

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2010年12月16日(16.12.2010)

(10) 国際公開番号
WO 2010/143686 A1

- (51) 国際特許分類:
H04L 29/10 (2006.01) H04L 29/06 (2006.01)
B60R 16/023 (2006.01)
- (21) 国際出願番号: PCT/JP2010/059841
- (22) 国際出願日: 2010年6月10日(10.06.2010)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2009-140890 2009年6月12日(12.06.2009) JP
- (71) 出願人 (米国を除く全ての指定国について): ルネサスエレクトロニクス株式会社 (Renesas Electronics Corporation) [JP/JP]; 〒2118668 神奈川県川崎市中原区下沼部1753番地 Kanagawa (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 辻本 拓哉 (TSUJIMOTO, Takuya) [JP/JP]; 〒6640005 兵庫県伊丹市瑞原四丁目1番地 株式会社ルネサスデザイン内 Hyogo (JP). 上村 俊之 (UEMURA,

Toshiyuki) [JP/JP]; 〒2118668 神奈川県川崎市中原区下沼部1753番地 ルネサスエレクトロニクス株式会社内 Kanagawa (JP). 三島 誠史 (MISHIMA, Seishi) [JP/JP]; 〒6640005 兵庫県伊丹市瑞原四丁目1番地 株式会社ルネサスデザイン内 Hyogo (JP). 木村 宏行 (KIMURA, Hiroyuki) [JP/JP]; 〒6640005 兵庫県伊丹市瑞原四丁目1番地 株式会社ルネサスデザイン内 Hyogo (JP). 境田 満成 (SAKAIDA, Michinari) [JP/JP]; 〒6640005 兵庫県伊丹市瑞原四丁目1番地 株式会社ルネサスデザイン内 Hyogo (JP).

(74) 代理人: 筒井 大和, 外 (TSUTSUI, Yamato et al.); 〒1600022 東京都新宿区新宿2丁目3番10号 新宿御苑ビル3階 筒井国際特許事務所 Tokyo (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR,

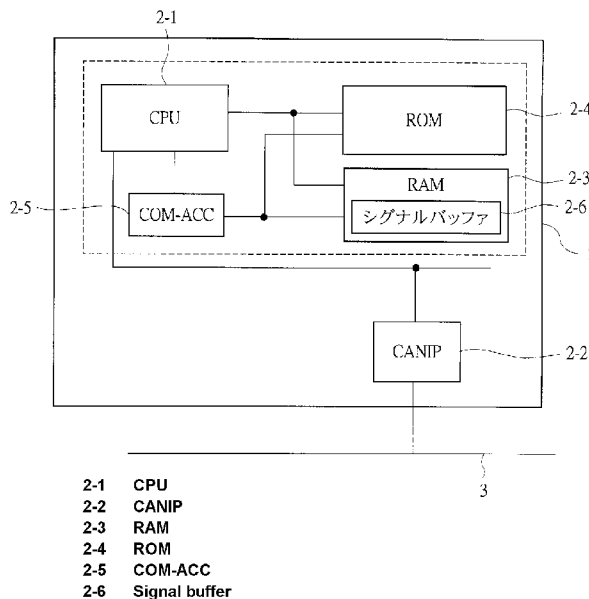
[続葉有]

(54) Title: AUTOMOBILE CONTROL SYSTEM AND ELECTRONIC CONTROL UNIT

(54) 発明の名称: 自動車用制御システム及び電子制御ユニット

[図3]

図 3



(57) Abstract: Provided is a means for increasing processing speed by using algorithms specialized for automobile software and dedicated hardware, compensating for processing speed losses accompanying abstraction in automobile software. A COM-ACC (COM accelerator) (2-5) is provided in parallel with a CPU (2-1). A protocol stack running on the CPU (2-1) extracts a PDU and PDU ID. The extracted PDU and PDU ID are sent to the COM-ACC (2-5), which decides whether to update the PDU.

(57) 要約: 自動車用ソフトウェアによる抽象化に伴う処理速度の低下を、専用のハードウェア及び自動車用ソフトウェアに特化したアルゴリズムを用いることで、処理速度の向上を図る手段を提供する。CPU 2-1に並存する形でCOM-ACC (COMアクセラレータ) 2-5を設ける。該CPU 2-1で動作するプロトコルスタックはPDU及びPDU IDを抽出する。抽出したPDU及びPDU IDはCOM-ACC 2-5に送出された後、COM-ACC 2-5がPDUの更新判断を行う。



LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：自動車用制御システム及び電子制御ユニット

技術分野

[0001] 本発明は、自動車用伝送アーキテクチャに関するAUTOSAR（登録商標）仕様に準拠した自動車用のソフトウェアコンポーネント、特にハードウェアアクセラレータを併用することで処理負荷の軽減を図る手段に関する。

背景技術

[0002] AUTOSAR（登録商標）（AUTomotive Open System ARchitecture）は自動車用伝送アーキテクチャに関するオープンなデファクトスタンダードの策定及び成立のために協業する自動車製造業者及び自動車サプライヤーのパートナーシップならびに該パートナーシップが策定する仕様群である。AUTOSAR（登録商標）は、各種の自動車用のソフトウェアコンポーネントを各種定義し、またそれらのソフトウェアコンポーネント間で用いるインターフェースを定義することで、自動車用ソフトウェアの再利用性、汎用性を確保することを目的に活動を行っている。

[0003] AUTOSAR（登録商標）はECU（電子制御ユニット）を単位としてソフトウェアを構成することを基本コンセプトとしている。ECU間は抽象的なVFB（Virtual Functional Bus：仮想機能バス）で接続され、実装上のハードウェアと切り離れた形でシステムの初期検討が可能となっている。実装に際しては、VFBはCAN（Control Area Network）やFlexRayといったより具体的なハードウェアを想定して開発を行う。その際にも、CANDライバやFlexRayドライバといった標準コンポーネントによってハードウェア（特にCPU）が相違しても上位層の再利用性を確保することが想定されている。従って、CANやFlexRayと言ったハードウェアを想定した規格と分離した形でデータの送受信を行う必要に迫られる。AUTOSAR（登録

商標)においてはPDU (Protocol Data Unit) がこれに当たる。

[0004] PDUは上位層であるアプリケーションソフトウェアで利用される各種データの最小単位であるシグナルをECU間で送受信するための論理的なデータコンテナである。PDUは8バイト (=64ビット) 幅のデータである。図1はECU間でのPDUの取り扱いを表す概念図である。

[0005] この図においては、送信側ECU1と受信側ECU2がCAN3を介して通信を行うことを想定する。送信側ECU1はPDUに適切なヘッダ、フッタを添付したCANフレーム4をCAN3経由で受信側ECU2に送信する。受信側ECU2のCANIP2-2は受信したCANフレーム4からヘッダ及びフッタを切り離し必要な処理を行う。受信側ECU2内のCPU2-1はヘッダ、フッタを切り離したPDUからシグナルの分離、フィルタ処理を行うこととなる。

[0006] ここで、フィルタ処理とは、特定の状態にメッセージの値が合致しないときにメッセージフィルタによって受信したメッセージを無視することをいう。

[0007] これらの一連の処理についてはAUTOSAR (登録商標) Specification of CAN Interface (非特許文献1)、AUTOSAR (登録商標) Specification of CAN Driver (非特許文献2)、Specification of CAN Communication (非特許文献3)、AUTOSAR (登録商標) Specification of Flexray Interface (非特許文献4)、AUTOSAR (登録商標) Specification of FlexRay Driver (非特許文献5)、AUTOSAR (登録商標) Specification of Communication (非特許文献6)などで規定されている。

先行技術文献

非特許文献

[0008] 非特許文献1: AUTOSAR (登録商標) Specification of CAN Interface v3.0.3

非特許文献2: AUTOSAR (登録商標) Specification of CAN Driver v2.2.3

非特許文献3: AUTOSAR (登録商標) Specification of CAN Communication v3.0.3

非特許文献4: AUTOSAR (登録商標) Specification of FlexRay Interface v3.0.2

非特許文献5: AUTOSAR (登録商標) Specification of FlexRay Driver v2.2.1

非特許文献6: AUTOSAR (登録商標) Specification of Communication v3.0.2

発明の概要

発明が解決しようとする課題

[0009] これらAUTOSAR (登録商標) の仕様では、自動車用ソフトウェアの再利用性、汎用性を確保する、という目的から、ハードウェアへの依存は制限されている。

[0010] しかし、パーソナルコンピュータの分野でもそうであったように、ソフトウェアによる抽象化は処理工程、処理時間の増大を招かざるを得ない。これに対して、パーソナルコンピュータの分野では、一般的な処理を行うCPU (中央処理装置) に対して、特定の処理に特化したアクセラレータチップを並存させ、上記のような抽象化された処理を高速に処理する手段が考えられている。

[0011] 一方、パーソナルコンピュータと異なり、自動車には特有な問題が多く存在する。すなわち、パーソナルコンピュータでは、送受信の対象となるのは、高位のアプリケーション層に位置するアプリケーションで取り扱われるデータがほとんどである。これに対し、自動車用ソフトウェアにおいては、自動車の状態を表す数ビット単位の短いデータ長を表すデータがやり取りされ

ることが多く、自動車用ソフトウェアで扱うデータは数ビット単位のもが多く、32ビットCPU等のビット幅の広いバスを採用したCPUを用いても一般的なCPU命令による処理では効率が上がらない、といった具合である。

[0012] また、AUTOSAR（登録商標）の仕様では、受信時には下位のプロトコル処理部（CAN—IP、FlexRay—IP等）にてPDU単位で扱われたデータ単位をシグナル単位へ分解する、送信時にはシグナル単位のデータをPDU単位のデータへ梱包する処理を行う、といったパーソナルコンピュータの分野ではあまり起こらない処理を行う。これらビット単位の梱包、分解処理はCPUにとって大きな負荷である。

[0013] 昨今、地球環境への影響を最小限に抑えるべく、自動車においては少燃費である事が必須課題となっている。一方でECUによる電子制御は益々増大している。このためECU自体の消費電力も厳しく制限されてきている。従ってECUの処理負荷が増大する事に対する解決策として、より高速なCPUを用いるという方法が困難になりつつある。このため、何らかのハードウェアによる高速かつ短時間での処理が必要となっている。

[0014] 本発明の目的は、自動車用ソフトウェアによるハードウェアの抽象化に伴う処理速度の低下を専用のハードウェアを用いて補助し、処理速度の高速化および低消費電力化を図る手段を提供する事にある。

[0015] 本発明の前記並びにその他の目的と新規な特徴は、本明細書の記述及び添付図面から明らかになるであろう。

課題を解決するための手段

[0016] 本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、次の通りである。

[0017] 本発明の代表的な実施の形態に関わる自動車用制御システムは、1又は2以上のシグナルを有するプロトコルデータユニットを含むデータフレームを受信する第1のハードウェアと、第1のハードウェアを接続しデータフレームの伝達に用いられる仮想機能バスと、を有し、第1のハードウェアは、仮

想機能バス経由で受信したデータフレームからプロトコルデータユニット及びプロトコルデータユニットのデータフォーマットを識別するプロトコルデータユニットIDを抽出するプロトコルスタックを実行し、プロトコルスタックは、1又は2以上のシグナルの更新の有無の判定をプロトコルデータユニットへの一度の処理で実行することを特徴とする。

[0018] 本発明の代表的な実施の形態に関わる電子制御ユニットは、CPUと、仮想機能バスインターフェースと、アクセラレータと、を有し、仮想機能バスインターフェースを介してプロトコルデータユニットを含むデータフレームを受信し、CPUは受信したデータフレームからプロトコルデータユニット及びプロトコルデータユニットのデータフォーマットを識別するプロトコルデータユニットIDを抽出するプロトコルスタックを実行し、CPUは、プロトコルデータユニット及びプロトコルデータユニットIDをアクセラレータに送信し、アクセラレータがシグナルの更新の有無の判定及びシグナルの更新を実行することを特徴とする。

[0019] この電子制御ユニットにおいて、アクセラレータは、プロトコルデータユニットに含まれるアップデートビットの判定と、プロトコルデータユニット内の各シグナルに設定されたアルゴリズムに従い実行されたフィルタリング処理の判定結果から、過去に処理したプロトコルデータユニットの値を更新するデータ更新用マスクを作成することを特徴としてもよい。

[0020] また、この電子制御ユニットにおいて、アクセラレータは、データ更新用マスクを用い、データの更新があったシグナルを抽出し、プロトコルスタックに前記データの更新があったシグナルを出力することを特徴としてもよい。

[0021] この電子制御ユニットにおいて、プロトコルスタックは、受信した前記データの更新があったシグナルを用いてプロトコルデータユニットを更新することを特徴としてもよい。

発明の効果

[0022] 本願において開示される発明のうち、代表的なものによって得られる効果

を簡単に説明すれば以下の通りである。

[0023] 本発明の代表的な実施の形態に関わるCOMアクセラレータ並びにそれに対応するCOMソフトウェアを採用することで、AUTOSAR（登録商標）で規定するAUTOSAR（登録商標）COMの一定の処理の動作単位をPDU単位にすることができる。これにより、従来シグナル毎に反復処理を求められたシグナル更新の判定を一度に行うことができるという論理的な処理工数の低減、及びハードウェア処理による処理工数の低減を図ることが可能となる。

[0024] また、この処理工数の低減により実動作時間が短縮されるため、処理負荷増大による消費電力増加を抑制する事が可能となる。

図面の簡単な説明

[0025] [図1] ECU間でのPDUの取り扱いを表す概念図である。

[図2] 従来のAUTOSAR（登録商標）の各種ソフトウェアを動作させるためのハードウェアの構成を表すブロック図である。

[図3] 本発明に関わるハードウェアの構成を表すブロック図である。

[図4] 一般的なAUTOSAR（登録商標）用の通信用プロトコルスタックの構成を表す概念図である。

[図5] COM-ACC内のフィルタ回路を8ビット幅で表した概念図である。

[図6] シグナル抽出処理におけるフィルタ処理を説明するためのフローチャートである。

[図7] PDU受信時のCOMソフトウェアとCOM-ACCとの処理の分担についての概念図である。

[図8] 本発明に関わるデータ更新用マスク生成回路の構成を表す概念図である。

[図9] 本発明に関わる受信フラグデータの圧縮回路の一部を表す回路図である。

[図10] 図9の圧縮回路の動作を説明するための図である。

[図11] COM-ACC内のシグナル挿入抽出回路におけるこのシグナル抽出

処理の手順を表す概念図である。

[図12] シグナル送信時のCOMソフトウェアとCOM-ACCとの処理の分担についての概念図である。

[図13] シグナル挿入抽出回路でのシグナル挿入処理の手順を表す概念図である。

[図14] アップデートビットの判定方法及び対応するシグナルへの結果の反映方法の概念図である。

[図15] COM-ACCの概略構成を示すブロック図である。

[図16] 本発明の第2の実施の形態にかかわるCOM-ACCの構成を表すブロック図である。

[図17] 本発明の第2の実施の形態にかかわるPDUデータの受信処理に係るフローチャートである。

[図18] シグナル境界情報、及びPDUデータのデータ構成を表す概念図である。

[図19] シグナル境界情報、アップデートビット許可情報、アップデートビット位置情報及びアップデートビットチェック結果情報のデータ構成を表す概念図である。

[図20] 本発明の第2の実施の形態の（合わせこみ処理1）にかかわるアップデートビットチェック部の模式図である。

[図21] 本発明の第2の実施の形態の（合わせこみ処理3）にかかわるアップデートビットチェック部の模式図である。

[図22] フィルタ処理部の構成を示すブロック図である。

[図23] フィルタアルゴリズム選択器の構成を示すブロック図である。

[図24] マスク生成回路および生成方法を示した図である。

[図25] 各設定データ及び入力データの流れを表す図である。

[図26] マスクデータ生成時のデータフローおよびデータ構成図を示す図である。

[図27] 図20の一連の手順を実現する構成を表す回路図である。

[図28] 右詰したアップデートビット情報をアップデートビット許可情報でデータ High に設定されているビット位置に、対応するアップデートビット情報を当てはめる場合の概念図である。

[図29] 図 28 の回路動作を詳細に説明したものである。

[図30] 図 29 で述べた右端検出回路の構成を表すブロック図である。

[図31] 処理単位 UBR の内部構成を表す回路図である。

[図32] 図 29 で述べたアップデートビット比較結果シフト回路の構成を表すブロック図である。

[図33] 処理単位 UBS の内部構成を表す回路図である。

[図34] (合わせこみ処理 2) で行う 64 ビット幅の動作に付いての概念図である。

[図35] 6 ビット幅 8 段加算器を用いたアップデートビットの位置あわせを行う概念図である。

[図36] 図 34 および図 35 の処理の流れをまとめたブロック構成図である。

発明を実施するための形態

[0026] 以下、本発明の実施の形態について図を用いて説明する。

[0027] (第 1 の実施の形態)

図 2 は従来の AUTOSAR (登録商標) の各種ソフトウェアを動作させるための受信側 ECU 2 の構成を表すブロック図である。図 3 は本発明に関わる受信側 ECU 2 の構成を表すブロック図である。

[0028] なお、図 2 および図 3 は説明用の図面であるので、I/O 等の回路等は省略している。

[0029] 従来の AUTOSAR (登録商標) 用の受信側 ECU 2 は CPU 2-1、CAN IP 2-2、RAM 2-3、ROM 2-4 を含んで構成される。

[0030] 本発明の実施の形態は CAN による通信にて構成されているが、他の通信プロトコルである FlexRay、Ether 等での利用も想定されている。図 1 に FlexRay による通信用の IP およびバスの構成を表しているが、ここでは CAN による構成を用いて説明を行う。

[0031] CPU 2-1はAUTOSAR（登録商標）の規定する各種仕様に従ったソフトウェアを動作させるためのCPUである。自動車には数多くの制御対象となるハードウェアが存在する。そしてそれらのハードウェアの制御には、それぞれにCPUを用意する必要がある。しかし、自動車のモデルチェンジは4年以上であり、CPU等の電子部品の製造期間と相違することが多々ある。また、自動車のマイナーチェンジ時に新たな機能を追加するなどの理由により異なるCPUを利用しなければならなくなった時に、そのCPUの変更に迅速に対応することができるようにするのがAUTOSAR（登録商標）の目的である。

[0032] また、このアプリケーションソフトウェアはC言語等の高級言語で記述されている場合、コンパイラの設定を変更する事で多くの場合、CPUの違いを超えてソフトウェアを共有する事が可能となっている。たとえば、パーソナルコンピュータの分野では、基本となるオペレーションシステム（OS）上で、動作するアプリケーションソフトウェアは、同一のOSが搭載されたパーソナルコンピュータであれば、製造メーカーおよびCPUの違いを区別せず使用する事が可能である。一方自動車用ECUに搭載されるマイクロコントロールユニット（MCU）の分野では、従来ではアプリケーションソフトウェアとMCUの各機能を動作させるためのドライバソフトウェア等が直接データの受け渡し処理を行っていた。このため、処理の最適化の意味では効率よく動作していたのだが、他方、アプリケーションソフトウェアの共有性を損なっていた。このため、制御システムの開発毎にドライバソフトウェア、アプリケーションソフトウェアの双方を作り直す必要があった。これは正もAUTOSAR（登録商標）の目的である。

[0033] CAN IP 2-2は、CANバス対応の通信ハードウェアである。CAN IP 2-2とCPU 2-1は図示しないブリッジチップ等を介して接続されている。CAN IP 2-2はMCU外部とはCANバスで接続され、MCU内部では、CPUと内部の専用接続線を用いて接続されている。

[0034] RAM 2-3は、CPU 2-1が動作する際の一時記憶領域として用いら

れる記憶媒体である。CPU 2-1が動作中に使用する一時記憶領域である。

[0035] ROM 2-4は、CPU 2-1で動作させるソフトウェアを記録するための不揮発性の記録媒体である。CPU 2-1が動作中に使用する不揮発性の記憶領域である。

[0036] CAN 3はVFB（仮想機能バス）の一種である。CAN 3はECUに接続されているCANバスであり、同一バス上には1又は2以上の別のECUが同様に接続されている。

[0037] これに対し、本発明のハードウェアはCOM-ACC 2-5が追加されている点が従来の物と相違する。

[0038] COM-ACC（COMアクセラレータ）2-5は、CPU 2-1で処理すべきAUTOSAR COMソフトウェアのうち、所定の処理を専用に担当するためのアクセラレータチップである。すなわち、従来CPU 2-1で処理されるAUTOSAR（登録商標）COMソフトウェアの処理のうち、反復的に実行される箇所を専用に処理するハードウェアモジュールである。また、COM-ACC 2-5はRAM 2-3やROM 2-4に接続され、COM-ACC内での動作に必要な設定データの取得および処理が完了したデータの一時保存に用いられる。このCOM-ACC 2-5での処理が本発明の特徴となる。

[0039] 図4は一般的なAUTOSAR（登録商標）用の通信用プロトコルスタックの構成を表す概念図である。

[0040] ハードウェアであるCANIP 2-2を除くと、CANDライバソフトウェア102、CANインターフェースソフトウェア103、PDUルータソフトウェア104、COMソフトウェア105及びRTE（Run Time Environment）ソフトウェア106を含んで構成される。

[0041] CANDライバソフトウェア102は、CANIP 2-2を直接制御し、PDUデータ及びCANプロトコルで規定されているデータパケット識別用のID（PDU-ID）をCANIP 2-2が管理するデータ保存領域から

取得し、データの取得を上位のCANインターフェースソフトウェア103へ送信するためのソフトウェアである。

[0042] CANインターフェースソフトウェア103は、CANDライバソフトウェア102によって検出され処理されたイベントが送信される。またPDUルータソフトウェア104などの上位層からCANDライバソフトウェア102のサービスへのアクセス手段を提供する。CANインターフェースソフトウェア103を介することで、上位のモジュール（COMソフトウェア105など）は複数の異なるCANハードウェアデバイスの管理に単一のインターフェースを用いることが可能となる。

[0043] PDUルータソフトウェア104は、モジュール間での（I）PDUユニット（Interaction Layer Protocol Data Unit：中間層プロトコルデータユニット）ルーティングサービスを提供するモジュールである。異なる通信プロトコル間の通信において、PDUデータを別のプロトコル用インターフェースソフトウェアに引き渡す処理を行う。自動車制御システムにおいては、異なる通信プロトコルを用いる通信ネットワークを越えたデータの送受信を行うゲートウェイ動作と呼ばれる処理を行う必要がある場合が存在する。AUTOSAR（登録商標）では、PDUルータソフトウェア104がゲートウェイ機能を提供している。

[0044] COMソフトウェア105は、上位のランタイム環境（RTEソフトウェア106）に対して、PDUルータソフトウェア104から送信されるPDUに対し適切な処置（例えばビッグエンディアンとリトルエンディアンの変換など）を行った後、PDU由来のデータ（例えばPDUに内包されるシグナル）を用いるランタイム環境に送信可能なようにするモジュールである。パソコンなどの技術分野では、送られたデータは全て上位層に送られることが多い。しかし自動車の場合は、このCOMソフトウェア105で送られたデータに変化があったか否かを判定し、変化があったときもしくは、PDUに内包されるシグナルの値が特定の条件を満たしている場合のみ図示しない上位のランタイム環境（COMソフトウェア105経由で信号の送受信を行

うソフトウェア)に送信される処理を行う。これによりCPU 2-1の負荷を減少させる。

[0045] すなわち、PDUルータソフトウェア104から転送されたPDUに対し、適切な処理を行い(たとえばエンディアン変換処理)、内包するシグナルを抽出し、更新情報であるアップデートビットを検査し、更新されているシグナルであれば、該当シグナルにあらかじめ設定されているフィルタリング条件に照らし合わせ、比較処理を行い、条件を満たしていれば、上位ソフトウェア層であるRTEソフトウェア106へと送信する。更新されていないシグナルについては、COMソフトウェア105は以降の処理を行わないことにより、CPU 2-1の負荷を軽減している。

[0046] RTEソフトウェア106はオペレーティングシステム、バス技術、通信レイヤ、ハードウェア抽象化レイヤ、API、標準ライブラリ関数などの構成要素を結び付け、サービスおよび基盤となるハードウェアへの標準ソフトウェア・インターフェースを管理するソフトウェアである。RTEソフトウェア106の更に上位に図示しないアプリケーションプログラムが存在する。

[0047] 図15はCOM-ACC 2-5の概略構成を示す。COM-ACC 2-5は、フィルタ回路、マスク生成回路、受信フラグデータ圧縮回路、シグナル挿入抽出回路、アップデートビット生成検出回路からなる。実線は受信したPDUからのシグナル抽出処理、破線は送信用の新たなPDUの生成処理でのデータの流れを示す。

