

(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
2012年10月11日(11.10.2012)



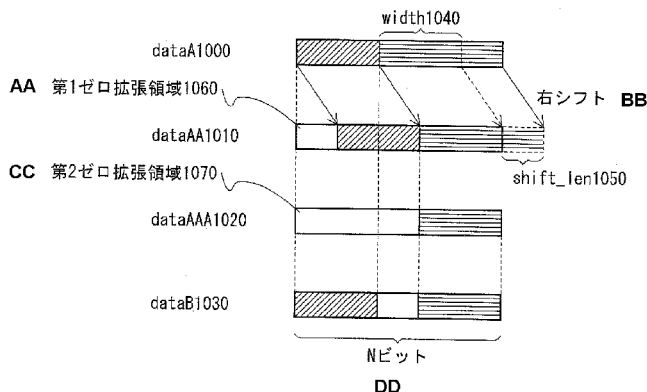
(10) 国際公開番号
WO 2012/137428 A1

- (51) 国際特許分類:
G06F 9/305 (2006.01)
 - (21) 国際出願番号: PCT/JP2012/001890
 - (22) 国際出願日: 2012年3月19日(19.03.2012)
 - (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 2011-086313 2011年4月8日(08.04.2011) JP
 - (71) 出願人(米国を除く全ての指定国について): パナソニック株式会社(PANASONIC CORPORATION) [JP/JP]; 〒5718501 大阪府門真市大字門真1006番地 Osaka (JP).
 - (72) 発明者; および
 - (75) 発明者/出願人(米国についてのみ): 馬場 大介 (BABA, Daisuke).
 - (74) 代理人: 中島 司朗, 外 (NAKAJIMA, Shiro et al.); 〒5310072 大阪府大阪市北区豊崎三丁目2番1号淀川5番館6F Osaka (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, 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, QA, RO, RS, RU, RW, 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.
 - (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, RW, 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 (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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- 添付公開書類:
— 国際調査報告(条約第21条(3))

(54) Title: DATA PROCESSING DEVICE AND DATA PROCESSING METHOD

(54) 発明の名称: データ処理装置、及びデータ処理方法

[図11]



- AA First zero extended area 1060
- BB Right-shift
- CC Second zero extended area 1070
- DD N bits

(57) Abstract: The present invention is provided with a decoder which, from one instruction, reads information indicating a bit string store region, information specifying a first bit range, and information specifying a second bit range consecutive to the first bit range in order to output a decode signal corresponding to the read information; and a bit operation circuit which, in accordance with the decode signal output from the decoder, inserts a bit string comprising identical predetermined values between the first bit range and the second bit range for the bit string stored in the bit string store region so as to generate and output an output bit string.

(57) 要約: 1命令から、ビット列格納領域を示す情報と第1ビット範囲を指定する情報と第1ビット範囲に連続する第2ビット範囲を指定する情報とを読み取って、読み取った情報に応じるデコード信号を出力するデコーダと、デコーダから出力されるデコード信号に応じて、ビット列格納領域に格納されるビット列に対して、第1ビット範囲と第2ビット範囲との間に同一の所定値からなるビット列を挿入することで出力ビット列を生成して出力するビット操作回路とを備える。

WO 2012/137428 A1

明 細 書

発明の名称：データ処理装置、及びデータ処理方法

技術分野

[0001] 本発明は、ビット列に対してビット操作を行うデータ処理技術に関する。

背景技術

[0002] 一般に、コンピュータシステムにおいて、記憶装置に対するデータの書き込み、及び読み出しは、8ビットの整数倍からなる1ワード（例えば、16ビット）を単位として行われる。

[0003] 従来、記憶装置における記憶領域を有効に利用するために、ワード単位未満の端数を含むビット数からなる複数のデータを記憶装置に対して書き込む場合には、これら複数のデータをワード単位の整数倍となるビット数のデータにパッキングしてからデータの書き込みを行うコンピュータシステムが知られている。

