

# Responsive Link をベースにした 高信頼なリアルタイム通信機構の設計と評価

水頭 一壽<sup>†1</sup> 吉住 修<sup>†1</sup>  
松谷 宏紀<sup>†1</sup> 山崎 信行<sup>†1</sup>

本論文では、信頼性の高いリアルタイム通信を実現するために、リアルタイム通信規格の Responsive Link をベースにした通信機構を設計し、評価を行う。提案手法は Responsive Link の誤り訂正符号と伝送路符号を拡張して耐ノイズ性能を向上するとともに、伝送路特性に応じた符号の組み合わせで通信できる機構を設計する。評価結果から、誤り訂正符号と伝送路符号を拡張することで、様々なノイズ特性においてノイズ耐性が向上することがわかった。また、面積評価により提案手法が現実的なハードウェアコストで実現可能であることがわかった。

## Design and Evaluation of Dependable Real-Time Communication Method Based on Responsive Link

KAZUTOSHI SUITO,<sup>†1</sup> OSAMU YOSHIZUMI,<sup>†1</sup>  
HIROKI MATSUTANI<sup>†1</sup> and NOBUYUKI YAMASAKI<sup>†1</sup>

In this paper, we design and evaluate a dependable real-time communication mechanism based on the Responsive Link. The proposed communication mechanism improves the noise immunity by combining error correction and line codes of the Responsive Link in response to the communication environment. The simulation results show that the noise immunity is improved by combining proper error correction and line codes for various communication environments. The proposed dependable real-time communication mechanism is implemented for the Responsive Link. Although the implementation results show that it increases the implementation area of the Responsive Link by 42%, it provides a favorable tradeoff between highly-dependable communication and reasonable area.

### 1. はじめに

近年、ヒューマノイドロボット等の複雑なリアルタイムシステムの出現により、単一プロセッサで行う処理の限界や、システムの耐故障性等を向上するために、プロセッサを複数用いた分散リアルタイムシステムへの要求が高まっている。分散リアルタイムシステムでは、ノード内の処理においてリアルタイム性を満たすだけでなく、ノード間の通信においてもリアルタイム性を保証しなければならない。このような背景から、我々は分散リアルタイムシステムのための通信機構として Responsive Link<sup>11)</sup> を研究開発しており、国内外で標準化も行っている<sup>6)4)</sup>。

我々はディペンダブルな分散リアルタイムシステムを実現するために、VLSI システムの基盤技術を研究しており<sup>13)</sup>、ディペンダビリティの実証検証として、ロボット小次郎<sup>3)</sup> に組み込むシステム LSI の研究開発を行っている。ディペンダブルな分散リアルタイムシステムを実現するための要素技術として、ノード間通信のディペンダビリティが必要不可欠である。ターゲットとするシステムでは定常時に 80V50A、高負荷時には 80V200A ものモータ駆動電流が使用され、これに伴い発生するノイズが通信に影響を与える。ディペンダブルなリアルタイム通信を実現するためには、リアルタイム性を保証すると共に、高い耐ノイズ性能を有する高信頼なリアルタイム通信規格が必要と考える。そこで、本論文では分散リアルタイム通信規格 Responsive Link をベースに、高信頼なリアルタイム通信機構を設計し、評価を行う。

本論文の構成は以下の様になっている。2 節で研究対象である Responsive Link の特徴について述べ、3 節で Responsive Link をベースにした高信頼なリアルタイム通信機構の設計について述べ、4 節でハードウェアコスト、対ノイズ性能の評価を行う。最後に 5 節で本稿をまとめる。

### 2. 分散リアルタイム通信規格 Responsive Link

Responsive Link は、分散リアルタイムシステムのための通信機構として研究開発されており、国内外で標準化されている。本節では Responsive Link の特徴について述べる。

<sup>†1</sup> 慶應義塾大学大学院理工学研究科開放環境科学専攻

Department of Computer Science, Graduate School of Science and Technology, Keio University

### 2.1 ハードリアルタイム通信とソフトリアルタイム通信の分離

ヒューマノイドロボットなどの分散リアルタイムシステムのノード間通信は、アクチュエータ制御のための通信や同期のための通信など、データ量は少ないが通信遅延に対する制約が厳格な通信と、カメラなどの各種センサから入力されるマルチメディアデータなど、データ量が多くバンド幅を必要とするが、通信遅延に対する制約が寛容な通信の2種類に大別できる。前者の通信をハードリアルタイム通信、後者の通信をソフトリアルタイム通信と呼ぶ。パケットサイズを大きくすると、スループットは向上するが通信遅延は増大する。パケットサイズを小さくすると、通信遅延は減少するがスループットは低下する。この様にスループットと通信遅延はトレードオフの関係にあるため、ハードリアルタイム通信とソフトリアルタイム通信の要求は相反している。そこで、ハードリアルタイム通信とソフトリアルタイム通信の通信ライン分離し、各ラインを point-to-point の双方向シリアル通信で結合する。ハードリアルタイム通信のための通信ラインをイベントリンク、ソフトリアルタイム通信のための通信ラインをデータリンクと呼ぶ。イベントリンクは通信遅延を短くするため、パケットサイズは 16Byte と小さくなっている。対してデータリンクはスループットを向上させるために、パケットサイズは 64Byte となっている。

