

Responsive Link を用いた分散リアルタイムシステム向け 高信頼な通信機構の設計と実装

水頭 一壽^{1,a)} 向後 卓磨^{1,b)} 松谷 宏紀^{1,c)} 山崎 信行^{1,d)}

受付日 2012年3月5日, 採録日 2012年9月10日

概要: ヒューマノイドロボットに代表される分散リアルタイムシステムにおいて, 誤り訂正符号もしくは伝送路符号を用いた通信の高信頼化は必須の機能である. これらの符号強度とスループット性能の間にはトレードオフがあるため, 伝送路特性および扱うデータに応じて適切な誤り訂正符号および伝送路符号を選ぶ必要がある. 本研究では, 実際のヒューマノイドロボットでの利用を想定し, 3種類の伝送路符号と3種類の誤り訂正符号を装備・切替え可能な通信機構を既存の Responsive Link を拡張するかたちで設計し, 130 nm プロセスを用いて実チップ化した. そのうえで, 設計した通信機構のハードウェア量, および, 各種符号化方式を組み合わせた際のパケットエラー率, ジッタについて解析し, ノイズ耐性に影響を与えるのは誤り訂正符号よりも伝送路符号であることを示した.

キーワード: 分散リアルタイムシステム, リアルタイム通信, Responsive Link

Design and Implementation of Dependable Communication Mechanism on Responsive Link for Distributed Real-time Systems

SUITO KAZUTOSHI^{1,a)} KOGO TAKUMA^{1,b)} MATSUTANI HIROKI^{1,c)} YAMASAKI NOBUYUKI^{1,d)}

Received: March 5, 2012, Accepted: September 10, 2012

Abstract: A dependable communication mechanism based on error correction and line codes is an essential part of distributed real-time systems, such as humanoid robots. Since strong error correction and line codes introduce performance overhead, combination of these codes should be carefully selected, depending on the transmission line and type of communication data. In this paper, a dependable communication mechanism that supports any combinations of three error correction codes and three line codes is designed by extending the Responsive Link for actual humanoid robots, and it has been fabricated as a real chip with a 130 nm process technology. We evaluated the proposed communication mechanism in terms of hardware amount, packet error ratio, and jitter for all combinations of the error correction and line codes implemented. The results show that the line codes significantly affect the packet error ratio.

Keywords: Distributed Real-Time System, Real-Time Communication, Responsive Link

1. はじめに

今日我々の生活の大部分は情報システムによって支えられており, それらのディペンダビリティの向上は社会的な

関心事となっている. たとえば, 自動車, 航空機, 医療用機器, ヒューマノイドロボットに代表される高機能な組み込み機器においては, 高い処理性能に加え, システムの耐故障性はすでに必須のものとなっている. このような状況を背景に, プロセッサを複数用いてシステムを分散制御する分散リアルタイムシステムおよびその高信頼化への期待が高まっている.

分散リアルタイムシステムは, 複数の分散制御用コントローラが, リアルタイム通信規格によって接続されてい

¹ 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University,
Yokohama, Kanagawa 223-8522, Japan

a) suito@ny.ics.keio.ac.jp

b) kogo@ny.ics.keio.ac.jp

c) matutani@ny.ics.keio.ac.jp

d) yamasaki@ny.ics.keio.ac.jp

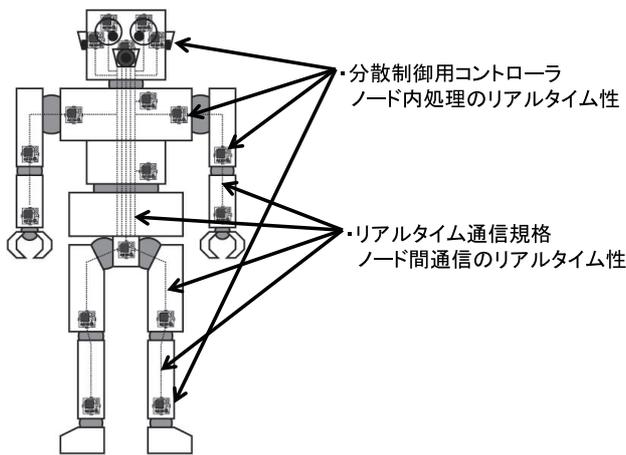


図 1 分散リアルタイムシステムの例

Fig. 1 Example of distributed real-time system.

る (図 1)。そのため、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。通信のリアルタイム性を保障可能な通信規格には、Controller Area Network (CAN) [16], FlexRay [14], Time Triggered Ethernet (TTEthernet) [2], Responsive Link [21] があり、自動車やヒューマノイドロボットなどの制御に応用されている。

このような通信規格を用いて高信頼な分散リアルタイムシステムを実現するうえで、データ伝送路におけるデータの誤りは致命的な障害となる。誤り訂正符号もしくは伝送路符号によってデータ転送の信頼性を保証する必要があるが、符号強度とスループット性能の間にはトレードオフがあり、ノイズ環境およびデータの重要度に応じて適切な誤り訂正符号および伝送路符号を選ぶ必要がある。様々な伝送路符号と誤り訂正符号の組合せの解析、さらにそれらを多数実装して用途に応じて切替え可能とするメリットは大きい。既存の分散リアルタイム通信規格の多くは1つもしくは少数の誤り訂正符号を装備するにとどまっている。

本研究では、実際のヒューマノイドロボットである小次郎 [7] に応用するため、伝送路特性に応じた符号の組合せで通信可能な通信機構を提案する。提案手法に採用する伝送路符号の1つとして、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ビットの誤り訂正と2ビットの誤り検出が可能な 4B10B を新たに提案する。提案した高信頼な通信機構を既存の Responsive Link を拡張するかたちで設計し、130nm プロセスを用いて実チップ化した。そのうえで、送路符号と誤り訂正符号の組合せについてハードウェア量、パケットエラー率、ジッタについて解析した。

本研究がターゲットとするヒューマノイドロボットでは定常時に 80V50A、高負荷時には 80V200A ものモータ駆動電流が使用され、これにともない発生するノイズが通信に影響を与える。このようなきわめて高いノイズ環境において動作するリアルタイム通信規格は、ヒューマノイドロ

