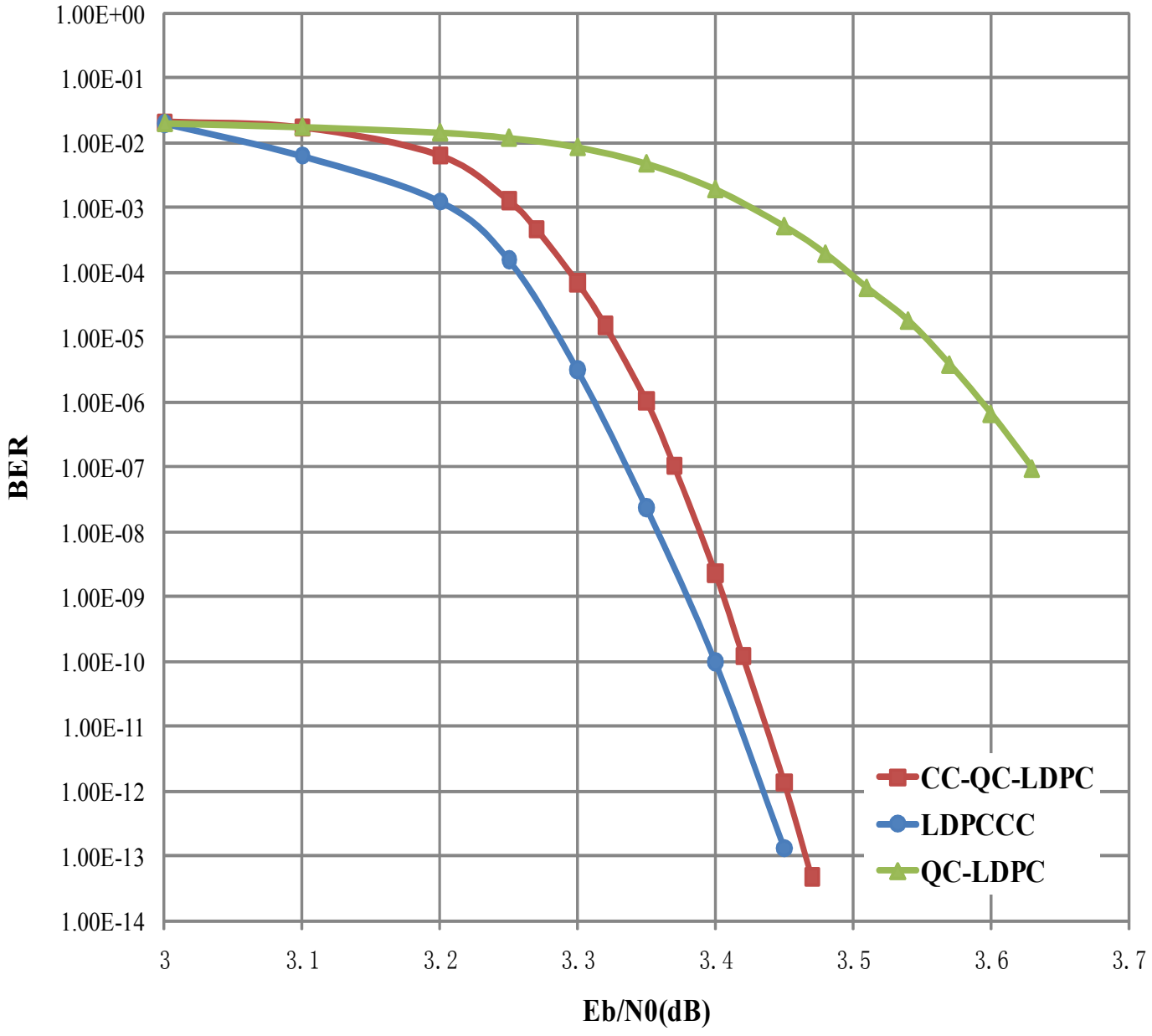


Fig. 7. The BER comparison of different decoders. The CC-QC-LDPC code is code D and the QC-LDPC code is the sub-code of code D. All the results are obtained from FPGA simulation under AWGN channels and 4-bit quantization.