### 2.2 優先度によるパケットの追い越し

Responsive Link では、優先度によるパケットの追い越しを行う。パケットの追い越しを行うため、優先度による調停器と追い越し用のバッファ、回避用の外部記憶インタフェースを内蔵したスイッチを持っている<sup>12)</sup>。図1に Responsive Link のスイッチを示す。スイッチは5つの入出力ポートを持ち、ポート0番は自ノード、ポート1~3番は他ノードへの入出力ポートとして使用される。パケットの出力ポートが衝突した際、優先度に基づいて調停を行い、優先度の高いパケットが先にポートに出力される。パケットの追い越しは、通信におけるプリエンプションに相当し、優先度によるパケットの追い越しを実現することで、通信のリアルタイム性を保証することが可能になる。

### 2.3 宛先アドレスと優先度によるルーティングと優先度付け替え

Responsive Link の経路制御はルーティングテーブルを用いて行う。Responsive Link の経路制御では、宛先アドレスと優先度の両方を用いて経路を決定する。優先度によって経路を制御することによって、同じあて先のパケット内でも優先度によって経路の設定が出来る。宛先アドレスと優先度の両方を用いた経路制御によって、図2に示す様に、優先度の高いパケットは他のパケットと衝突が起きない専用回線を実現したり、迂回経路を設けたりすることが可能になる。

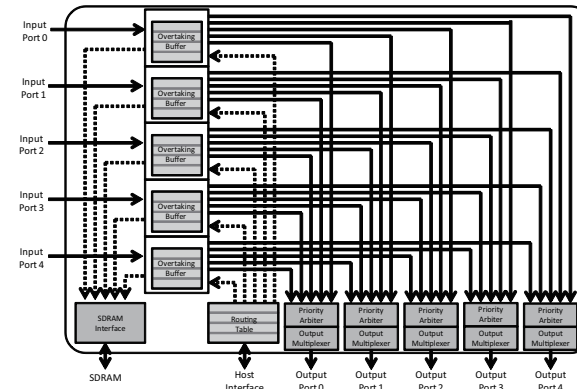


図1 Responsive Link のスイッチ

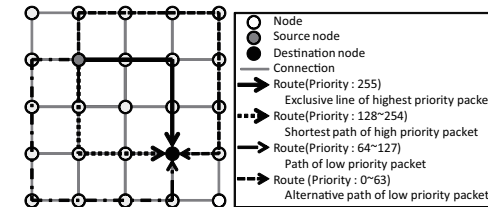


図2 Responsive Link の経路制御

Responsive Link では、リアルタイム通信を分散管理型で制御するために、ノード毎にパケットの優先度を付け替えることが可能である。優先度付け替えの制御はルーティングテーブルを用いて行う。ルーティングテーブルには経路情報と共に、出力時に設定する優先度が格納されている。優先度付け替えによって、低優先度パケットへの優先度継承や、通信途中でのパケットの加減速が可能である。

### 2.4 誤り訂正符号と伝送路符号

Responsive Link では、誤り訂正符号としてハミング符号<sup>8)</sup>を採用している。生成多項式は  $x^4 + x + 1$  を使用し、8ビットのデータに対して4ビットの冗長ビットを付加し、任意の1ビットの誤りを訂正可能である。伝送路符号には、0を送信する場合には信号レベルを反転し、1を送信する場合には信号レベルを保持する NRZI (Non Return to Zero Inverted)

符号を使用する．伝送にはエンベデッドクロック方式を用いたシリアル通信を使用する．このため、NRZI を用いる際に信号レベルが連続しないように、送信データに 1 が 5 個連続した場合 0 を挿入する BS (Bit Stuffing) を組み合わせる．受信側は DPLL (Digital Phase Lock Loop) を使用してクロックデータリカバリを行い、データを受信する．伝送時の変調レートは伝送路特性に応じて 50, 100, 200, 400, 800Mbaud の中から選択可能である．

### 3. 高信頼リアルタイム通信機構の設計

Responsive Link の耐ノイズ性能を向上させるには、誤り訂正符号の強化と伝送路符号の変更が考えられる．提案手法では新たに誤り訂正符号と伝送路符号を実装し、通信時に伝送路特性に合わせて任意の符号を組み合わせ使用可能な機構を設計する．

#### 3.1 誤り訂正符号の拡張

##### 3.1.1 BCH 符号

Responsive Link に採用されているハミング符号は、データを 8 ビットに対して 4 ビットの冗長ビットを付加し、1 ビットの誤り訂正を実現していた．耐ノイズ性能を向上させるために、新たに BCH 符号<sup>7)</sup> を採用する．BCH 符号は符号化利得が大きくかつ符号長や符号強度が柔軟である．また、エンコーダ・デコーダの構成が簡単な為、ハードウェアによる実装が比較的容易である．このため無線通信やフラッシュメモリのエラー訂正などで用いられている．提案手法では生成多項式に  $x^8 + x^7 + x^6 + x^4 + x^2 + x + 1$  を使用し、8 ビットのデータに 8 ビットの冗長符号を付加し、2 ビットまでのランダム誤り、3 ビットまでのバースト誤り、全ビットが 1 になった場合の誤りを検出可能にする．

