A low-complexity decoder based on LDPC

Yun Feilong[1,a] , Zhu Hongpeng[1], Du Feng[1] , Lv Jing[1]

[1]PLA University of Science and Technology；Nanjing China；210000

[a]1157962870@qq.com

Keywords: LDPC；low-complexity；FPGA

Abstract : This paper designs a low-complexity decoder based on LDPC of GPS standards.In the design ,the decoder we only uses a CNU(Check Node Unit) in the decoding process,which reduce the hardware sources effectively.Besides,we decreases the decoding complexity through the check matrix equivalence transformation,At last ,we realize the decoder based on Xilinx Kintex7 XC7K325T FPGA.The result suggests that the resources only consumes 124 slices after the software ISE layout and wire.

Introduction

LDPC code was proposed by Doctor Gallager at 1960s[1].LPDC code was considered an important development in channel code due to its good performance which is closed to Shannon limit, lower decoding complexity than other codes and decoding in parallel.Now it has been adopted by many international communication standards,such as EEE 802.16e,DVB-S2 ,802.11 ,GPS and CCSDS.

In GPS standards ,LDPC code adopts the lower triangle matrix[2][3],which is obtained by the code constructing algorithm PEG[4].This paper designs a low complexity decoder based on LDPC of the GPS standards .The decoder only uses a CNU(check node unit) repeatedly which reduce the hardware resources effectively.Besides, this paper also decreases the decoding complexity through the check matrix equivalence transformation.At last ,the results suggests that the decoder only costs 124 slices based on (548,274)LDPC after the software ISE layout and wire.

LDPC and algorithm

LDPC in GPS standards

LDPC adopts lower triangle matrix in GPS standards.There two kinds of LDPC in the standards,including (548,274)LDPC and (1200,600)LDPC[5].They have the same check matrix architecture.In figure 1,we give the check matrix $\boldsymbol{H}_{548}$ architecture based on the (548,274)LDPC.

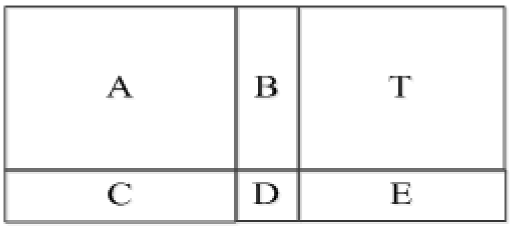


Figure 1.the check matrix architecture

In figure 1,A stands for a 273×274 matrix where the number of ‘1’ is sparse .B is a 273×1 matrix .C is a 1×274 matrix.D is a 1×1 matrix.E is a 1×273 matrix.And T is a 273×273 reversible lower triangle matrix,where the element above the diagonal is ‘0’.

Encoding follows as:we suppose the information vector is $\boldsymbol{s}$ and the check vector is $\boldsymbol{s}1$ and $\boldsymbol{s}2$. $\boldsymbol{s}1$ is a 1×1 matrix and $\boldsymbol{s}2$ is a 1×273 matrix.The final encoding vector $\boldsymbol{c}=(\boldsymbol{s},\boldsymbol{s}_1,\boldsymbol{s}_2)$.We can get (1) and (2) according to the equation $\boldsymbol{H}_{548}\boldsymbol{c}^T=\boldsymbol{0}$.

$$\boldsymbol{A}\boldsymbol{s}^T + \boldsymbol{B}\boldsymbol{s}_1{}^T + \boldsymbol{T}\boldsymbol{s}_2{}^T = \boldsymbol{0} \tag{1}$$

$$\boldsymbol{C}\boldsymbol{s}^T + \boldsymbol{D}\boldsymbol{s}_1{}^T + \boldsymbol{E}\boldsymbol{s}_2{}^T = \boldsymbol{0} \tag{2}$$

Solving (1) and (2),we can get (3) and (4)

$$\boldsymbol{s}_1{}^T = (\boldsymbol{D} + \boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B})^{-1}(\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A} + \boldsymbol{C})\boldsymbol{s}^T \tag{3}$$

$$\boldsymbol{s}_2{}^T = \boldsymbol{T}^{-1}(\boldsymbol{A}\boldsymbol{s}^T + \boldsymbol{B}\boldsymbol{s}_1{}^T) = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{s}^T + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{s}_1{}^T \tag{4}$$

When solving $\boldsymbol{s}1$,we can compute and store the value of $(\boldsymbol{D} + \boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B})^{-1}(\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{A} + \boldsymbol{C})$ ahead of schedule.When solving $\boldsymbol{s}2$,we can also compute and store the value of $\boldsymbol{T}^{-1}\boldsymbol{A}$ and $\boldsymbol{T}^{-1}\boldsymbol{B}$ ahead of schedule[6].