ットだけでなく自動車や航空機に代表される輸送機器など幅広い用途に応用できると考えられる。

本論文の構成は以下のとおりである。2章で分散リアルタイムシステムで利用可能な通信規格について紹介し、3章で本通信機構のベースラインとなった Responsive Link の特徴について述べる。4章および5章で 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と実装について述べ、6章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7章で本論文をまとめる。

2. 分散リアルタイムシステム向け通信規格

本章では、ヒューマノイドロボットや自動車などで利用されているリアルタイム通信規格について調査する。

Controller Area Network (CAN) [16] はマルチマスタ方式のシリアルバスで結合され、Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) 方式を用いて通信の調停を行う。また、ID を用いた優先度アービトレーションによって、リアルタイム通信を実現しようとする。CAN は ISO で標準化されており [9]、自動車や航空機、FA (Factory Automation) など様々な分野で利用されている。

FlexRay [14] は、CAN と同様に、車載通信向けのネットワークとして FlexRay Consortium [4] によって開発されている。FlexRay は Time Division Multiple Access (TDMA) を用いたタイム・トリガ型の通信によってリアルタイム性を保証する。時間同期については全ノードが単一のクロックを参照することで実現している。

Time Triggered Ethernet (TTEthernet) [2] は Ethernet 上でリアルタイム通信を実現する通信規格である。TTEthernet も FlexRay 同様 TDMA を用いたタイム・トリガ型の通信によってリアルタイム性を保証する。TTEthernet では IEEE 1588 という時間同期プロトコルを使用し [1]、時間同期を実現している。

これらリアルタイム通信規格における信頼性を向上のために、すでにいくつかの検討が行われている。Emani ら [5] は、CAN 通信における性能と信頼性を向上させるために、リードソロモン符号と Cyclic Redundancy Check (CRC) を組み合わせた通信手法を提案している。また Maxino ら [12] は、組込み向けの制御ネットワークにおいて、チェックサムによるエラー検出の有効性と、そのトレードオフについて述べている。

しかし、これらの研究では単一もしくは少数の誤り訂正符号にのみ着目しており、ビットエラーやバーストエラーに適した誤り訂正符号および 8b10b に代表される伝送路符号との組合せについて十分に検討していない。本研究では、実際のヒューマノイドロボットでの利用を想定し、3種類の伝送路符号と3種類の誤り訂正符号を装備し、用途に応じて組合せを変更できる通信機構を Responsive Link

を拡張するかたちで設計・実装する。

3. 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、国内では IPSJ-TS 0006:2003 [8], 国際的には ISO/IEC 24740 として標準化されている [10]. 本章では Responsive Link の特徴について述べる。

3.1 Responsive Link の特徴

リアルタイム通信では、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、データ量が多くバンド幅を必要とするが、通信遅延に対しての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。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を分離し、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 16 バイト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 64 バイトとなっている。

Responsive Link ではパケットには優先度が付加され、優先度によるパケットの追い越しを行う。複数のパケットの出力ポートが競合した場合、優先度の高いパケットが優先的に送信される。優先度によるパケットの追い越しによって、パケットの到着時間は Worst-Case Execution Time (WCET) 解析によって予測可能となり、通信のリアルタイム性を保証することができる。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、リアルタイムタスクのスケジューリング理論を通信に応用することができる。

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、退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 (図 2)。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。

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、同じ宛先のパケット

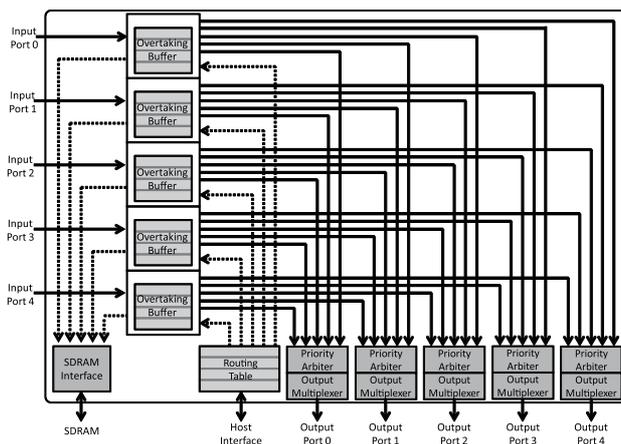


図 2 Responsive Link のネットワークスイッチ
Fig. 2 Network switch of Responsive Link.

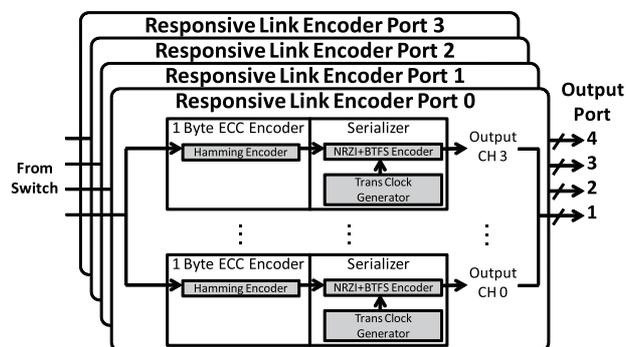


図 3 エンコーダのブロック図
Fig. 3 Block diagram of encoder.

内でも優先度によって個別に経路の設定ができる。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、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、迂回路を設けたりすることが可能になる。

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、ノードごと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ともに、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。

3.2 Responsive Link の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 [17] を採用している。ハミング符号生成多項式に $x^4 + x + 1$ 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 ビットの誤り訂正を実現している。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

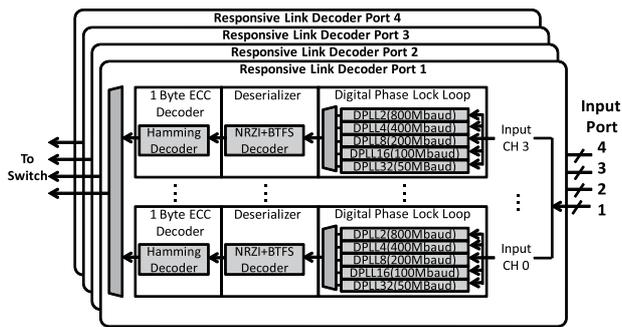


図 4 デコーダのブロック図
Fig. 4 Block diagram of decoder.

