

伝送フレームに着目したデジタル通信システム設計手法

白川千洋 樋口一茂 宮崎敏明 林一博 山田一久
shira@exa.onlab.ntt.jp

NTT 光ネットワークシステム研究所 伝送処理研究部
〒 238-03 横須賀市武 1-2356, Y-807C

高度化するマルチメディアサービスの要求にタイムリーに応えるためには、その基盤となる伝送装置の設計コスト削減が大きな課題である。本報告では、伝送システムの効率的な設計開発を行うためFORM (Frame-Oriented Representation Method) と呼ぶ新しい設計手法を提案する。FORM は機能/ハードウェアモデルの二つ異なるレベルのモデルを持ち、機能設計とタイミング設計を独立して行うことができる。FORMを用いることで、機能レベルのシステム仕様をRTLに変換することができる。本手法をSDH/ATMインタフェース回路に適用した結果、設計複雑度を軽減しシステム設計を単純化することができた。

A Frame Base Design Method for Digital Telecommunication System

K. Shirakawa, K. Higuchi, T. Miyazaki, K. Hayashi and K. Yamada
shira@exa.onlab.ntt.jp

NTT Optical Network Systems Laboratories
Y-807C, 1-2356 Take, Yokosuka-shi, Kanagawa, 238-03 JAPAN

This paper proposes a new design method called FORM (Frame-Oriented Representation Method) for digital telecommunication systems with the aiming of efficient system design and development. FORM has the unique feature wherein timing design and function design are performed independently. FORM can translate system-level specifications at the behavioral level into RTL. This method is applied to the SDH/ATM interface and it is proved that FORM relaxes design complications and simplifies system design.

1. はじめに

近年の Broadband Integrated Services Digital Network (B-ISDN)におけるマルチメディアサービスの多様化により、多機能性、柔軟性、高速性がネットワークに要求されている。そのニーズにタイムリーに答えるためには、複雑化するデジタル通信システム的设计コスト削減が不可欠である。

通信システムにおいては、一般的に端末側における情報ソースに対するデジタル信号処理とそれらを正しく伝送する伝送信号処理の二種類のデジタルデータ処理が存在する。ここでは後者の機能を「伝送処理」と呼ぶ。デジタル信号処理では、Digital Signal Processorがしばしば用いられ、多くの高位設計手法とそれに基づくCADシステムが研究開発されてきた[1,2,3]。しかしながら、既存のCADツールは高位の伝送処理システム設計には適していない(今でもシステムレベルの伝送処理設計は手作業で行われている)。システムレベル仕様はしばしば図、表、フローチャートや自然言語を用いて記述されており、従来のトップダウン設計環境を利用するためには、その仕様がRegister Transfer Level (RTL)に変換されなければならない。この作業は、設計者に対して多大な労力を要求することになる。これが設計コスト増大の大きな要因となっている。この現状を打破するためには、伝送処理に適したモデルを持つ設計手法とそれに基づくCADシステムが必要である。

通信処理に関わる多くのモデルが提案され、その幾つかはSDL [4] や LOTSO [5] のように標準化されている。また、幾つかの通信プロトコルを扱うCADシステムも報告されている [6,7]。そのモデルとCAD環境は相対的なシーケンスを表現することを主眼としており、通信サービスやより高位のプロトコル処理を表すのに適している。一方、伝送処理システム設計においてはリアルタイム処理が本質的なものであり、上記モデルを直接適用することができない。RTLモデルを適用することも考えられるが、いかに洗練されたCADシステムを導入したとしても、タイミングの詳細設計は不可欠でありシステムレベル設計に不向きである。したがって、伝送処理に特化したモデルに基づく新しい設計手法の開発が不可欠である。

本報告では、Frame-Oriented Representation Method (FORM) と呼ぶ伝送フレームに基づく新設計手法を提案する。第二章で伝送システム的设计時の問題点について述べた後、FORMのシステム概念とそのモデルに