##### 3.1.2 リードソロモン符号

1 バイト単位の誤り訂正では回復できないバースト誤りへ対応するため、多バイトでの誤り訂正符号を行うため、リードソロモン符号<sup>5)</sup> を採用する．リードソロモン符号はバースト誤りに有効で、光学ディスクや磁気ディスクなどの記憶装置や、宇宙通信などの分野で採用されている．提案手法では生成多項式に  $g_15(x)$  を使用し、4 バイトのデータに 2 バイトの冗長符号を付加し、1 バイト以内の誤り訂正を可能にする．Responsive Link はカッツスルーでのパケット追い越しを行うため、符号化するサイズをヘッダと同じ 4 バイトとしている．

#### 3.2 伝送路符号の拡張

##### 3.2.1 8B/10B 符号化

Responsive Link に採用されている伝送路符号は NRZI と BS を組み合わせたものである．NRZI と BS を組み合わせた伝送の特徴は、送信データに 1 が 5 個連続した場合にのみ 0 を挿入するという点で、符号化の効率が非常に高い．8 ビットのデータを符号化した場合、8~9 ビットになる．しかし、送信データによって動的に符号長が変わるため、データの切れ目が判別しにくいという問題がある．図 3 に示すように、エラーが発生した際に意に反してビットの挿入・削除が発生し、同期が崩れるという問題がある．最悪の場合、エラー発生以降ずっと通信不能に陥る場合もある．この点を考慮し、伝送路符号にエンベデッドクロック方式でかつ固定長の符号化である 8B10B<sup>9)</sup> を採用する．8B10B は 8 ビットのデータを 10 ビットに変換して DC バランシングすることで、データ中にクロックを埋め込んでいる．また、ランニングディスペリティを使用することでエラーの検出も可能である．8B10B は PCI Express<sup>1)</sup> や InfiniBand<sup>2)</sup> などの高速シリアル通信で使用されている．

み 0 を挿入するという点で、符号化の効率が非常に高い．8 ビットのデータを符号化した場合、8~9 ビットになる．しかし、送信データによって動的に符号長が変わるため、データの切れ目が判別しにくいという問題がある．図 3 に示すように、エラーが発生した際に意に反してビットの挿入・削除が発生し、同期が崩れるという問題がある．最悪の場合、エラー発生以降ずっと通信不能に陥る場合もある．この点を考慮し、伝送路符号にエンベデッドクロック方式でかつ固定長の符号化である 8B10B<sup>9)</sup> を採用する．8B10B は 8 ビットのデータを 10 ビットに変換して DC バランシングすることで、データ中にクロックを埋め込んでいる．また、ランニングディスペリティを使用することでエラーの検出も可能である．8B10B は PCI Express<sup>1)</sup> や InfiniBand<sup>2)</sup> などの高速シリアル通信で使用されている．

TX Data	11101000	11111101
Encoded Data	11101000	111110101
	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
RX Data	11111000	110110101
Decoded Data	11111000	110110101

図 3 NRZI+BS によるエラーの例

##### 3.2.2 4B10B 符号化

送路符号に使用する NRZI や 8B10B では、伝送路上のデータ 1 ビット化けた場合でも、復号後に複数ビットのエラーとして検出される場合がある．この問題を回避するためには、伝送路符号自体にエラー訂正機能を持たせる必要がある．そこで、DC バランシングによるエンベデッドクロックと同時に、1 ビットのエラー訂正が可能な 4B10B を新たに提案する．4B10B は変換テーブルを用いて 4 ビットのデータを 10 ビットに変換する．4B10B の変換テーブルを表 1 に示す．符号化後の 10 ビットは、お互いハミング距離が 3 以上離れている．復号時には伝送されてきた 10 ビットをテーブル内から検索し、ハミング距離が 1 以内のものがあれば正しく復号され、それ以外の場合はエラーとして処理される．4B10B では 2bit 以上のエラーが発生すると必ず 0 を出力する仕様とした．この仕様はエラー発生時に意に反してハミング距離が離れる可能性があるため、改善の余地があるだろう．

#### 3.3 符号化方式の組み合わせと符号化率

提案手法では、誤り訂正符号にリードソロモン符号と BCH 符号、伝送路符号に 8B10B と 4B10B を新たに追加した．これに加えて、従来の Responsive Link で使用されているハ

表 1 4B10B 変換テーブル

4B	10B	4B	10B	4B	10B	4B	10B
0000	1100101100	0100	0111010001	1000	1001110001	1100	1011010010
0001	1011001100	0101	1100011001	1001	0111000110	1101	1001100110
0010	1100110010	0110	0101110100	1010	1010110100	1110	1010101001
0011	0110011100	0111	1101000101	1011	1101001010	1111	01110101010

ミング符号と NRZI+BS も使用可能である．表 2 に選択可能な符号化の組み合わせと，符号化率を示す．複数ある誤り訂正符号と伝送路符号を組み合わせることで，様々な伝送路特性に柔軟に対応可能である．