Decoding algorithm

The algorithm adopts the revised min-sum algorithm(MSA)[7],which comes from BP algorithm[8].It is widely used ,because it's easy to realize for hardware. The algorithm as follows:

1)Initializing the LLR of every variable node $L_n^0 = L_{ch,n}$ ($L_{ch,n}$ is the Number n data received from channel). Initializing the data $q_{nm}^0 = L_n^0$,which the variable node V_n transmits to the check node C_m at first.Initializing the iteration time $k = 1$;(usually the row of matrix corresponds to the check node,the column corresponds to the variable node)

2)Check node update:updating the data which C_m transmits to V_n, $n \in B(m), B(m)$ is the variable node collection which connect to the check node C_m.

$$r_{mn,MS}^k = \beta\left(\prod_{n' \in B(m)\backslash n} \operatorname{sgn}\left(q_{n'm}^{k-1}\right)\right)\min_{n' \in B(m)\backslash n}\left|q_{n'm}^{k-1}\right| \tag{5}$$

In(5), β is the revised factor which is constant. $n' \in B(m)\backslash n$ means the collection that eliminates n; $sgn()$ suggests taking the sign; $r_{mn,MS}^k$ is the data which C_m transmits to V_n, also is the produced middle data; q_{nm}^k is the data which V_n transmits to C_m at the Number k time iteration;

3)Variable node update:updating the LLR data L_n^k of variable node V_n and q_{nm}^k which V_n transmits to C_m, $m \in A(n)$ after all check nodes have updated. $A(n)$ is the check node collection which connect to the check node V_n.

$$L_n^k = L_{ch,n} + \sum_{m \in A(n)} r_{mn,MS}^k \tag{6}$$

$$q_{nm}^k = L_{ch,n} + \sum_{m' \in A(n)\backslash m} r_{m'n,MS}^k \quad \text{or} \quad q_{nm}^k = L_n^k - r_{mn,MS}^k \tag{7}$$

In(6), $m \in A(n)$ is the collection including m; L_n^k is the Number n LLR at the Number k time iteration; In(7) $m' \in A(n)\backslash m$ is the collection that removes m;

4)Judging and stopping:we get the sequence W^k at Number k time iteration according to L_n^k;

$$w_n^k = (1 - sgn(L_n^k))/2 \tag{8}$$

In(8), w_n^k is the Number n data through hard judging at Number k time iteration ;

5)We can get check sequence S^k through multiplying the check matrix $\boldsymbol{H}$ and W^k.If $S^k = \theta$ (θ is all 0 sequence), the iteration is stopped and outputting the sequence $W = W^k$;

$$S^k = W^k \boldsymbol{H}^T mod \tag{9}$$

If the check equation(9) can't be satisfied and the time of iteration reach the maximum,the iteration is stopped and decoding is failed.Otherwise, the iteration is continued, $k = k+1$,then jump to Step 2.

The design of decoder

The decoder of this paper adopts all-serial architecture,which only has a CNU (check node unit).Aiming at a check matrix, the decoding process of the architecture follows as:firstly,we take the VN(variable node) data in the first line of the matrix and put the data into CNU to dispose.Then,we can get the new VN data after CNU disposes.Secondly,we use the new VN data to update the old.After updating,we take the data in the next line of the matrix from the new VN data and put them into the same CNU.The decoding process will repeat above steps until the decoder stops.From the decoding process,we can see that the decoding architecture is simple relatively . The process doesn't refer to the VNU(variable node unit) and only need one CNU,which saves the hardware resources effectively.The figure 2 gives the decoding architecture.

