

差集合巡回符号 エラー訂正回路設計仕様書

和田知久

今回はテレビの文字多重放送に用いられているエラー訂正方式である差集合巡回符号 (Difference-set Cyclic Code) を用いたエラー訂正回路の設計を行います。とはいえ、コンテストの課題ですので、小さなデジタル回路を設計することを念頭に、符号長を実際に使用されている長さよりかなり短くし、取り扱いやすい大きさとします。また、いくつかの設計オプションを設定し、各個人やチームによっていろいろなくふうができるように、実現方法にはある程度自由度を与えています。要求される設計内容は、HDL (VHDL もしくは Verilog HDL) による設計と論理合成です。論理合成ツールに制約はなく、無償で提供されている論理合成ツールでも参加できます。HDL 設計に興味のある方はどしどし参加してください。また、余裕のある方は FPGA などへ実装すれば、努力を認めて高い評価が得られると思います。ぜひトライしてみてください。

課題の概要

図1に今回設計するデジタル・データの送受信システムのブロック図を示します。システムは大きく分けて、送信機 (transmitter) と受信機 (receiver) に分かれますが、今回の設計課題は受信機 (receiver) です。

送信者はビット列を出力します。送信機はシリアルに入力されるビット列11ビットに対して差集合巡回符号の符

号化処理 (encode) を行い、11ビットの入力にエラー訂正用のビット列10ビットを付加して21ビットの符号とします。この21ビットが伝送されるわけですが、伝送中にビット・エラーが発生します。図1ではエラー・ジェネレータのブロックがエラーを発生させます。

受信機ではこのエラーを含んだ21ビットの符号を受け取り、エラー訂正処理を行います。エラー訂正処理には有名などころではリード・ソロモン方式などがありますが、処理が複雑でコンテストの課題には不向きです。しかし、差集合巡回符号方式では、多数決回路でエラー訂正を行うことができ、回路的には非常にシンプルです。

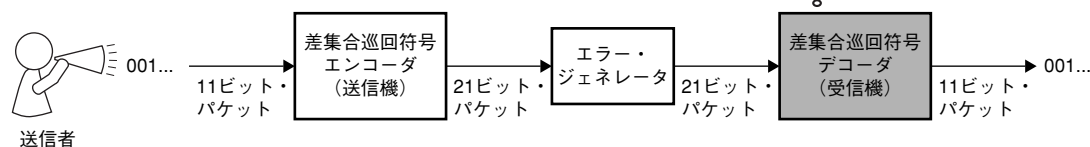
今回の課題では差集合巡回符号の細かい知識がなくても設計できるように、仕様をくふうしています。設計すべきことは比較的単純ですので、デジタル設計の知識のある方は自信を持って、課題に取り組んでください。

差集合巡回符号について

●パリティ・チェックの復習

デジタル・データの伝送におけるエラーの有無をチェックする場合、パリティ・チェックという方式がよく用いられます。表1に、8ビットのデータに1ビットのパリ

〔図1〕
送受信システム図



ティ・ビットを付加した例を示します。8ビット・データの中の‘1’の個数が偶数であればパリティ・ビットは‘0’であり、‘1’の個数が奇数であればパリティ・ビットは‘1’となります。したがって、パリティ・ビット生成回路は非常に簡単であり、多入力のXOR(排他的論理和)となります。

したがって、パリティ・ビットを付加した9ビット幅のデータを送信し、受信側で{D₇, D₆, D₅, D₄, D₃, D₂, D₁, D₀, Parity}のXORをとれば、その出力が‘1’のときはどこかにエラーが発生したことが検知でき、XORの出力が‘0’のときはエラーが発生しなかったと判断できます。ただし、2ビットのエラーは検知できませんし、送信した9ビットのうちどのビットがエラーで反転したかを知ることにはできません。

●差集合巡回符号のイントロダクション

差集合巡回符号を文献などで調査すると、ガロア体の多項式などを用いたややこしい理論が説明されています。しかし、ここでは符号理論に詳しくない方、むしろディジタル回路設計者のような方に理解しやすいように、差集合巡回符号を用いたエラー訂正方法を紹介します。

今回設計するシステムでは先にも説明したように、11ビットの情報ビットに対して10ビットのエラー訂正用

のビットを付加し、21ビットのデータを送信します。この21ビットには以下に示すようなおもしろい特性があります。

表2に、11ビットの情報ビットとして{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}というすべて‘1’からなる情報を送る場合の例を示します。エラー訂正用の10ビットは表2のようになります。この生成方法は後ほど説明します。

この21ビットの送信ビットに対して以下の表3に示すような5種類A₁～A₅のパリティ・チェックを考えます。

パリティ・チェックA₁ではビット番号で9, 12, 13, 18, 20のところに1があります。この五つの位置に対応する送信ビットSB(9), SB(12), SB(13), SB(18), SB(20)のXORを取ると‘0’になります。同様にパリティ・チェックA₂～A₅でもすべて‘0’となります。まとめると表4のようになります。

ここで送信ビットの20ビット目SB(20)がエラーで反転した場合を考えます。SB(20)は5種類のパリティ・チェック式にすべて含まれています。したがって、五つのパリティ・チェック式A₁からA₅はすべて‘1’となります。

また、送信ビットの20ビット目以外で1カ所のエラーが発生した場合を考えます。表3を見てわかるように、送信ビットの20ビット目以外のビットは5種類のパリティ・チェック式のどれかに1回のみ使用されます。つまり、A₁からA₅のうちどれか一つが‘1’となります。

したがって、上記パリティ・チェック式を計算し、その多数が‘1’となるとSB(20)にエラーが発生していることが検知され、SB(20)のビットを反転することでエラーの訂正を行うことができます。

本設計コンテストでは、

$$A_1 + A_2 + A_3 + A_4 + A_5 \geq 3$$

〔表1〕8ビット・パリティの例とパリティ発生回路

8ビットのデータ								パリティ・ビット	パリティ発生回路
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Parity	
1	0	1	1	0	0	0	1	0	
0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	

〔表2〕

情報ビット(11ビット)と送信ビット(21ビット)の例

	エラー訂正用(10ビット)										情報ビット(11ビット)									
情報ビット(11ビット)											1	1	1	1	1	1	1	1	1	1
送信ビット(21ビット)	0	0	1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1

〔表3〕

情報ビット(11ビット)と送信ビット(21ビット)の例

ビットの位置番号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
送信ビット(SB)	0	0	1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
パリティ・チェックA ₁										1			1	1					1		1
パリティ・チェックA ₂		1										1			1	1					1
パリティ・チェックA ₃					1		1										1			1	1
パリティ・チェックA ₄	1					1		1										1			1
パリティ・チェックA ₅			1	1					1		1										1

〔表4〕

5種類のパリティ・チェック

パリティ・チェックA ₁	A ₁ =SB(9) xor SB(12) xor SB(13) xor SB(18) xor SB(20)=0
パリティ・チェックA ₂	A ₂ =SB(1) xor SB(11) xor SB(14) xor SB(15) xor SB(20)=0
パリティ・チェックA ₃	A ₃ =SB(4) xor SB(6) xor SB(16) xor SB(19) xor SB(20)=0
パリティ・チェックA ₄	A ₄ =SB(0) xor SB(5) xor SB(7) xor SB(17) xor SB(20)=0
パリティ・チェックA ₅	A ₅ =SB(2) xor SB(3) xor SB(8) xor SB(10) xor SB(20)=0