[0048] それぞれの回路で処理されたデータは、PDUを下位へ送信する場合のモードを決定する送信モードコンディションTMC(Transmission Mode Condition)、更新されたシグナルを示す受信フラグ、送信用に新たに生成されたPDUに保存する送信PDUバッファ、更新されたシグナルを保存するシグナルバッファ2-6へ格納される。

[0049] COM-ACC 2-5は、受信したPDUに含まれるシグナルの抽出処理においては、COMソフトウェア105からPDUと当該PDUを弁別する

ためのPDUIDが供給され、COMソフトウェア105からの起動信号によりフィルタリング処理を開始する。COM-ACC2-5は、フィルタリング処理によりCOMソフトウェア105において処理すべきシグナルをシグナルバッファに格納した後、CPU2-1に割り込み等もしくは受信フラグによる通知を行う。

[0050] また、COM-ACC2-5は他のECUに送信するためのPDUを生成する。このPDU生成処理においては、COMソフトウェア105から他のECUに送信するためのシグナルと生成するPDUを示すPDUIDを供給され、COMソフトウェア105からの起動信号によりPDU生成処理を開始する。COM-ACC2-5は、生成したPDUをPDU送信バッファに格納した後、CPU2-1に割り込み等による通知を行う。

[0051] フィルタ回路、マスク生成回路、受信フラグ回路、シグナル挿入抽出回路、アップデートビット生成検出回路のそれぞれの回路動作については、以下に説明する。

[0052] (シグナル抽出処理)

以下、別のECUから受信したPDUに含まれるシグナルを抽出する処理について説明する。

[0053] 本実施の形態では、このCOMソフトウェア105におけるフィルタ処理にCOM-ACC(COMアクセラレータ)2-5を用いることが特徴である。逆を言えば、PDUルータソフトウェア104よりも物理層側のプロトコルスタックはCAN用のものであってもFlexRay用のものであっても、VFBに関係なく適用することが可能となる。

[0054] 加えて、RAM2-3の記憶領域の一部として、後述するOLDデータ(図5のOLDデータ12)を保存するシグナルバッファ2-6を有する。

[0055] このCOM-ACC2-5における処理について説明する。

[0056] 図1でも述べたとおり、CANバス上を経由して送信された64ビット固定長のPDUがCOMソフトウェア105に到達する。このPDU内には1ないし64ビットの可変長データである「シグナル」が含まれている。1つ

のPDU内には、容量の許す限り、複数のシグナルを含めることが可能である。それぞれのシグナルのPDU内での位置は設定データとして各ECUが保持している。各シグナルはシグナルIDにより区別されている。

[0057] シグナルの有意の容量は「シグナルの種別」によって決定される。この「シグナルの種別」及び2以上のシグナルがある場合の「シグナルの順序」はPDU毎に定義されている。このように1又は2以上のシグナルを検出し、それぞれを抽出する処理をReception Filteringと仕様上定義されている。

[0058] 抽出したシグナルに対して、フィルタ処理等のチェックを行う。具体的には、該信号が更新されたかなどを判断の対象とする。このフィルタ処理後、条件をクリアしたシグナルはシグナルバッファへ保存される。

[0059] これらの処理を全てのシグナルに対して行い、通知すべきものについては図示しない上位のランタイム環境に通知されることになる。

[0060] しかし、この処理には問題がある。

[0061] すなわち、PDU内のシグナルの数が増えれば、その分1つのPDUの処理に時間を要することである。特に、シグナル長が1ないし64ビットの可変長であり、シグナル毎にフィルタ処理を行う場合、シグナル毎の処理設定に時間を要する。

[0062] 本発明は、このフィルタリングの遅延の解消を図るものである。そして、COM-ACC2-5はハードウェア的にこれらの処理をPDU内の前シグナルに対して一括・並行して実行する。

[0063] 本発明の実施の形態では、新たにシグナル境界情報10を定義する。これによりPDU内の全てのフィルタリングを一括して実行することが可能となる。

[0064] 図5はCOM-ACC2-5内のフィルタ回路を8ビット幅で表した概念図である。このフィルタ回路は、CPU2-1から供給されたPDUのPDU IDからフィルタリング処理を行う条件と必要な情報を取得し、PDUの64ビットに含まれる1以上のシグナルのフィルタリング処理を並列に実行

する。

- [0065] このフィルタ回路の概念図はPDUIDから取得するフィルタリング処理の条件と情報を設定するシグナル境界情報10、OLDデータ12、X/MAXデータ13、Mask/Minデータ14、フィルタアルゴリズム選択データ15、フィルタリングの処理の対象となるPDUを格納する元データ格納部11、フィルタリング処理を行う比較器16とフィルタアルゴリズム選択回路17、及びフィルタリング処理結果を格納するフィルタ演算結果出力領域18を含む。なお、本図の各データにおいては、一番左が第1ビットであり、一番右が第8ビットであるとする。
- [0066] シグナル境界情報10は、同一のPDUに含まれる各シグナルの境界点（各シグナルのMSB）に識別用のデータがセットされた64ビットのデータである。このシグナル境界情報により、シグナル毎に次のビットに接続する比較結果の制御を行っている。
- [0067] PDU中には複数のシグナルが含まれるのは既述の通りである。シグナル境界情報10では各シグナルの最上位のビットを「1」にすることで、その境界を表す。
- [0068] 元データ格納部11は、今回の処理に関わるPDUの値を格納する記憶領域である。
- [0069] OLDデータ12は前回受信処理を行って、シグナルデータを更新したPDUの値をいう。OLDデータ12の値はRAM2-3上のシグナルバッファに記録されているが、COM-ACC2-5はフィルタ処理を行う毎にRAM2-3に対応するOLDデータを読み出しにいき、一連の処理が終わりOLDデータ12の値を更新する必要があるときには、RAM上のシグナルバッファに更新値を書き込む。
- [0070] X/MAXデータ13は、フィルタ処理時に用いる比較用データである。COMソフトウェア105におけるフィルタ処理は、AUTOSAR（登録商標）仕様上8つのフィルタが規定されている。X/MAXデータ13はこの8つのフィルタの特定のフィルタで用いるデータ群である。Mask/M

i nデータ14も同様であり、この8つのフィルタの特定のフィルタで用いるデータ群である。なお、XとMAX、MaskとMinは同時に用いることがない。このため、X/MAXデータ13及びMask/Minデータ14は1つのデータ領域を共用することができる。

[0071] フィルタアルゴリズム選択データ15は上述のAUTOSAR（登録商標）仕様の8つのフィルタのいずれを用いるかを決定するデータである。フィルタアルゴリズム選択回路17は、このフィルタアルゴリズム選択データ15を用いて動作させるフィルタアルゴリズムを選択する回路である。

[0072] このフィルタアルゴリズム選択データ15はシグナル毎に設定される。また、X/MAXデータ13及びMask/Minデータ14は対応するシグナルによってデータの意味が相違することに注意する。

[0073] 比較器16は上述のOLDデータ12、X/MAXデータ13、Mask/Minデータ14及び各ビットの比較結果を隣接ビットに転送するか否かを表すシグナル境界情報10によって、元データ格納部11に格納された値をフィルタ演算するための比較器である。

[0074] フィルタ演算結果出力領域18は、上位層及び次回のフィルタ処理で利用可能なようにフィルタ処理の結果（更新用のOLDデータ）を記録するためのCOMハードウェア内の記憶領域である。

[0075] 次にこのフィルタ回路の動作について図6を用いて説明する。

[0076] 図6はシグナル抽出処理におけるフィルタ処理を説明するためのフローチャートである。

[0077] まず、CANIP2-2がCANバス経由でPDUを含むパケットを受信する。CANIP2-2は、このパケットからヘッダ、フッタを削除した後、CPU2-1にヘッダ等削除後のPDUを送信する。

[0078] 該PDUを受け取ったCPU2-1上では図4に示したプロトコルスタックを持つAUTOSAR（登録商標）基本ソフトウェアで処理を行う。すなわち、CANDライバソフトウェア102、CANインターフェースソフトウェア103、PDUルータソフトウェア104、COMソフトウェア10

5の順で処理がなされる。この一連の処理の間に、PDU及びPDUIDを抽出する。このPDUIDはPDUのデータフォーマットを決定する際の識別子となる。また本発明におけるシグナル境界情報10やアップデートフラグイネイブル情報31、アップデートフラグビット位置情報32（全て後述）を決定する際にもPDUIDが用いられる。

[0079] この抽出したPDU及びPDUIDをCPU2-1で動作中のCOMソフトウェア105がCOM-ACC2-5に対して送信する（ステップS1001）。

[0080] COM-ACC2-5は受け取ったPDUを元データ格納部11にセットする。また、ここで渡されたPDUIDを検索キーとして、OLDデータ12をRAMから読み出す（ステップS1002）。

[0081] PDUIDによって、PDUにいくつかのシグナルが含まれているか、各シグナルがどのようなフィルタ処理を行うか判断できる。そこでCOM-ACC2-5のフィルタアルゴリズム選択回路17はPDUIDを元にフィルタアルゴリズム選択データ15、シグナル境界情報10の設定を行う（ステップS1003）。さらに、フィルタアルゴリズムに合わせて、X/MAXデータ13、Mask/Minデータ14の設定を行う（ステップS1004）。

[0082] ここまで終了したところで比較器16が元データ格納部11、OLDデータ12及び他の設定に基づきフィルタ処理を行う（ステップS1005）。

[0083] このフィルタ処理の結果はフィルタ演算結果出力領域18に格納される。このフィルタ演算結果出力領域18に格納されたフィルタ処理の結果は次のフィルタ処理においてOLDデータとして使用される。従って、RAM上のシグナルバッファ2-6に存在するOLDデータを更新する（ステップS1006）。

[0084] このステップS1006でRAM上に存在するOLDデータを更新する際、後述する書き込み用の「データ更新用マスク」を用いる。

[0085] また、フィルタ演算結果出力領域18に格納されたフィルタ処理の結果は

シグナル単位でCOMソフトウェア105に戻る（ステップS1007）。

[0086] この一連の処理において、COMソフトウェア105とCOM-ACC2-5との処理の分担について図7を用いて説明する。図7はCOMソフトウェア105とCOM-ACC2-5との処理の分担についての概念図である。

[0087] まず、PDUルータソフトウェア104からCOMソフトウェア105に対して送信されるPDU及びPDUIDをCOMソフトウェア105は受信する（#1）。COMソフトウェアは、PDUに対して所定の処理を行う。処理後のPDUデータ及びPDUIDをCOMソフトウェア105はCOM-ACC2-5に対して送信する（#2（S1001））。送信後、COMソフトウェア105はCOM-ACC2-5に対して動作トリガを発行し（#3）、COM-ACC2-5の処理を開始する。

[0088] COM-ACC2-5はRAM2-3上のシグナルバッファからOLDデータを読み出し（#4（S1002））、フィルタ処理に必要なX/MAXデータ13、Mask/Minデータ14、フィルタアルゴリズム選択データ15をROM2-4から読み出し、フィルタ処理を開始する。COM-ACC2-5はフィルタ処理、アップデートビットチェック、データ更新用マスク作成、データ更新、シグナル抽出を行う。この際、アップデートビットチェックに用いるアップデートフラグビットイネイブル情報31とアップデートフラグビット位置情報32をROM2-4などから読み出す（#5）。

[0089] データ更新用マスクを作成した場合、COM-ACC2-5はRAM2-3上のシグナルバッファのOLDデータを更新する（#6（S1006））。

[0090] COM-ACC2-5はその後、抽出したシグナルデータをCOMソフトウェア105に送信する（#7（S1007））。COMソフトウェア105は受け取ったシグナルデータを上位に通知し、図示していない上位からの問い合わせに応じ上位へ送信する。なお抽出したシグナルデータの受け渡し

を、直接インターフェースを通して行うか、アドレスの番地の引渡しで行うかは設計事項である。

- [0091] 次に、マスク生成回路でのデータ更新用マスクの作成方法について図 8 を用いて説明する。図 8 は本発明に関わるデータ更新用マスク生成回路の構成を表す概念図である。本図の各データにおいても、一番左が第 1 ビットであり、一番右が第 8 ビットであるとする。
- [0092] このデータ更新用マスク生成回路は、データとして比較結果レジスタ 2 1、シグナル境界情報 1 0、書き込み用マスク 2 2、書き込み用反転マスク 2 3、元データ格納部 1 1、O L D データ 1 2、シグナルバッファ入力値 2 4 を取り扱う。
- [0093] 本図においては、図 5 と共通の入力データとして、書き込みの元になるフィルタ演算結果出力領域 1 8 のデータだけでなく、シグナル境界情報 1 0、元データ格納部の値 1 1、O L D データ 1 2 も用いられる。
- [0094] 比較結果レジスタ 2 1 はフィルタ演算結果出力領域 1 8 のデータに対してアップデートビットチェック結果（後述）を反映させた値（例えば論理積演算結果）である。このフィルタ演算結果出力領域 1 8 のデータにアップデートビットチェック結果を反映させた値を「受信フラグデータ」と規定する。
- [0095] この比較結果レジスタ 2 1 の値とシグナル境界情報 1 0 のデータを書き込み用マスク生成器 1 9 で対比する。
- [0096] 既述の通り、シグナル境界情報 1 0 では各シグナルの最上位のビットを「1」にすることで、その境界を表す。書き込み用マスク生成器 1 9 には受信フラグデータとシグナル境界情報 1 0 のデータが入力される。該書き込み用マスク生成器 1 9 はシグナル境界情報 1 0 のシグナルの「区切り」を識別すること、及び、シグナルの対応するビットにフィルタ演算結果出力領域 1 8 に「1」がセットされているとき、出力である書き込み用マスク 2 2 の該当するシグナル（すなわちフィルタ処理をパスしたシグナル）に対応するビット全てを「1」にセットすることを特徴とする。
- [0097] 書き込み用マスク生成器 1 9 は、フィルタ処理をパスしたシグナルのビッ

ト全てを「1」に、フィルタ処理をパスしなかったシグナルのビットを全て「0」として書き込み用マスク22に出力する。図8では比較結果レジスタ21の第3ビット及び第5ビットで「1」が立っている。従って、書き込み用マスク22は16進数表記で「3F」となる。

[0098] 書き込み用反転マスク23は書き込み用マスク22の出力を反転させたものである。従って、上述の例では書き込み用反転マスク23は16進数表記で「E0」となる。

[0099] その後、書き込み用マスク22と書き込み用反転マスク23を用いて、シグナルバッファ入力値生成回路20がシグナルバッファ入力値24を導出する。

[0100] シグナルバッファ入力値生成回路20は、このシグナルバッファ入力値24を生成するための論理回路である。すなわち、書き込み用マスク22と元データ格納部11の値の論理積で元データ格納部11の値のうちフィルタをパスしたシグナルデータのみを残し、他のシグナルのビットを全て「0」にする。

[0101] また、書き込み用反転マスク23とOLDデータ12の論理積でOLDデータ12のうち上書きされる部分のみを「0」にクリアし、残りのシグナルのビットはそのままの状態に維持する。

[0102] そして、書き込み用マスク22及び書き込み用反転マスク23に関する上記論理演算の結果の論理和をとることで、シグナルバッファ入力値24を生成する。このシグナルバッファ入力値24がデータ更新用マスクであり、図7のOLDデータ更新(#6)に用いる値である。すなわちRAM2-3上のシグナルバッファ2-6に、このシグナルバッファ入力値24が書き込まれることとなる。

[0103] 次に、この受信フラグデータの格納について検討する。

[0104] 既述の通り、受信フラグデータはフィルタ演算結果出力領域18のデータとアップデートビットチェック結果の論理積をとった値である。この受信フラグデータは最大64ビットのデータであり、COM-ACC2-5内で受

信フラグデータをこのままの形で取り扱くと、受信フラグデータの取り扱いが複雑となり、メモリを多く消費する。そこで、COM-ACC2-5内で必要な受信フラグデータのみを圧縮する必要がある。

[0105] この受信フラグデータの圧縮に際して、シグナル境界情報10を用いる。すなわち、シグナル境界情報10中で「1」が立っているビットのみが受信フラグデータで変化する可能性のあるビットである。逆を言えば、シグナル境界情報10中で「1」が立っているビットのみを取り扱いの対象として、他のビットを削除すれば必要な受信フラグデータのみを圧縮することが可能となる。

[0106] 図9は本発明に関わる受信フラグデータの圧縮回路の一部を表す回路図である。また図10はこの図9の圧縮回路の動作を説明するための8ビット分の概念図である。図10であげられている「10000010」という2進数のデータは8ビットの受信フラグデータである。また、この受信フラグデータとの対比の対象となるシグナル境界情報の値は「10010010」である。

[0107] 図9にあらわす回路を図10の斜線の升目の様に8段43個の階段状に構成する。このように構成することで、右端から順にシグナル境界情報の値と受信フラグデータの値のANDを取って左端を検出する。「1」が埋まっていなければ、一段上の左となりから値を取得し、「1」が埋まっていれば「1」を選択データ分の段数重ねて実行する。このようにすれば、シグナル境界情報のうち1がセットされている数だけ右詰で1が埋まる。

[0108] このビットの動きにフィルタの結果を連動させると、フィルタの結果が右詰でセットされる。

[0109] 次にCOM-ACC2-5におけるシグナル抽出処理について説明する。

[0110] 図7でも述べたとおり、フィルタ処理後COM-ACC2-5は受信フラグデータを得る。ここからシグナルデータを抽出するのがここで述べるシグナル挿入抽出回路でのシグナル抽出処理である。

[0111] 図11はCOM-ACC2-5におけるこのシグナル抽出処理の手順を表

す概念図である。

- [0112] まず、受信フラグデータ中の取得を欲するシグナルの右端をPDUの右端までシフトさせる（図11（a））。
- [0113] その後、PDUの右端から取得を欲するシグナルのデータ長だけ「1」を挿入し、他のビットを「0」としたマスクデータを作成する（図11（b））。
- [0114] この図11（a）のビットシフト後のPDUデータと、図11（b）のマスクデータの論理積を取る（図11（c））。これにより、取得を欲するシグナルのみを得ることが可能になる（図11（d））。
- [0115] これとは逆に、COMソフトウェア105は、複数のシグナルを内包するPDUデータを構成する場合がある。すなわち、COMソフトウェア105より物理層に近いモジュール（例えばPDUルータソフトウェア104）に送信する場合である。この際、COMソフトウェア105によるシグナル挿入処理については後述するPDU生成処理で説明する。
- [0116] 最後にアップデートビットのチェックについて説明する。
- [0117] このアップデートビットのチェックは、マスク作成回路の比較結果レジスタ21の値を導出するためのアップデートチェック結果を求める回路である。
- [0118] アップデートビットはPDU内の任意の位置に設定可能なシグナルの更新情報である。アップデートビットはPDUの送信元である別のECUがセットする。
- [0119] PDU内には送信側で更新したシグナル、更新しなかったシグナルが混在する場合がある。この際、更新されなかったシグナルについては受信側での利用価値が無い。従って、COMソフトウェア105ではこのような更新されていないシグナルの処理は行わない。
- [0120] 従来は、COMソフトウェア105が、反復してPDUデータからシグナルを切り出し、アップデートビットによりその変更の有無を判定し、アップデートビットが変更されていることを示すシグナルに対してフィルタ処理を

行っていた。しかし、本実施の形態では、PDU単位での一括フィルタ処理を行うため、更新されていないシグナルのフィルタ処理を行わない、という判断はできない。そこで、受信したシグナルは全てフィルタ処理し、フィルタ及びアップデートビットがセットされているシグナルのみCOMソフトウェア105で処理を行うために送信する。さらにCOMソフトウェア105で処理を行ったシグナルのOLDデータを次のフィルタ処理で用いるために更新するために、各シグナルのフィルタ結果とアップデートビットのチェック結果の論理積を取った情報でマスクを作成し、OLDデータの更新を行う。

- [0121] このアップデートビットは、シグナル毎にアップデートビットを持つか否かを設定できる。従って、全シグナル対応してアップデートビットが設定されているわけではない。また、PDU内でのシグナルデータの位置とアップデートビットの位置に関連性がないため、シグナル境界情報10では、チェックできない。このため、アップデートフラグイネイブル情報31とアップデートフラグビット位置情報32が必要となる。
- [0122] アップデートフラグイネイブル情報31は、PDU内の各シグナルのアップデートビット設定情報の有無を表す64ビットの情報である。該シグナルのシグナル境界情報10で「1」がセットされているビットに該シグナルのアップデートフラグイネイブル情報31が格納される。アップデートフラグイネイブル情報31が「1」のシグナルについては、PDU内のいずれかの位置にアップデートフラグが存在する。
- [0123] アップデートフラグビット位置情報32は、PDU内の任意の位置に設定できるアップデートビットの設定場所に関する64ビット情報である。このアップデートフラグイネイブル情報31で「1」が立っているビットであれば、対応するPDUのビットにアップデートビット自体が存在することとなる。
- [0124] まとめると次のようになる。
- [0125] アップデートフラグイネイブル情報31及びシグナル境界情報10の双方

共に「1」が立っているシグナルは、アップデートフラグを有するシグナルに関する情報であると判断する（状況1）。

[0126] 一方、シグナル境界情報10が「1」で、アップデートフラグイネイブル情報31が「0」のPDUのビットであれば、アップデートフラグを持たないシグナルに関する情報であると判断する（状況2）。

[0127] また、アップデートフラグビット位置情報32が「1」であるPDUのビットは、いずれかのシグナルのアップデートフラグビットであると認識される。

[0128] なお、いずれのアップデートフラグビットがいずれのシグナルに対応するかを判定する手段については設計事項であり、本明細書では限定しない。例えば、アップデートフラグビットを有するシグナルの並び順と、アップデートフラグビットの並び順をかならず一致させるなどの解決方法が考えられる。

[0129] これらのアップデートフラグイネイブル情報31及びアップデートフラグビット位置情報32はPDU IDを参照してCOM-ACC2-5がRAM 2-3もしくはROM 2-4から読み出すこととなる。

[0130] 本回路においては、PDU内のアップデートフラグビット位置情報32が指し示す位置のデータが「1」であればアップデートされていると判断する。これらのアップデートビット情報を集め、フィルタの結果と論理積を取るため、アップデートフラグビット位置情報32をシグナル境界情報10の位置に反映させたアップデートフラグチェック結果33を生成する（図14）。

[0131] その結果、1）フィルタ演算結果（フィルタ演算結果出力領域18に格納されている値）が「1」かつアップデートビットが設定され（状況1）、該シグナルに対応するアップデートフラグビットが「1」のとき、もしくは2）シグナル境界情報10が「1」で、アップデートフラグイネイブル情報31が「0」のとき（状況2）は更新されたシグナルであると判断する。そしてこれらをOLDデータ12の上書きの対象とし、既述の通り、データ更新

用マスク生成回路の前段階で用いられる（比較結果レジスタ 21 の説明を参照）。

[0132] 以上のような構成を取ることで、PDUデータ中に含まれるシグナル単位で処理を繰り返すことなく、PDUデータで一括してフィルタ処理、更新用マスクの作成、及び一時記憶への上書きを行うことが可能となる。

[0133] （PDU生成処理）

図 12 は PDU 生成処理における COM ソフトウェア 105 と COM-ACC 2-5 との処理の分担について説明する概念図である。

[0134] まず、COM ソフトウェア 105 が他の ECU に送信するシグナル（1 以上）を上位アプリケーションより受け取る。COM ソフトウェア 105 は受け取ったシグナルと生成すべき PDU を示す PDU ID もしくは PDU ID が関連付けされたシグナル ID を COM-ACC 2-5 に対して送信する（#8（S2001））。送信後、COM ソフトウェア 105 は COM-ACC 2-5 に対して動作トリガを発行し（#9）、COM-ACC 2-5 の処理を開始する。

[0135] COM-ACC 2-5 は RAM 2-3 上のシグナルバッファから PDU ID に対応する OLD データを読み出す（#10（S2002））。送信用の新たな PDU として、シグナル挿入抽出回路を用いて、新たに生成されたシグナルが設定されるべき場所に設定する。また、新たに生成されたシグナルのアップデートビットを、アップデートビット生成抽出回路を用いて、送信用の新たな PDU に設定する。送信用の新たな PDU のうち、新たに生成されたシグナルではないシグナルが設定される場所には RAM 2-3 から読み出した OLD データを設定すればよい。

