Specification

Hamming Error Correction Module

Kamath O., Nebhrajani A. V.

Contents

1 Theory

1.1 Characterization of Hamming Codes

Hamming codes are a class of linear systematic single-error correcting codes.

Definition 1.1. *A linear (n, k) code is a code whose 2^k codewords form a k-dimensional subspace of the vector space of all n-tuples over the field $GF(2)$.*

Note that for any (n, k) linear code[1], this means that the modulo-2 sum of any two codewords (or any other linear combination in $GF(2)$) results in another codeword, since the codewords themselves form a subspace.

Definition 1.2. *A systematic linear code is one whose codewords contain the message symbols unmodified, and have the redundant checking part created using linear sums of the message symbols.*

Hamming codes can correct exactly one error, and can be used to detect two errors using an extra overall parity bit. For a Hamming code using m parity bits:

Property	Number of Symbols
Code length	$n = 2^m - 1$
Data length	$k = 2^m - m - 1$
Distance	3

This describes a Hamming($2^m - 1$, $2^m - m - 1$) code. We now study general linear systematic codes, then specialize in the last subsection to Hamming codes.

1.2 Generator Matrix and Parity Check Matrix

Linear codes can entirely be described by their generator matrix G and parity check matrix H. The generator matrix for a linear systematic code is:

$$G = \begin{bmatrix} P \mid I_k \end{bmatrix}$$

where I_k is a $k \times k$ identity matrix and P is a submatrix that generates the parity-check equations of the code. The rows of G must be linearly independent. To encode a message $\mathbf{u}$, compute $\mathbf{u} \cdot G = \mathbf{v}$, where $\mathbf{v}$ is a codeword. Since the right submatrix is the identity matrix, this simply replicates the k information bits. The leftmost $n - k$ bits form the parity bits. The generator matrix describes entirely how to encode a message using the code.

The parity check matrix is a $(n - k) \times n$ matrix

$$H = \begin{bmatrix} I_{n-k} \mid P^T \end{bmatrix}$$

so that any codeword $\mathbf{v}$ in the code satisfies $\mathbf{v} \cdot H^T = 0$. The columns are linearly independent. Clearly, $G \cdot H^T$ is the zero matrix.

[1] n refers to the total number of symbols, while k refers to the number of symbols used only for information.

1.3 Syndrome and Error Detection

Let us transmit an encoded message $\mathbf{v}$ over a noisy channel, and let $\mathbf{r}$ be the received message at the output. If $\mathbf{e}$ is the error caused by the channel, clearly, $\mathbf{r} = \mathbf{v} + \mathbf{e}$. On receiving $\mathbf{r}$, the receiver computes

$$\mathbf{s} = \mathbf{r} \cdot H^T$$

where $\mathbf{s}$ is called the syndrome of $\mathbf{r}$. If the syndrome is zero, $\mathbf{r}$ is a codeword, otherwise, the presence of errors has been detected. It is possible that $\mathbf{s} = \mathbf{0}$ but errors still occurred, but causing one codeword to turn into another (nonzero) codeword. For linear codes, there are $2^k - 1$ such undetectable error patterns. Now, see that:

$$\mathbf{s} = \mathbf{r} \cdot H^T = (\mathbf{v} + \mathbf{e}) \cdot H^T = \mathbf{v} \cdot H^T + \mathbf{e} \cdot H^T = \mathbf{e} \cdot H^T$$

The syndrome is therefore a linear combination of the error digits, and therefore solving the system of $n - k$ linear equations yields the precise locations of the errors. However, there is no unique solution. We select the error with the least Hamming weight.