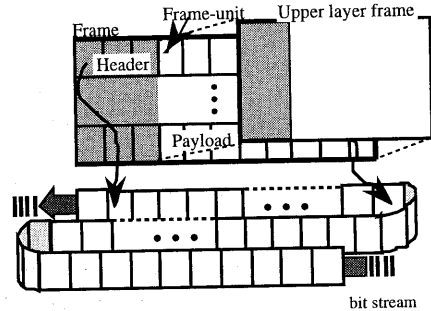


図1: 伝送データマッピングモデル

ついて第三章と第四章でそれぞれ説明する。第五章では本手法を Synchronous Digital Hierarchy (SDH) / Asynchronous Transfer Mode (ATM) インタフェースに適用した結果について述べる。

2. 伝送処理システム設計時の問題点

伝送処理システムは、いくらかの処理遅延は許されるが一ラインあたり数Mbpsから数Gbpsの高スループットが要求される。このようなシステムでは、絶対的なリアルタイム処理が要求されるため、設計時に以下のような問題を生じる。

- ・ **タイミング制御**: 多重化階層毎の伝送データが主信号上に一度にマッピングされているため、各階層毎の多様な伝送処理が同時に行われる必要がある。各々の処理は、異なるタイミング制約を持っておりタイミング制御設計が非常に複雑になる。
- ・ **回路規模**: SDH/ATMインタフェース終端回路においては、STM-1、VC-4、ATMセルの各多重化階層毎の処理回路規模は1~10Kゲートと様々であった。このように伝送処理機能を多重化階層毎に単純に分割したとしても簡単な設計を行うには大きすぎる。

3. 伝送処理システム概念

図1に示すように伝送データは論理的にはビット流にマッピングされた多くのフレームユニットからなる空間的なフレーム構造とみなすことができる。フレームはヘッダと呼ばれるその階層で処理されるべき部分とペイロードと呼ばれる上位階層処理のために残される部分の二つからなる。上位階層のフレームはペイロードに多重化され、それが各階層で繰り返される。

このような伝送データを扱う伝送処理システムは全体としては非常に複雑ではあるが、各多重化階層においては以下の二つを本質的な機能と考えることができる。

- ・ **Synchronization** : 下位階層のペイロードに格納されたビット流から同期情報を抽出することによりフレームを認識する。
- ・ **Transport Processing Action (TPA)** : 制御情報抽出/挿入とビット操作をそれぞれのフレームユニットに対して行い、そのフレームに埋め込まれた情報を他の階層に知らせる。

ここでは簡単化のために一ライン処理を行うシステムを考える。各階層毎に上記した同期機能によって同期が確立された後に各々のフレームが認識される。データは、各多重化階層におけるフレームアドレス (local address) によって管理される。TPA は、同一の local address で参照されるフレームユニットのデータに対して実行される。ここでは、local address で管理される処理機能のグループを総称して Transport Processing Group (TPG) と呼ぶ。TPG は各多重化階層毎に存在し、伝送処理システムはこれらの TPG を組み合わせて表現することができる。

システム設計者は、ふつう各多重化階層毎のフレームを定義してからそのフレームに対するデータ操作として個々の処理機能を記述する。FORM はその設計スタイルにあった効率的な設計を実現するために提案されており、システム設計者が伝送システム機能を表現するのに適したモデルに基づいている。FORM には、機能モデルとハードウェアモデルの二つがある。前者は、後者に比べて機能レベルで定義するという点で高位モデルとして位置づけられる。ハードウェアモデルは、機能モデルから抽出されたタイミング制御情報付加して生成され RTL 言語で記述される。これらのモデルと設計手法の詳細を次章で述べる。

4. 新しいシステム設計モデルの提案

4.1 FORM 機能モデル

TPG は図 2 に示すように Synchronization、Frame、TPA の三つの部分からなり、それらは設計者によって定義される。Synchronization は、フレームの先頭をそのフレームが格納される下位階層フレームのペイロード位置にリンクする。フレームが認識された後は、local address によって参照されたフレームユニットデータに