[0136] 新たに生成された送信用の PDU に対して、他の ECU への送信タイミングを決定する等のフィルタ処理を図 5 のフィルタ回路を用いて行った後、シグナル・PDU 送信バッファにフィルタ処理後の送信用の PDU を設定して COM ソフトウェア 105 に送信する（#11（S2007））。COM ソフトウェア 105 は生成した送信用の PDU を他の ECU へ送信するために

、PDUルータソフトウェア104、CANインターフェースソフトウェア103、CANDライバソフトウェア102、CANIP2-2を経由してCANネットワークに出力する(#12)。

[0137] 図13はシグナル挿入抽出回路でのシグナル挿入処理の手順を表す概念図である。

[0138] まず、COM-ACC2-5から受信したシグナルを所定のNビットシフトレジスタへ下位詰めで格納する。その際、該受信したシグナル以外のビットには「0」をセットする。その後、そのNビットシフトレジスタを、該シグナルを挿入する所定のビット数だけ左シフトする(図13(a))。

[0139] 次に、図13(a)のシフト後のNビットシフトレジスタに対応して、シグナルデータ部が「0」で他のビットが「1」のマスクデータを作成する(図13(b))。

[0140] その後、該受信したシグナルを挿入するPDUとNビットシフトレジスタを、図13(b)のマスクデータを用いてマージする(図13(c))。

[0141] これによりNビットシフトレジスタに格納された受信したシグナルが挿入されたPDUの作成が完了する(図13(d))。

[0142] 以上、PDUからのシグナル抽出処理と送信用のPDUの生成処理とにおいて、共通する回路を用いて処理を行うことが可能となる。

[0143] (第2の実施の形態)

次に本発明の第2の実施の形態に付いて、図を用いて説明する。本実施の形態は第1の実施の形態の応用に当たる。

[0144] 図16は、本発明の第2の実施の形態にかかわるCOM-ACCの構成を表すブロック図である。本図は図15と置き換わる形で用いられる。

[0145] COM-ACC2-5は、動作制御部2-5-1、インターフェース部2-5-2、レジスタ部2-5-3、データ処理部2-5-4で構成され、データ処理部2-5-4は、フィルタ処理部2-5-5、マスク生成部2-5-6、アップデートビットチェック部2-5-7、汎用処理部2-5-8で構成されている。インターフェース部2-5-2は、図3で示した様に、C

P U 2 - 1、R A M 2 - 3、R O M 2 - 4へとそれぞれ接続されている。

[0146] 動作制御部 2 - 5 - 1 は C O M ソフトウェア 1 0 5 からの動作要求に従い、C O M - A C C 2 - 5 全体の動作を管理する。

[0147] インターフェース部 2 - 5 - 2 は C P U 2 - 1、R A M 2 - 3、R O M 2 - 4 との接続のインターフェースを提供する。

[0148] レジスタ部 2 - 5 - 3 は C O M - A C C 2 - 5 の動作に必要な設定データおよび、R A M 2 - 3 に保存されている P D U データ、シグナルデータを動作中に使用するために一時保管し、さらに C O M - A C C 2 - 5 動作中の加工途中のデータを一時保存するために用いられる。データ処理部 2 - 5 - 4 はデータの加工を行う回路であり、内部にはそれぞれの加工内容に応じた処理ブロックが内在している。

[0149] フィルタ処理部 2 - 5 - 5 は P D U データを一括してフィルタリング処理するための回路であり、C A N で設定される 6 4 ビットの P D U 中に含まれる全てのシグナルに対するフィルタリング処理を一括して処理する事が可能である。

[0150] マスク生成部 2 - 5 - 6 はフィルタ処理部 2 - 5 - 5 で処理された結果を元に抽出すべきシグナルを選択するためのマスクデータを生成する回路である。マスクデータは 6 4 ビットで構成され、P D U 内の全てのシグナルに対するマスクデータを生成する。

[0151] アップデートビットチェック部 2 - 5 - 7 は、P D U 内の任意の位置に存在するアップデートビットを全て一括でチェックし、結果をフィルタ処理結果と共にマスクデータ処理部 2 - 5 - 6 へと引き渡す。

[0152] 汎用処理部 2 - 5 - 8 では、上記各データ処理部以外のデータ加工処理を行う回路であり、データのシフト処理、論理積、論理和の処理を行う。

[0153] 各データ処理部でのデータ処理方法については、受信時の動作を用いて以下に説明する。

[0154] (受信処理)

図 1 7 は、本発明の第 2 の実施の形態にかかわる P D U データの受信処理

に係るフローチャートである。

- [0155] CAN通信にて（他の通信プロトコルも同様）CANIP2-2で受信されたCANフレーム4のうち、ヘッダ部に含まれるPDU固有のPDU-IDデータおよびPDUデータはCANDライバソフトウェア104により、CANIP2-2から転送される。CANインターフェースソフトウェア103を経由して、PDUルータソフトウェア104によりCOMソフトウェア105へ転送すべきPDUであると判断されたPDUデータは、IDデータと共にCOMソフトウェア105へ転送される。
- [0156] 本来AUTOSAR（登録商標）で規定されているCOM層での処理はすべてCOMソフトウェア105にて実行されるが、本発明であるCOM-ACC2-5を搭載したシステムにおいては、COMソフトウェア105およびCOM-ACC2-5が連携してCOM層での処理を実行する。
- [0157] COMソフトウェア105がPDUデータおよびPDU-IDを送信することでCOM-ACC2-5は起動される（ステップS201）。COM-ACC2-5では図17に示すフローに従って処理を行う。
- [0158] まずCOM-ACC2-5は設定データおよび、OLDデータを取得する（ステップS202）。設定データおよびOLDデータはPDUと共にCOMソフトウェア105より与えられたIDの値を元に、COM-ACC2-5は記憶領域であるROM2-4から設定データを、またRAM2-3よりOLDデータを取得する。
- [0159] OLDデータとは、前回受信したPDUデータである。OLDデータは、後述のフィルタリング機能および更新されなかったシグナルの前回受信時のデータを保持する目的で用いられる。
- [0160] COM-ACC205におけるPDUの処理は、PDU内に含まれるシグナルの数に寄らず一括で行われる。このため、COM-ACC205はPDU内のシグナルを識別する情報を必要としている。本実施の形態では、新たにシグナル境界情報301を定義する。アップデートビットチェック処理（ステップS203）、インバリッドチェック処理（ステップS204）、フ

ィルタリング処理（ステップS 2 0 5）、マスク作成処理（ステップS 2 0 6）のいずれの処理もこのシグナル境界情報 3 0 1 を元に処理を行うことで、P D Uを一括して処理する事を可能としている。

[0161] 図 1 8 は、シグナル境界情報 3 0 1、及びP D Uデータ 3 0 2のデータ構成を表す概念図である。

[0162] シグナル境界情報 3 0 1 は、P D Uデータ 3 0 2 内の各シグナルの最上位ビット（M S B）の位置にH i g hデータ（1）を設定し、その他のビットを全てL o w（0）に設定した8バイト（6 4ビット）のデータとして定義する。

[0163] 各々のシグナルのP D Uデータ 3 0 2 内での設定位置はA U T O S A R（登録商標）C O Mの仕様に従って、シグナル毎にE C U内で静的に設定されている。この情報を元にC O M－A C C 2－5 専用に新たに生成したデータをシグナル境界情報 3 0 1 として使用する。

[0164] C O M－A C C 2－5 で受信時の最初に実行されるアップデートビットのチェック処理（ステップS 2 0 3）は、P D Uデータ 3 0 2 内のどのシグナルとも重複しないビット位置に設定される各シグナルの送信側でのデータ更新情報であり、1 がセットされていれば、送信側でデータが更新された事を示し、0 であれば、更新されていない事を示す。もし、データが更新されていなければ、C O Mソフトウェア 1 0 5 より上位層のソフトウェアでは不必要なデータである事から、以降の処理を行わず、データを破棄する処理を行う。

[0165] A U T O S A R（登録商標）によれば、アップデートビットを各シグナルに定義するか否かは任意であり、P D Uデータ 3 0 2 内の各シグナルにアップデートビットが定義されているか否かを識別する目的で、新たにアップデートビット許可情報 4 0 1 を定義する。このアップデートビット許可情報 4 0 1 はシグナル境界情報 3 0 1 を元に行っている。

[0166] 図 1 9 は、シグナル境界情報 3 0 1、アップデートビット許可情報 4 0 1、アップデートビット位置情報 4 0 2 及びアップデートビットチェック結果

情報 403 のデータ構成を表す概念図である。

- [0167] アップデートビット許可情報 401 は、各シグナルのうちアップデートビットが定義されているシグナルに対応するシグナル境界情報 301 の中で各シグナルの MSB 位置でデータ High が設定されているビットを High と設定し、アップデートビットが定義されていないシグナルの MSB 位置の値を Low に設定したデータである。また、アップデートビットの設定位置は任意であり、設定位置を示す情報として、アップデートビット位置情報 402 を新たに定義する。アップデートビット位置情報 402 は、アップデートビットの設定位置のビットを High (1) に設定し、他のビットを全て Low (0) データに設定した 64 ビットで構成されたデータである。
- [0168] アップデートビットのチェック処理 (ステップ S203) は、アップデートビット許可情報 401 およびアップデートビット位置情報 402 を用いる。COM-ACC2-5 はアップデートビット位置情報 402 が示す PDU データ 302 内のビット状態をチェックする。もしアップデートビット許可情報 401 の対応するシグナルの情報が High に設定されていれば、COM-ACC2-5 は同一位置にアップデートビットの状態を反映させたアップデートビットチェック結果情報 403 を生成する。
- [0169] このため、アップデートビット許可情報 401 に設定されている High データの数は、アップデートビット位置情報 402 に設定されている High と同一である必要がある。
- [0170] アップデートビット位置情報 402 に設定される High データの位置と、アップデートビット許可情報 401 に設定されている High データの位置が同一ではない。このため、一括処理を行う上で、アップデートビット位置情報を元に PDU データ 302 のビット状態をチェックしたアップデートビットチェック結果情報 403 とアップデートビット許可情報 401 で設定されているシグナル位置とのビット位置の合わせこみを実行する必要がある。元来全ての処理を COM ソフトウェア 105 で実行する場合であれば、処理はシグナル毎に PDU 内に設定されるシグナルの数だけ繰り返し処理をす

ればよく、アップデートビットチェック結果403とアップデートビット許可情報401のデータ合わせこみを必要としない。

[0171] 最終的に、PDUデータ302内のいずれかのシグナルが有効にアップデートされていれば、該当するシグナルのアップデート処理を行う（ステップS207）。その後COM-ACC2-5の処理は終了する（ステップS208）。

[0172] データ位置の合わせこみ処理に関しては、以下の様に複数の実施形態が考えられる。

[0173] （合わせこみ処理1）

上述の様にアップデートビット位置情報402はPDU内の任意の位置にあり、アップデートビット許可情報401の各シグナルのMSBポジションに設定されているイネイブル情報との関連性を持たない。このため、アップデートビット情報に対応するアップデートビットイネイブル情報がアップデートビットのビット位置より上位側のビット位置に存在するのか、下位側に存在するのかが不明なため、そのままでは位置あわせ処理が困難となる。このため、まずアップデートビット情報だけを抽出する処理が必要となる。

[0174] 図20は本発明の第2の実施の形態の（合わせこみ処理1）にかかわるアップデートビットチェック部2-5-7の模式図である。なお、本図は説明のため8ビット幅で構成されているが、実際には64ビットで構成されている。

[0175] 図20の様な階段状に構成された2つのシフト回路を用いる。図上では#1と#2で表記されている。#1はPDUデータ501（図18のPDUデータ302を8ビット対応したもの）を右詰にシフトするシフト回路である。また#2はアップデートビット位置情報部（図19のアップデートビット位置情報401を8ビット対応したもの）を右詰にシフトするシフト回路である。

[0176] まず、#2のアップデートビット位置情報402のHighデータが設定されているビットの情報を用いて、PDU内のアップデートビットデータの

みを右詰にて抽出する。これに「同期」する形で、# 1のPDUデータ部501もPDU内のアップデートビットデータのみを右詰にて抽出する。

[0177] 具体的には、# 2のアップデートビット位置情報部502はアップデートビット位置のみHighデータがセットされることとなる。このため、上から2段目以下の各ビットは、自ビットより右側の全ビットに既にHighデータがセットされている場合は、アップデートビットが右端までシフトしていると判断し、自ビットの直上のデータを取り込む。そうでない場合には、まだシフト可能であると判断し、自ビットの左上のビットデータを取り込む。

[0178] この動作を64ビット幅であれば、63段繰り返すことにより、全てのアップデートビット位置情報で設定されているHighデータが右詰状態で抽出される。

[0179] しかしながらこれでは、アップデートビット位置情報のみを抽出したに過ぎない。実際のPDU内のアップデートビット情報を抽出した事にはならない。

[0180] このアップデートビット位置情報部502の動作をそのままPDUデータ部501に対して実行する事で、アップデートビット位置情報502で設定されたポジションに対応するPDU内のビットデータのみが右詰にて抽出される。これにより全てのアップデートビット情報は、右詰状態にて抽出される。この右詰状態になったデータを右詰結果510と称する。以下の説明ではビット幅も相違するが、右詰処理後の結果は全て右詰結果510である。

[0181] これにより既述の「同期」が達成できる。

[0182] 図27は、この図20の一連の手順を実現する構成を表す回路図である。なお、本回路構成は同一構成の繰り返しとなるので、図20の# 1、# 2の一点鎖線で囲われた4ビットを対象に回路構成を表している。

[0183] 上述する64ビット幅の場合、63段処理を繰り返すことは説明した通りである。1段の処理を行う際には、64ビット分の処理単位が必要となる。

[0184] 図27では、4ビット幅3段構成となっているため、12個の処理単位が

含まれる。なお、この図では、異なる処理単位であっても「内部構成」については同じ符号を用いる。本図では、内部構成については処理単位 $U_n 6$ に符号が付いているが、他の処理単位も同じ内部構成を採るものとする。

[0185] 一つの処理単位 U_n には、一つのアンドゲート AND と 2 つのスイッチ SW 1、SW 2 が含まれる。

[0186] アンドゲート AND は、1) 同一の処理段の 1 つ下位のビットの処理単位 U_{n-1} 内に存在するアンドゲート AND が出力する値と、2) 1 つ手前の処理段の同じビットの処理単位 U_{n-2} 内に存在するスイッチ SW 1 の出力との論理積をとる回路である。

[0187] このアンドゲート AND の出力はスイッチ SW 1 の切り替え信号として用いられる。また、同一処理段の 1 つ上位のビットの処理単位 U_{n+1} 内に存在するアンドゲート AND の入力として用いられる。

[0188] このように構成することで、アンドゲート AND は、同一の処理段の 1 つ下位のビットの処理単位 U_{n-1} 内に存在するアンドゲート AND が出力する値が「1」で、かつ 1 つ手前の処理段の同じビットの処理単位 U_{n-2} 内に存在するスイッチ SW 1 の出力が「1」の場合のみ「1」を出力する。

[0189] なお、各処理段の最も下位のビットの 1) は固定値として「1」が入力される。

[0190] スイッチ SW 1 はアンドゲート AND の出力により 2 つの入力端子の内の一方を出力する選択スイッチである。このスイッチ SW 1 の入力の一方は 1) 固定値「1」である。他方の入力は、2) 1 つ前段の処理の 1 つ上位のビットの処理単位 U_{n+1} 内のスイッチ SW 1 の出力、である。

[0191] アンドゲート AND の出力が「1」の時は、スイッチ SW 1 は 1) 固定値「1」を出力する。また、アンドゲート AND の出力が「0」の時は、2) 処理単位 U_{n+1} 内のスイッチ SW 1 の出力をスイッチ SW 1 の出力とする。

[0192] この出力はスイッチ SW 2 の切り替え信号として用いられる。また、一つ後段の処理段の同じビットの処理単位 U_{n+1} のアンドゲート AND の入力の 1 つとして、及び一つ後段の処理段の一つ下位のビットの処理単位 U_{n+2} の

スイッチSW1の入力の一つとして用いられる。

[0193] スイッチSW2は、一つ前段の処理段の同じビットの処理単位 U_n2 のスイッチSW2の出力と、一つ前段の処理段の一つ上位のビットの処理単位 U_n4 のスイッチSW2の出力とのいずれを選択するかを決定するスイッチである。スイッチSW2の切り替え信号としてスイッチSW1の出力を用いる。

[0194] スイッチSW1の出力が「0」のとき、一つ前段の処理段の一つ上位のビットの処理単位 U_n4 のスイッチSW2の出力を、処理単位 U_n のスイッチSW2の出力とする。またスイッチSW1の出力が「1」のとき、一つ前段の処理段の同じビットの処理単位 U_n2 のスイッチSW2の出力を、処理単位 U_n のスイッチSW2の出力とする。このスイッチSW2の出力は、一つ前段の処理段の同じビットの処理単位 U_n5 のスイッチSW2及び一つ下位のビットの処理単位 U_n6 のスイッチSW2の双方に入力される。

[0195] このような回路構成とすることで、上記のような処理が可能となる。

[0196] 次に、右詰したアップデートビットデータをアップデートビット許可情報401が設定されているビット位置へセットする必要がある。図28は、右詰したアップデートビット情報をアップデートビット許可情報503でデータHighに設定されているビット位置に、対応するアップデートビット情報を当てはめる場合の概念図である。なお、この図も便宜上16ビットで表現しているが、もっと長いビット幅としても問題は無い。

[0197] 図28の左最上段は、図20及び図27で説明した「右詰」を行った結果（右詰結果510）である。また左最下段は、アップデートビットチェック結果504である。右はアップデートビット結果許可情報503を表す。

[0198] ここでは、アップデートビット許可情報503を使い、上述の右詰したアップデートビット情報をアップデートビット許可情報503でデータHighに設定されているビット位置に、対応するアップデートビット情報を当てはめる。

[0199] 図28では、図20とは逆向きの階段状の回路を用いる。各段の各ビット

は自ビットの最上段にあるアップデートビット許可情報503のビットがHighに設定され、かつ自ビット上段のデータ列のうち自ビットの位置が最も右端の有効ビットであれば、上段のデータを取得し、そうでなければ右上の段のデータをセットする。

[0200] 右詰した結果を左にシフトさせていき、結果の右端のデータ位置が許可情報の右から一番目の位置に重なったときに位置を固定し、残りのデータをさらにシフトさせていく。以下同様にそれぞれの許可情報位置で固定し、左端までシフトさせる。

[0201] この図28のシフト回路も15段重ねる事で、16ビットデータであるPDUデータ内のアップデートビット情報を同様に16ビットで構成されるアップデートビット許可情報503の位置に合わせる事が可能となる。64ビットの場合には、63段の処理を重ねることも図20と同様である。図29は、図28の回路動作を詳細に説明したものである。

[0202] 図29の#1は、検索対象の右端を特定する右端検知回路の動作を表す図である。また#2は、アップデートビット比較結果シフト回路の動作を表す図である。

[0203] これらの右端検知回路とアップデートビット比較結果シフト回路は同期して動作する。右端検知回路の初期値は16進数表記の「0001」であり、アップデートビット比較結果シフト回路の初期値は右詰結果510である。ここでは16進数表記の「0015」を初期値とする。また許可情報504の初期値は16進数表記の「8432」である。

[0204] まず右端検知回路について説明する。

[0205] 右端検知回路で「1」が立っているビットの内最も左のビットと対応する許可情報504の対応するビットの値が「1」であるか確認する。図29A段では、16進数表記「0001」について対応を確認する。

[0206] 対応する許可情報504が「0」であれば、右端検知回路の値を1ビット上位のビットを1加算し、1段処理を進める。従って、後述するP1、P3の箇所を除けば、1段処理する毎に1ビットだけシフトする。

- [0207] 一方、「1」である場合には、さらに1段上位のビットも「1」であるかを確認する。図29#1では、B段P1及びD段P3がこの処理に当たる。P1では、許可情報504の対応するビットのみが「1」であったため2ビット分シフトしている。これに対し、P3では、アップデートビット許可情報503が2ビット連続で「1」となっているため、計3ビットシフトする。
- [0208] アップデートビット比較結果シフト回路は上述の通り、右端検知回路と同期して動作する。
- [0209] 右端検知回路で、許可情報504の所定のビットが「0」であれば、アップデートビット比較結果シフト回路にセットされた値に関わらず、現在セットされている値を1ビット上位側にシフトし、アップデートビット比較結果シフト回路の対応するビットに「0」をセットする。一方、許可情報504の所定のビットが「1」であれば、対応するビットの値はそのままに、判定の対象を1ビット上位の物とする。
- [0210] 具体的には、許可情報504の所定のビットが「1」のときの動作例としては、図29#2のB段がこれに当たる。右端検知回路でP1のあるB段では、アップデートビット比較結果シフト回路のP2が対応する処理となる。アップデートビット比較結果シフト回路の下位2ビット目が「1」であるため、この2ビット目の値に初期値の最下位ビットの値である「1」がそのまま残される。そして、3ビット目の「0」が以降の判定の対象となる。
- [0211] また、P3のあるD段では、アップデートビット比較結果シフト回路の操作の対象となるビットが2ビット分シフトすることとなる。
- [0212] 次に、この処理を行うための回路構成について説明する。
- [0213] 図30は、図29で述べた右端検出回路の構成を表すブロック図である。
- [0214] この回路も、上述の通り各ビットをサポートする多段処理の構成を採っている。64ビットの処理の場合には63段、16ビット処理する場合には15段の処理が必要となる。この回路の1段の処理には、取り扱うビット数分の処理単位UBRを用いて行う。

- [0215] 図 3 1 は処理単位 U B R の内部構成を表す回路図である。この処理単位 U B R 内には 2 つのアンドゲート AND 1、AND 2 とスイッチ回路 SW 3 を含んで構成される。この回路をマトリクス上に並べることで、右端検出回路は構成される。
- [0216] 図 3 2 は、図 2 9 で述べたアップデートビット比較結果シフト回路の構成を表すブロック図である。この回路も右端検出回路同様、処理単位 U B S をマトリクス上に並べることで構成される。
- [0217] 図 3 3 は、処理単位 U B S の内部構成を表す回路図である。この処理単位 U B S 内には 2 つのアンドゲート AND 3、AND 4 と、2 つのスイッチ回路 SW 4、SW 5 を含んで構成される。
- [0218] 以上のような処理により、生成されたアップデートビットチェック結果 5 0 4 は図 1 7 のステップ S 2 0 4 と S 2 0 5 の処理完了後、マスクデータ作成処理であるステップ S 2 0 6 において使用される。
- [0219] (合わせこみ処理 2)
- (合わせこみ処理 1) では、6 4 ビット幅の P D U データおよびアップデートビット位置情報 5 0 2、アップデートビット許可情報 5 0 3 全体を用いて右詰回路および位置合わせ回路を構成していたため、回路の段数が 6 3 段と、大きな回路規模を必要としていた。これに対し、合わせこみ処理 2 では、8 ビット単位に分割し、図 3 6 で表す 8 ビットの右詰め回路 9 0 1 が、それぞれに右詰結果 5 1 0 b を導出する。8 回の処理を行うことで、同様に 6 4 ビット幅のデータを扱う構成としている。
- [0220] 8 回に分けて右詰したアップデートビットデータを 8 ビット内に含まれるアップデートビットの数をカウントし、カウント数だけ左にずらしながら重ね合わせる事で、全アップデートビットを右詰にしたデータを作成する。
- [0221] 今度はアップデートビット許可情報 5 0 3 を 8 分割したそれぞれの 8 ビットデータ内にあるアップデートビット許可ビットの数、すなわち 8 ビットの右詰め回路 9 0 1 動作後のデータの実行長を後述する図 3 6 で表す 6 ビットカウンタ 9 0 2 がカウントする。そして、この 6 ビットカウンタに記録され

た必要ビット数だけ、前述の右詰したアップデートビットデータから切り出し、実施例 1 と同様にアップデートビットのデータをアップデートビット許可情報 5 0 3 の位置に合わせる。

[0222] 64 ビット幅の PDU データを 8 ビット単位で処理する場合、上記の処理を 8 回繰り返すこととなる。図 3 4 は、（合わせこみ処理 2）で行う 64 ビット幅の動作に付いての概念図である。

[0223] この図での「テンポラリバッファ」903 は 8 ビット単位に分割して処理したデータを格納する一次記憶メモリである。全てのビットがデータとして有効な場合、64 ビット記録する必要がある。したがってテンポラリバッファは 64 ビット記憶可能なように構成されている。

[0224] 8 ビット単位での右詰め処理を 1 度行くと、8 ビットの右詰結果 5 1 0 b が得られる。また、6 ビットカウンタ内には、実データ長が格納されている。