表 2 符号化方式の組み合わせ表

4 バイト単位の誤り訂正符号	1 バイト単位の誤り訂正符号	伝送路符号	符号化率
RS (48, 32)	BCH (16, 8)	NRZI+BS (9, 8)	29.6%
		8B10B (10, 8)	26.7%
		4B10B (10, 4)	13.3%
	HAM (12, 8)	NRZI+BS (9, 8)	39.5%
		8B10B (10, 8)	35.6%
		4B10B (10, 4)	17.8%
No ECC	8B10B (10, 8)	53.3%	
	4B10B (10, 4)	26.7%	
No RS	BCH (16, 8)	NRZI+BS (9, 8)	44.4%
		8B10B (10, 8)	40.0%
		4B10B (10, 4)	20.0%
	HAM (12, 8)	NRZI+BS (9, 8)	59.3%
		8B10B (10, 8)	53.3%
		4B10B (10, 4)	26.7%
	No ECC	8B10B (10, 8)	80.0%
		4B10B (10, 4)	40.0%

### 3.4 エンコーダ・デコーダの設計

Responsive Link をベースに，新たに BCH 符号，リードソロン符号，8B10B，4B10B を採用したエンコーダ，デコーダを設計する．図 4 にエンコーダ，図 5 にデコーダのブロック図を示す．エンコーダ，デコーダ共に 4 バイト単位の誤り訂正符号であるリードソロンの符号器は 1 つ実装されており，1 バイト単位の誤り訂正符号であるハミング符号，BCH 符号，および伝送路符号である NRZI+BS，8B10B，4B10B の符号器はそれぞれ出力ポートと同数の 4 つずつ実装されている．

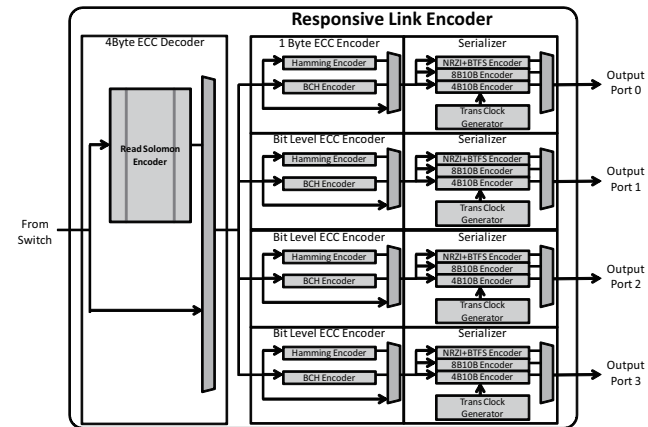


図 4 エンコーダのブロック図

エンコーダはスイッチ側からデータが入力されると，まず 4 バイト単位の誤り訂正符号のエンコーダに入力される．リードソロン符号が有効な場合，リードソロン符号のエンコーダにデータが入力される．リードソロン符号が無効な場合は，入力データはバイパスされる．リードソロン符号のエンコーダはパイプライン化されており，連続的にパケットを処理できる．次にデータは 1 バイト単位の誤り訂正符号のエンコーダに送られる．ビット単位の誤り訂正符号のエンコーダは設定に応じてハミング符号か BCH 符号を選択する．ビット単位の誤り訂正符号が無効の場合，データはバイパスされる．次にデータはシリアライズに入力され，伝送路符号の設定に応じて NRZI+BS，8B10B，4B10B のいずれかで符号化され，出力ポートへ送信される．各エンコーダの入力部分には FIFO があり，符号間のビット幅の違いを吸収している．

デコーダは入力ポートからデータが入力されると，信号が DPLL へ入力され，受信波形からクロックを抽出する．次に受信データと抽出したクロックをデシリアライズに入力し，受信データの切り出しと伝送路符号の復号を行う．次に 1 バイト単位の誤り訂正符号のデコードを行う．1 バイト単位の誤り訂正符号が使用されていない場合は，データはバイパスされる．次にリードソロン符号が有効な場合は，リードソロン符号のデコーダへ送信される．リードソロン符号のデコーダはエンコーダ同様パイプライン化されている．リードソロン符号が無効の場合はデータがバイパスされ，最終的にデコードされたデータはスイッチに送られる．

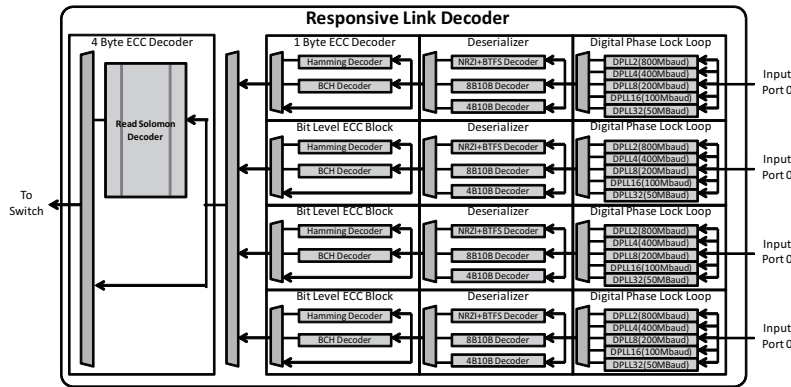


図 5 デコーダのブロック図

## 4. 評価

### 4.1 ハードウェアコスト

提案手法の実装に伴うハードウェアコストについて評価を行う。評価には Synopsys 社 Design Compiler 2010.12 を用い、TSMC 社 130nm のプロセスをターゲットに論理合成を行い、面積とゲート数を比較する。タイミング制約は、Serializer/Deserializer など通信に使用するクロックで動作する箇所は 800MHz、それ以外の部分は 200MHz としている。