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DPLLの分解能に応じて50, 100, 200, 400, 800 [Mbaud] が使用可能であり、伝送路特性に応じてこれらの中から選択可能である。図3にエンコーダのブロック図、図4にデコーダのブロック図を示す。Responsive Linkは4ポートのシリアル通信であるが、1つのポートに4つのチャンネルを束ねたパラレル通信によるマルチリンクが可能である。

4. 設計

本論文で提案する通信機構は、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を可能にするように設計する。ヒューマノイドロボットの体内のようなノイズの多い環境下で高信頼な通信を実現するためには、各種ノイズに対する耐性が重要となる。そこで、Responsive Linkのノイズ耐性を向上させるため、誤り訂正符号と伝送路符号を拡張する。

現在、ターゲットアプリケーションであるヒューマノイドロボット小次郎の通信系には、リードソロモン符号[11]とシンボル単位のパリティを使用した通信が使用されている。8バイトのデータに4バイトの冗長符号を付加し、2バイトまでのエラー訂正をするとともに、8ビットごとに2ビットのパリティを付加し、1ビットのエラー検出が可能である。そこで、提案手法では現在使用されている通信規格と同程度以上の誤り訂正能力が必要と考えられる。また、実機環境では、様々な周波数成分のノイズが発生する。高周波成分のノイズはランダムエラーとして検出され、低周波成分のノイズはバーストエラーとして検出される。また、時間方向へのノイズはジッタとして検出される。このような各種ノイズに対する耐性を向上させるために、複数のレベルでの誤り訂正、およびジッタ対策を行う。以上のような要求仕様を考慮し、提案手法に採用する誤り訂正符号と伝送路符号を拡張する。

誤り訂正符号では、ビット単位とブロック単位の2つのレベルで誤り訂正を行う。これは、ランダム誤りとバースト誤りの両方に柔軟に対応するためである。ビット単位の誤り訂正符号には、従来のハミング符号に加え、新たにBCH符号[15]を追加する。また、ブロック単位の誤り訂正符号には、リードソロモン符号を追加する。

伝送路符号では、従来のNRZI+BS符号に加えて、8B10B符号化[20]を追加する。さらに、伝送路符号レベルでの誤り訂正を実現する4B10B符号化を新たに提案し、追加する。

ノイズ耐性と符号長はトレードオフの関係にあるため、必要以上のノイズ耐性は通信のスループットを低下させてしまう。そこで、提案手法では複数の符号化方式を用意し、伝送路特性に応じて最適な符号化の組合せによって通信可能な機構を設計する。

4.1 誤り訂正符号

4.1.1 BCH符号

ノイズ耐性を向上させるために、誤り訂正符号にBCH符号を追加する。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のため、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。

設計するBCH符号化回路では、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する。BCH符号により、2ビットまでのランダム誤りと3ビットまでのバースト誤りの訂正、そして全ビットが1になった場合の誤り検出を実現する。

4.1.2 リードソロモン符号

ハミング符号やBCH符号では訂正不可能なバースト誤りへ対応するため、リードソロモン符号を追加する。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使用されている。

設計するリードソロモン符号化回路では、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はバーチャルカットスルー転送でのパケット追い越しを行うため、ヘッダのみを即座に復号する必要がある。そのため、リードソロモン符号によって符号化するデータのサイズは、パケットヘッダと同じ4バイトとしている。

4.2 伝送路符号

4.2.1 8B10B符号化

ノイズ耐性を向上させるため、伝送路符号に8B10Bを追加する。オリジナルのResponsive Linkに採用されている

表 1 4B10B 変換テーブル
Table 1 4B10B conversion table.

4B	10B	4B	10B
0000	1100101100	1000	1001110001
0001	1011001100	1001	0111000110
0010	1100110010	1010	1010110100
0011	0110011100	1011	1101001010
0100	0111010001	1100	1011010010
0101	1100011001	1101	1001100110
0110	0101110100	1110	1010101001
0111	1101000101	1111	0110101010

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。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。ビット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、フレーム同期が崩れるという問題がある。そこで、固定長の符号化である 8B10B を採用する。また、8B10B 符号化では 2 ビットの冗長ビットを追加することで、20%以下の低周波成分が排除される。そのため、シンボル間干渉ジッタを低減させることができる。

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 [3] や InfiniBand [6] などの高速シリアル通信で使用されている。

4.2.2 4B10B 符号化

伝送符号レベルでノイズ耐性を向上させるために、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 ビットの誤り訂正と 2 ビットの誤り検出が可能な 4B10B を新たに提案する。NRZI や 8B10B などの送路符号では、伝送路上のデータが 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。そこで、伝送路符号自体にエラー訂正機能を組み込むことで、より高いノイズ耐性を達成することができる。

提案する 4B10B はエンベデッドクロック方式と DC バランシング、誤り訂正符号の 3 つの機能を実現する。既存の符号化でエンベデッドクロックと同時に誤り検出・訂正が実現可能なものは存在しなかった。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。

4B10B は、符号化された 10 ビット内において、連続する 0 または 1 の数を 3 ビット以内、1 ビットのエラーが発生した場合にも 5 ビット以内であることを保障し、エンベデッドクロックを実現する。また、符号化された 10 ビット内の 0 と 1 の数を同じにすることで、DC バランシングを実現する。符号化された 10 ビットは、互いにハミング

表 2 符号化方式の組合せ

Table 2 Combination of encoding method.