[0225] テンポラリバッファは、処理の基準となるビット（バッファ処理基準ビット）から 8 ビット分、右詰結果 5 1 0 b に格納されているデータをそのまま書き込む。書き込み後は、バッファ処理基準ビットを 6 ビットカウンタ内のデータ長分上位にビットシフトさせることで、次の処理に備える。

[0226] これを 8 回繰り返すことで、合わせこみ処理 1 と同じ 64 ビットの右詰結果 5 1 0 が得られる。

[0227] 逆に、アップデートビット許可情報 5 0 3 を使いアップデートビットの位置あわせを行うことも可能である。図 3 5 は 6 ビット幅 8 段加算器を用いたアップデートビットの位置あわせを行う概念図である。

[0228] 6 ビット幅 8 段加算器 9 0 4 は 6 ビットの加算器を 8 個有する加算回路である。この 6 ビット幅 8 段加算器は負の値を取ることができる。6 ビット幅 8 段加算器の初期値は「-1」である。

[0229] まず、書き込み回路 9 0 5 は、アップデートビット許可情報 5 0 3 を 8 ビット単位で分割する。そして、6 ビット幅 8 段加算器の格段の加算器は 1 段下位の加算器の出力にアップデートビット許可情報の対応するビットの値を

足すことで、許可情報のビット位置を確認する。

[0230] すなわち、アップデートビット許可情報の所定のビットが「1」であれば、6ビット幅8段加算器の該当する加算器の出力が前段の加算器の出力と相違する。

[0231] 書き込み回路905は、この相違箇所の数だけテンポラリバッファからデータを抽出し、該当する変化点ビットに対応したアップデートビットチェック結果のビットをテンポラリバッファの対応するビットの値に、他を「0」にする。

[0232] この後、テンポラリバッファバッファ処理基準ビットを検出した変化点の数だけ上位側にシフトさせることで、8ビット分の処理が終了する。

[0233] 図36は、上記の図34および図35の処理の流れをまとめたブロック構成図である。

[0234] 以上のように構成することで、合わせこみ処理1同様の結果を小さな回路規模で実現することが可能となる。

[0235] (合わせこみ処理3)

実施例1、2では、多段にシフト回路を重ね合わせる事で一括してPDU内の全シグナルに対するアップデートビットチェックを実施していた。

[0236] これに対し、合わせこみ処理3では、64to6ビットエンコーダ602および605の2つ、ビットチェック回路603、ビットセット回路607を使い、PDU内に存在するアップデートビットの状態をチェックする事を可能としている。

[0237] 図21は本実施の形態の(合わせこみ処理3)にかかわるアップデートビットチェック部2-5-7の模式図である。

[0238] まず、64to6ビットエンコーダ602、605はアップデートビット位置情報601およびアップデートビット許可情報605それぞれをエンコードする。このとき両ビットエンコーダはプライオリティー機能付きのエンコーダを利用する。このため、エンコードされた結果はアップデートビット位置情報601およびアップデートビット許可情報605それぞれの、Hi

g hにセットされたビットのうち、最も下位ビット側にセットされたビット位置を6ビット（0から64を表示）でエンコードする。

[0239] 次にビットチェック回路603は、アップデートビット位置情報601をエンコードした結果を用いて、PDUデータ604のエンコード結果が指し示すビット位置のデータをチェックする。

[0240] ビットチェック回路603は、このチェック結果はビットセット回路607に伝える。ビットセット回路607は、アップデートビット許可情報605をエンコードした結果が示すビット位置に値をセットする。この結果がビットセット結果608となり、アップデートビットが許可されたシグナルに対するアップデートビットの状態がPDU内のそれぞれのシグナルのMSBの位置にセットされる。

[0241] 同時にアップデートビット位置情報601およびアップデートビット許可情報605それぞれのエンコードされたビットはクリアされ、次に下位にあるセットされたビット位置のエンコードに備える。

[0242] （以降の処理）

アップデートビットチェック処理完了後、インバリッドチェック処理であるステップS204が実行される。

[0243] インバリッドチェック204とはAUTOSAR（登録商標）で規定されているフィルタリング機能の一つであり、送信側で何らかの異常が発生し（センサ異常など）有効な値のシグナルを送信できなくなった場合に用いられる。

[0244] ステップ204ではあらかじめ送受信ECU間双方で同一に定められたインバリッド値を用い、送信側で有効な値を送信できないシグナルの値をこのインバリッド値に置き換えて送信し、受信側では受信したシグナルの値が自己の持つインバリッド値と同一か否かを判定する。異なるインバリッド値である場合、設定に応じて初期値に置き換える処理を行うものである。

[0245] COM-ACC2-5では、値の比較にフィルタ処理部2-5-5の回路を併用しており、インバリッドチェック204の処理については、データフ

フィルタリング205の処理についての記述の後に述べる。

- [0246] データフィルタリング205はAUTOSAR（登録商標）に規定されているフィルタリングアルゴリズムである、F__Always、F__Never、F__MaskedNewWqualsX、F__MaskedNewDiffersX、F__MaskedNewDiffersMaskedOld、F__NewIsWithin、F__NewIsOutside、F__OneEveryNの8種のフィルタリング処理の事である。これらのフィルタアルゴリズムはOSEK/VDX Communicationで規定される。
- [0247] F__Alwaysは、一切のフィルタリング処理を行わないアルゴリズムである。
- [0248] F__Neverは、全てのメッセージを排除するアルゴリズムである。
- [0249] F__MaskedNewWqualsXは、マスクされたが特定の値と等しければメッセージを通過させるアルゴリズムである。
- [0250] F__MaskedNewDiffersXは、マスクされた値が特定の値と異なればメッセージを通過させるアルゴリズムである。
- [0251] F__MaskedNewDiffersMaskedOldは、マスクされた値に変化が無い場合に通過させるアルゴリズムである。
- [0252] F__NewIsWithinは、事前に設定された境界値（MAX、MIN）の範囲内にマスクされた値が含まれる場合にメッセージを通過させるアルゴリズムである。
- [0253] F__NewIsOutsideは、事前に設定された境界値（MAX、MIN）の範囲外にマスクされた値が含まれる場合にメッセージを通過させるアルゴリズムである。
- [0254] F__OneEveryNは、所定数N回発生する毎にメッセージを通過させるアルゴリズムである。
- [0255] 各シグナルに設定されているアルゴリズムに応じてフィルタリングチェックを実施する。本実施案である、COM-ACC2-5では、F__OneE

very Nについては、COMソフトウェア105にて処理を実行している。

[0256] 図22はフィルタ処理部2-5-5の構成を示すブロック図である。

[0257] 本実施の形態では、フィルタ処理部2-5-5は2ビット比較器が64回路接続された不一致検出回路1701と同様に2ビット比較器が64回路接続された不一致検出回路2702および、2ビット減算器が64回路接続された減算回路1703、同様に2ビット減算器が64回路接続された減算回路2704、それぞれの比較結果からフィルタリング結果を選別するためのフィルタアルゴリズム選択器705で構成されている。

[0258] それぞれの回路の結果は境界切断回路706を挟んで上位ビットへ接続される。演算結果はそれぞれフィルタアルゴリズム選択器705へ接続される。

[0259] これら演算回路への入力データとして、全て64ビットで構成された、シグナル境界情報301、MAX/Maskデータ、MIN/Xデータ、PDUデータ(Newデータ)、Oldデータ、フィルタアルゴリズム選択データ(64×3)を新たに定義する。

[0260] フィルタアルゴリズムの設定は、固定であり、MAX値とMaskデータは同一のアルゴリズムで使用しない。このため、同一の64ビットのデータにどちらか使用する方のデータをセットし、データ領域を共有している。同様に、MIN値とXデータもデータ領域を共有している。

[0261] 図22の様に、不一致検出回路1701にはビットマスク処理をされたPDUデータ(Newデータ)とデータXが入力され、データ比較がビット毎に実施される。値が不一致の場合もしくは、下位ビットからの結果信号pre_eq0[x]が1の場合、上位ビットに1が伝えられる。また、その結果はアルゴリズム選択回路705へ伝えられる。上位ビットへの結果の転送はシグナル境界情報[x]により境界切断回路706で、PDU内の別のシグナルが存在するビットへ伝わらないように制御している。以下同様に、不一致検出回路2702には、ビットマスク処理をされたPDUデータ(New

wデータ)とOldデータが入力され、減算回路1 703には、PDUデータ(Newデータ)とMAXデータが入力される。また、減算回路2 704には、PDUデータ(Newデータ)とMINデータが入力され、それぞれ演算される。

[0262] 図23はフィルタアルゴリズム選択器705の構成を示すブロック図である。

[0263] アルゴリズム選択器705では、各々の演算結果が入力され、各ビットのアルゴリズム選択信号により図23で表すアルゴリズム選択回路705-1の様に選択されたアルゴリズムにより、各演算回路の結果を選択している。

[0264] アルゴリズム選択器705が出力結果とシグナル境界情報301との間で論理積を取る事により、PDUデータ内の各シグナルのMSBの位置にのみフィルタ結果が反映される。この回路により、PDU内の全シグナルについて、一括してフィルタ処理を行うことができ、PDU内のシグナルの数に依存せず、一括しての処理が可能となる。

[0265] 前述のステップS204に関しては、フィルタ処理部2-5-5のうち、不一致検出回路1 701のみを使用してインバリッドチェックを行っている。インバリッドチェック時にはNewデータおよび、インバリッド値が入力値として回路に与えられる。比較結果とシグナル境界情報301との論理積を演算する事により、各シグナルのMSB位置に結果が反映されたインバリッドチェック結果を求める事が可能となる。

[0266] 図24は、マスク生成回路および生成方法を示した図である。また、図25は、各設定データ及び入力データの流れを表す図である。

[0267] アップデートビット演算結果およびデータフィルタリングの結果より、更新するシグナルのみを抽出するマスクデータを生成する。図24の様に、シグナル境界情報801とデータフィルタリングの結果とアップデートビットチェック結果(共に64ビット幅)の論理積を取ったデータ802を入力とし、フィルタリング条件に合致し、かつアップデートビットがセットされているシグナルのビット位置にのみHighデータがセットされるマスクデー

タを生成する。

[0268] 図24より、シグナル境界情報にhighがセットされている場合、データフィルタリングの結果とアップデートビットチェック結果の論理積のデータが選択される。データの更新条件に合致している場合、Highデータが選択される。次に、シグナル境界情報にLowがセットされている場合、上位ビット側のデータが選択される。このため、シグナルのMSB以外のビット位置については、MSBの状態を引き継ぐ。このため、該当シグナルが更新条件に合致していれば、シグナルの全ビット位置にHighがセットされることになる。逆に、更新条件に合致していなければ、全てLowがセットされる。状態の引継ぎは次のシグナル境界情報にHighがセットされているビット位置まで継続され、次のシグナル境界位置からは、次のシグナルの状態を新たに引き継ぐ。

[0269] これにより、シグナル境界情報を801、フィルタ演算結果&アップデートビットチェック結果802を入力とした場合、マスクデータ803が生成される。

[0270] 生成されたマスクデータのHighがセットされているビット部分のみ、前回受信時のPDUデータ（Oldデータ）上にNewデータをマージする事で、更新されたシグナルのデータのみが更新される。

[0271] この一連の処理により、PDU内に属するシグナルの数に依存する事なく、全シグナルに対し一括して、アップデートビットチェック処理に関するステップS203、インバリッドチェック処理に関するステップS204、フィルタリング処理に関するステップS205、マスクデータ生成処理に関するステップ206、シグナルアップデート処理に関するステップS207を行う手段を提供する。

[0272] 図26はマスクデータ生成時のデータフローおよびデータ構成を示す図である。

[0273] 以上、本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は前記の実施の形態に限定されるものではなく、その要旨を

逸脱しない範囲で種々変更が可能であることは言うまでもない。

産業上の利用可能性

[0274] 本発明は、AUTOSAR（登録商標）のCOMソフトウェア並びに該COMソフトウェアを有するECUの負荷軽減を目的とする。

符号の説明

[0275] 1・・・送信側ECU、2・・・受信側ECU
2-1・・・CPU、2-2・・・CANIP、2-3・・・RAM、
2-4・・・ROM、2-5・・・COM-ACC、
2-6・・・シグナルバッファ、
3・・・CANバス、4・・・CANフレーム、
6・・・PDU、10・・・シグナル境界情報、11・・・元データ格納部、
12・・・OLDデータ、13・・・X/MAXデータ、
14・・・Mask/Minデータ、15・・・フィルタアルゴリズム選択データ、
16・・・フィルタリング処理を行う比較器、
17・・・フィルタアルゴリズム選択回路、
18・・・フィルタリング処理結果を格納するフィルタ演算結果出力領域、
19・・・書き込みマスク生成器、
21・・・データとして比較結果レジスタ、22・・・書き込み用マスク、
23・・・書き込み用反転マスク、24・・・シグナルバッファ入力値、
102・・・CANDライバソフトウェア、
103・・・CANインターフェースソフトウェア、
104・・・PDUルータソフトウェア、
105・・・COMソフトウェア、106・・・RTEソフトウェア、
2-5-1・・・動作制御部、2-5-2・・・インターフェース部、

2-5-3・・・レジスタ部、2-5-4・・・データ処理部、
2-5-5・・・フィルタ処理部、2-5-6・・・マスク生成部、
2-5-7・・・アップデートビットチェック部、
2-5-8・・・汎用データ処理部、
301・・・シグナル境界情報、302・・・PDUデータ構成、
401・・・アップデートビット許可情報、
402・・・アップデートビット位置情報、
403・・・アップデートビットチェック結果、
501・・・アップデートビットチェック時PDUデータ処理部、
502・・・アップデートビットチェック時アップデートビット位置情報
処理部、
503・・・アップデートビットチェック時アップデートビット許可情報
、
504・・・アップデートビットチェック結果、
601・・・アップデートビット位置情報、
602・・・64 to 6ビットエンコーダ、
603・・・ビットチェック回路、
604・・・ビットエンコーダ使用時PDUデータ、
605・・・ビットエンコーダ使用時アップデートビット許可情報、
606・・・64 to 6ビットエンコーダ、
607・・・ビットセット回路、608・・・ビットセット結果、
701・・・不一致検出回路1、702・・・不一致検出回路2、
703・・・減算回路1、
704・・・減算回路2、705・・・フィルタアルゴリズム選択回路、
705-1・・・フィルタアルゴリズム選択回路詳細、
706・・・境界切断回路、
801・・・マスク生成時シグナル境界情報、
802・・・フィルタ演算結果&アップデートビットチェック結果、

803・・・マスクデータ、
902・・・6ビットカウンタ、
903・・・テンポラリバッファ、
904・・・6ビット幅8段加算器、
905・・・書き込み回路。

請求の範囲

[請求項1]

1又は2以上のシグナルを有するプロトコルデータユニットを含むデータフレームを受信する第1のハードウェアと、前記第1のハードウェアを接続し前記データフレームの伝達に用いられる仮想機能バスと、を有する自動車用制御システムであって、

前記第1のハードウェアは、前記仮想機能バス経由で受信した前記データフレームからプロトコルデータユニット及び前記プロトコルデータユニットのデータフォーマットを識別するプロトコルデータユニットIDを抽出するプロトコルスタックを実行し、

前記プロトコルスタックは、前記1又は2以上のシグナルの更新の有無をプロトコルデータユニットへの一度の処理で実行することを特徴とする自動車用制御システム。

[請求項2]

CPUと、仮想機能バスインターフェースと、アクセラレータと、を有し、前記仮想機能バスインターフェースを介してプロトコルデータユニットを含むデータフレームを受信する電子制御ユニットであって、

前記CPUは受信した前記データフレームから前記プロトコルデータユニット及び前記プロトコルデータユニットのデータフォーマットを識別するプロトコルデータユニットIDを抽出するプロトコルスタックを実行し、

前記プロトコルスタックによるシグナルの更新の有無の判定を行う前記CPUは、前記プロトコルデータユニット及び前記プロトコルデータユニットIDを前記アクセラレータに送信し、前記アクセラレータがシグナルの更新の有無の判定を実行することを特徴とする電子制御ユニット。

[請求項3]

請求項2に記載の電子制御ユニットにおいて、

前記アクセラレータは、

前記プロトコルデータユニットに含まれるアップデートビットの判

定と、

前記プロトコルデータユニットの値と前記プロトコルデータユニットに内在する上位モジュールで利用されるデータの最小単位であるシグナル毎に設定されたフィルタ条件に従い判定を実行し前記アップデータビットの判定の結果と、前記プロトコルデータユニットのフィルタ判定の結果から、過去に処理した前記プロトコルデータユニットの値を更新するデータ更新用マスクを作成することを特徴とする電子制御ユニット。

[請求項4]

請求項3に記載の電子制御ユニットにおいて、

前記アクセラレータは、

前記データ更新用マスクからデータの更新があったシグナルを抽出し、

前記プロトコルスタックに前記データの更新があったシグナルを出力することを特徴とする電子制御ユニット。

[請求項5]

請求項3に記載の電子制御ユニットにおいて、

前記アクセラレータは、

前記データ更新用マスクからデータの更新があったシグナルを抽出し、

前記プロトコルスタックに前記データの更新があったシグナルを出力することを特徴とする電子制御ユニット。

[請求項6]

請求項5に記載の電子制御ユニットにおいて、

前記プロトコルスタックは、受信した前記データの更新があったシグナルを用いて前記プロトコルデータユニットを更新することを特徴とする電子制御ユニット。

[請求項7]

請求項3に記載の電子制御ユニットにおいて、

前記アクセラレータは、更に所定の回数を記録する回数記録レジスタを有し、

前記データ更新用マスクによりシグナルを抽出し、

前記プロトコルスタックに前記シグナルの値が前記回数記録レジスタに記録された回数繰り返されたときに前記シグナルの値を出力することを特徴とする電子制御ユニット。

[請求項8]

請求項3に記載の電子制御ユニットにおいて、

前記アクセラレータは、更に上限値を記録する上限値レジスタ及び下限値を記録する下限値レジスタを有し、

前記データ更新用マスクによりシグナルを抽出し、

前記プロトコルスタックに前記データの更新があったシグナルの値が前記上限値レジスタに記録された値と前記下限値レジスタに記録された値の範囲内にある場合に前記シグナルの値を出力することを特徴とする電子制御ユニット。

[請求項9]

請求項3に記載の電子制御ユニットにおいて、

前記アクセラレータは、更に上限値を記録する上限値レジスタ及び下限値を記録する下限値レジスタを有し、

前記データ更新用マスクによりシグナルを抽出し、

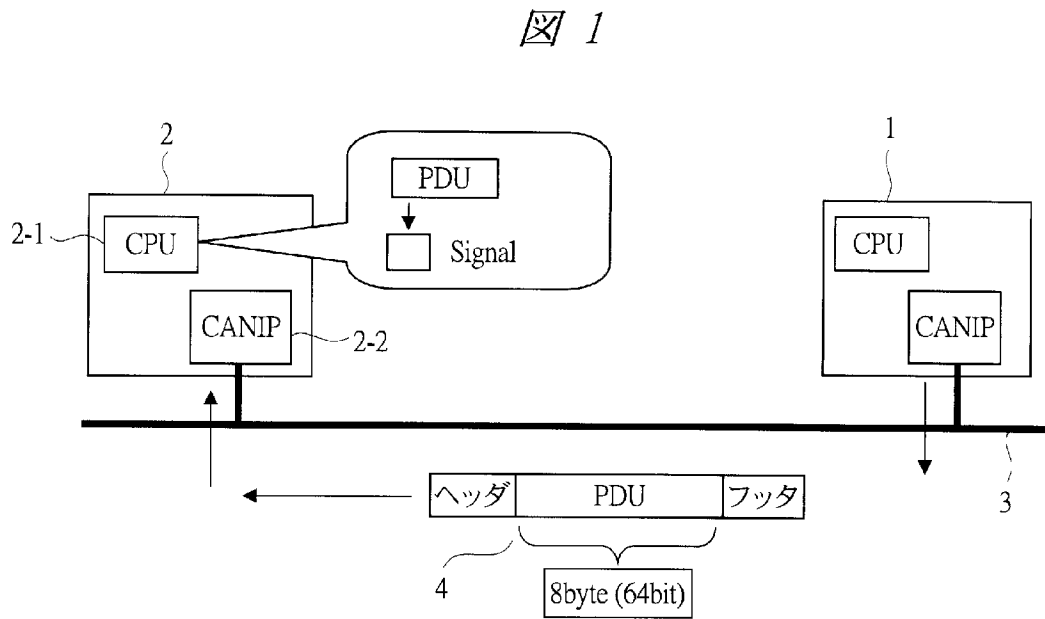
前記プロトコルスタックに前記データの更新があったシグナルの値が前記上限値レジスタに記録された値と前記下限値レジスタに記録された値の範囲外にある場合に前記シグナルの値を出力することを特徴とする電子制御ユニット。

[請求項10]

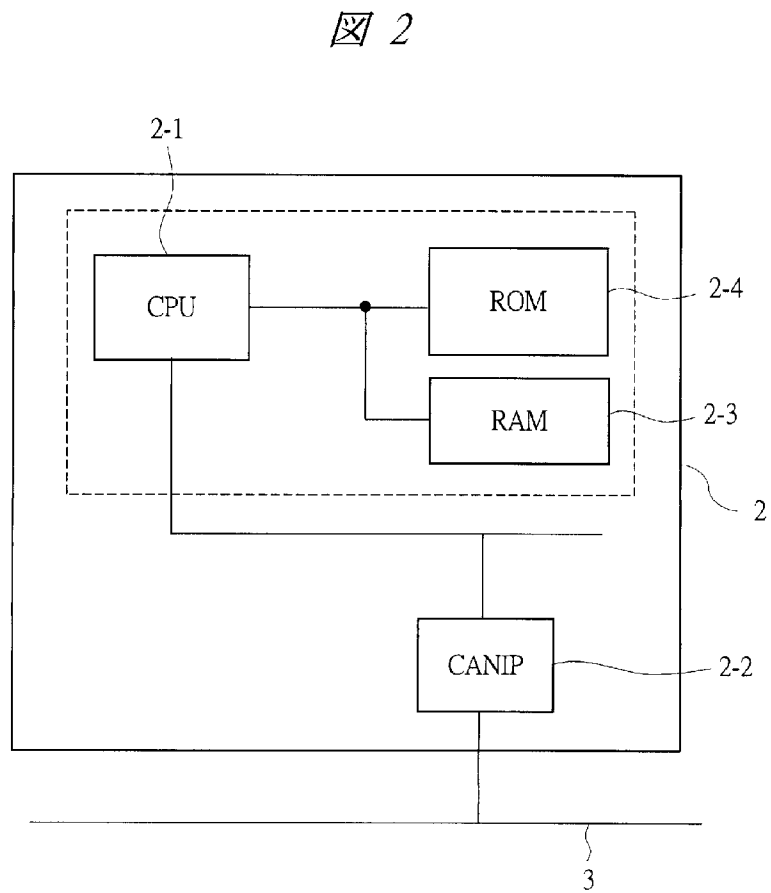
請求項2に記載の電子制御ユニットにおいて、

前記アクセラレータは通信経路を介して外部に接続される他の電子制御ユニットへ送信するためのデータフレームの生成において使用されることを特徴とする電子制御ユニット。

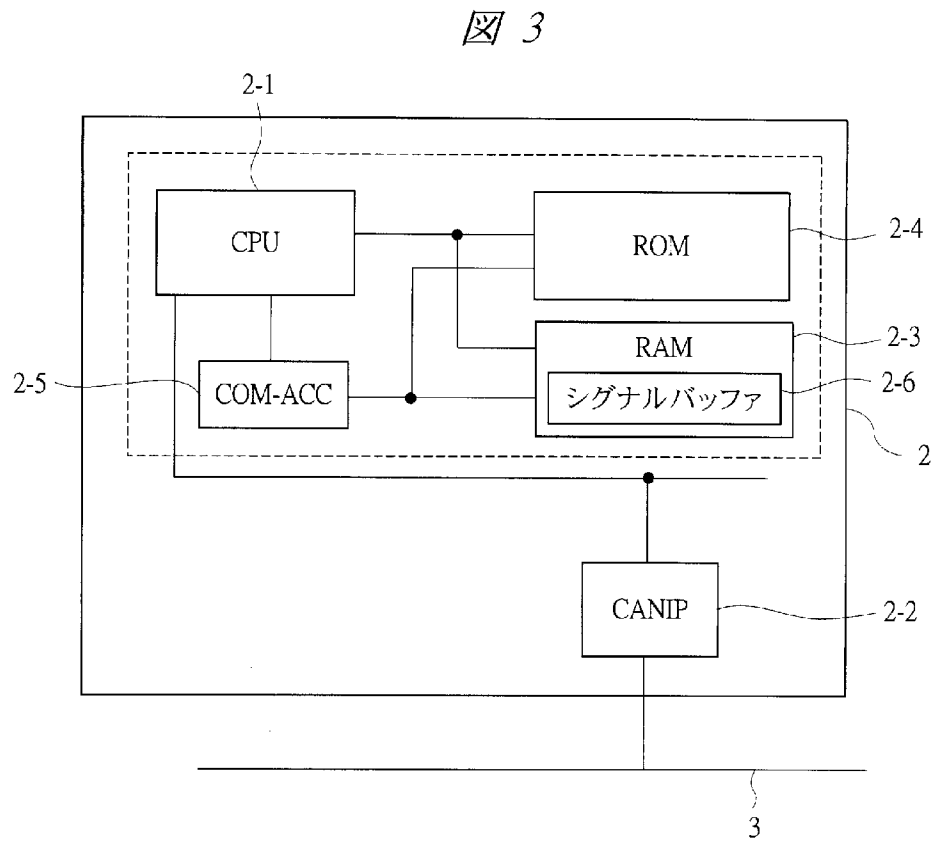
[図1]