[0004] コンピュータシステムにおいて利用される、データのパッキングを比較的効率的に行うための装置として、例えば、特許文献1、特許文献2に記載されるビットフィールド操作回路がある。

[0005] 特許文献1に記載されているビットフィールド操作回路（以後、「ビットフィールド操作回路A」と呼ぶ。）は、図21に示されるビットフィールド操作を行う。

[0006] 同図において、データA2000とデータB2001とは、ビットフィールド操作回路Aがビットフィールド操作を行う場合において、ビットフィールド操作回路Aに入力されるnビットのビット列であり、データC2002は、ビットフィールド操作回路Aがビットフィールド操作を行う場合において、ビットフィールド操作回路Aから出力されるnビットのビット列である。

[0007] ビットフィールド操作回路Aは、データA2000とデータB2001とオフセット量2010とワイド量2011とを指定する命令が入力されると

、データA 2 0 0 0における最下位ビットの位置からオフセット量2 0 1 0だけ左シフトした位置に、データB 2 0 0 1における、最下位ビットを含む、ワイド量2 0 1 1で示されるビット数のビット列が挿入されたデータC 2 0 0 2を生成するビットフィールド操作を行う。

[0008] 特許文献2に記載されているビットフィールド操作回路（以後、「ビットフィールド操作回路B」と呼ぶ。）は、図22に示されるビットフィールド操作を行う。

[0009] 同図において、データA 2 1 0 0は、ビットフィールド操作回路Bがビットフィールド操作を行う場合において、ビットフィールド操作回路Bに入力されるnビットのビット列であり、データB 2 1 0 1は、ビットフィールド操作回路Aがビットフィールド操作を行う場合において、ビットフィールド操作回路Bから出力されるnビットのビット列である。

[0010] ビットフィールド操作回路Bは、データA 2 1 0 0とオフセット量2 1 1 0とシフト量2 1 1 1とを指定する命令が入力されると、データA 2 1 0 0を、シフト量だけ右に論理シフトしたデータに対して、データA 2 1 0 0におけるオフセット量2 1 1 0で示されるビット数の最下位ビットを含むビット列が挿入されてなるデータB 2 1 0 1を生成するビットフィールド操作を行う。

先行技術文献

特許文献

[0011] 特許文献1：特開2008-83795号公報

特許文献2：特開2000-99327号公報

発明の概要

発明が解決しようとする課題

[0012] ところで、パッキングされたデータを記憶装置から読み出して利用する場合には、読み出したデータをアンパッキングする必要がある。

[0013] 従って、データをパッキング及びアンパッキングして利用するコンピュー

タシステムにおいては、データのパッキングを比較的効率良く実行することに加えて、パッキングされたデータのアンパッキングを比較的効率良く実行することが望まれる。

[0014] そこで、本発明は係る問題に鑑みてなされたものであり、パッキングされたデータのアンパッキングを比較的効率良く実行することができるデータ処理装置を提供することを目的とする。

課題を解決するための手段

[0015] 上記課題を解決するために本発明に係るデータ処理装置は、1命令から、 N (N は2以上の整数) ビットの対象ビット列を格納するビット列格納領域を指定する情報と、 N ビットのビット列に含まれる、当該 N ビットのビット列における第1の端ビットを含む第1ビット範囲を示す情報と、当該 N ビットのビット列に含まれる、当該第1ビット範囲に連続するビット範囲であって、当該 N ビットのビット列における第2の端ビットを含まない第2ビット範囲を示す情報とを読み取って、読み取った情報に応じたデコード信号を出力するデコーダと、前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に格納される前記対象ビット列に基づく、 N ビットからなる出力ビット列を生成して出力するビット操作回路とを備え、前記ビット操作回路は、前記出力ビット列における前記第1ビット範囲に、前記対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第2の端ビットを含む、前記第2ビット範囲のビット長からなる第3ビット範囲に、前記対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の所定値を配置することで、前記出力ビット列の生成を行うことを特徴とする。