誤り訂正符号 4 バイト単位	誤り訂正符号 1 バイト単位	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 RS	No ECC	8B10B (10, 8)	53.3%
		4B10B (10, 4)	26.7%
		BCH (16, 8)	NRZI+BS (9, 8)
	HAM (12, 8)	8B10B (10, 8)	40.0%
		4B10B (10, 4)	20.0%
		No ECC	NRZI+BS (9, 8)
No ECC	8B10B (10, 8)	53.3%	
	4B10B (10, 4)	26.7%	
	8B10B (10, 8)	80.0%	
		4B10B (10, 4)	40.0%

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、ハミング距離に基づく最小距離復号を行う。1 ビットのビットエラーは最小距離復号によって正しい符号語が一意に決まるため、エラー訂正が可能である。2 ビットのビットエラーは正しい符号語が一意に決まらないため、エラー訂正はできないが、検出は可能である。3 ビット以上のエラーは訂正も検出も不可能である。

4.3 符号化方式の組合せと符号化率

提案した通信機構では、従来の Responsive Link で使用可能なハミング符号と NRZI+BS に加え、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。表 2 に選択可能な符号化の組合せと、符号化率を示す。誤り訂正符号と伝送路符号は任意の組合せで使用可能である。提案した通信機構は、エラー訂正能力と符号化効率のトレードオフに基づき、伝送路特性に応じて最適な符号を選択することができる。

4.4 エンコーダ・デコーダの設計

従来の Responsive Link をベースに、BCH 符号、リードソロモン符号、8B10B、4B10B を追加したエンコーダ、デコーダを設計する。本論文で設計する通信機構は、提案した誤り訂正符号、および伝送路符号をすべて実装し、任意の組合せでの通信を実現する。これによってハードウェアコストは増大するが、様々な伝送路符号と誤り訂正符号の組合せの解析、さらにそれらを多数実装して用途に応じて切替え可能とするメリットは大きいと考えられる。拡張した Resopnsive Link のエンコーダ、およびデコーダのプ

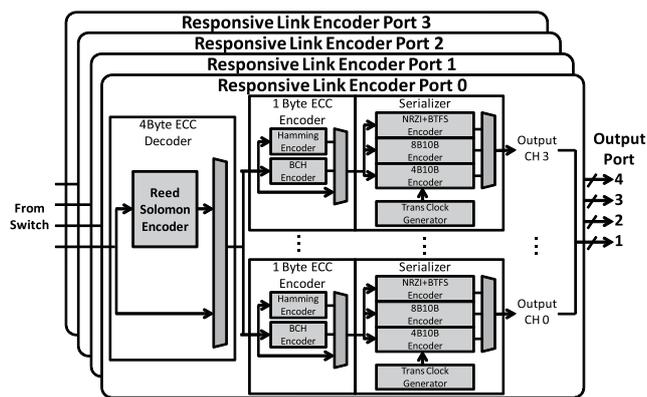


図 5 拡張したエンコーダのブロック図
Fig. 5 Block diagram of expanded encoder.

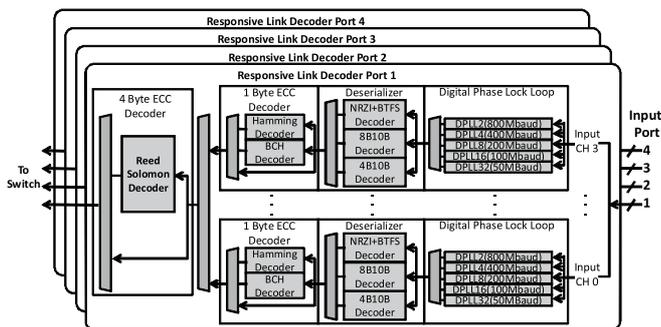


図 6 拡張したデコーダのブロック図
Fig. 6 Block diagram of expanded decoder.

ロック図をそれぞれ図 5 と図 6 に示す。

エンコーダ、デコーダともに 4 バイト単位の誤り訂正符号であるリードソロモンの符号器は 1 ポートに 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 1 ポートに 4 つずつ実装されている。

エンコーダはスイッチ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。リードソロモン符号が有効な場合、リードソロモン符号のエンコーダにデータが入力される。リードソロモン符号が無効な場合は、入力データはバイパスされる。リードソロモン符号のエンコーダは 4 段の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ザに入力され、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する。

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ザに入力し、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。次にリードソロモン符号が有効な場合は、リードソロモン符号のデコーダへ送信される。リードソロモ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。リードソロモ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。

5. Dependable Responsive Multithreaded Processor

提案した通信機構を分散リアルタイムシステム向け SoC (System-on-a-Chip) である D-RMTP (Dependable Responsive Multithreaded Processor) I [18] に実装し 2012 年 1 月にテープアウトした。D-RMTP I は分散リアルタイムシステムを実現するため、以下に示すデバイスを集積している。

- リアルタイム処理用プロセッサ RMT Processing Unit (RMTPU)
 - 8 way 優先度付き SMT [19]
 - IPC (Instruction Per Clock) 制御機構 [13]
 - 32 エントリのコンテキストキャッシュ
 - トレース機構
- リアルタイム通信規格 Responsive Link
- コンピュータ用周辺機器
 - PCI-X, IEEE1394, DMA Controller, DDR SDRAM Interface, など
- ロボット制御用 I/O
 - PWM (Pulse Width Modulation), Pulse Counter, RTC (Real Time Clock), など

図 7 に D-RMTP I のレイアウト、図 8 にチップ写真を示す。中央から右下にかけての部分に、リアルタイム処理用プロセッサ RMTPU が配置されている。右上の長方形の部分に Responsive Link であり、この部分に提案した通信機構が実装されている。左下の部分はロボット制御用 I/O や SRAM が配置されており、これらは電池などの補助電源によってバックアップされている。そのため、電源遮断時にもデータやロボット制御用 I/O の状態を保持しておくことができる。プロセッサ用のトレースバッファもバッテリーバックアップ領域に配置されている。左上の部分には IEEE1394 や Ethernet, DMA Controller など、コンピュータ制御用の I/O が配置されている。

D-RMTP I の諸元を表 3 に示す。

実装した通信機構は、オリジナルの Responsive Link の規格の上位互換である。最大変調速度は 800 [Mbaud] で、パケット追い越しバッファをイベントリンク、データリン

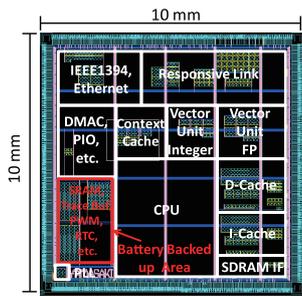


図 7 D-RMTP I のレイアウト
Fig. 7 Layout of D-RMTP I.