1.4 Decoding and Error Correction

The received vector $\mathbf{r}$ could be one of 2^n possible messages. A decoding is a partition of this set of 2^n possibilities into 2^k disjoint subsets such that each subset corresponds to a single codeword. $\mathbf{r}$ gets mapped to whichever codeword's subset it happens to be in. A way to represent this partition is the "standard array" representation:

$$\begin{bmatrix}
\mathbf{v_1 = 0} & \mathbf{v_2} & \cdots & \mathbf{v_i} & \cdots & \mathbf{v_{2^k}} \\
\mathbf{e_2} & \mathbf{e_2 + v_2} & \cdots & \mathbf{e_2 + v_i} & \cdots & \mathbf{e_2 + v_{2^k}} \\
\mathbf{e_3} & \mathbf{e_3 + v_2} & \cdots & \mathbf{e_3 + v_i} & \cdots & \mathbf{e_3 + v_{2^k}} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{e_j} & \mathbf{e_j + v_2} & \cdots & \mathbf{e_j + v_i} & \cdots & \mathbf{e_j + v_{2^k}} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{e_{2^{n-k}}} & \mathbf{e_{2^{n-k}} + v_2} & \cdots & \mathbf{e_{2^{n-k}} + v_i} & \cdots & \mathbf{e_{2^{n-k}} + v_{2^k}}
\end{bmatrix}$$

where all the $\mathbf{e_j}$ are distinct. The 2^{n-k} disjoint rows of this matrix are called the cosets of the code C, and the first tuple e_j of each coset is called the coset leader.

Theorem 1.3. *All the 2^k elements of a coset have the same syndrome. Different cosets have different syndromes.*

Proof. Consider an element $\mathbf{v_i} + \mathbf{e_j}$ in the coset with leader $\mathbf{e_j}$. Then,

$$\mathbf{s} = (\mathbf{v_i} + \mathbf{e_j})H^T = \mathbf{v_i}H^T + \mathbf{e_j}H^T = \mathbf{e_j}H^T$$

Thus all vectors of a coset have the same syndrome. To show that different cosets have different syndromes, by contradiction, let e_j and e_l have the same syndrome and $j < l$. Then,

$$\mathbf{e_j}H^T = \mathbf{e_l}H^T \implies \mathbf{e_j} + \mathbf{e_l}H^T = 0$$

This means $\mathbf{e_j} + \mathbf{e_l} = \mathbf{v_i}$ where $\mathbf{v_i}$ is some codeword. This means $\mathbf{e_l} = \mathbf{e_j} + \mathbf{v_i}$, that is, $\mathbf{e_l}$ is in the coset of $\mathbf{e_j}$, which would violate the selection of the $\mathbf{e_i}$'s as unique. □

Therefore, if we have a map from syndromes to coset leaders to correct with, we know how to decode.

1.5 Hamming Codes are Perfect Codes

Hamming codes are linear codes where the primary restriction is that the coset leaders all having Hamming weight less than or equal to 1. Thus, if we visualize each column of the standard array as being a hypersphere of radius $e = 1$ with the codeword at the center, any other codeword is at least $2e + 1 = 3$ units away. Thus, the distance of a Hamming code is 3. Such a code is called a *perfect code* since it achieves maximum packing density of the space without overlapping the spheres.

The parity check matrix $H = \begin{bmatrix} I_m \mid Q \end{bmatrix}$ simply has submatrix Q whose columns are the m tuples of weight 2 or more, acting as a parity check. Indeed, the column vectors of H are simply a permutation of all nonzero m tuples in $GF(2)$.

1.6 Worked Example

An example serves to solidify the theory covered, and gives us insight into how the implementation would look. A Hamming(7, 4) code can be defined using the following generator and parity check matrices:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

1.6.1 Encoding

To encode a message $\mathbf{u} = (1\ 0\ 1\ 1)$, we simply multiply it with G:

$$(1\ 0\ 1\ 1) \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = (1\ 0\ 0\ 1\ 0\ 1\ 1)$$

In hardware, this can simply be done using a combinational circuit: the rightmost four symbols are simply the information bits, and the leftmost three symbols are XOR operations.

Let's now transmit this message over a noisy channel.

