A New Probabilistic Gradient Descent Bit Flipping Decoder for LDPC Codes

Adepu Adarsh Sai

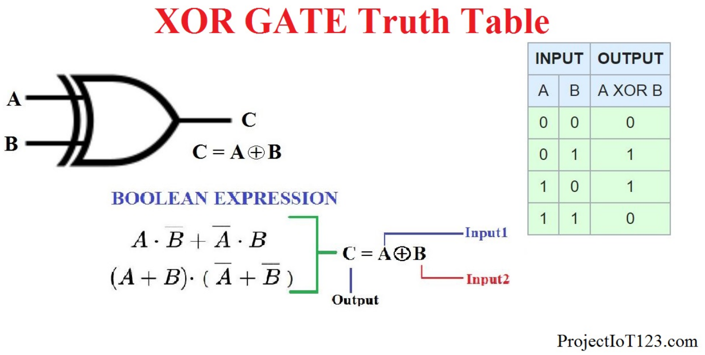
IITH(AI)

June 14, 2021

Exclusive-OR operation(XOR)

XOR operation

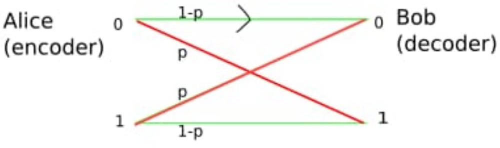
If number of 1s at it inputs is odd then the output is 1
If number of 1s at it inputs is even then the output is 0

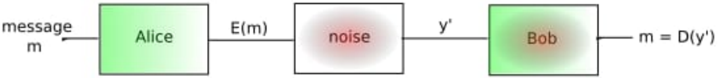


INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Binary Symmetric Channel (BSC_p)

• It is a communications channel model used in coding theory and information theory.
• A transmitter wishes to send a bit(0 or 1). and the receiver will receive a bit.
• The bit will be flipped with a "crossover probability" of p. and otherwise is received correctly.



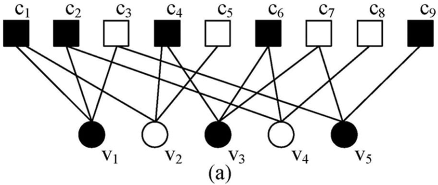


Low-Density Parity-Check code(LDPC)

LDPC

A low-density parity-check code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel.

We can represent an LDPC code graphically, for example



Parity-Check matrix

Parity-check matrix

In coding theory, a parity-check matrix $\mathbf{H}_{(M \times N)}$ of a linear block code C is a matrix which describes the linear relations that the components of a codeword must satisfy.

- Each row of $\mathbf{H}$ corresponds to a parity check equation which is represented by a Check Node(CN)
- Each column represents a codeword symbol which is denoted by a Variable Node(VN)
- Code rate $R = 1 - \frac{M}{N}$
- A length-N binary vector $\mathbf{u}$ is a codeword if and only if $\mathbf{u} \odot \mathbf{H}^\top = 0$
- $\mathrm{N}_c(i)$ is the set of VNs connected to i-th CN, i.e $\mathrm{N}_c(i) = \{j|h_{i.j} = 1\}$.Similarly we can define $\mathrm{N}_v(j)$.
- Degree of j-th VN,$d_v^j = |\mathrm{N}_v(j)|$.Similarly we can define d_c^i
- $i \in \{0, 1, .., M-1\}$ and $j \in \{0, 1, .., N-1\}$

NOTE: $\odot$ is symbol of mod 2 multiplication.

Inversion Function

Inversion function for BSC channel is given by

$$\Lambda_j = c_j \oplus r_j + \sum_{i \in \mathrm{N}_v(j)} s_i \tag{1}$$

where

- $j \in \{0, 1, .., N-1\}$
- **c** is estimated codeword(N$\times$ 1)
- **r** is received codeword(N$\times$1)
- **s** is syndrome vector(M$\times$1)

$$s_i = \oplus_{j \in \mathrm{N}_c(i)} c_j \tag{2}$$

- $\oplus$ represents bit-wise XOR operation

The value of the inversion function is called energy value of the respective bit.

Hierarchy of Bit Flipping decoders

T-PGDBF

↑

PGDBF

↑

GDBF

↑

BF

BF algorithm:

- Energy value of each bit is calculated from inversion function.
- If the energy value is above certain value then the bit is flipped.
- The process is iterated.