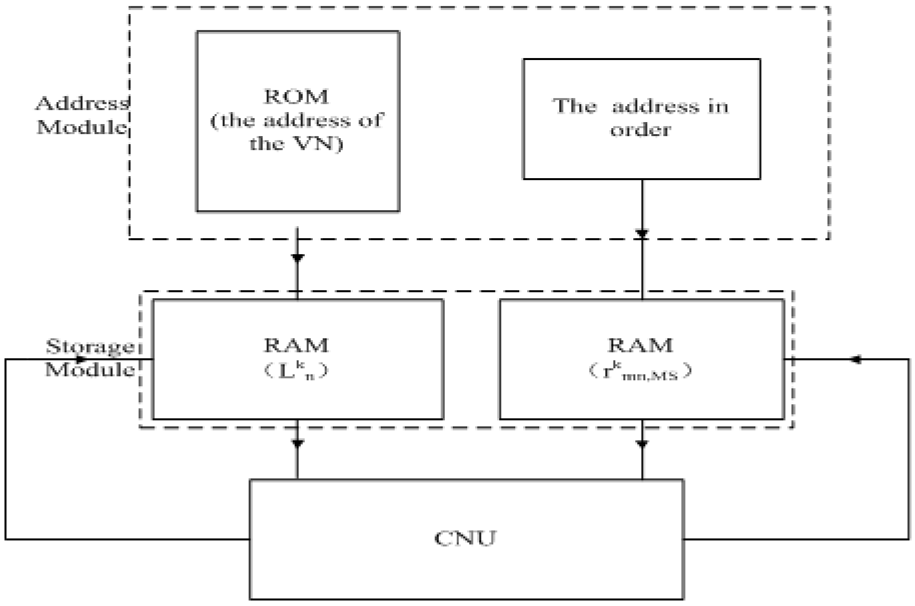
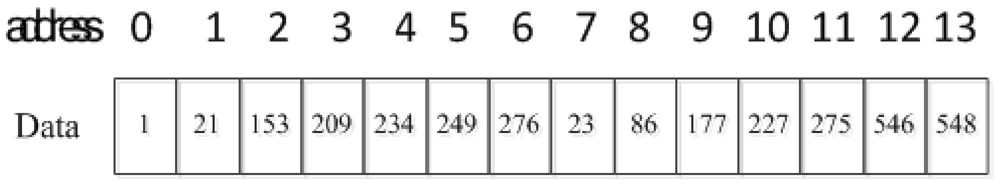


Figure 2. The decoding architecture

In figure 2,the architecture includes the address module, the storage module and the CNU module.

There are two parts in the AM(Address Module).One is the address module of writing and reading the RAM of $r^k_{mn,MS}$.In this module, AM only needs to produce the address from address 0 to last address in order ,which isn't affected by the different VN data positions.The other is the address module of writing and reading the RAM of L^k_n.In this module ,because the n in L^k_n is not sequential in decoding process,so AM should produces the corresponding address according to the different VN data positions.In the design,we put the different VN data positions into one ROM ahead of schedule .So the decoder can read the ROM in order from address 0 to last address.The VN data position in ROM follows in figure 3.In figure 3, the data 1,21,153,209,234,249,276 is the VN data position in Line 0 of the matrix and the data 23,86,177,227,546,548 is the the VN data position in Line 1 of the matrix.The VN data position in other lines is also stored in the ROM .



address	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Data	1	21	153	209	234	249	276	23	86	177	227	275	546	548

Figure 3. The ROM storage

There are two RAMs in the storage module.One is used to store L^k_n,which the depth of is the length of encoding bit,548.The other is used to store $r^{k}_{mn,MS}$,which the depth of is equal to the number of element '1' in check matrix.

We find that the degree is '7' or '8' in matrix $\boldsymbol{H}_{548}$.Due to the different degree,the decoder needs to switch different architecture,which increases the decoding complexity.So,we transform the matrix equivalently to solve this problem.

CNU module main includes subtracting,taking absolute value,comparing and adding module.We adopts pipeline process in CNU module.The CNU module architecture is given in figure 4.

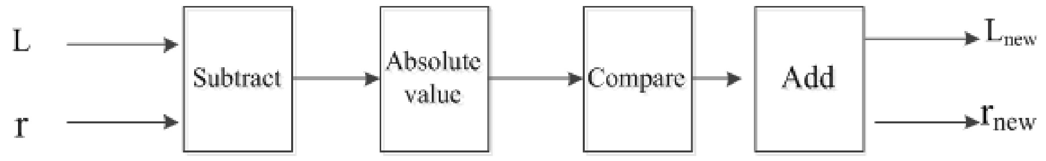


Figure 4.The CNU module

We know that when decoding,the decoder begins from the first line of the matrix.In fact,because the algorithm can be carried out in parallel,so the dispose of every line in matrix is independent.In other words ,the we can exchange any two lines of the matrix and the performance is not affected.According to this idea,we put the lines of Degree 7 into the front lines and put the lines of Degree 8 into the back lines.There are 121 lines of Degree 7 and 153 lines of Degree 8 in matrix $\boldsymbol{H}_{548}$ .The complexity of the decoder is reduced through above matrix transformation.Meanwhile,the decoder doesn't need to mark different degree lines and saves the switching resources between different degree lines.

The performance and the hardware resources

The performance

In figure 5,we give the performance of the decoder based the (548,274)LDPC. The simulation result is got under the condition of the BPSK modulation and AWGN channel.The simulation software is VC6.0.Meanwhile,the received data is quantified by 6bit and the iteration time is 20.