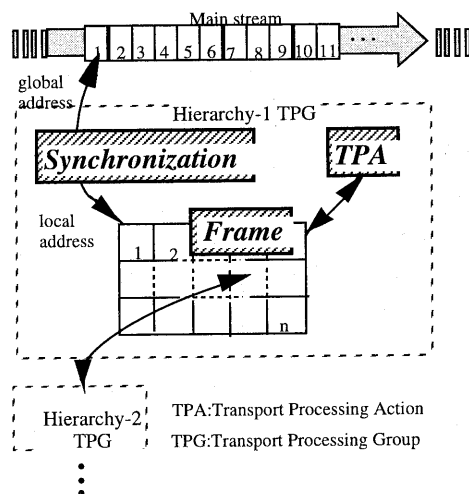


図 2: FORM 機能モデル

対して TPA が実行される。International Telecommunication Union (ITU) の仕様を分析した結果、TPA 機能はパターン比較、スクランブル、パリティ計算、cyclic redundancy check (CRC) 法のようなフレームユニットデータのビット操作として表すことができる [8,9]。システム全体は、このような TPG を組み合わせることにより実現することができる。主信号のデータ管理は、システム全体にわたる同期リンクによって生成された global address を用いて一元的に行われる。システム設計者が意識する必要はないが、本モデルにおいてはタイミング制御に必要な情報は間接的に格納されている。

上記した機能モデルでは、処理機能がシステム設計者に馴染みやすいフレームに基づいて定義されており、同様な構造の伝送データを扱う他のシステムに対しても容易に適用できる。設計者が一度 TPG を定義しておけば、それらの TPG をリンクするだけでいつでも新しいシステムを構築することができる。本モデルでは、タイミング制御については全く記述する必要がないため設計者は従来と同様な抽象度の機能レベルで簡単にシステム機能を記述できる。

4.2 FORM ハードウェアモデル

フレーム構造は下位階層のフレームのペイロードの主信号流から抽出され、それぞれの TPG の中で local address が付与されなければならない。機能モデルをハードウェアとして実現するためには、機能モデルで

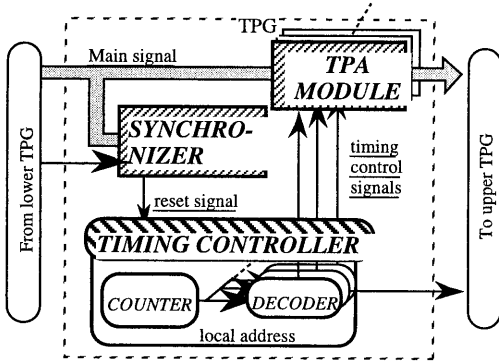


図 3: FORMハードウェアモデル

定義されたTPAの起動タイミング制御信号が各フレーム毎の local address に基づいて管理される。

FORMハードウェアモデルを図3に示す。機能モデルにおける Synchronization と TPA はそれぞれ SYNCHRONIZER と TPA MODULE として実現される。TIMING CONTROLLER は Frame と TPA 定義から生成することができる。SYNCHRONIZER は、COUNTER と DECODER から成る TIMING CONTROLLER に対して reset signal を送る。COUNTER はフレーム長まで数え上げるフレームカウンタであり、reset signal を受けてフレームカウンタをリセットした後に local address を生成する。タイミングは、各 DECODER で local address をデコードして生成された各 TPA MODULE 管理用の timing control signal により制御される。同一フレームにおける全ての TPA MODULE に対しては共通の COUNTER が用いることができ、各 TPA 毎に単に DECODER のみを生成すればよい。timing control signal は、上位・下位階層の TPG と同期をとるためにも用いられる。伝送処理システムは、これら TPG を多重化階層順に連結することで実現できる。