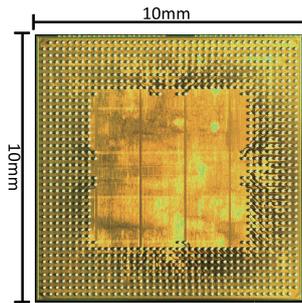


図 8 D-RMTP I のチップ写真
Fig. 8 Chip micrograph of D-RMTP I.

表 3 D-RMTP I の諸元

Table 3 Specification of D-RMTP I.

プロセスルール	: TSMC 130 nm CMOS 8 層銅配線
ダイサイズ	: 10.0 [mm] × 10.0 [mm] = 100 [mm ²]
電圧	: Core 1.0 [V], I/O 3.3 [V], DRAM 2.5 [V]

クそれぞれに 8 パケット分実装している。パケット退避用の外部記憶装置には、最大 256 [MB] の DDR SDRAM をサポートしている。ルーティングテーブルのエントリ数は 256 エントリとしている。

6. 評価

6.1 ハードウェアコスト

提案手法の実装にともな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 130 nm のプロセスをターゲットに論理合成を行い、面積を比較する。Responsive Link の規格では最大ボーレートが 800 [Mbaud] と定義されており、Serializer と Deserializer は 800 [MHz] での動作が要求される。そのため、Serializer と Deserializer には 800 [MHz] のタイミング制約を与える。それ以外の部分は Serializer と Deserializer の 1/4 以下のクロックで動作するので、200 [MHz] のタイミング制約を与える。

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を表 4 に示す。1 バイト単位の誤り訂正符号において、BCH 符号

表 4 符号化モジュールのハードウェアコスト

Table 4 Hardware cost of codec.

符号化モジュール		面積 [μm^2]
NRZI/BTFS	Encoder	6,711.6
	Decoder	4,841.0
4B10B	Encoder	7,791.1
	Decoder	5,226.3
8B10B	Encoder	9,699.0
	Decoder	4,998.9
Hamming	Encoder	215.6
	Decoder	1,240.8
BCH	Encoder	329.3
	Decoder	2,417.1
Reed Solomon	Encoder	22,375.2
	Decoder	38,654.9

表 5 符号化モジュールの面積比率

Table 5 Area ratio of codec.

符号化モジュール	実装数 [個]	面積 [μm^2]	面積比率 [%]
NRZI/BTFS	4	46,210.0	19.6
4B10B	4	52,069.4	22.2
8B10B	4	58,791.2	25.0
Hamming	4	5,825.5	2.5
BCH	4	10,985.6	4.7
Reed Solomon	1	61,030.0	26.0

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% 増加しており、デコーダでは面積が 94.8% 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% 増加しており、デコーダでは面積が 3.2% 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% 増加しており、デコーダでは面積が 8.0% 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、デコーダでは面積が約 300 倍と大きくなっている。

1 バイト単位の誤り訂正符号と伝送路符号は、エンコーダとデコーダの各チャンネルにそれぞれ 4 個ずつ実装されている。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はエンコーダとデコーダの各チャンネルに 1 個ずつ実装されている。符号化モジュールの面積比率を表 5 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最大の割合を占めてはいるが、伝送符号の 8B10B、4B10B、NRZI+BS との差は面積で約 0.95%~7% 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% を占めている。この結果より、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分かる。

表 6 Responsive Link 全体のハードウェアコスト
Table 6 Total hardware cost of Responsive Link.

	面積 [mm ²]
オリジナルの Responsive Link	3.21770
提案手法を実装した Responsive Link	4.55141
面積増加率	41.4492%

表 7 ノイズ耐性評価のパラメータ
Table 7 Parameter of noise immunity test.

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	1 bit, 2 bit, 4 bit, 8 bit

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 6 に示す。提案手法を実装することで、従来と比較して 41.4%面積が増加することが分かった。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。

6.2 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。本評価では、Responsive Link をループバックで接続し、パケットの送受信を行う。その際、ノイズを発生させるモジュールを伝送路上に接続し、パケット転送のエラー率を観測する。シミュレータには Cadence 社 NC-Sim を用いた。

本論文で想定するロボット環境においては、通常のビット誤りに加えて、大出力モータドライバの駆動にともなうノイズ、および、供給電圧の変動にともなうバースト誤りに対処する必要がある。したがって、本論文では、ビット単位のビット誤り、および、バイト単位のバースト誤りを想定した評価を行う。本評価で用いるノイズモデルのパラメータとして、ビットエラー率とビット長を定義する。伝送したビット数に対して、反転したビット数の割合をビットエラー率と定義し、伝送路の品質を表すパラメータとする。ノイズ発生時に連続して反転するビットの数をビット長と定義し、ノイズの周波数特性を表すパラメータとする。反転したビットの総数はノイズ発生回数×ビット長となるため、ノイズ発生モジュールはビットエラー率をビット長で割った確率でノイズを生成し、ビット長で指定したビット数だけ連続してビットを反転させる。

評価ではパケット転送開始直後の 10 パケット (640 バイト) をウォームアップサイクルとして結果から除外し、その後の 1,024 パケット (64 キロバイト) を評価対象とした。

評価項目のパラメータを表 7 に示す。ビットエラーレートはすべての符号化の組合せでパケットエラーレートが 0% となった 1/10⁶ から、パケットエラーレートが 100% と

なった 1/10¹ を評価の範囲とした。また、ノイズのビット長は、エラー訂正の最小単位がバイト単位のため、1 ビットから 8 ビットの範囲とした。

伝送路符号に NRZI+BS を用いた際のパケットエラーレートを図 9 に示す。

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 1/10³% 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。1/10²% 以上のエラーではすべ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見かけ上のエラー発生回数が低くなる。このため、エラー率が低くなると考えられる。NRZI+BS を伝送路符号として用いる際にリードソロモン符号を加えた場合、ノイズ長が長い場面においてノイズ耐性が改善している。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率が上がったと推測できる。

伝送路符号に 8B10B を用いた際のパケットエラーレートを図 10 に示す。8B10B を伝送路符号として使用した場合、おおむ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 1/10³% でもすべ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 使用するうえで最も符号強度の強いリードソロモン符号と BCH 符号を組み合わせたものは、ビットエラー率が 1/10³% でもパケットエラー率はすべてのビット長で 20% 前後、ビットエラー率が 1/10²% 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。