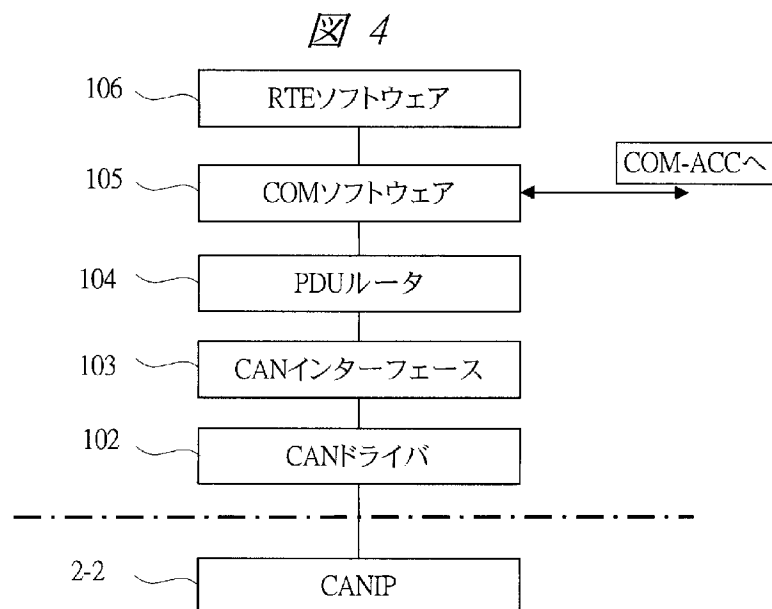
[図2]



[図3]

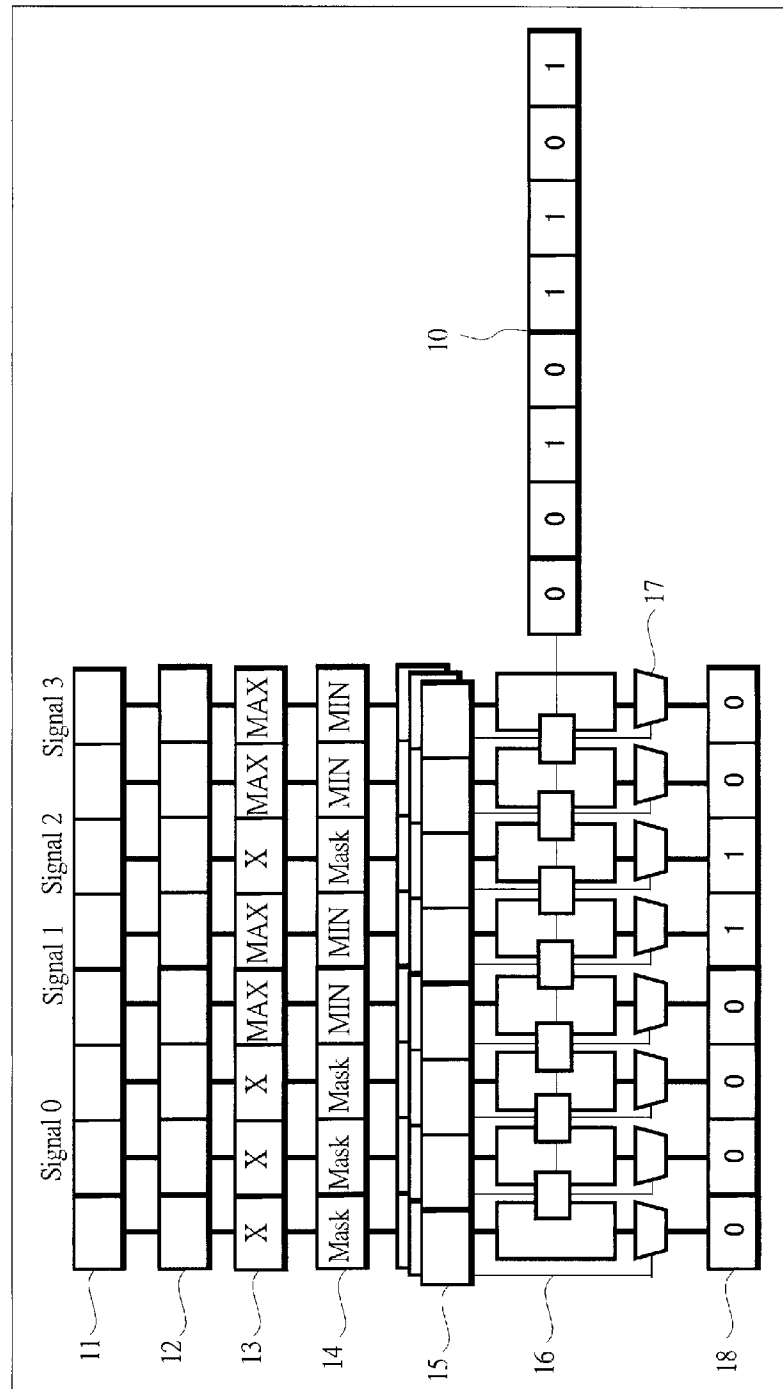


[図4]



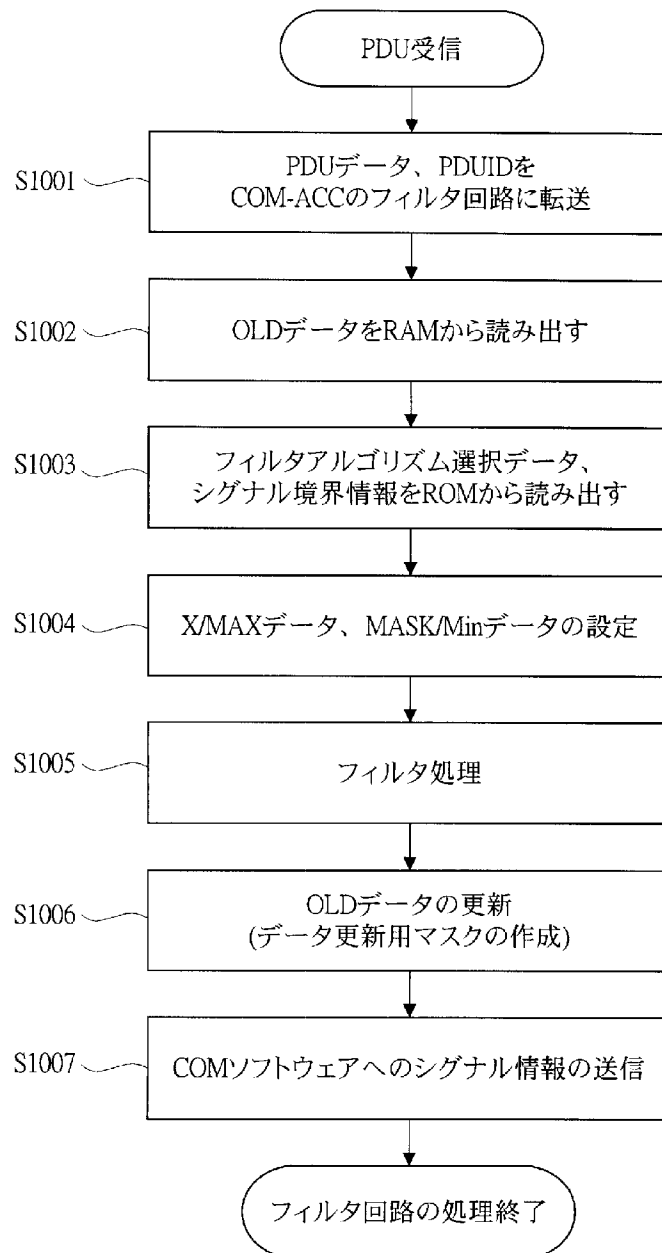
[図5]

図 5



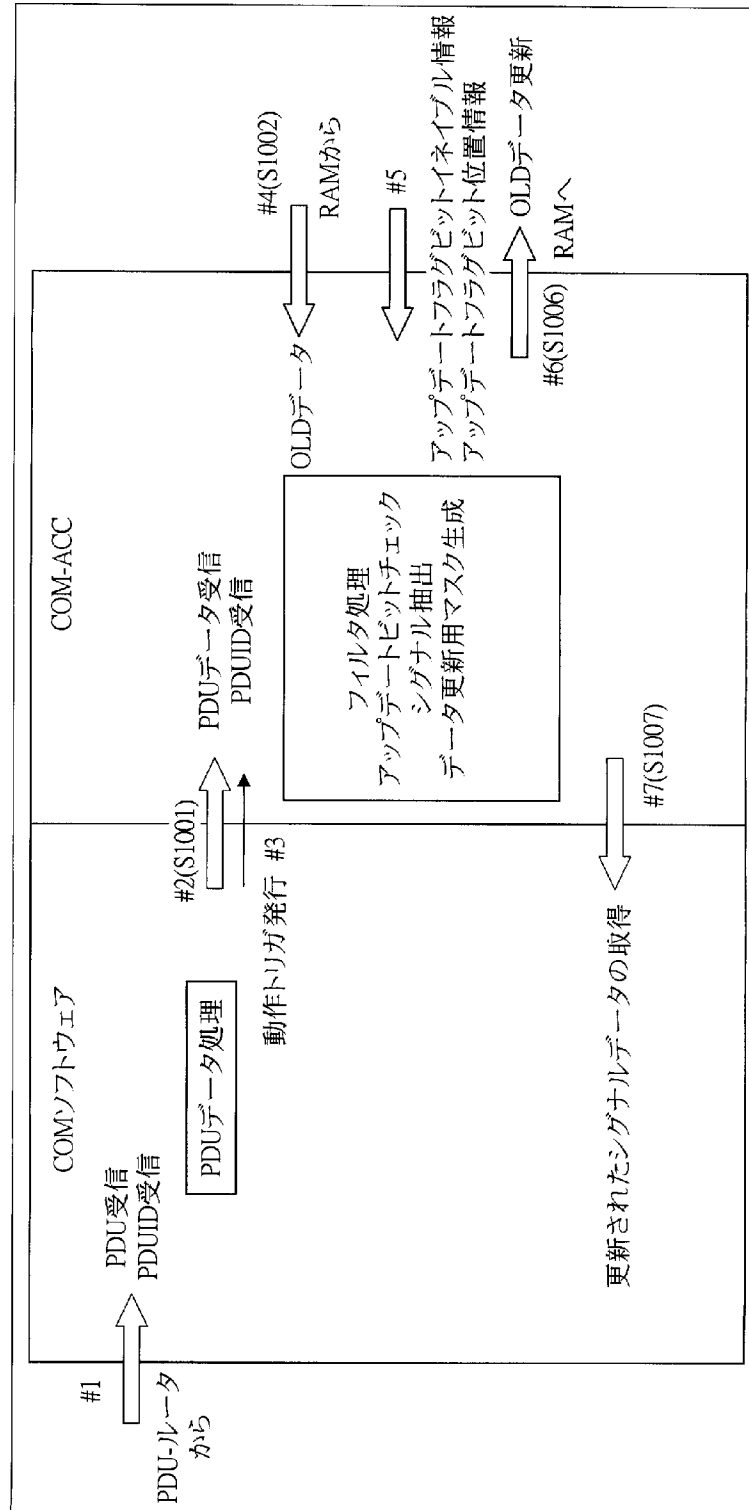
[図6]

図 6



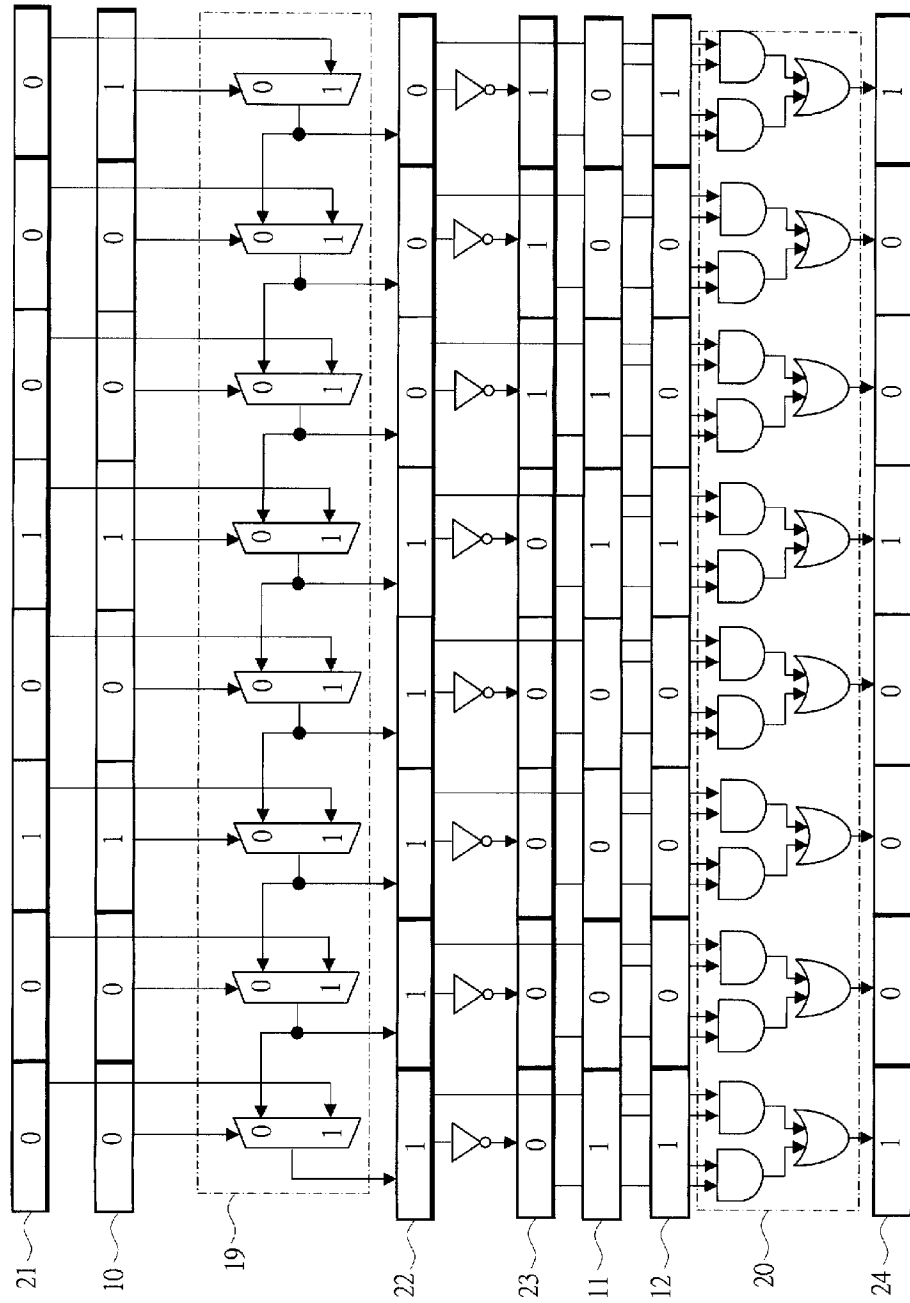
[図7]

図 7



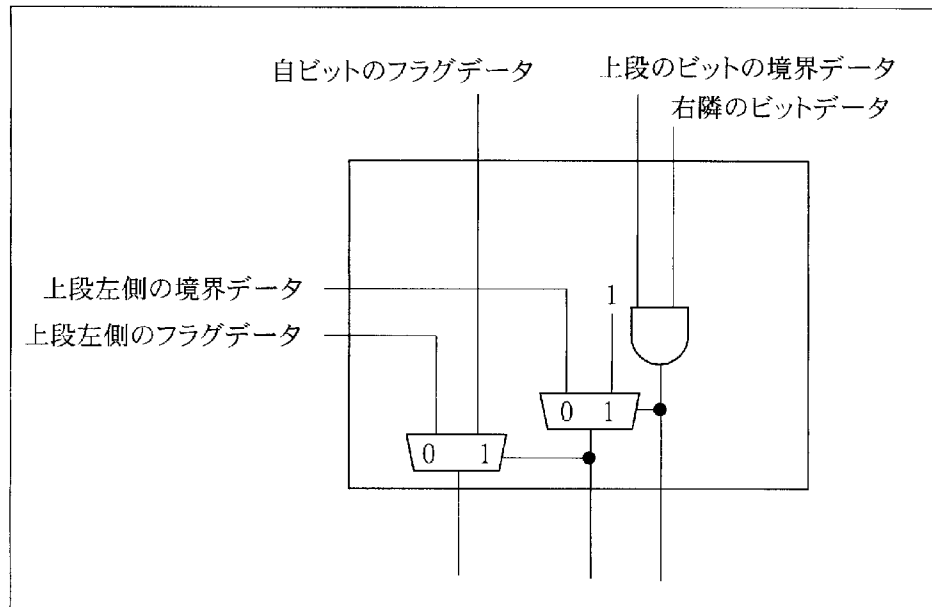
[図8]

図 8



[図9]

図 9

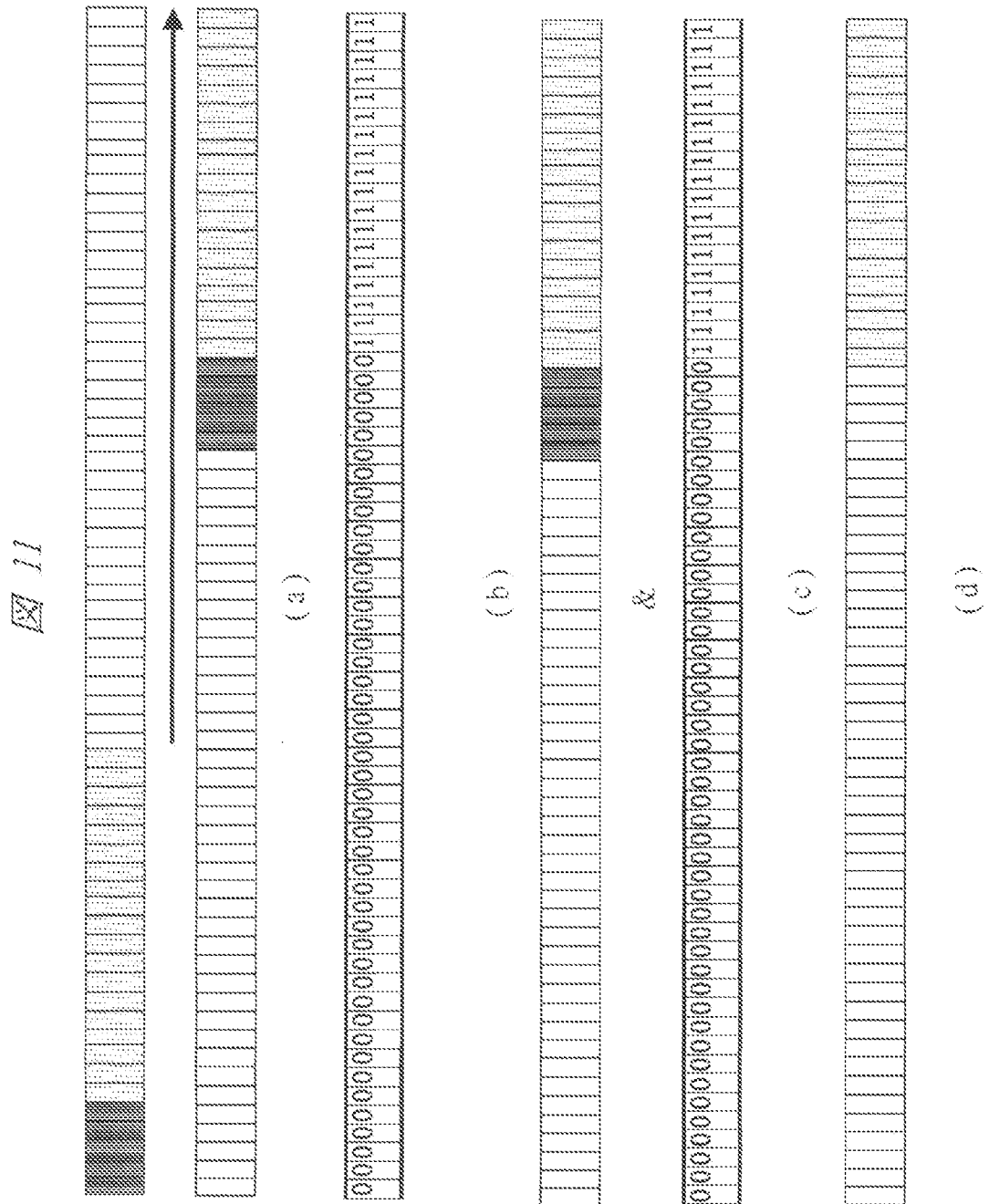


[図10]

図 10

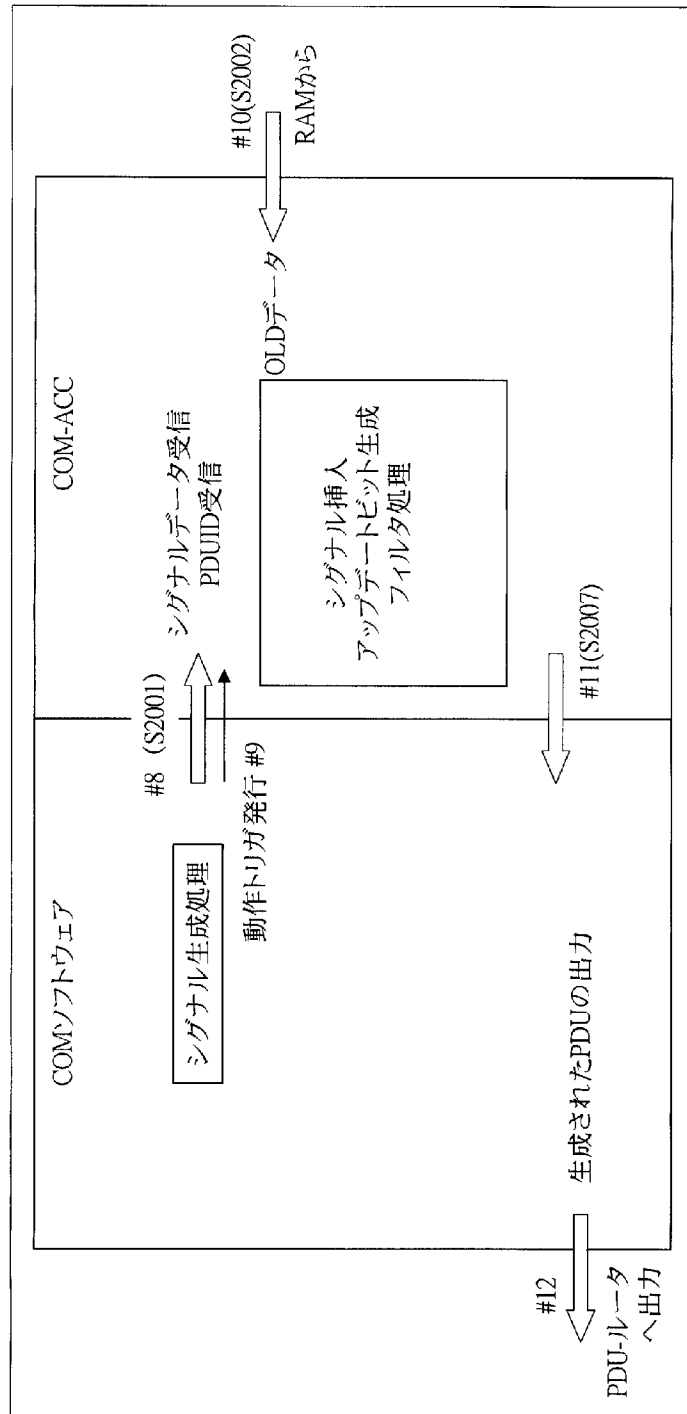
1	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1
	0	1	0	0	0	0	1
		0	1	0	0	0	1
			0	1	0	0	1
				0	1	0	1
					0	1	0
						1	0
							0
0	0	0	0	0	0	1	0
							1