新たに追加した誤り訂正符号、および伝送路符号がそれぞれどの程度ハードウェアコストに影響を及ぼすかを明らかにするため、各符号のエンコーダとデコーダの面積とゲート数を図 6 に示す。1 バイト単位の誤り訂正符号において、BCH 符号はオリジナルのハミング符号と比べ、エンコーダでは面積が 52.8%、ゲート数が 64.3%増加しており、デコーダでは面積が 94.8%、ゲート数が 264.6%増加している。符号長が 12 ビットのハミング符号に対し、BCH 符号の符号長は 16 ビットと長いため、ハードウェアコストが増加している。伝送路符号において、8B10B はオリジナルの NRZI+BS と比べ、エンコーダでは面積が 44.5%、ゲート数が 51.9%増加しており、デコーダでは面積が 3.2%、ゲート数が 4.1%増加している。4B10B はオリジナルの NRZI+BS と比べ、エンコーダでは面積が 16.1%ゲート数が 23.6%増加しており、デコーダでは面積が 8.0%、ゲート数が 13.5%増加している。4 バイト単位の誤り訂正符号であるリードソロモン符号では、1 バイト単位の誤り訂正符号であるハミング符号と比較すると、エンコーダでは面積が約 100 倍、ゲート数が 150 倍、デコー

ダでは面積が約 300 倍、ゲート数が 380 倍と大きくなっている。

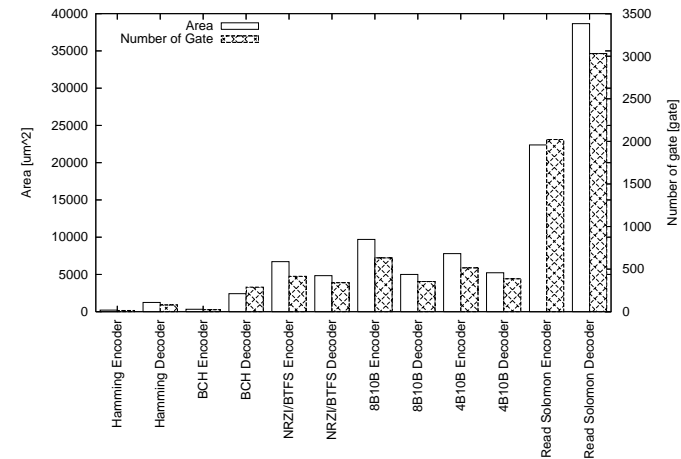


図 6 符号化モジュールのハードウェアコスト

実装数を考慮し、誤り訂正符号と伝送路符号がそれぞれハードウェアコストに及ぼす割合を図 7 に示す。最も単体のサイズが大きかったリードソロモン符号がゲート数・面積共に最大の割合を占めてはいるが、伝送路符号の 8B10B、4B10B、NRZI+BS との差は面積で約 0.95%~7%、ゲート数で約 7%~12%程度の差である。リードソロモン符号と伝送符号 3 種の合計で、面積全体の約 93%、全ゲート数の約 91%を閉めている。このことより、提案手法のハードウェアコストは、リードソロモン符号と伝送符号の 8B10B、4B10B、NRZI+BS が支配的であることがわかる。

次に全体のハードウェアコストの増加を見積もるため、提案手法を実装した Responsive Link と、従来の Responsive Link の面積とゲート数を表 3 に示す。提案手法を実装することで、従来と比較して 41.4%面積が増加し、73.5%ゲート数が増加することがわかった。ハードウェアコストの増加は小さくはないが、提案手法を実装した Responsive Link は約 2.2mm 角、後工程のマージンを 5 割見込んだ場合でも約 3.0mm 角の面積であり、十分に実装可能である。

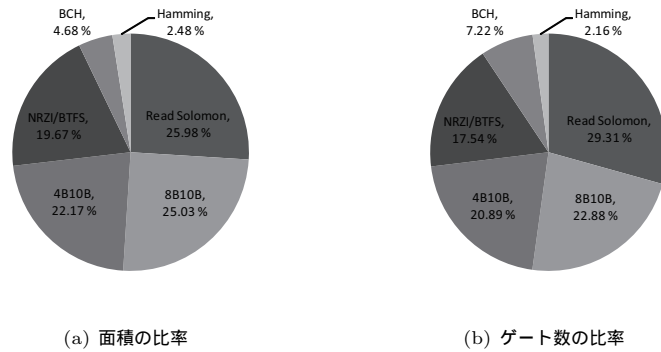


図 7 符号化モジュールのハードウェアコスト比率

表 3 Responsive Link 全体のハードウェアコスト

	面積 [ $mm^2$ ]	ゲート数 [K gate]
オリジナルの Responsive Link	3.21770	133.626
提案手法を実装した Responsive Link	4.55141	231.843
増加率	41.4492 %	73.5015 %