伝送路符号に 4B10B を用いた際のパケットエラーレートを図 11 に示す。4B10B を伝送路符号として使用した場合、ビットエラー率が 1/10⁴% 以下の場合、パケットエラー率がほぼすべての組合せで 0% に抑えられている。ビットエラー率が 1/10³% 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 使用するうえで最も符号強度の強いリードソロモン符号と BCH 符号を組み合わせたものは、ビットエラー率が 1/10³% でもパケットエラー率はすべてのビット長で約 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 1/10²% の場合、ノイズ長が 1 ビットのとき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 と同程度か、それ以下の性能になってしまう。

以上の評価結果より、ノイズ耐性に最も影響を及ぼすの

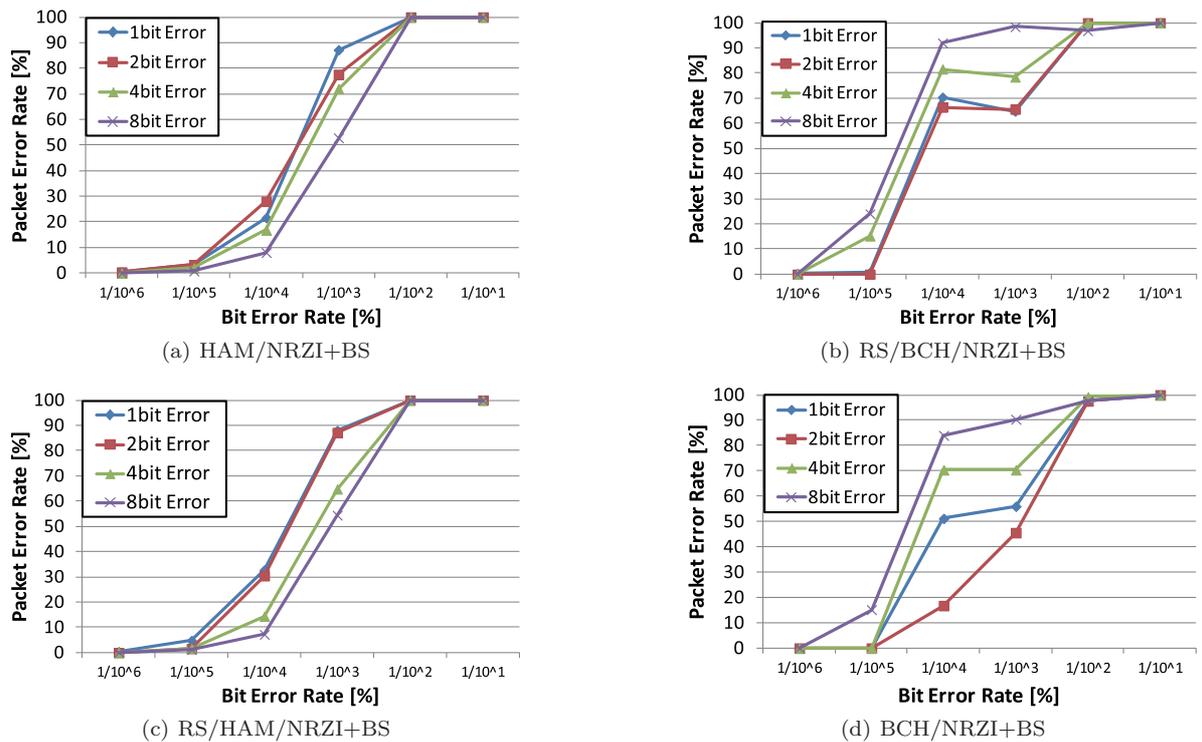


図 9 パケットエラーレート (NRZI+BS)
Fig. 9 Packet error rate (NRZI+BS).

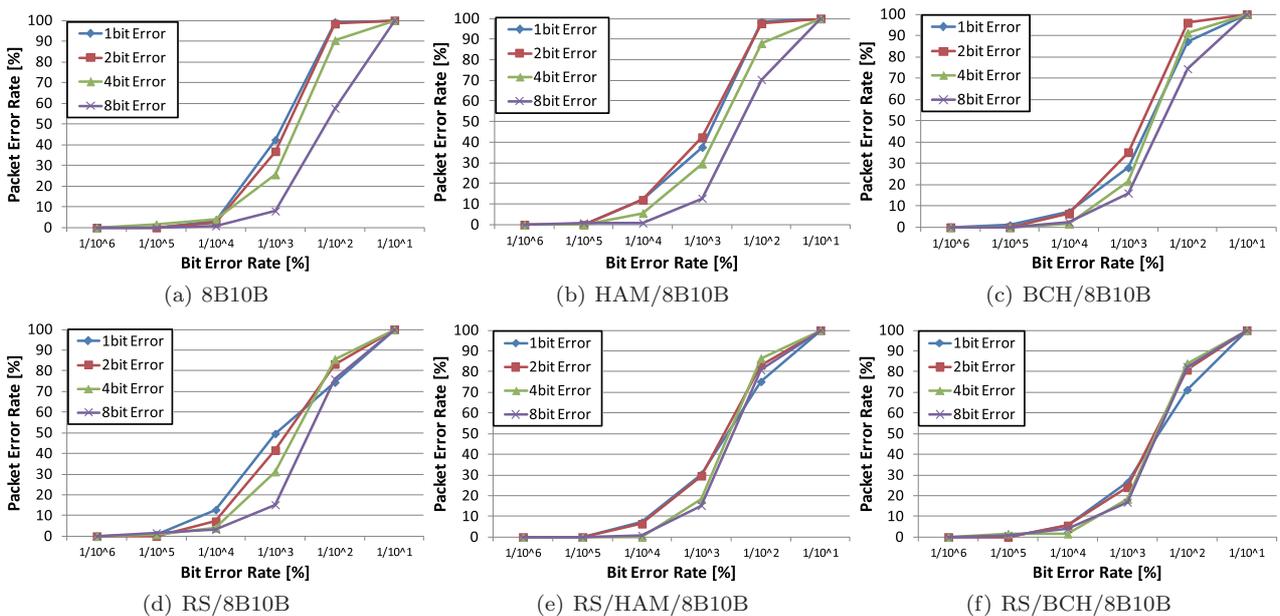


図 10 パケットエラーレート (8B10B)
Fig. 10 Packet error rate (8B10B).