[図11]



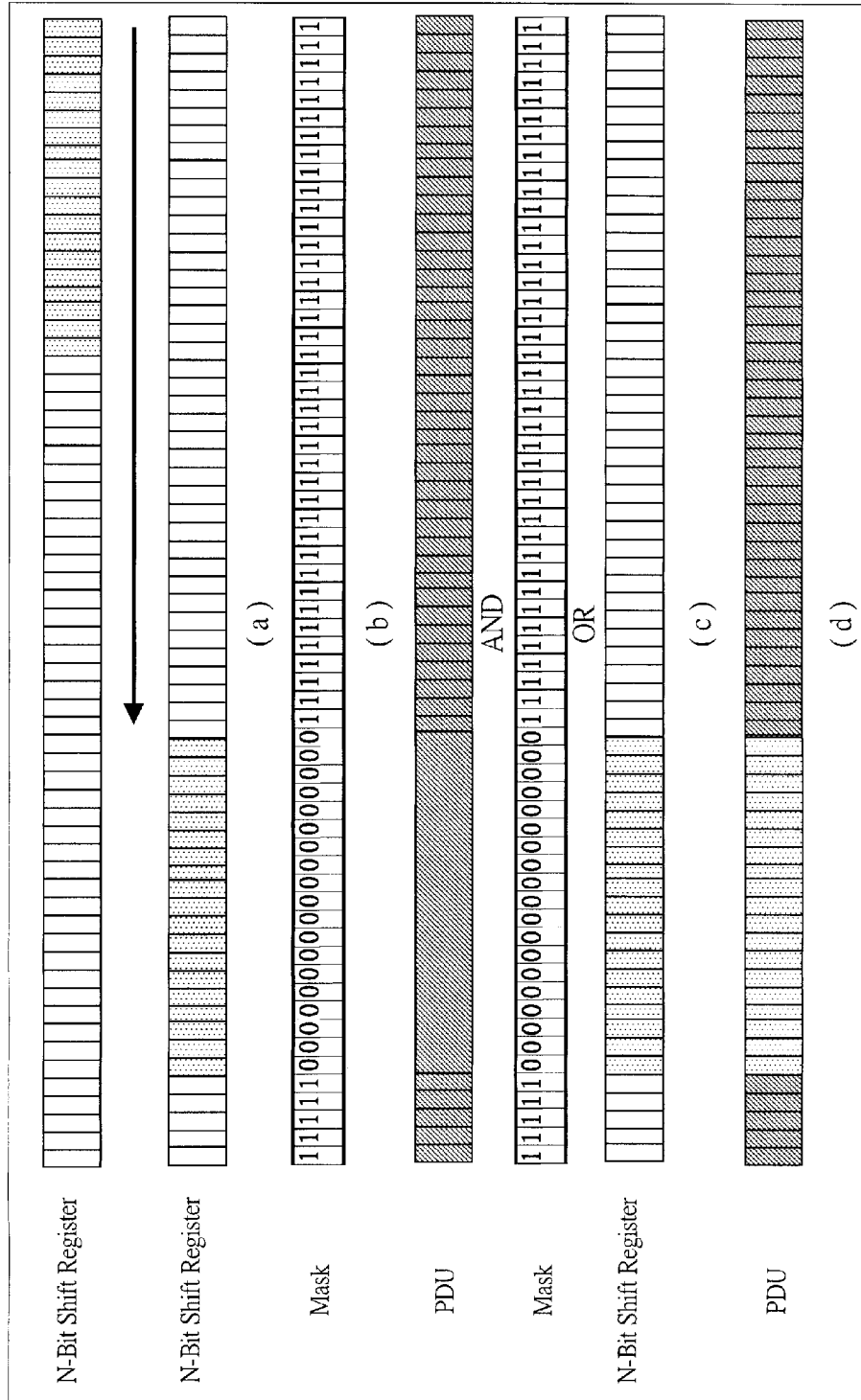
[図12]

図 12



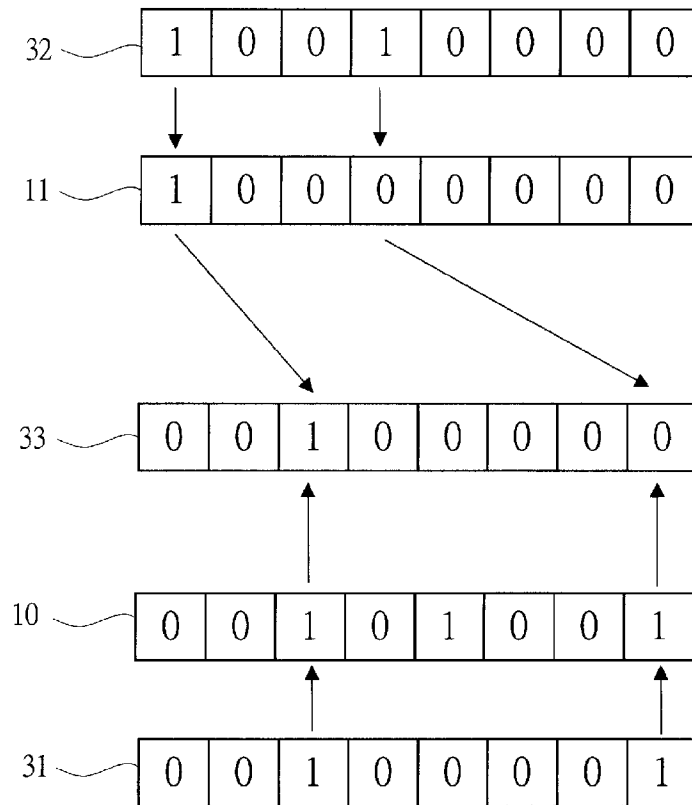
[図13]

図 13



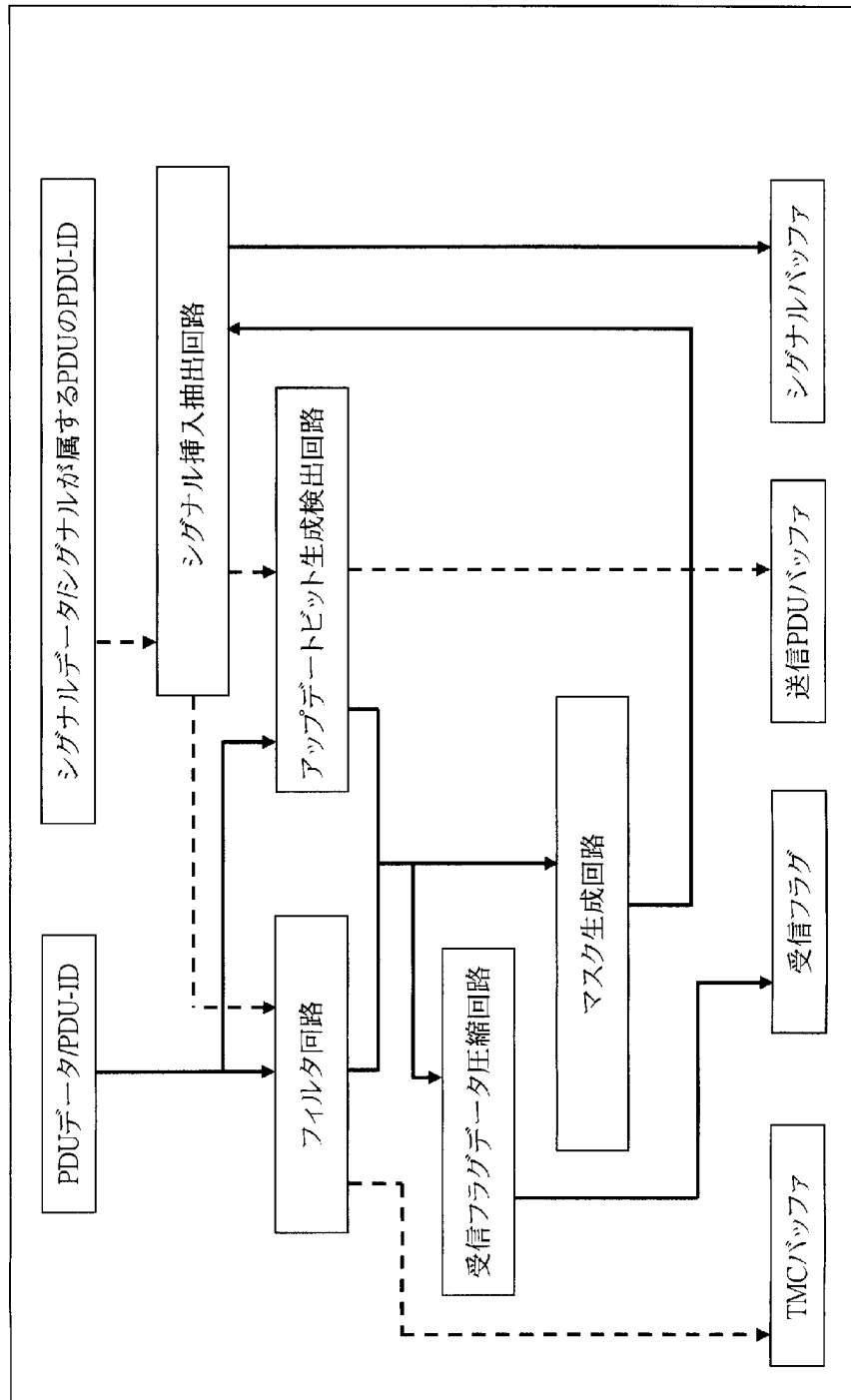
[図14]

図 14

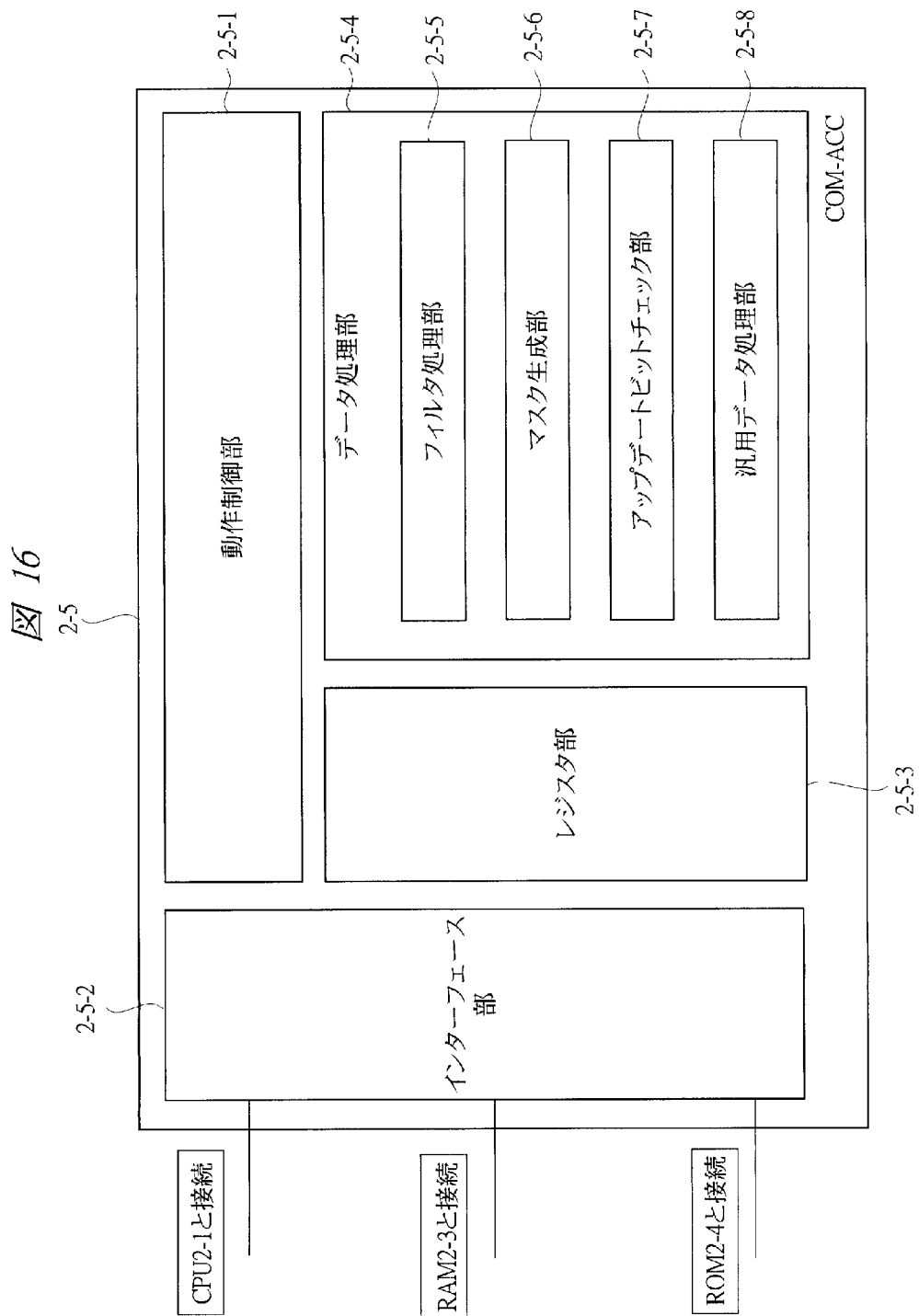


[図15]

図 15

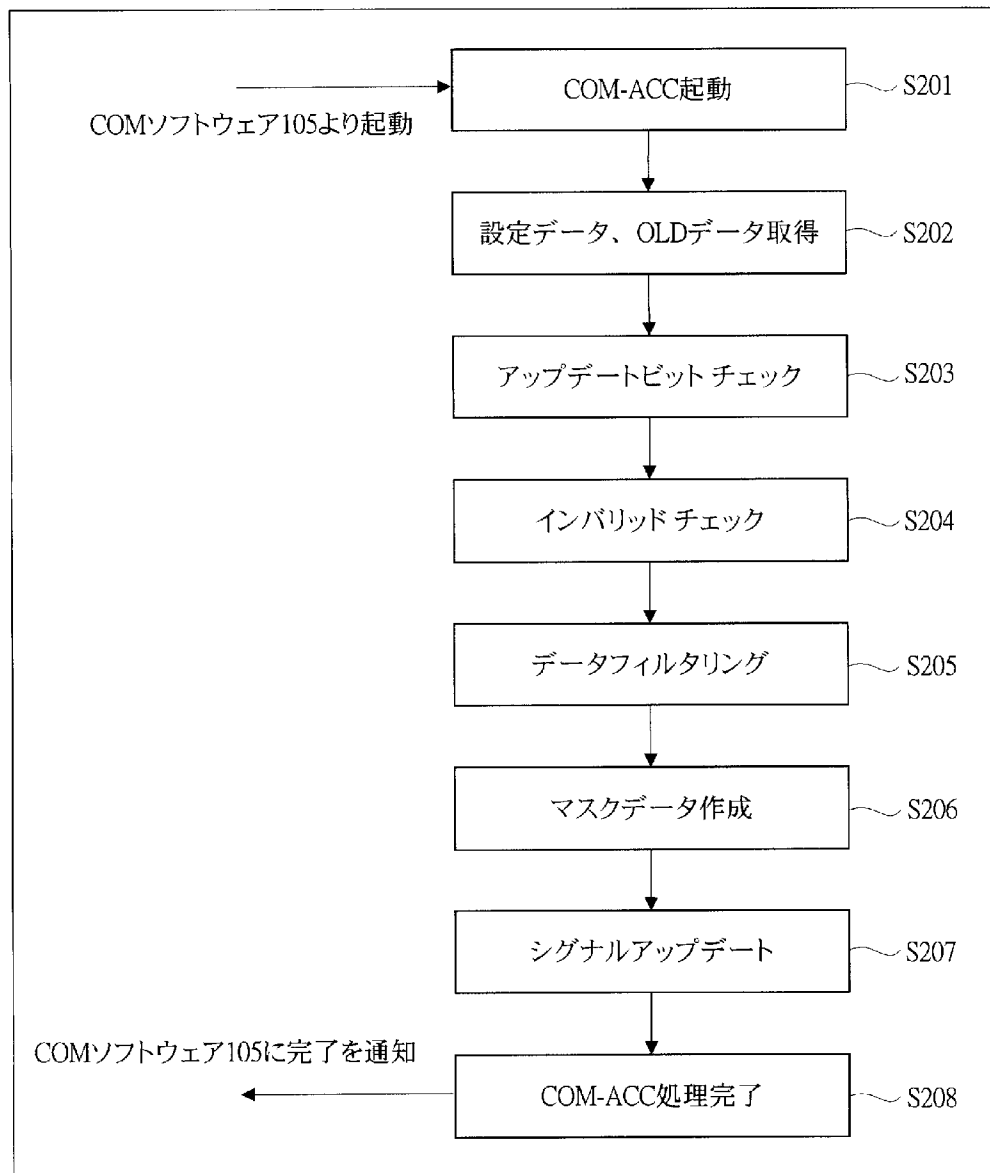


[図16]



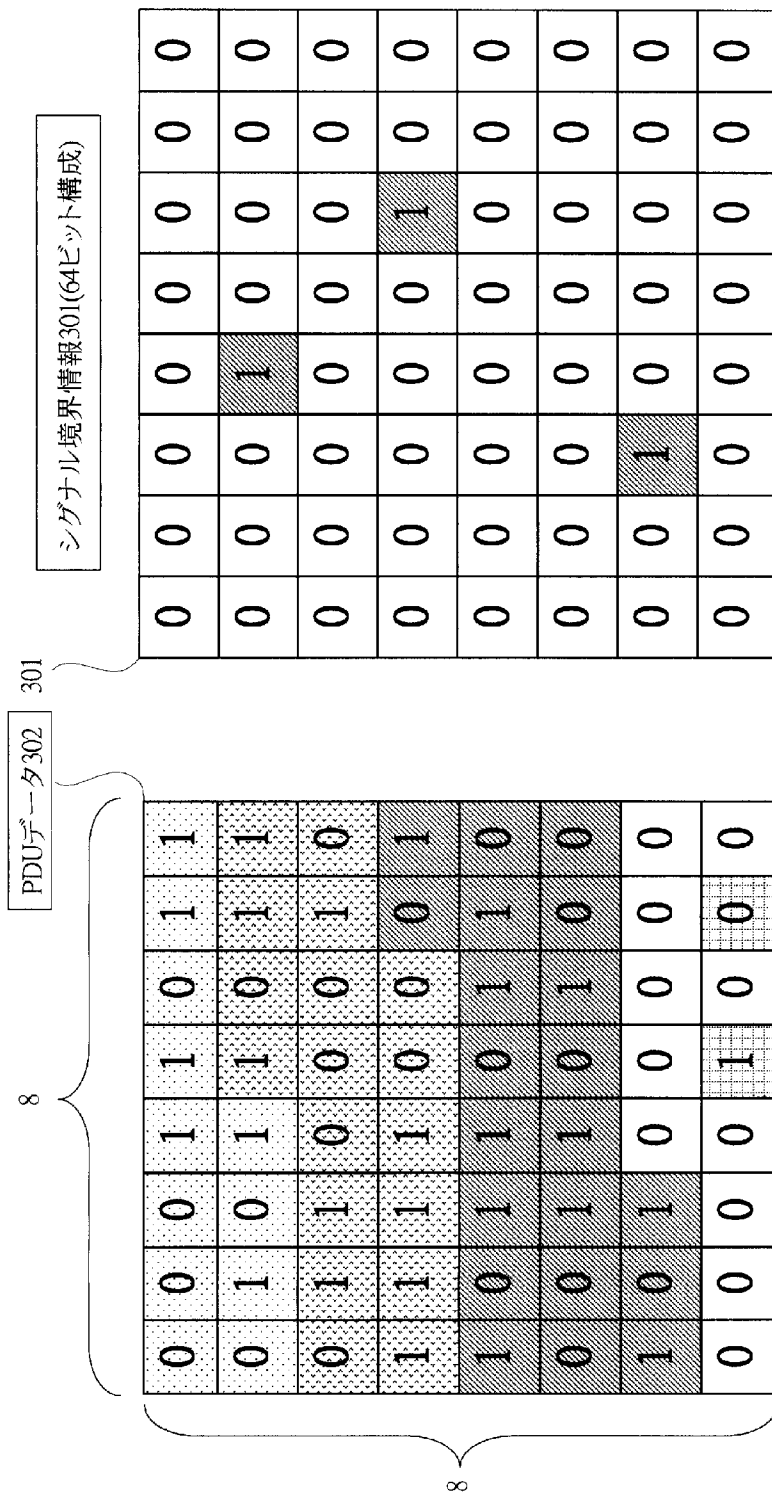
[図17]

図 17



[図18]

図 18



61 ☒

シグナル境界情報301

0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

301

アップデートビット許可情報401

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

401

設定あり → (Row 2, Col 4)

設定なし → (Row 4, Col 8)

設定あり → (Row 8, Col 5)