GDBF algorithm:

- Energy value of each bit is calculated from inversion function.
- The bits which have the maximum energy value are flipped.

PGDBF algorithm:

• Compared to the GDBF algorithm, only a randomly selected subset of bits which have the maximum energy value are flipped in the PGDBF algorithm.

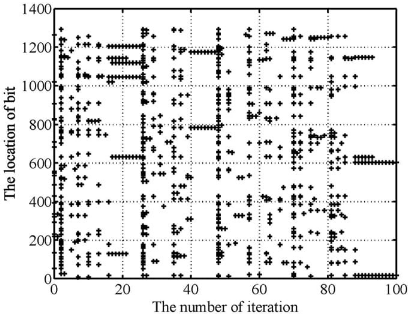


Fig. 2. The locations of flipped bits in each iteration.

A $d_v = 4,\ d_c = 8,\ R = 0.5,\ N = 1296(dv4R050N1296)$ code is used and $T_{max} = 300$.

T-PGDBF algorithm

• From Fig-2 it can be seen that some bits are flipped repeatedly in several iterations,which are called trapping sets.

T-PGDBF algorithm:

• So a tabu-list is defined and the bits which are flipped in the current iteration will be added to this list.

• Only the bits which are not in the tabu-list and have the maximum energy value are flipped with probability p_o.

T-PGDBF algorithm

Algorithm 1: Tabu-list aided PGDBF Algorithm

input : $\boldsymbol{r} = (r_0, r_1, \cdots, r_{N-1})$
initialize: $\boldsymbol{c}^{(0)} = \boldsymbol{r}$, $t = 0$, $(l_0, l_1, \cdots, l_{N-1}) = \mathbf{0}$

for $t = 0$ **to** $T_{max} - 1$ **do**
  for $j = 0$ **to** $M - 1$ **do**
    $s_j = \oplus_{i \in \mathcal{N}_c(j)} c_i^{(t)}$
  if $\boldsymbol{s} = \mathbf{0}$ **then**
    break
  $\Lambda_{max} = 0$
  for $i = 0$ **to** $N - 1$ **do**
    $\Lambda_i(\boldsymbol{c}) = c_i \oplus r_i + \sum_{j \in \mathcal{N}_v(i)} s_j$
    if $\Lambda_{max} < \Lambda_i$ *and* $l_i = 0$ **then**
      $\Lambda_{max} = \Lambda_i$
  Generate the binary random sequence $\boldsymbol{R}$
  for $i = 0$ **to** $N - 1$ **do**
    if $\Lambda_i = \Lambda_{max}$, $R_i = 1$, *and* $l_i = 0$ **then**
      $c_i^{(t+1)} = 1 - c_i^{(t)}$
      $l_i = 1$
    else $l_i = 0$

output : $\boldsymbol{c}^{(t)}$

- **r** is received codeword.
- $c^{(t)}$ denotes the estimated codeword in the t-th iteration.
- l_i indicates whether the i-th bit is in tabu-list or not.
- M = Number of CNs.
- N = Number of VNs.
- **s** is syndrome vector.
- Λ_i = energy value of i-th bit.
- R is the random sequence and $Pr\left\{R_i = 1\right\} = p_o$

Simulation Results

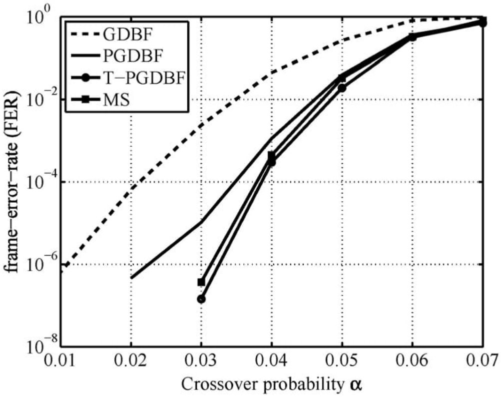


Fig. 3. The simulation results for the dv4R050N1296 LDPC code.