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路特性では 8B10B と同等か、それ以下の性能となる。

6.3 ジッタ耐性の評価

RTL シミュレーションを用いてクロックデータリカバ

リ回路のジッタ耐性を評価する。本評価で用いるジッタのモデルは、送信ノードから受信ノードへ到着するまでの時間変動を表す TIE (Time Interval Error) とする。図 12 に送受信波形と TIE の関係を示す。図 12 の $T_1 \sim T_3$ の変動が TIE として観測される。

本評価におけるジッタの生成には、送信ノードから受信ノードに到着するまでの時間を振幅、到着時間の時間方向への変動を周波数とした正弦関数を用いた。振幅

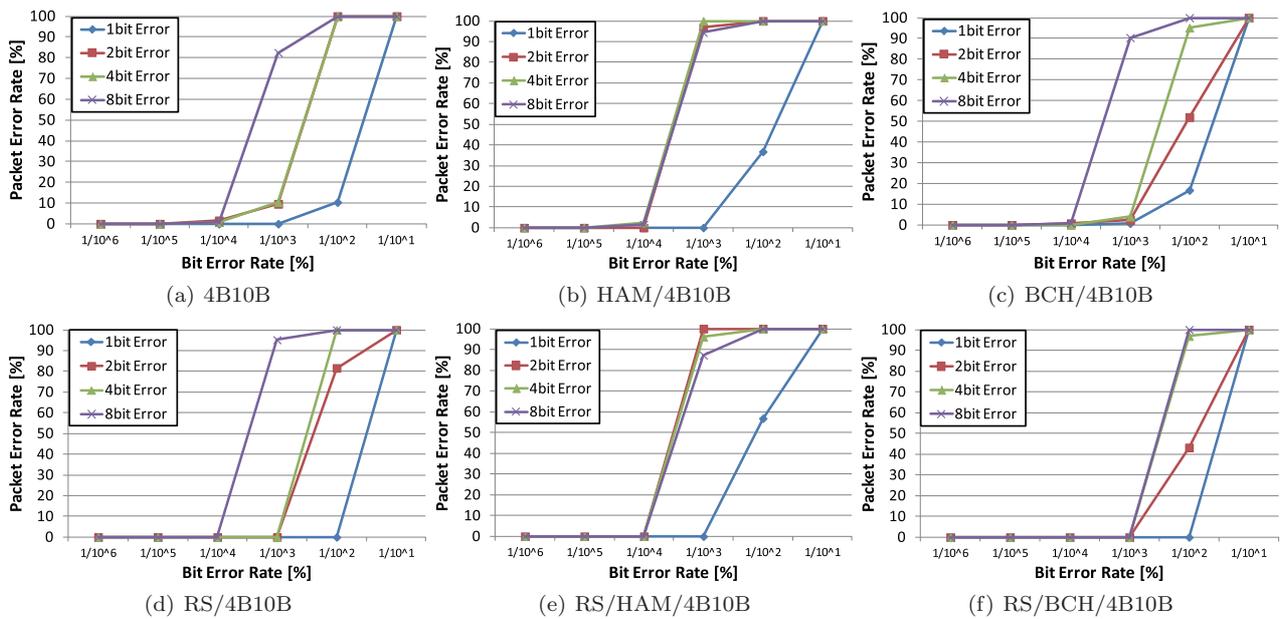


図 11 パケットエラーレート (4B10B)
 Fig. 11 Packet error rate (4B10B).

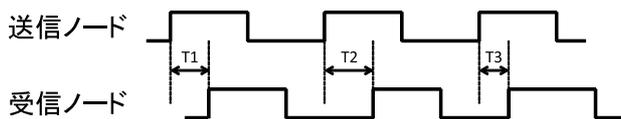


図 12 Time interval error.
 Fig. 12 Time interval error.

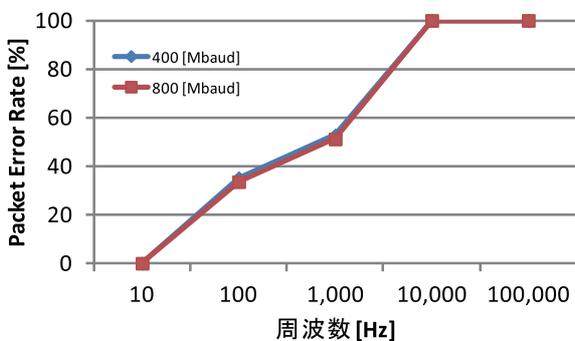


図 13 ジッタ耐性の評価結果
 Fig. 13 Evaluation result of jitter tolerance test.

は 0 [sec]~1/boudrate [sec] の範囲で、周波数は 10 [Hz]~100,000 [Hz] の範囲で変動するものとした。評価では Responsive Link をループバックで接続し、その間にジッタモデルを挿入し、パケットの送受信を行う。伝送レートは 400 [Mbaud] と 800 [Mbaud] を使用し、比較する。ジッタモデルを使用して伝送路上でジッタを生成し、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 (640 バイト) は無視し、その後の 1,024 パケット (64 キロバイト) を評価対象とする。使用する符号化の組合せは、ジッタによる影響を明確にするために誤り訂正符号は使用せず、伝送路符号に 8B10B を使用した。

図 13 に評価結果を示す。評価結果より、いずれの伝送

レートにおいても、変動周期が 10 Hz ではジッタの影響はほとんどなかった。100 Hz では約 35%、1,000 Hz では約 50% のパケットがエラーとなった。10,000 Hz を超えるとほぼすべてのパケットがエラーとなり、通信ができない状態となった。

実システムにおけるジッタの発生要因は、伝送路でのクロストーク遅延があげられる。また、LSI 自体の動作クロックを生成する水晶発振子、または PLL のジッタも要因の一つである。クロックデータリカバリ回路のジッタ耐性を超えるジッタが発生した場合、誤ったタイミングでデータがラッチされ、結果として受信データのビットエラーとして検出される。そのため、ジッタ耐性が低い周波数帯においては、誤り訂正符号を使用して通信する必要がある。