#### 4.2 ノイズ耐性の評価

RTL シミュレーションを用いてノイズ耐性を評価する．シミュレータには Cadence 社 NC-Sim を用いた．本評価で用いるノイズのモデルは，ビットあたりのエラー率とビット長で定義する．ビットあたりのエラー率をビット長で割った確率でノイズが発生し，ビット長で指定した長さ連続してビットが反転する．評価項目のパラメータを表 4 に示す．Responsive Link をループバックで接続し，400Mbaud でパケットの送受信を行う．ノイズモデルを使用して伝送路上でノイズを生成し，パケットのエラー率を観測する．評価ではパケット転送開始直後の 10 パケット (640Byte) は無視し，その後の 1024 パケット (64KByte) を評価対象とする．評価結果は伝送路符号化ごとにまとめ，図 4.2 に NRZI+BS，図 9 に 8B10B，図 10 に 4B10B を使用した場合の結果を示す．

評価結果より，オリジナルのハミング符号と NRZI+BS を使用した場合，ビットエラー率が  $1/10^3\%$  の時点でノイズ長が 8 ビットのものを除いてパケットエラー率 50%を上回っている． $1/10^2\%$  以上のエラーでは全てのノイズ長でまったく通信ができない状態となる．ノイズ長とエラーの関係は，ノイズ長が長いほどビットエラー率が低くなる傾向にある．ノイ

表 4 ノイズ耐性評価のパラメータ

ビットエラーレート	$1/10^1, 1/10^2, 1/10^3, 1/10^4, 1/10^5, 1/10^6$
ノイズのビット長	1bit, 2bit, 4bit, 8bit

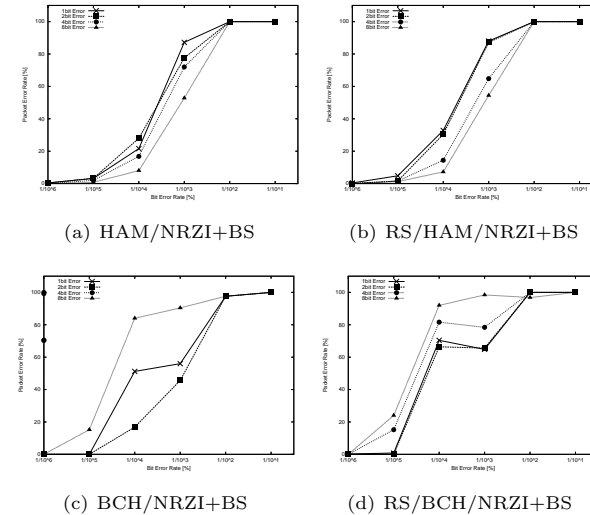


図 8 パケットエラーレート (NRZI+BS)

ズ長が長いほど見かけ上のノイズ発生率が低くなるため，エラー率が低くなると考えられる．これにリードソロモン符号を加えた場合，ノイズ長が長い場合にノイズ耐性が良くなっている．NRZI+BS に BCH 符号を組み合わせた場合，ハミング符号を組み合わせた場合よりも悪い結果となった．これは，ハミング符号と比較して符号化効率が下がったため，相対的にパケットエラーが発生する確立が上がったと推測できる．

8B10B を伝送路符号として使用した場合，概ね組み合わせた符号の強度に従ったパケットエラー率となった．8B10B ではビットエラー率が  $1/10^3\%$  でも全ての場合でパケットエラー率が 50%を下回っている．誤り訂正符号を使用しない場合でも，NRZI+BS と誤り訂正符号を組み合わせたものより良い結果となっている．これは，符号化効率が良いからと考えられる．8B10B 使用する上で最も符号強度の強いリードソロモン符号と BCH 符号を

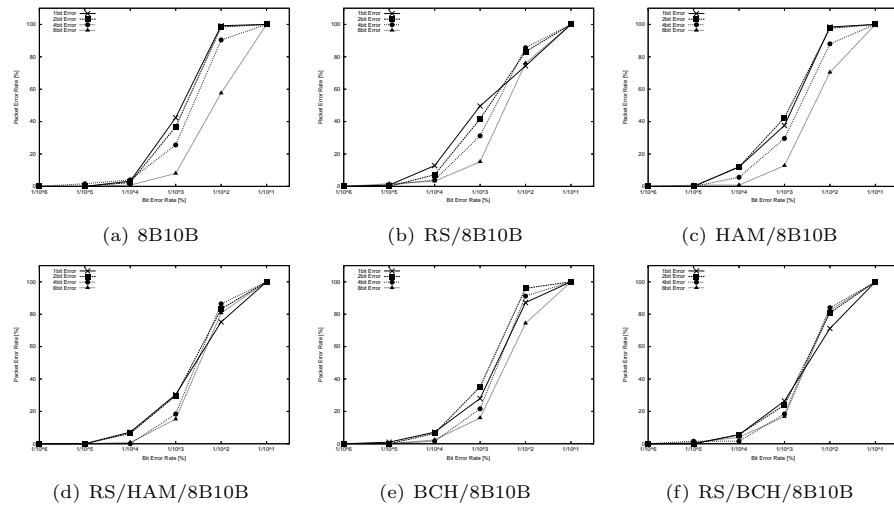


図9 パケットエラーレート(8B10B)