1.6.2 Decoding

We receive $\mathbf{r} = (1\ 0\ 0\ 1\ 1\ 1\ 1)$. We first compute the syndrome $\mathbf{r}H^T$:

$$\mathbf{s} = (1\ 0\ 0\ 1\ 1\ 1\ 1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = (0\ 1\ 1)$$

Which coset leader does this syndrome correspond to? By checking the syndrome of each coset, we get the following table:

Syndrome	Coset leader
(1 0 0)	(1 0 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0 0)
(1 1 0)	(0 0 0 1 0 0 0)
(0 1 1)	(0 0 0 0 1 0 0)
(1 1 1)	(0 0 0 0 0 1 0)
(1 0 1)	(0 0 0 0 0 0 1)

Note the coset leaders are chosen to have minimum Hamming weight. Since this is a Hamming code, the weights are at most 1. Now, to fix the error, we compute $\mathbf{v}^* = \mathbf{r} + \mathbf{e}$. Thus,

$$(1\ 0\ 0\ 1\ 1\ 1\ 1) + (0\ 0\ 0\ 0\ 1\ 0\ 0) = (1\ 0\ 0\ 1\ 0\ 1\ 1)$$

Which is indeed the transmitted message.

The combinational hardware implementation is simply to compute the syndrome using the parity check matrix, then use a truth table such as the one above to map the syndrome to the error digits. Then, modulo-2 add the error digits to the received digits to get the corrected output.

2 Interface

The design consists of two main modules: the encoder and the decoder. The encoder takes a input data of width k, and returns Hamming-encoded output of width $k + m$ where m is the number of parity bits. Parity bits are placed at one less then power-of-two indexes.

	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	d9	d8	d7	d6	d5	d4	p8	d3	d2	d1	p4	d0	p2	p1	
		✓		✓		✓		✓		✓		✓		✓	p1
...	✓			✓	✓			✓	✓			✓	✓		p2
	✓	✓	✓					✓	✓	✓	✓				p4
	✓	✓	✓	✓	✓	✓	✓								p8

Parity bit `pX` is the parity of codeword bits having index whose $(\log_2(\texttt{X} + 1))^{\text{th}}$ bit is set. Permuting the parity bits and data bits this way has the advantage of structuring the Hamming code as a binary search: the sum of the incorrect parity bits of the received codeword is the index of the flipped bit, or is 0 in case there are no errors.

The parameter used is the number of data bits, k. The number of parity bits m is computed using k to satisfy the inequality

$$k \leq 2^m - m - 1$$

Unused bits are zero-filled. This implementation chooses to ignore any bits that are "zero filled", since the only operation on the bits is an XOR. Moreover, reducing the number of bits reduces the probability of error.

2.1 Signals

2.1.1 Encoder

For an input of width k data bits, and output width $n = k + m$, where m is the number of parity bits, the signal table is:

Signal Name	Width	Type	Description	Drive
`CLK`	1	Input	User clock signal	Square wave
`RST`	1	Input	Reset the device	Active low
`DIN_VAL`	1	Input	Data input validity	Active high
`DIN`	k	Input	Data bits	Active high
`EOUT_VAL`	1	Output	Encoded output validity	Active high
`EOUT`	n	Output	Encoded output	Active high

The default state of the encoder is the output flip-flops reset and `EOUT_VAL` set to `0`.

The largest Hamming code supported for a 1Ghz clock is Hamming(1023, 1013). Our expected latency is 1 nanosecond (1 clock cycle). Hamming(1023,1013) requires an XOR of 512 bits, which when synthesized as a tree of gates has a delay of $9 \times 100\text{ps} = 900\text{ps}$, assuming an XOR gate has a delay of 100 picoseconds.

2.1.2 Decoder

For an input of $n = k + m$ bits, and output width k bits, the signal table is:

Signal Name	Width	Type	Description	Drive
CLK	1	Input	User clock signal	Square wave
RST	1	Input	Reset the device	Active low
EIN_VAL	1	Input	Codeword input validity	Active high
EIN	n	Input	Codeword bits	Active high
DOUT_VAL	1	Output	Data output validity	Active high
DOUT	k	Output	Data output	Active high

The latency of the decoder is also 1 clock cycle of a 1Ghz clock.

2.2 Timing Diagrams

The expected behavior for a Hamming(7,4) code encoding is:

DIN	EOUT
0000	0000000
0001	0000111
0010	0011001
0011	0011110
0100	0101010
0101	0101101
0110	0110011
0111	0110100
1000	1001011
1001	1001100
1010	1010010
1011	1010101
1100	1100001
1101	1100110
1110	1111000
1111	1111111