発明の効果

[0016] 上述の構成を備える本発明に係るデータ処理装置によると、1命令からデコードされたデコード信号に応じて動作するビット操作回路が、ビット列格

納領域に格納されるビット列のうちの第1ビット範囲と第2ビット範囲とにパッキングされているデータを、第1ビット範囲と第2ビット範囲との間に同一の所定値からなるビット列を挿入するようにアンパッキングして出力することとなる。このことにより、このデータ処理装置は、パッキングされたデータのアンパッキングを比較的効率良く実行することができるようになる。

図面の簡単な説明

- [0017] [図1]データ処理装置100の主要なハードウェア構成を示すブロック図
- [図2]4バイトからなる1命令のビットフィールド構成図
- [図3]ビット操作種命令とデコード信号との対応関係を示すデコード信号対応表
- [図4]ビット操作回路150の回路構成を示す回路構成図
- [図5]符号拡張回路330の回路構成を示す回路構成図
- [図6]デコーダ450の行うデコードの真理値表
- [図7]マスクデータ生成回路350の行うデコードの真理値表
- [図8]shrn_N命令デコード時におけるデータ構成図
- [図9]shrn_{LN}命令デコード時におけるデータ構成図
- [図10]shrn_{LS}命令デコード時におけるデータ構成図
- [図11]extr_N命令デコード時におけるデータ構成図
- [図12]extr_{LN}命令デコード時におけるデータ構成図
- [図13]extr_{LS}命令デコード時におけるデータ構成図
- [図14]データ処理装置100の行うビット操作処理のフローチャート
- [図15]具体例におけるデータ構成図
- [図16]データ処理装置1500の主要なハードウェア構成を示すブロック図
- [図17]SIMD型ビット操作種命令とデコード信号との対応関係を示すデコード信号対応表
- [図18]データ処理装置1700の主要なハードウェア構成を示すブロック図
- [図19] (a) 命令形式をニーモニック形式で模式的に示す模式図その1、 (

b) 命令形式をニーモニック形式で模式的に示す模式図その2

[図20]データ処理装置2200の構成を示すブロック図

[図21]従来のビットフィールド操作回路Aにおけるデータ構成図

[図22]従来のビットフィールド操作回路Bにおけるデータ構成図

[図23]下詰めアンパッキング処理におけるデータ構成図

[図24]上詰めアンパッキング処理におけるデータ構成図

発明を実施するための形態

[0018] <実施の形態1>

本発明に係るデータ処理装置の一実施形態であるデータ処理装置100の構成についての詳細な説明に先立って、ここでは、まず、データ処理装置100が開発されるに至った経緯について説明する。

[0019] 図23は、データアンパッキング処理の一例である下詰めデータアンパッキング処理における、下詰めデータアンパッキング処理前のデータ構造と、下詰めデータアンパッキング処理後のデータ構造との一具体例を示す図である。

[0020] ここで、この下詰めデータアンパッキング処理は、データパッキングされている複数のビット列のそれぞれが、ワード単位の整数倍のビット列のそれぞれにおける最下位ビット側に配置されるようにアンパッキングされる処理のことである。

[0021] 同図において、データA2300は、下詰めデータアンパッキング処理前の32ビットのビット列であって、7ビットの有効ビット列2301と9ビットの有効ビット列2302とからなる16ビットのビット列と、16ビットの論理値“0”となる不要ビット列2303とから構成されている。

[0022] データB2310は、下詰めデータアンパッキング処理後の32ビットのビット列であって、上位16ビットの中の最下位ビット側位置に有効ビット列2301が配置され、下位16ビットの中の最下位ビット側位置に有効ビット列2302が配置されるように、下詰めデータアンパッキング処理がなされたデータとなっている。