4.3 設計例

4.3.1 機能記述

SDH/ATM インタフェースにおける多重化階層を図4に示す。多重化の順序は ATMセル->VC-4->STM-1 であり、これらは各々フレーム名である。この例を用いて FORM 設計手法 を説明する。図5に SDH/ATM インタフェース終端回路 "stm1_atm_t" の FORM 機能記述例を示す。

本例では、設計者は "\$TPG_DEF" において図4の各

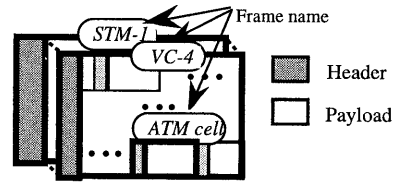


図 4: SDH/ATMインタフェースにおける多重化階層例

```

SYSTEM stm1_atm_t {
  global vpi_set(8:1) = 11b';
  global vci_set(16:1) = 0011b';
  global vci_mis_msg, vpi_mis_msg;

  $TPG_DEF stm1 {
    $SYNCHRONIZATION_DEF { ..... }
    $FRAME_DEF { ..... }
    $TPA_DEF { ..... }
  }

  $TPG_DEF vc-4 {
    $SYNCHRONIZATION_DEF { ..... }
    $FRAME_DEF { ..... }
    $STPA_DEF { ..... }
  }

  $TPG_DEF atm {
    $SYNCHRONIZATION_DEF {
      $SYNC_POINT vc-4.payload;
      $SYNC_METHOD atm_sync;
    }
    $FRAME_DEF {
      $LENGTH_UNIT byte;
      $LENGTH 53;
      header (1:5);
      payload (6:53);
      vpi (1.4:2.3);
      vci (2.6:4.5);
    }
    $STPA_DEF {
      $SLAYER L1 {
        IF position(payload)
        THEN {
          atm_dscr(get(payload));
        }
      }
      $SLAYER L2 {
        IF position(vpi)
        THEN {
          IF cmp(get(vpi), vpi_set)
          THEN vpi_mis_msg = 0;
          ELSE vpi_mis_msg = 1;
        }
        IF position(vci)
        THEN {
          IF cmp(get(vci), vci_set)
          THEN vci_mis_msg = 0;
          ELSE vci_mis_msg = 1;
        }
      }
    }
    $STATE_DEF vpi_mis {
      $STATE {vpi_mis_state,
              normal}
      vpi_mis_state {
        IF cmp(vpi_mis_msg, 0)
        THEN GOTO normal;
      }
      normal {
        IF cmp(vpi_mis_msg, 1)
        THEN GOTO vpi_mis_state;
      }
    }
  }
}

```

図 5: SDH/ATMインタフェースにおけるFORM機能記述例

多重化階層に対応する "stm1"、"vc-4"、"atm" の三つの TPG を定義する。各々の TPG 定義は、図2に示した FORM 機能モデルにおける Synchronization、Frame、

表 1: SDH/ATMインタフェースにおける機能記述用TPAプリミティブ関数

Function name	Action
atm_scr/atm_dscr	scramble/descramble with the generating polynomial $1+X^{43}$
stm_scr/stm_dscr	scramble with the generating polynomial $1+X^6+X^7$
bip_24/8	calculate bit interleaved parity-24/8
crc_3/8/10	calculate cyclic redundancy check with the generating polynomial $1+X+X^3/1+X+X^2+X^3/1+X+X^4+X^5+X^9+X^{10}$
cmp	compare some bit data
get/put	get/put bit data from/to bit stream according to the position defined by the unique name
position	check whether the desired position has come or not