組み合わせたものは、ビットエラー率が  $1/10^3\%$  でもパケットエラー率は全てのビット長で 20%前後、ビットエラー率が  $1/10^2\%$  でもパケットエラー率は 90%程度と非常に高いノイズ耐性を示した。

4B10B を伝送路符号として使用した場合、ビットエラー率が  $1/10^4\%$  以下の場合、パケットエラー率がほぼ全ての組み合わせで 0%に抑えられている。ビットエラー率が  $1/10^3\%$  以上の場合、ノイズ長が伝送路符号の訂正限界である 1 ビットのときは、非常に高いノイズ耐性を示すが、2 ビット以上のノイズが発生する場合はあノイズ耐性が大きく低下する。4B10B 使用する上で最も符号強度の強いリードソロン符号と BCH 符号を組み合わせたものは、ビットエラー率が  $1/10^3\%$  でもパケットエラー率は全てのビット長で約 0%と、8B10B と比較しても非常に高いノイズ耐性を示した。ビットエラー率が  $1/10^2\%$  の場合、ノイズ長が 1 ビットの時はパケットエラー率は約 0%と非常に優秀だが、ノイズ長が 2 ビット以上になると 8B10B と同等か、それ以下の性能となってしまう。

以上の評価結果より、ノイズ耐性に最も影響を及ぼすのは伝送路符号と考えられ、NRZI+BS と比較して、8B10B は同程度の符号化効率で高いノイズ耐性を示すといえる。4B10B はノイズ長が短い伝送路特性では 8B10B よりも高いノイズ耐性を示すが、ノイズ長が長い伝送

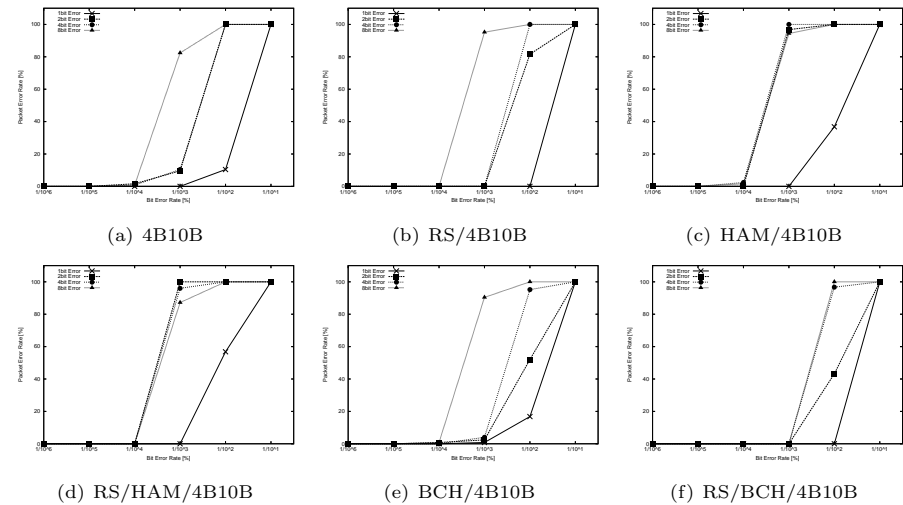


図10 パケットエラーレート(4B10B)

路特性では 8B10B と同等か、それ以下の性能となる。

#### 4.3 実機評価による妥当性の評価

RTL による耐ノイズ性能評価の妥当性を示すために、実機による耐ノイズ性能の評価を行う。評価には従来の Responsive Link が実装された RMTP (Responsive Multithreaded Processor)<sup>10)</sup> を使用する。Responsive Link をループバックで接続し、通信経路上に U\_RD 社製の設備診断用分割型電流注入 CT CTL-28-S330-001FCL を取り付け、電流を注入することでノイズを再現する。評価に使用するプログラムは前節同様のものを使用する。伝送路に注入するノイズには、周波数 10MHz、デューティ比 50%、電圧振幅が 0V~10V の矩形波を用いる。11 に実機評価環境を示す。

オシロスコープを用いて測定した Responsive Link の伝送波形を図 12 に示す。電流を注入してノイズを発生させた場合とそうでない場合の受信波形を比べると、電流を注入した受信波形は送信波形と比べてノイズの影響を受けて受信波形が乱れていることがわかる。転送したパケットのうち 96.6%が伝送成功し、そのうち 0.3%がハミング符号によって正しくエラー訂正が行われていることがわかる。転送失敗した 3.4%のうち、0.1%はエラーを検出したが訂正が出来ず、残りの 3.3%はエラー検出が出来なかった。