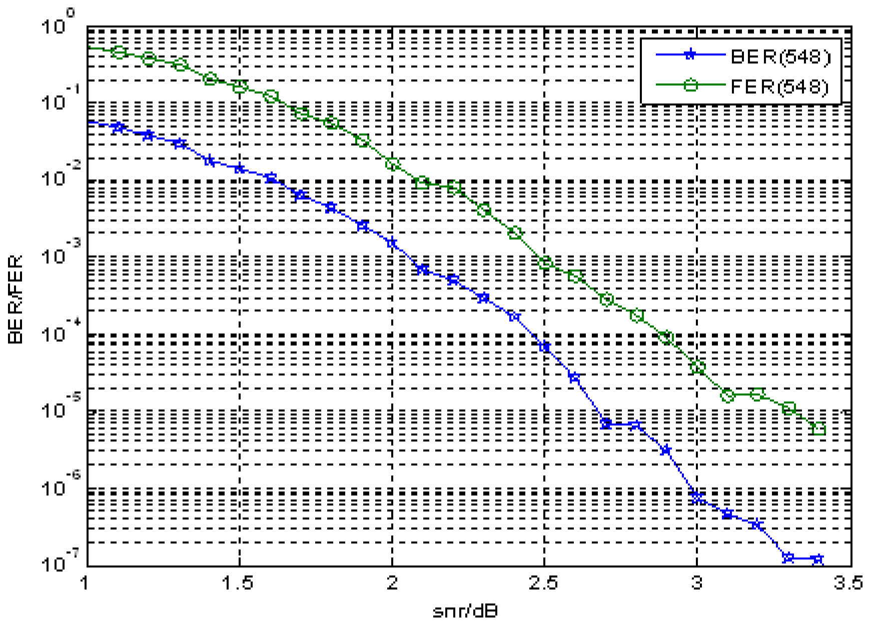


Figure 5.The performance of the decoder

We can see from figure 4 that the BER(Bit Error Ratio) can reach to the level of 10^{-6} when the SNR is 2.5dB and the FER(Frame Error Ratio) can reach to the level of 10^{-6} when the SNR is 2.9dB.So the performance is very good and can be applied into the practice.

The hardware resources

We realize the decoder in the hardware based on the (548,274)LDPC.The decoder adopts the Xilinx Kintex7 XC7K325T FPGA.After the software ISE layout and wire,the resources is given in table 1.

Table 1.The cost resources

resource	number	percent
Slices	124	1%
RAM	2 pieces of 18K 2 pieces of 36K	2%

The maximum frequency can reach to 304.414MHz.So the throughput can reach to 0.1232Mbps.The formula of throughput is given in (10).

$$Throughput = \frac{n_{LDPC} \times f_{max}}{C \times Iter} \qquad (10)$$

In (10), $f_{\max}$ is the maximum frequency which the decoder can reach to. n_{LDPC} is the number of information bit. C is the number of the clock in one iteration time. $Iter$ is the number of the iteration time.

Conclusion

This paper designs a low complexity decoder based on LDPC of GPS standards. The decoder only uses a CNU in the decoding process,which reduces the resources effectively.At last ,the result suggests that the resources only costs 124 slices after FPGA implement and the throughput can reaches to 0.1232Mbps.The research have important value in the low speed and limited resources filed.

Reference

[1] Gallager R G. Low-density parity-check codes[J]. Information Theory, IRE Transactions on, 1962, 8(1):21-28.

[2] GPS Joint Program Office. Navstar global positioning system interface specification: Draft IS-GPS-800[R], 2006:26-27.

[3] Betz J W,Blanco M A,Cahn C R,rt al. Description of the L1C signal[C]//Proceedings of ION GNSS,2006:2080-2091.

[4] Hu Yingpeng,Wang jiang,Cheng Wen. Research on encoding and decoding based on short length of LDPC[J].Communication Technology,2013,45（9):25-28.

[5] Navstar GPS.Space Segment/User Segment L1C Interfaces[R].Draft IS-GPS-800,19April,2006.

[6] Wang Jianhui,Li Jingyuan,Ni Shaojie，et al.The design of LDPC decoder based on the signal of GPS L1C[J]. Journal of National University of Defense Technology. 2013.1:014.

[7] Jiang Ming. Decoding Algorithm and Applied Research of Low-Density Parity-Check Code[D] .Nanjing:Southeast University,2006.(in China)

[8] Chen J,Dholakia A,Eleftheriou E,et al .Reduced-complexity decoding of LDPC codes[J].Communications,IEEE Transactions on,2005,53(8):1288-1299.