アップデートビット位置情報402

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

402

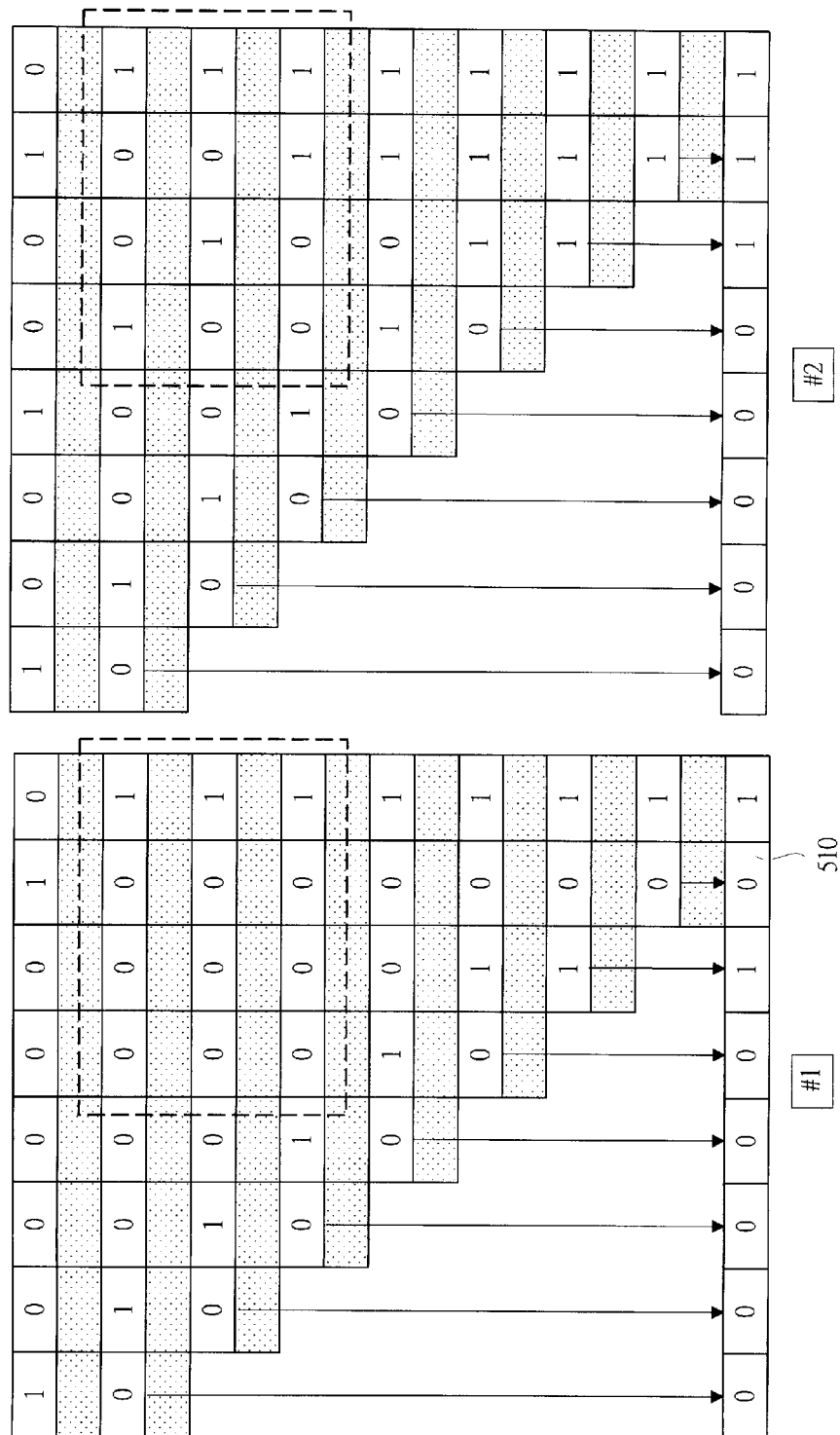
アップデートビットチェック結果403

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

403

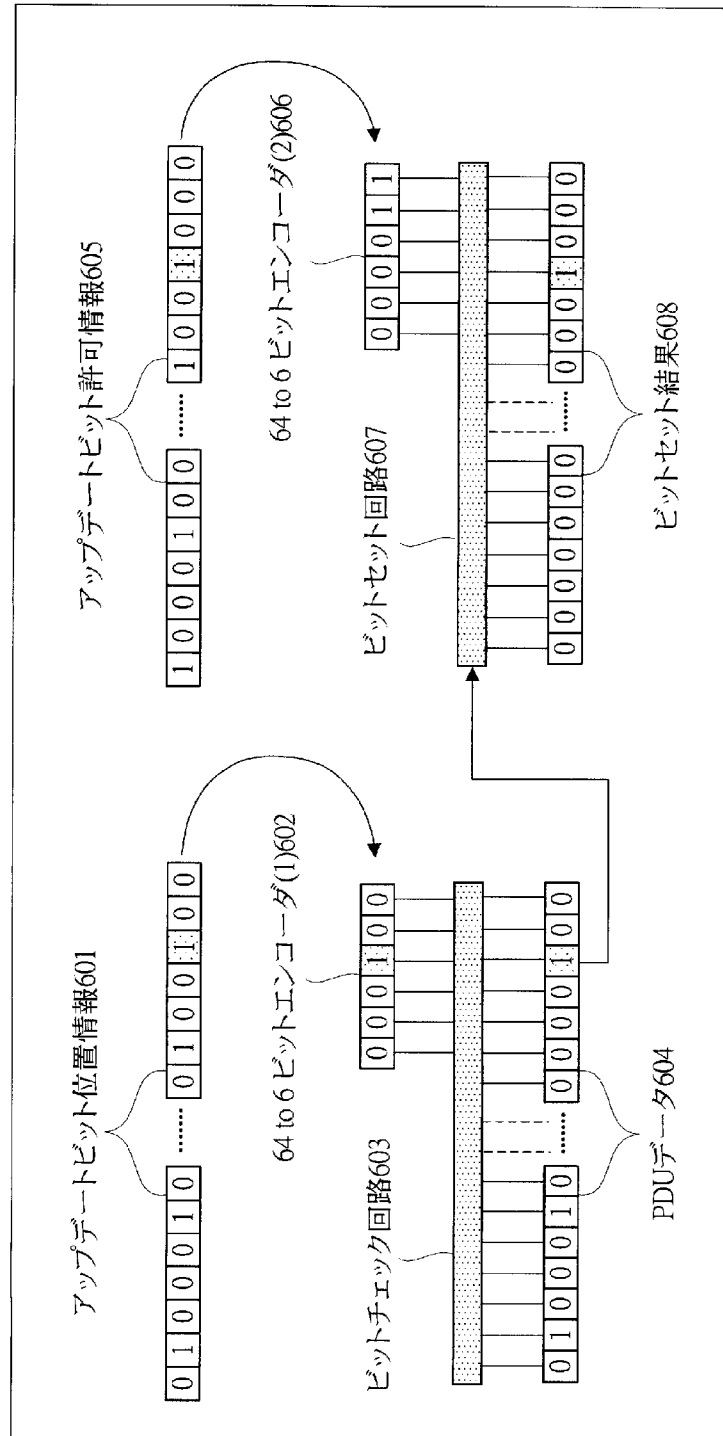
[図20]

図 20



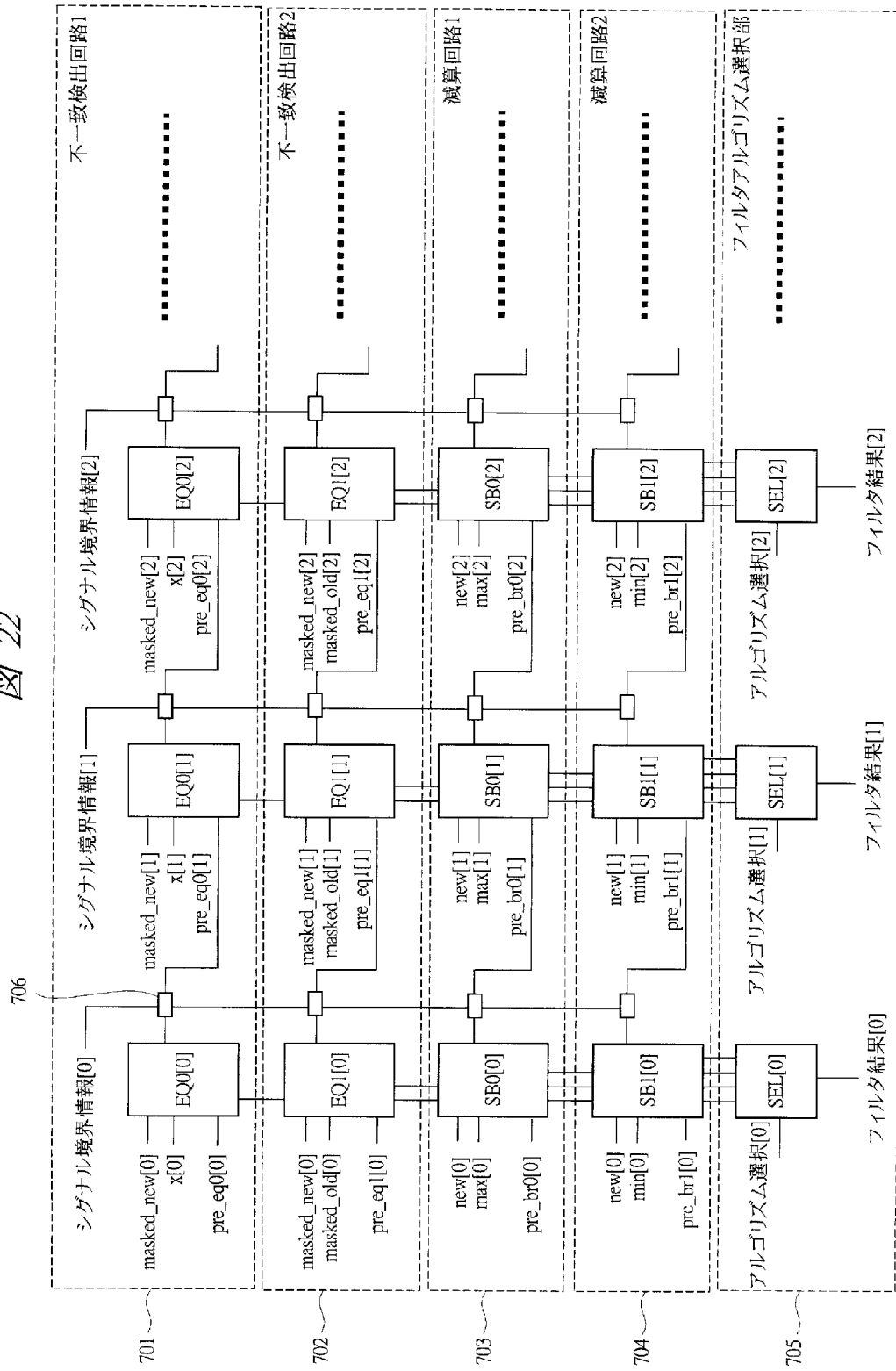
[図21]

図 21



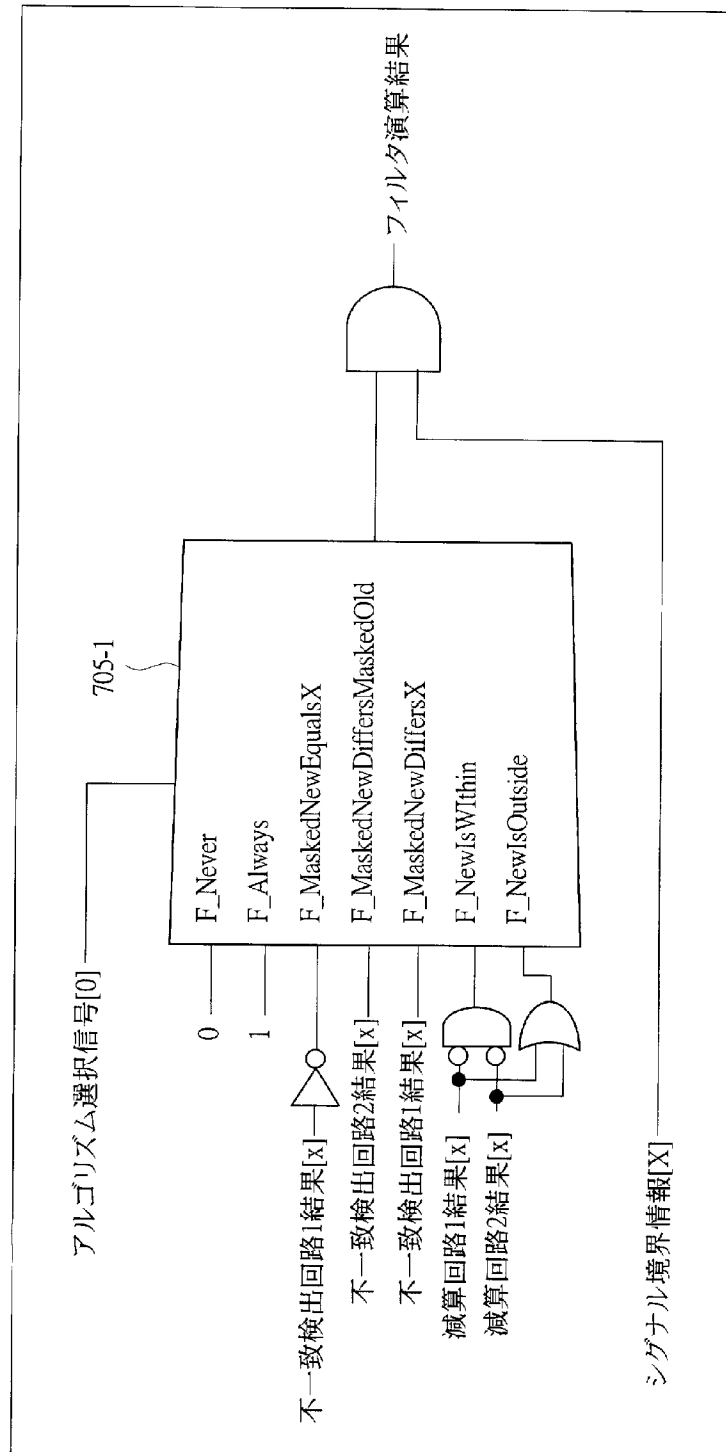
[図22]

図 22



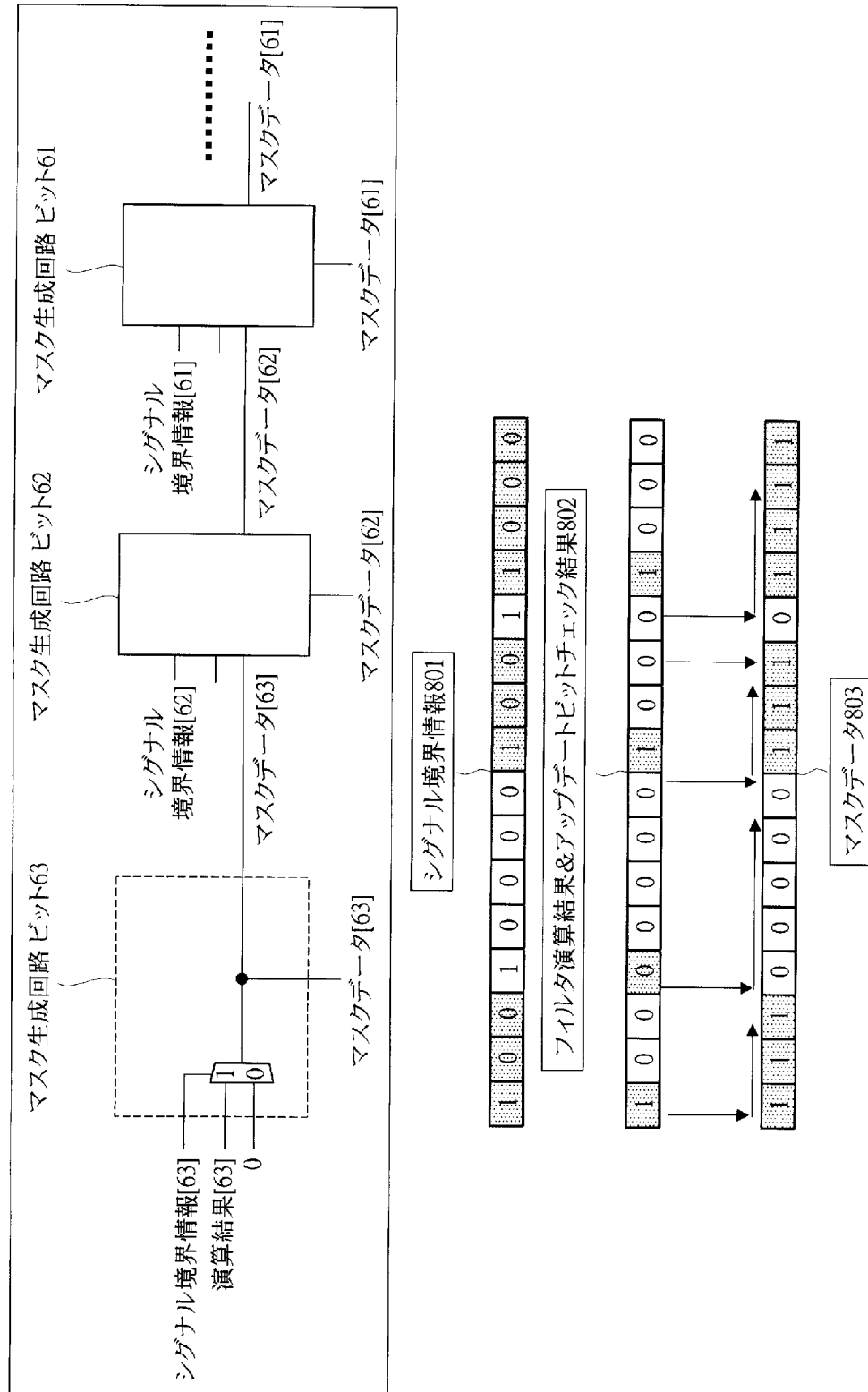
[図23]

図 23

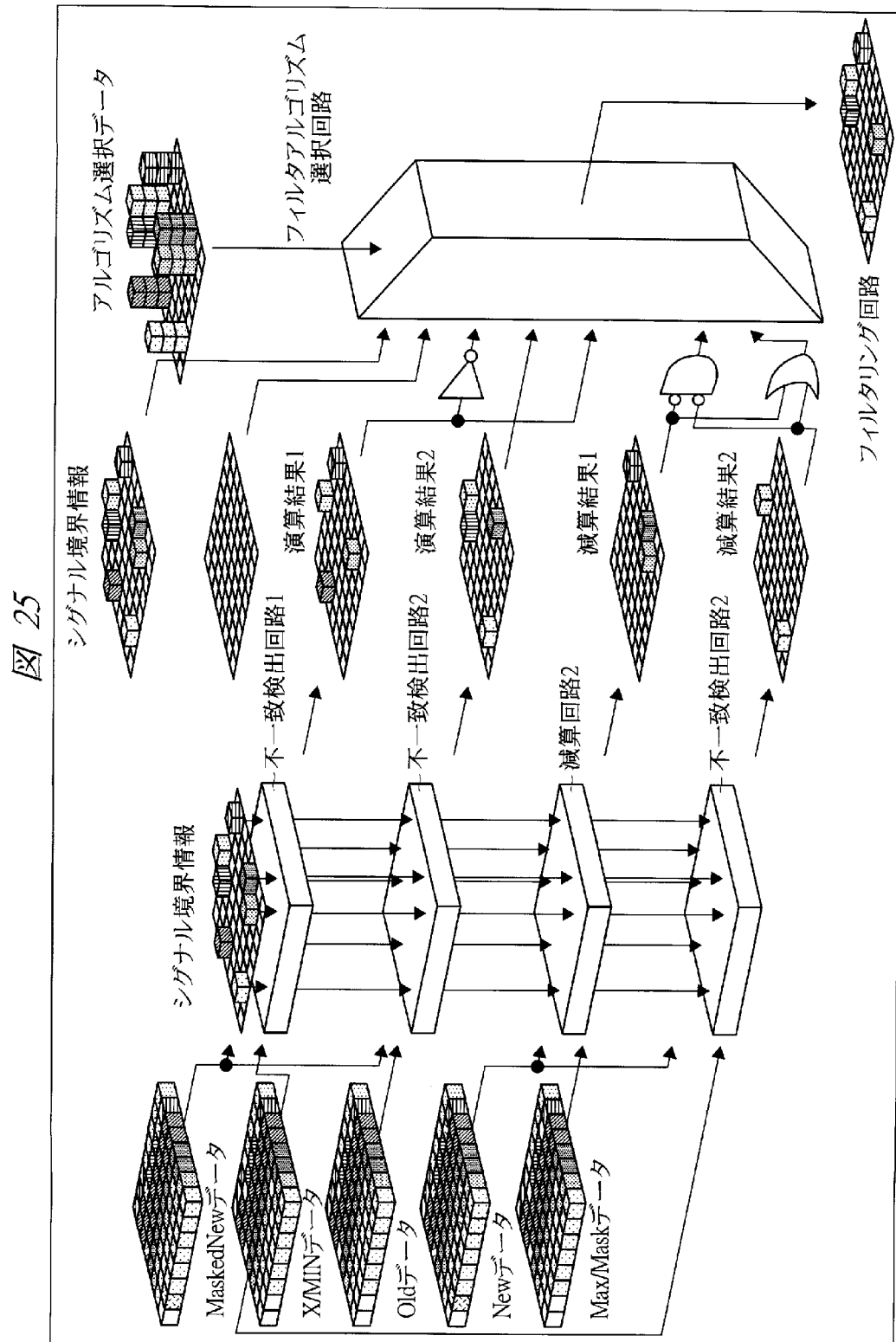


[図24]

図 24

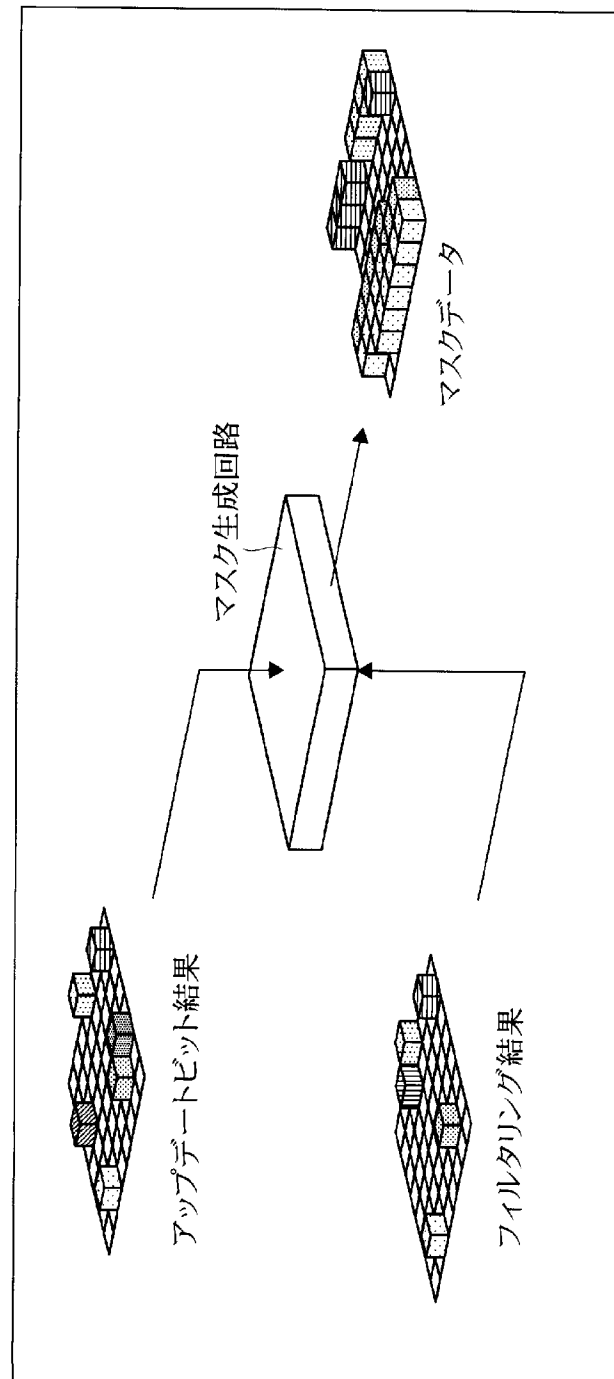


[図25]



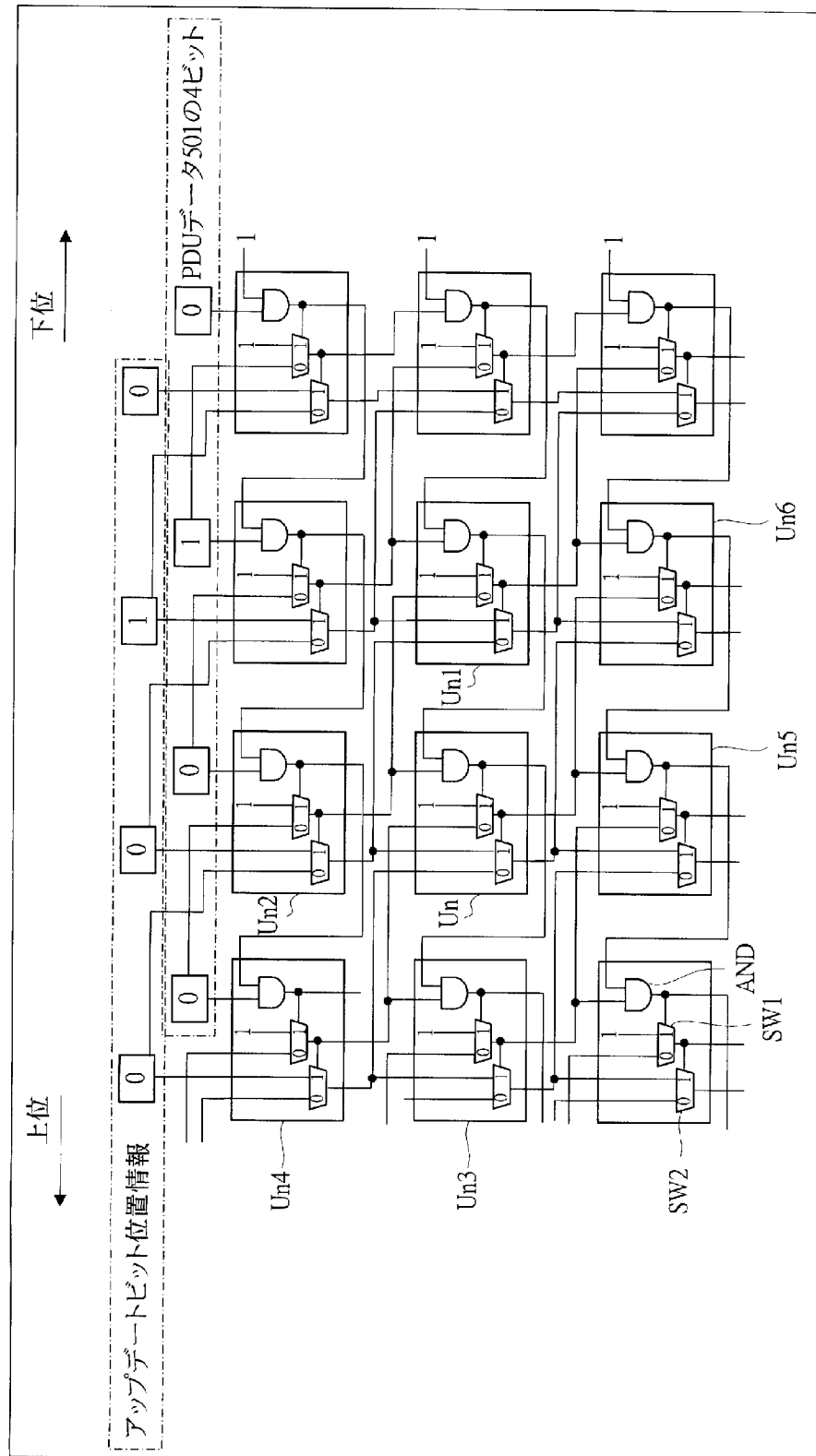
[図26]