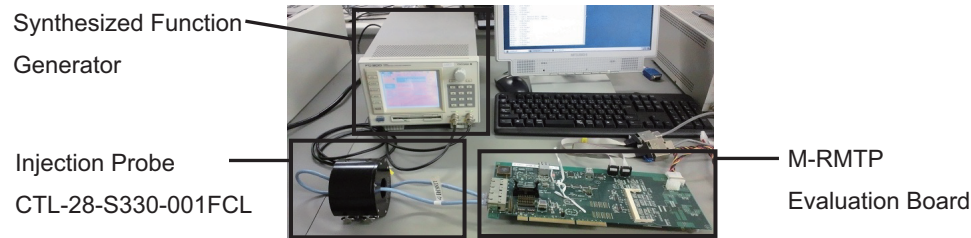
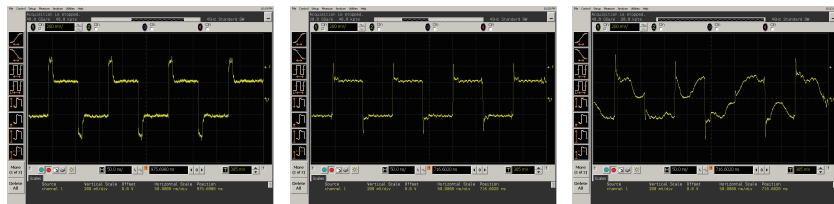


図 11 実機評価環境



(a) 送信波形 (b) 受信波形 (ノイズなし) (c) 受信波形 (ノイズあり)

図 12 Responsive Link の伝送波形

今回の実機評価では、電流注入 CT を用いてノイズが発生するという事実と、既存の Responsive Link を用いて通信を行い、エラーの発生および訂正が行われていることが確認できた。今後は、電流注入 CT を用いて発生させるノイズをモデル化した上でシミュレーション評価との対応を取り、シミュレーション評価の妥当性の検証を行う予定である。また、提案手法を FPGA や ASIC として実装して評価を行い、有用性を示すことを更なる課題としたい。

## 5. まとめ

本論文では、信頼性の高いリアルタイム通信を実現するために、リアルタイム通信規格の Responsive Link をベースにした通信機構を設計し、評価を行った。提案手法では、Responsive Link の誤り訂正符号と伝送路符号を拡張して耐ノイズ性能を向上するとともに、伝送路特性に応じた符号の組み合わせで通信可能である。

評価結果より、オリジナルの Responsive Link と比較して、8B10B を伝送路符号に使用した場合、は同程度の符号化効率で高いノイズ耐性を実現できることが示された。4B10B

を伝送路符号に使用した場合、ノイズ長が短い伝送路特性では 8B10B よりも高いノイズ耐性を示すが、ノイズ長が長い伝送路特性では 8B10B と同等か、それ以下の性能となることが示された。また、提案手法を実装することで従来の Responsive Link と比較して面積が約 42%、ゲート数が 73.5% 増加することがわかった。提案手法を実装した Responsive Link 全体の面積は  $4.55\text{mm}^2$  であり、十分に実装可能であることが示された。

以上の結果より、ディペンダブルな分散リアルタイムシステムを実現するために、提案手法が有用であることがわかった。

謝辞 本研究は科学技術振興機構 CREST の支援によるものであることを記し、謝意を表す。また、本研究の一部は文部科学省グローバル COE プログラム「環境共生・安全システムデザイン」の先導拠点に依るものであることを記し、謝意を表す。

## 参考文献

- 1) Anderson, R.D. and T.Shanley: PCI Express System Architecture (2003). Mind-Share, Inc.
- 2) IBTA: InfiniBand Trade Association (2011). <http://www.infinibandta.org/>.
- 3) I.Mizuuchi, Y.Nakanishi, Y.Sodeyama, Y.Namiki, T.Nishino, and J.Urata, N., K.Hongo, T.Yoshikai and M.Inaba: An advanced musculoskeletal Humanoid Kojiro, *IEEE RTCSA 7* (2005).
- 4) ISO/IEC: *JTC1 SC25 WG4*.
- 5) I.S.Reed and G.Solomon: Polynomial Codes over Certain Finite Fields, *SIAM Journal of Applied Mathematics*, Vol.8, pp.300-304 (1960).
- 6) ITSJ-TS: *0006:2003*.
- 7) R.C.Bose and D.K.Ray-Chaudhuri: On A Class of Error Correcting Binary Group Codes, *Information and Control 3*, Vol.1, pp.68-79 (1960).
- 8) R.W.Hamming: Error Detecting and Error Correcting Codes, *Bell System Technical Journal*, Vol.29, pp.147-160 (1950).
- 9) Widmer, A. and P.A.Franzcek: A DC-Balanced Partitioned-block, 8B/10B Transmission Code, *IBM Journal of research and development*, Vol.27 (1983).
- 10) Yamasaki, N.: Responsive Multithreaded Processor for Distributed Real-Time Systems, *Journal of Robotics and Mechatronics*, pp.130-141 (2005).
- 11) 山崎信行: 分散制御用リアルタイム通信 Responsive Link の設計及び実装, 情報処理学会論文誌コンピューティングシステム, Vol.45, No.SIG 3 (ACS 5), pp.50-63 (2004).
- 12) 山崎信行: 分散管理通信方法及び装置 (2003). 特許平 11-343139.
- 13) 山崎信行: ディペンダブル VLSI システムワークショップ 2009, 科学技術振興機構 (2009). <http://www.dvlsi.jst.go.jp/list/pdf/0912yamasaki.pdf>.