TPA に相当する "\$SYNCHRONIZATION_DEF"、"\$FRAME_DEF"、"\$TPA_DEF" からなる。ここでは、TPG"atm"に着目してその定義について説明する。"atm"フレームは、"vc-4"の"payload"に格納されており、ITU勧告 I361 [9] に規定されている同期方法"atm_sync"によって認識される。"\$LENGTH"と"\$LENGTH_UNIT"で定義されているように、フレーム長は53でその単位はバイトである。先頭から5バイトと残りの部分はそれぞれ"header"と"payload"として定義されている。"vpi"は一バイト目の#4ビットから二バイト目の#5ビットの間の位置として定義されている(一バイトは、#8、#7、#6、...)。"vci"も同様な手順で定義されている。図5に示すように"atm"は"payload"、"vci"、"vpi"にそれぞれ対応するA1、A2、A3の三つのTPAを含んでいる。他のシステムには多少異なる関数が必要であるかもしれないが、これらのTPAは、表1に示すような幾つかのビット操作関数で表現できる。TPAの機能は、これらの関数と"IF"-"THEN"条件で表現できる。"payload"の位置が来たときに、A1で規定された"atm_dscr()"がそこに格納されたデータに対して実行される。"vpi"の位置が来たときには、そのデータに対して定数のビットデータ"11h"と比較される。もしそれらの値が等しいなら、メッセージ変数"vpi_mis_msg"に"0"がセットされ、メッセージが伝達される。さらに、状態遷移"vpi_mis"は"\$STATE_DEF"で定義される。この例では、"vpi_mis_state"と"normal"の二状態が定義されている。"vpi_mis_msg"が"0"かつ状態が"vpi_mis_state"であった時は、状態は"normal"に遷移する。"\$LAYER"はTPAの順序を定義するために導入した。"\$LAYER"で記述された処理は、その記述順に実行

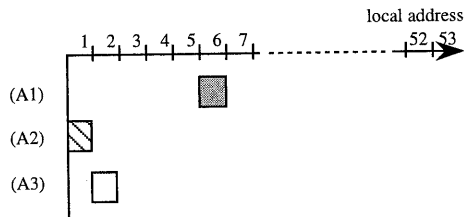


図 6: ATMセルにおけるタイミング抽出例

される。図5においては最初にA1が実行された後、他のTPA (A2、A3)が処理データに対して実行される。

4.3.2 タイミングコントローラ生成

TIMING CONTROLLERは設計者の手を煩わすことなく自動的に生成することができる。図5に示した機能モデル記述のTPA定義"\$TPA_DEF"及びフレーム定義"FRAME_DEF"を参照することにより、TPAが処理を開始すべき時間を見つけたことができる。そのステップは以下ようになる。

- 1) フレームカウンタ長53を"\$LENGTH"から決定
- 2) "\$TPA_DEF"からTPA"A1"が実行されるべき"payload"と命名された位置を発見
- 3) "\$FRAME_DEF"からその位置のlocal address 6を付与

図6にTPA A1、A2、A3に対するタイミング抽出例を示す。このタイミング情報は、各々のTPGから別々に生成することができる。システムのタイミング制御設計を行うには、これらの情報を収集すればよい。システムが単相同期システムでTPA処理が必ず一クロックサイクルで終了すると仮定すれば、TIMING CONTROLLERは簡単に実現できる。そのコンポーネントはCOUNTERとDECODERであり、上記したフレームカウンタ長とタイミング情報によって生成できる。TPA MODULEの機能は表1に示した幾つかの関数で表されており、TPAとSYNCHRONIZERの回路はRTLライブラリとして提供される。したがって、機能モデルからハードウェアモデルへの変換は比較的簡単に行える。