T_{max} is set to 20 for MS algorithm,otherwise $T_{max} = 300$, $p_o = 0.9$

Hardware Architecture

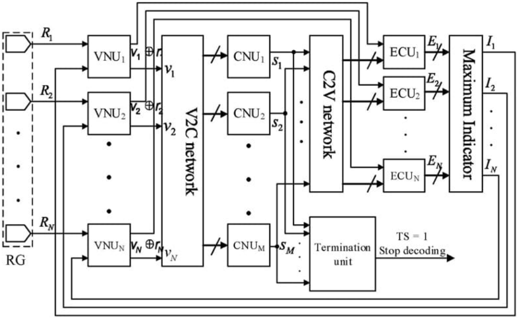


Fig. 4. The top-level architecture of the T-PGDBF decoder.

- At the initializaion of the decoding, **r** is assigned to the estimated codeword **v** in VN units(VNUs).
- M CN units(CNUs) are used to calculated the syndrome vector **s**.

- The energy values are calculated in N ECUs and the maximum indicator produces the indicator value I_i of each bit.
- For the bits which have maximum energy value, $I_i = 1$, otherwise $I_i = 0$.
- The random sequence R is pre-generated and cyclically shifted once in each iteration.
- In each iteration, the termination unit will perform an OR operation among the whole syndrome vector to verify whether the current estimated codeword is valid or not.
- When the valid codeword is obtained or the maximum number of iterations is reached, TS=1 and the decoder is stopped.

Additional Architecture

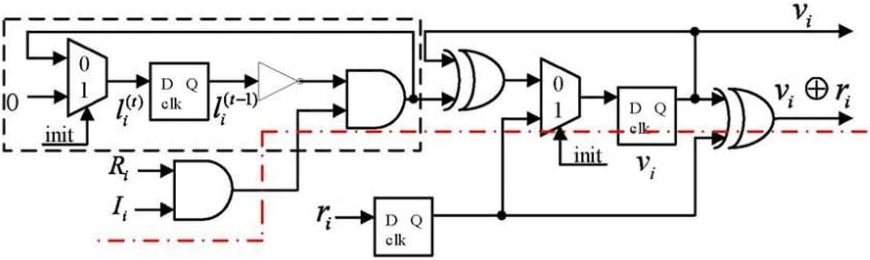


Fig. 5. The architecture of the VNU.

- The signal *init* is only triggered once to initialize l_i and v_i.

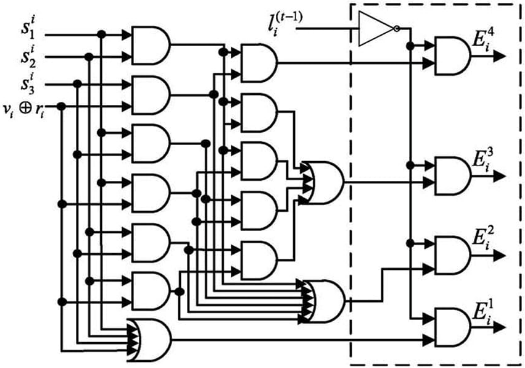


Fig. 6. The architecture of the ECU.

- Fig-6 shows the ECU which is designed for the code with $d_v = 3$.

s_1^i, s_2^i, s_3^i are the syndromes connected to the i-th VN.

- The energy value is expressed in one-hot format. i.e $E_i^{d_v+1} \ldots E_i^2 E_i^1$ where $E_i^k = 1$ if and only if $E_i = k$.

Synthesis and comparison results

THE SYNTHESIS RESULTS FOR THE DV3R050N1296 CODE

	T-PGDBF	PGDBF [8]	MS [9]
Technology	90nm	90nm	65nm
Total Area(mm^2)	0.406	0.367	1.38†
Power(mW)	56.3	51.8	-
Frequency(MHz)	323	357	130.4†
Average Iteration Number	6.10	5.15	1.29
Cycles per Iteration	1	1	6
Throughput (Gbps)	68.6	89.8	21.8†
Area Efficiency (Gbps/mm^2)	169.0	244.7	15.8†

†These are the scaled results under the TSMC 90nm CMOS technology.

- Compared to the PGDBF decoder, the T-PGDBF decoder requires 10.6% extra area and 8.7% extra power.
- T-PGDBF operates at lower frequency.
- Therefore, the proposed T-PGDBF algorithm has greater potential in practical applications due to its high-performance and high-throughput property.