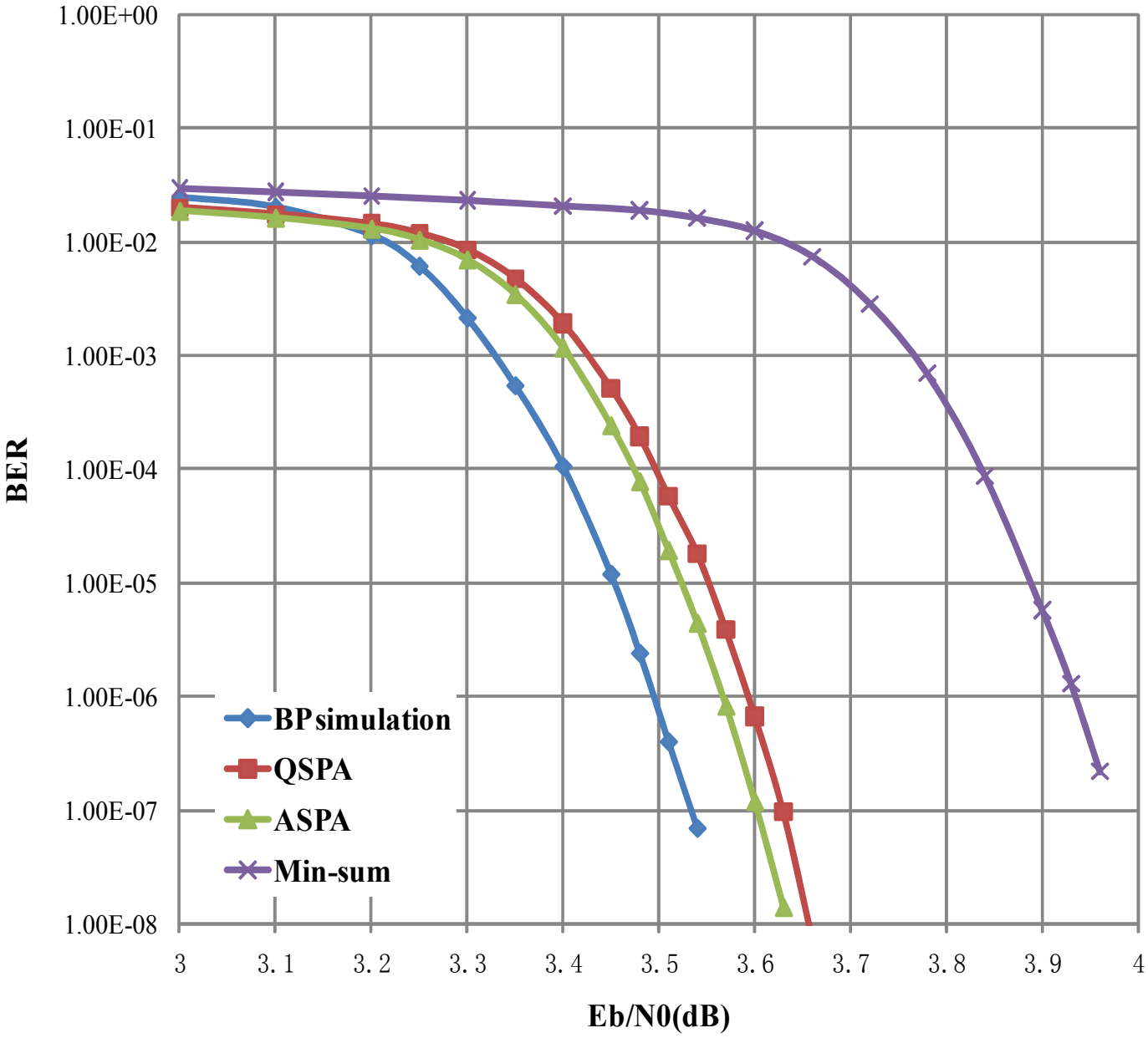


Fig. 8. The BER comparison of various decoders using different CNPs: QSPA using the LUTs, ASPA using mappers and adders, and min-sum using comparators. All the hardware simulations are under 4-bit quantization. BP is the result of computer-based simulation with double-precision floating-point data. The simulations are all based on the same QC-LDPC code with a check-node degree of 28.