5. SDH/ATMインタフェースへの適用

FORMハードウェアモデルを回路として実現しようとした時、一つの問題に直面した。提案したタイミン

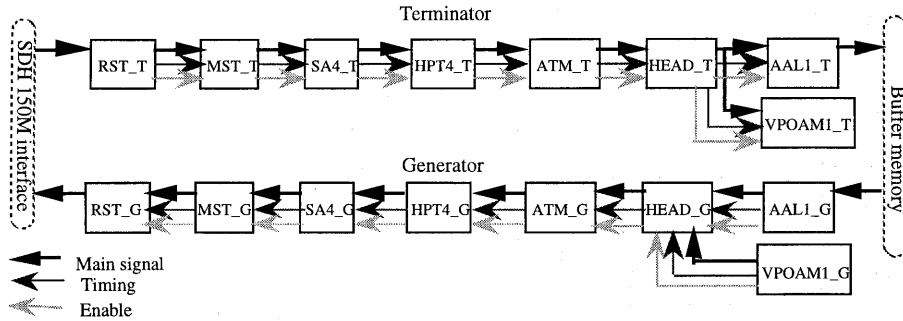


図 7: SDH/ATMインタフェースへの適用

グ制御手法は洗練されていたが、全 TPA を制御するには多くの timing control signal が必要であった。そのため、TPG が SYNCHRONIZER、TIMING CONTROLLER と幾つかの TPA を含むブロックに単純に分割されれば、その間を渡る多くの timing control signal を取り扱ことは非常に大変である。そこで、ここでは "Timing" と "Enable" の二つの共通タイミング制御インタフェースを導入する。前者は、TIMING CONTROLLER をリセットし、後者はアドレスのカウントを可能にするものである。各々のブロックは自前の TIMING CONTROLLER を持っており、TPA を制御する timing control signal はブロックの外側からは隠蔽されている。この共通インタフェースを用いる限り、設計者は外部ピンを増加させることなく TPG を幾つかのブロックに自由に分割できる。

FORM を SDH/ATM インタフェースの開発に適用した。図 7 に本回路のブロック図を示す。それぞれのブロックは図 5 の L1 や L2 のような異なる "\$SLAYER" に対応している。設計者は上記した共通インタフェースを用いて TPG を自由に分割し、各々のブロックの回路規模を適当に調整できる。これらのインタフェース回路は、上記した共通インタフェースを用いて三人の設計者によって独立して設計された。これらの記述を論理合成ツールによってゲートに変換してプログラマブルなりアルタイム伝送実験システム [10] にマッピングし、20MHz で動作することを確認した。

6. まとめ

新しいフレームベースの設計手法 FORM を提案した。本手法は、機能モデルとハードウェアモデルを持ち、機能設計とタイミング制御設計を分離して行うことができる。後者は、前者からタイミング制御情報を

自動的に抽出することにより行われる。FORM を SDH/ATM インタフェースシステムの設計に適用し、提案した共通タイミング制御インタフェースを用いて三人の設計者による協調設計を行った。設計した回路をプログラマブルな伝送実験システムにマッピングした結果、20MHz で動作することを確認した。

参考文献

- [1] H. De Man et al., "Cathedral II: A Silicon Compiler for Digital Signal Processing," IEEE Design & Test, Vol. 3, No. 6, Dec. 1986.
- [2] A. Kalavade et al., "A Hardware-Software Codesign Methodology for DSP," IEEE Design & Test, pp. 16-28, Sep. 1993.
- [3] S. Note et al., "Combined Hardware Selection and Pipelining in High-Level Performance Data-Path Design," IEEE Trans. on CAD, Vol. 11, No. 4, pp. 413-423, Apr. 1992.
- [4] R. Saracco et al., "CCITT SDL: Overview of the Language and its Application," Comp. Networks ISDN Syst. Special Issue, Vol. 13, No. 2, pp. 65-74, 1987.
- [5] International Standards Organization, "LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behavior," ISO, IS 8807, Feb. 1989.
- [6] A. Jerraya et al., "Towards System-Level Modeling And Synthesis," Proc. VLSI'92, India, Feb. 1992.
- [7] A. Jerraya et al., "Linking System Design Tools and Hardware Design Tools," Proc. CHDL'93, Ottawa, Canada, pp. 331-338, Apr. 1993.
- [8] ITU Recommendation, G707-709, 1992.
- [9] ITU Recommendation, I361, 1992.
- [10] K. Hayashi et al., "Reconfigurable Real-time Signal Transport System Using Custom FPGAs," Proc. FCCM'95, Napa, USA, Apr. 1995.