7. おわりに

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させるとともに、伝送路特性に応じた符号の組合せで通信可能である。提案手法に採用する伝送路符号の一つとして、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 ビットの誤り訂正と 2 ビットの誤り検出が可能な 4B10B を新たに提案した。

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、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。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%増加することが分かった。提案手法を実装した Responsive Link 全体の面積は 4.55 mm² であり、十分に実装可能であることが示された。提案手法によって拡張した Responsive Link を分散リアルタイムシステム向け SoC である D-RMTP I に搭載し、LSI として実装した。これらの結果より、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分かった。本研究では実装対象に Responsive Link を選択したが、ここで得られた知見は他の通信規格にも適用可能である。

今後、様々な伝送路特性に対応する最適な符号化の組合せを明らかにするとともに、伝送路符号に応じて動的に符号化を選択・変更し、ディペンダブルなリアルタイム通信を実現するためのソフトウェア機構を開発する予定である。また、D-RMTP I の実機を用いた評価を行う予定である。

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」によるものであることを記し、謝意を表す。

参考文献

[1] Ademaj, A. and Kopetz, H.: Time-Triggered Ethernet and IEEE 1588 Clock Synchronization, *International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (TU Wien)*, pp.41–43 (2010).

[2] TTTech Computertechnik AG: TTEthernet – A Powerful Network Solution for All Purposes (2009). available from http://www.tttech.com/fileadmin/content/white/TTEthernet/TTEthernet_Article.pdf.

[3] Anderson, R.D. and Shanley, T.: *PCI Express System Architecture*, MindShare, Inc. (2003).

[4] FlexRay Consortium, available from <http://www.flexray.com/>.

[5] Emani, K., Kam, K. and Zawodniok, M.: Improvement of CAN BUS Performance by Using Error-Correction Codes, *Region 5 Technical Conference, 2007 IEEE*, pp.205–210 (2007).

[6] IBTA: InfiniBand Trade Association (2011). available from <http://www.infinibandta.org/>.

[7] Mizuuchi, I., Nakanishi, Y., Sodeyama, Y., Namiki, Y., Nishino, T., Urata, N.J., Hongo, K., Yoshikai, T. and Inaba, M.: An advanced musculoskeletal Humanoid Kojiro, *IEEE RTCSA 7* (2005).

[8] IPSJ-TS: 0006:2003, available from <http://www.itscj.ipsj.or.jp/ipsj-ts/02-06/toc.htm>.

[9] ISO: 11898:1992, Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication (1992).

[10] ISO/IEC: 24840:2008, available from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50352.

[11] Reed, I.S. and Solomon, G.: Polynomial Codes over Certain Finite Fields, *SIAM Journal of Applied Mathematics*, Vol.8, pp.300–304 (1960).

[12] Maxino, T. and Koopman, P.: The Effectiveness of Checksums for Embedded Control Networks, *IEEE Trans. Dependable and Secure Computing*, Vol.6, pp.59–72 (2009).

[13] Yamasaki, N., Magaki, I. and Itou, T.: Prioritized SMT Architecture with IPC Control Method for Real-Time Processing, *The 13th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp.12–21 (2007).

[14] Nolte, T., Hansson, H. and Bello, L.L.: Automotive Communications – Past, Current and Future, *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '05)*, Vol.1, pp.985–992 (2005).

[15] Bose, R.C. and Ray-Chaudhuri, D.K.: On A Class of Error Correcting Binary Group Codes, *Information and Control 3*, Vol.1, pp.68–79 (1960).

[16] BOSCH: GmbH, Postfach 300240: CAN Specification Version 2.0 (1991).

[17] Hamming, R.W.: Error Detecting and Error Correcting Codes, *Bell System Technical Journal*, Vol.29, pp.147–160 (1950).

[18] Suito, K., Fujii, K., Matutani, H. and Yamasaki, N.: Dependable Responsive Multithreaded Processor for Distributed Real-Time Systems, *Proc. International Symposium on Low-Power and High-Speed Chips* (2012).

[19] Tullsen, D.M., Eggers, S.J. and Levy, H.M.: Simultaneous Multithreading: Maximizing On-Chip Parallelism, *Proc. 22nd Annual International Symposium on Computer Architecture*, pp.392–403 (1995).

[20] Widmer, A. and Franszek, P.A.: A DC-Balanced Partitioned-block, 8B/10B Transmission Code, *IBM Journal of Research and Development*, Vol.27 (1983).

[21] Yamasaki, N.: Responsive Link for Distributed Real-Time Processing, *International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA)*, pp.20–29 (2007).



水頭 一壽 (学生会員)

2008 年慶應義塾大学理工学部情報工学科卒業。2010 年同大学大学院理工学研究科開放環境科学専攻修士課程修了。現在、同大学院後期博士課程に在籍。リアルタイム通信の研究に従事。



向後 卓磨

2009 年慶應義塾大学理工学部情報工学科卒業。2011 年同大学大学院理工学研究科開放環境科学専攻修士課程修了。現在、日本電気株式会社スマートエネルギー研究所勤務。マイクロログリッド等の研究に従事。



松谷 宏紀 (正会員)

2004年慶應義塾大学環境情報学部卒業。2008年同大学大学院理工学研究科開放環境科学専攻博士課程修了。博士(工学)。現在、慶應義塾大学理工学部情報工学科専任講師。2009年度より2010年度まで日本学術振興会特別研究員SPD。計算機アーキテクチャ、オンチップネットワークの研究に従事。



山崎 信行 (正会員)

1991年慶應義塾大学理工学部物理学科卒業。1996年同大学大学院理工学研究科計算機科学専攻博士課程修了。博士(工学)。同年電子技術総合研究所入所。1998年10月慶應義塾大学理工学部情報工学科助手。同専任講師を経て、2004年4月より同助教授(現、准教授)。リアルタイムシステム、プロセッサアーキテクチャ、並列分散処理、システムLSI、ロボティクス等の研究に従事。日本ロボット学会、電子情報通信学会、IEEE各会員。