図 26

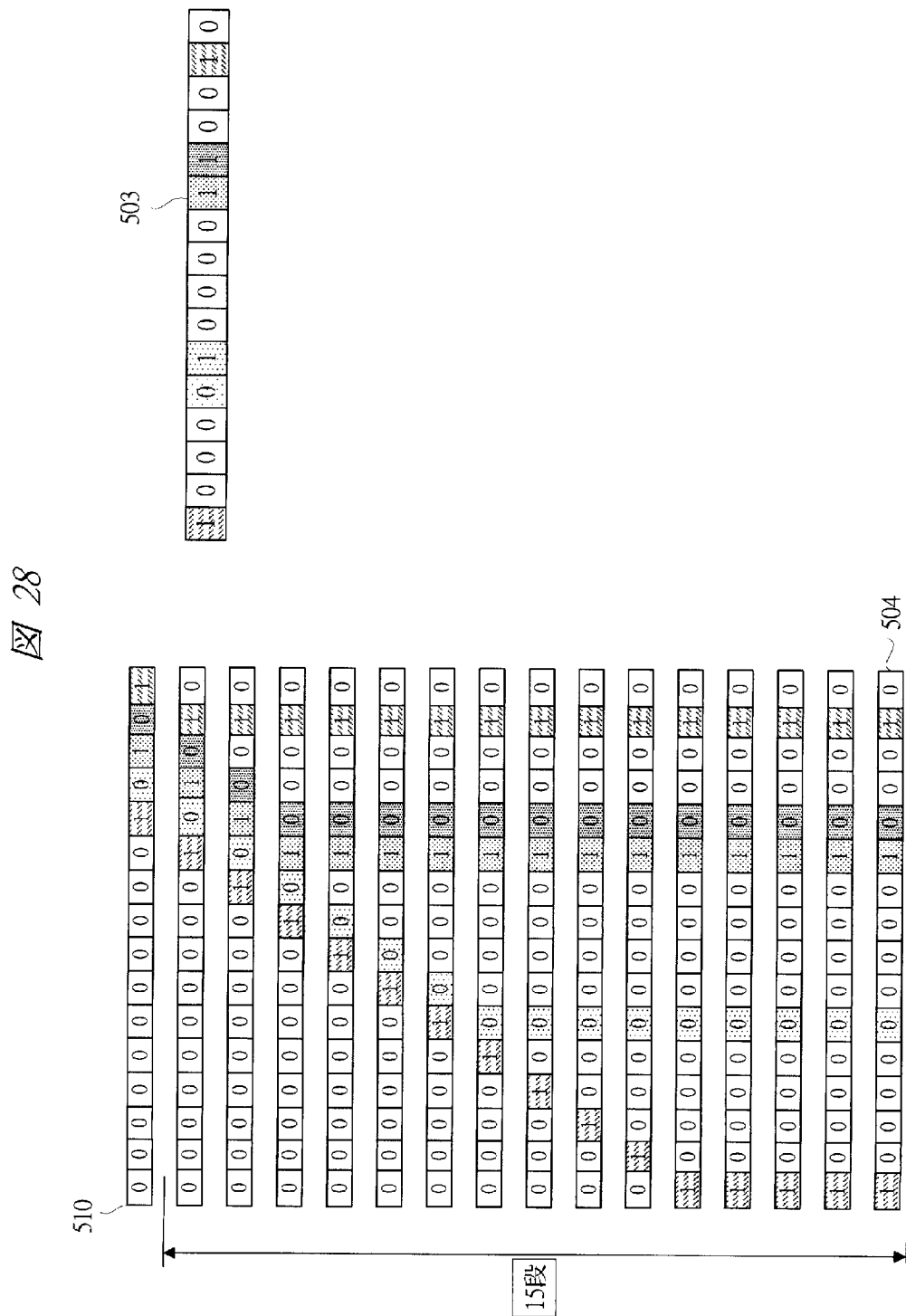


[図27]

図 27

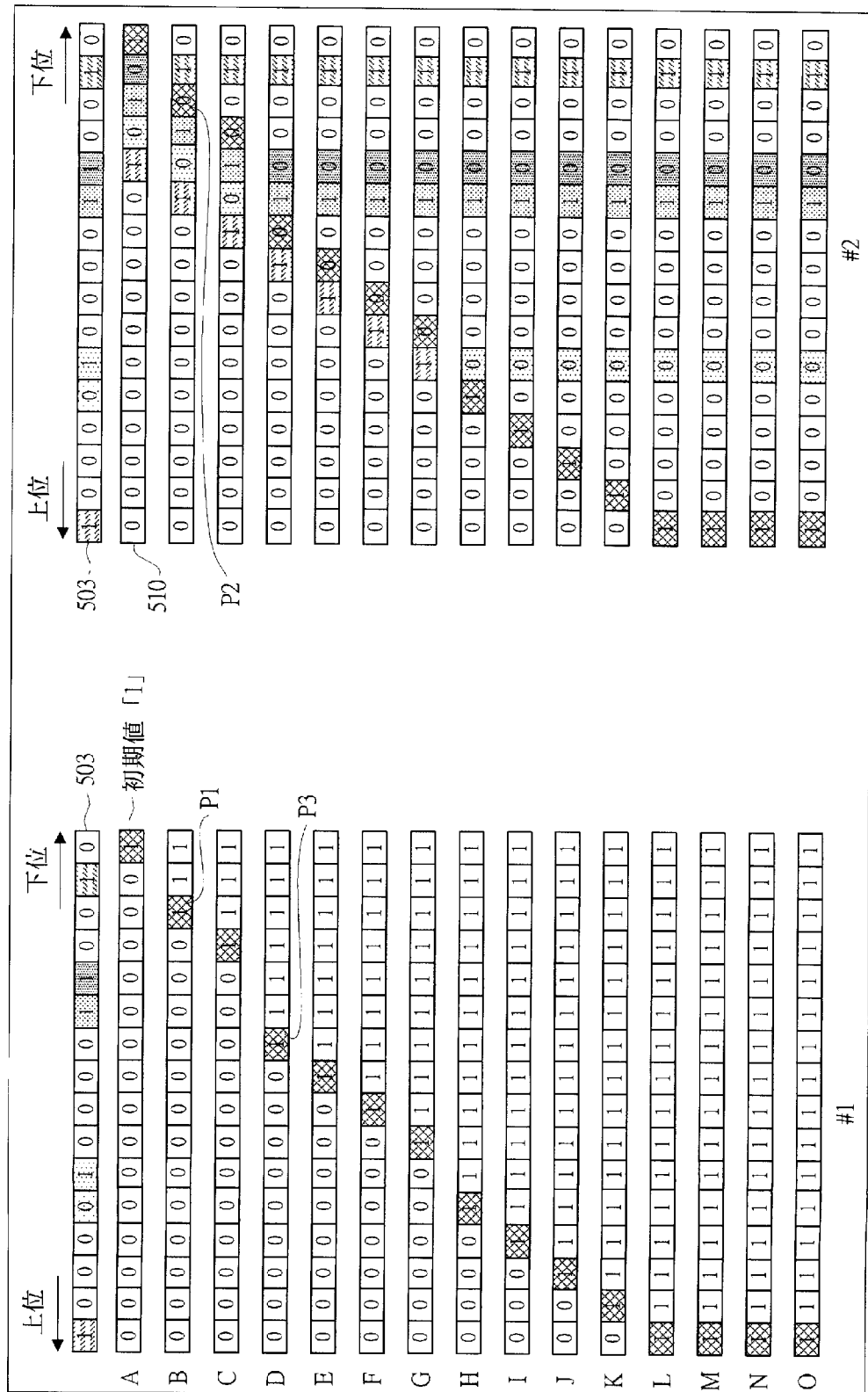


[図28]

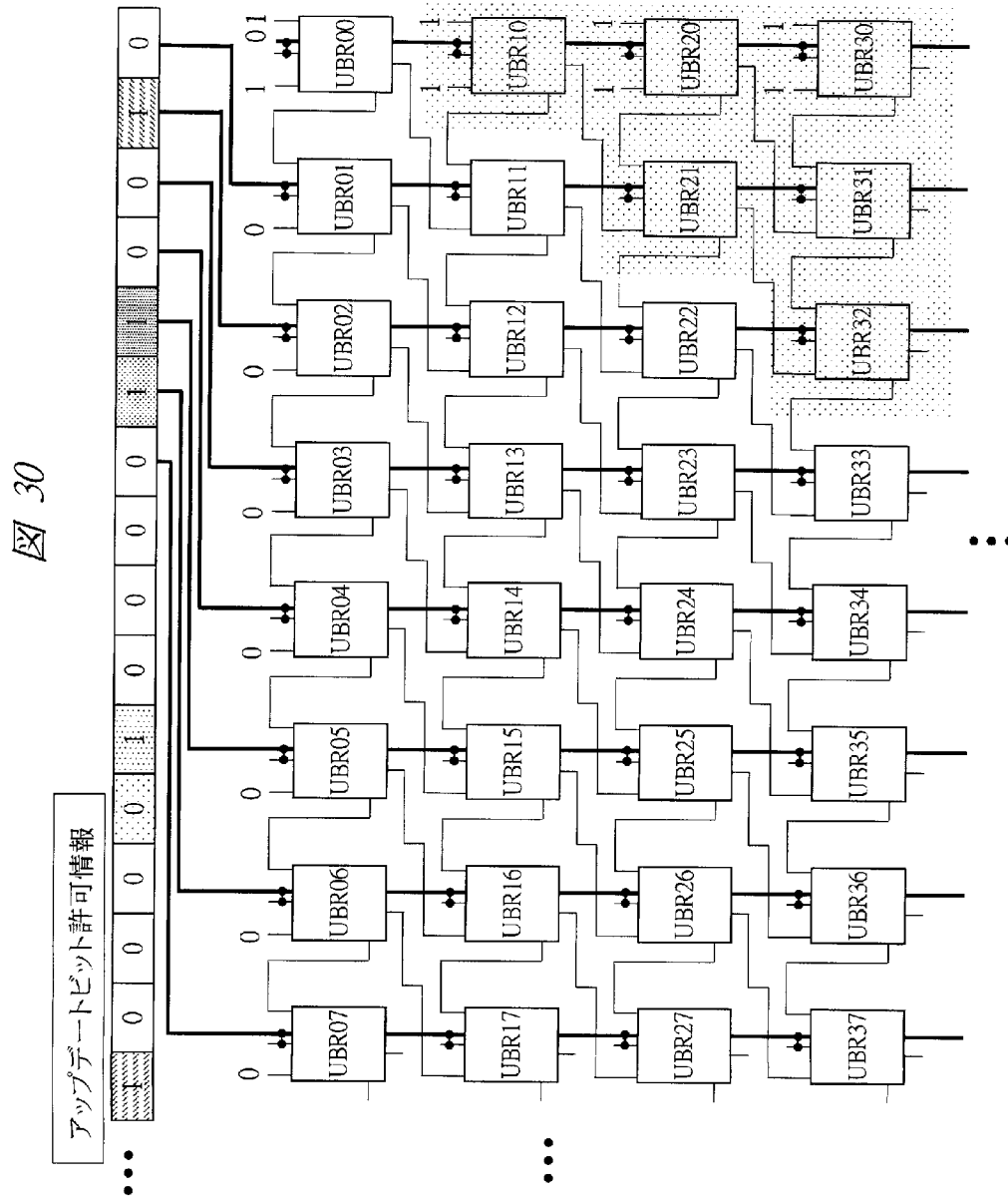


[図29]

図 29

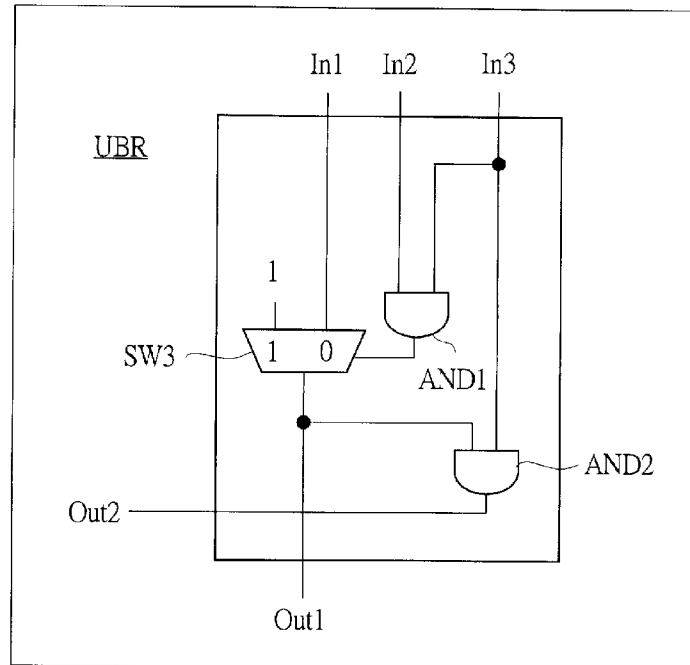


[図30]



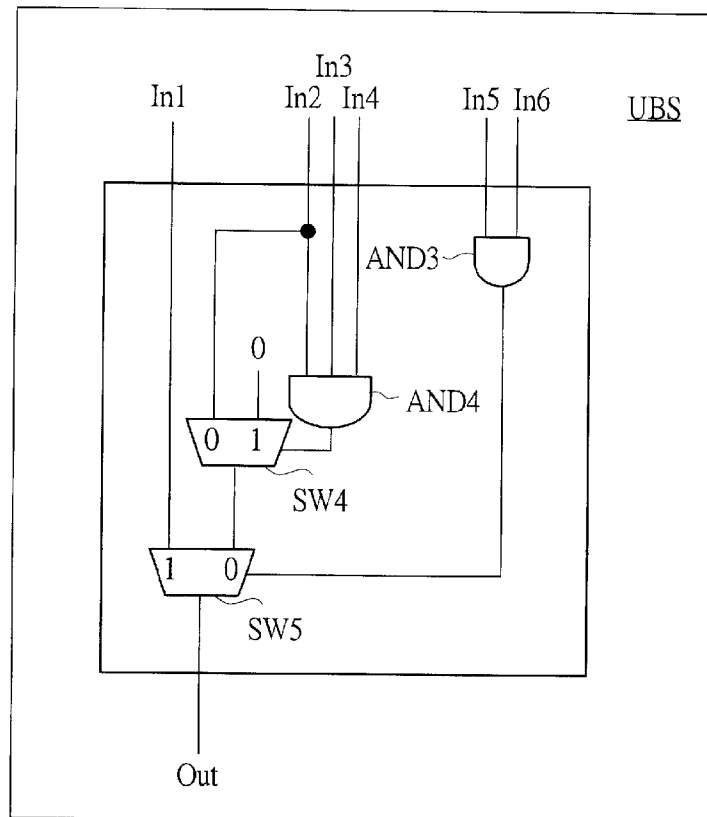
[図31]

図 31



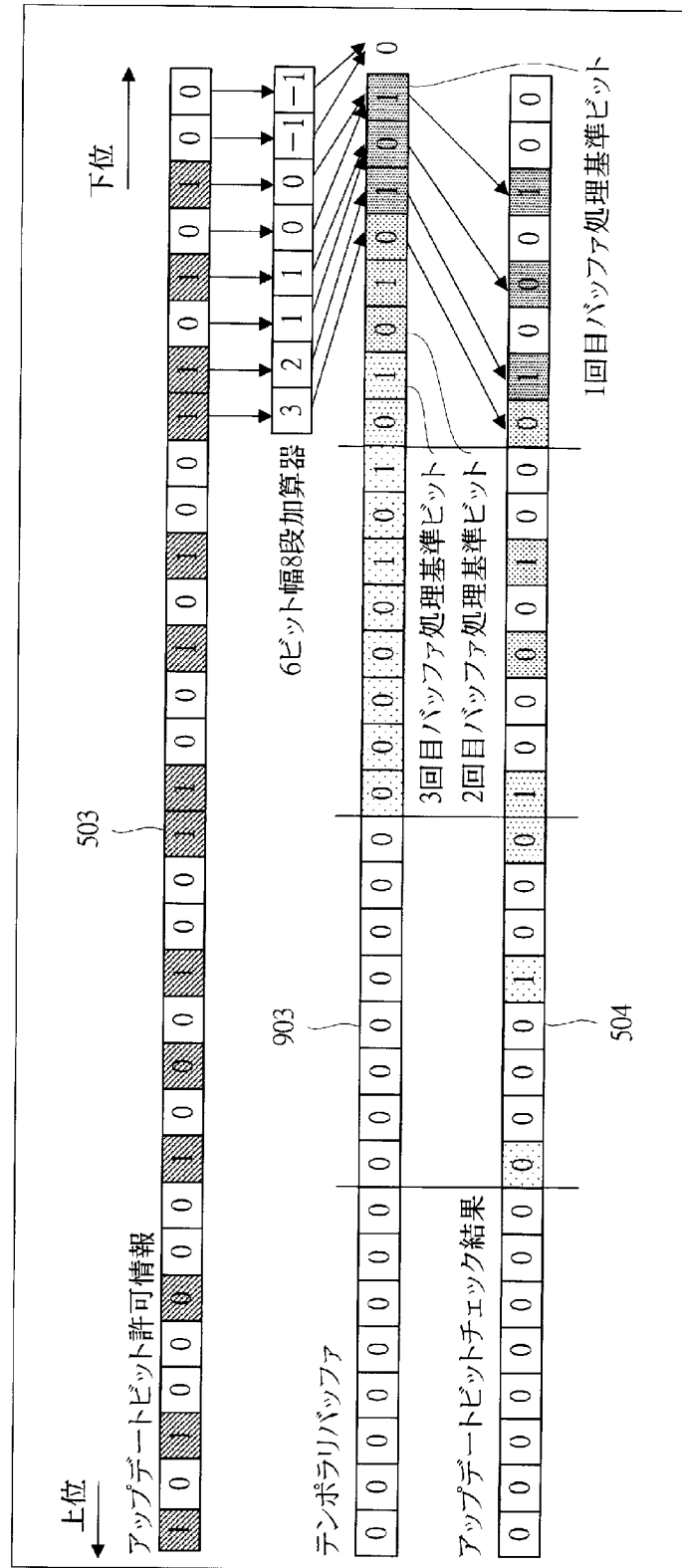
[図33]

図 33



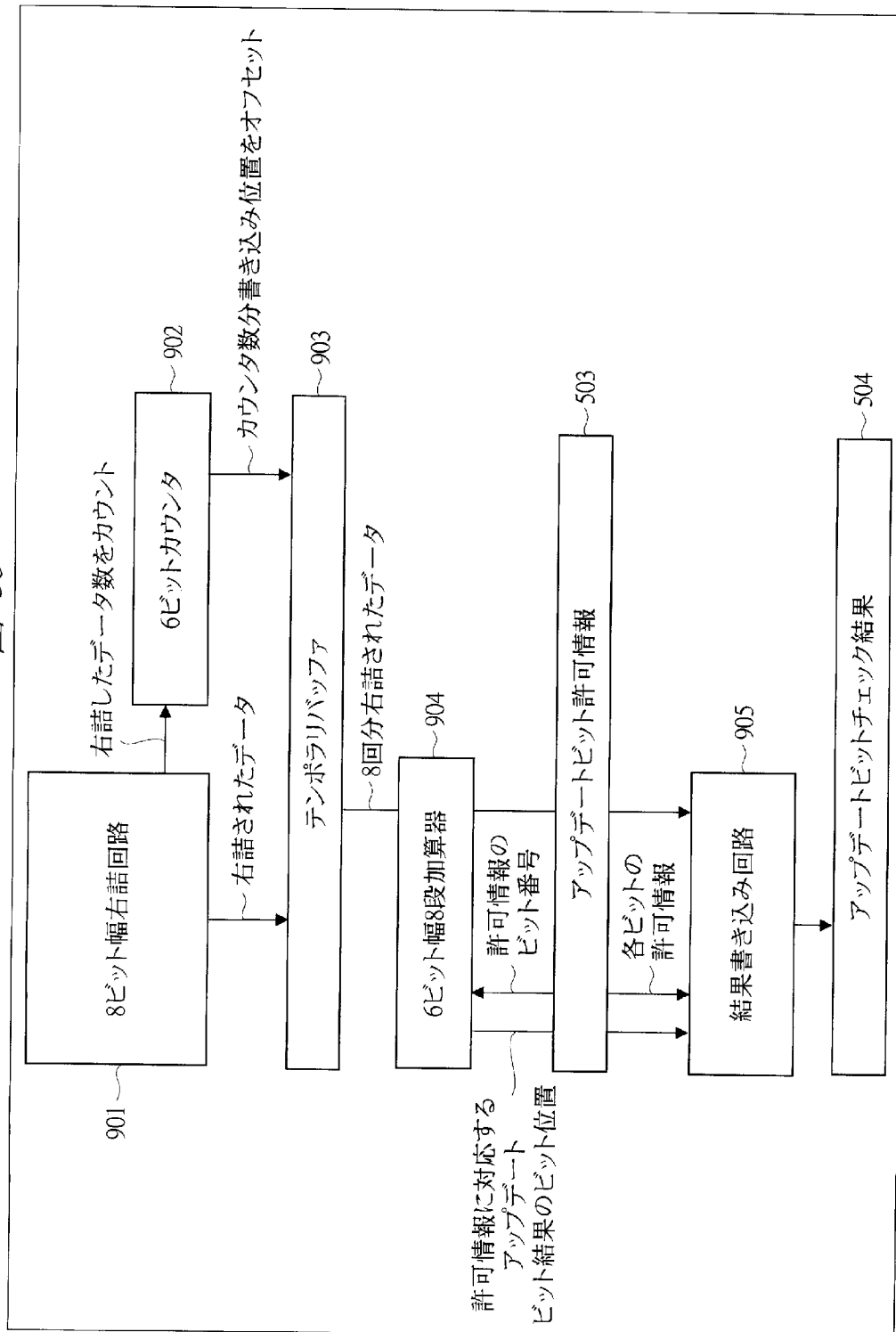
[図35]

図 35



[図36]

図 36



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/059841

A. CLASSIFICATION OF SUBJECT MATTER

H04L29/10(2006.01)i, B60R16/023(2006.01)i, H04L29/06(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L29/10, B60R16/023, H04L29/06

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2010
Kokai Jitsuyo Shinan Koho	1971-2010	Toroku Jitsuyo Shinan Koho	1994-2010

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	OSEK/VDX Communication, Version 3.0.3, 2004.07.20, pp.12-13	1 2-10
A	JP 11-163907 A (Yazaki Corp.), 18 June 1999 (18.06.1999), entire text; all drawings (Family: none)	1-10
A	JP 2006-256457 A (Fujitsu Ten Ltd.), 28 September 2006 (28.09.2006), entire text; all drawings (Family: none)	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
25 August, 2010 (25.08.10)

Date of mailing of the international search report
07 September, 2010 (07.09.10)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. H04L29/10(2006.01)i, B60R16/023(2006.01)i, H04L29/06(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. H04L29/10, B60R16/023, H04L29/06

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 0 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 0 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 0 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X	O S E K / V D X C o m m u n i c a t i o n , V e r s i o n 3 . 0 . 3 , 2004. 07. 20, pp. 12-13	1
A		2-10
A	JP 11-163907 A（矢崎総業株式会社）1999. 06. 18, 全文, 全図（フ ァミリーなし）	1-10
A	JP 2006-256457 A（富士通テン株式会社）2006. 09. 28, 全文, 全図 （ファミリーなし）	1-10

C 欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

2 5 . 0 8 . 2 0 1 0

国際調査報告の発送日

0 7 . 0 9 . 2 0 1 0

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

谷岡 佳彦

5 K

4 7 7 7

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 5 6