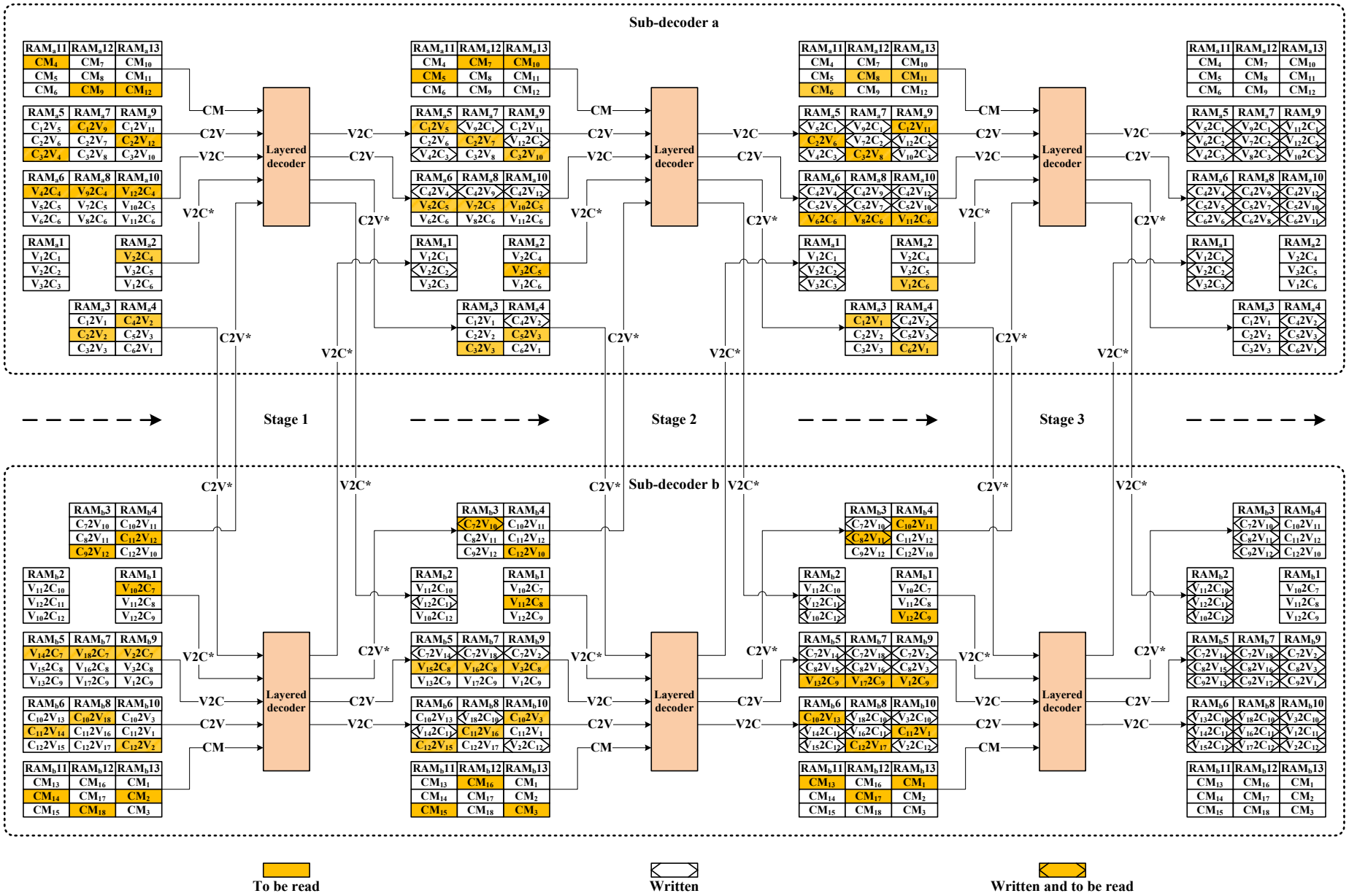


Fig. 9. State 2 of the decoding procedure in the example.