- [0023] ところで、データB 2 3 1 0における下位16ビットのビット列を、例えば、16ビット単位でなされる算術演算等で利用される整数として取り扱う場合には、ビット列2 3 1 1の部分（以下、「拡張ビット列2 3 1 1」と呼ぶ。）が同一の所定値で満たされている必要がある。すなわち、符号なし整数として取り扱う場合には、拡張ビット列2 3 1 1が値“0”で満たされている必要があり、符号付き整数として取り扱う場合には、拡張ビット列2 3 1 1が、有効ビット列2 3 0 2における最上位ビットの値で満たされている必要がある。
- [0024] しかしながら、従来のビットフィールド操作回路A、及び従来のビットフィールド操作回路Bは、拡張ビット列2 3 1 1を所定の値で満たすタイプの下詰めデータアンパッキング処理（以下、「第1データアンパッキング処理」と呼ぶ。）を1命令で実現することができない。従って、従来のデータ処理装置では、この第1データアンパッキング処理を実現するためには、少なくとも2つの命令を組み合わせて実行する必要がある。
- [0025] 図24は、データアンパッキング処理の一例である上詰めデータアンパッキング処理における、上詰めデータアンパッキング処理前のデータ構造と、上詰めデータアンパッキング処理後のデータ構造との一具体例を示す図である。
- [0026] ここで、この上詰めデータアンパッキング処理は、データパッキングされている複数のビット列のそれぞれが、ワード単位の整数倍のビット列のそれぞれにおける最上位ビット側に配置されるようにアンパッキングされる処理のことである。
- [0027] 同図において、データA 2 4 0 0は、上詰めデータアンパッキング処理前の32ビットのビット列であって、7ビットの有効ビット列2 4 0 1と9ビットの有効ビット列2 4 0 2とからなる16ビットのビット列と、16ビットの論理値“0”となる不要ビット列2 4 0 3とから構成されている。
- [0028] データB 2 4 1 0は、上詰めデータアンパッキング処理後の32ビットのビット列であって、上位16ビットの中の最上位ビット側位置に有効ビット

列 2 4 0 1 が配置され、下位 1 6 ビットの中の最上位ビット側位置に有効ビット列 2 4 0 2 が配置されるように、上詰めデータアンパッキング処理がなされたデータとなっている。

[0029] ところで、データ B 2 4 1 0 における、有効ビット列 2 4 0 2 とビット列 2 4 1 1 とからなる 1 6 ビットのビット列を、例えば、1 6 ビット単位でなされる算術演算等で利用される整数として取り扱う場合には、ビット列 2 4 1 1 の部分（以下、「拡張ビット列 2 4 1 1」と呼ぶ。）が同一の所定値で満たされている必要がある。すなわち、符号なし整数として取り扱う場合には、拡張ビット列 2 4 1 1 が値“0”で満たされている必要があり、符号付き整数として取り扱う場合には、拡張ビット列 2 4 1 1 が、有効ビット列 2 4 0 2 における最上位ビットの値で満たされている必要がある。

[0030] しかしながら、従来のビットフィールド操作回路 A、及び従来のビットフィールド操作回路 B は、拡張ビット列 2 4 1 1 を所定値で満たすタイプの上詰めデータアンパッキング処理（以下、「第 2 データアンパッキング処理」と呼ぶ。）を 1 命令で実現することができない。従って、従来のデータ処理装置では、この第 2 データアンパッキング処理を実現するためには、少なくとも 2 つの命令を組み合わせて実行する必要がある。

[0031] 一般に、コンピュータシステムにおいて、単位時間あたりに実行できる命令数は限られている。よって、従来のデータ処理装置では、少なくとも 2 つの命令を組み合わせて実行する必要がある、上述の第 1 データアンパッキング処理と第 2 データアンパッキング処理とは、1 命令で実行されることが望まれる。

[0032] そこで、データ処理装置 1 0 0 は、上述の第 1 データアンパッキング処理と第 2 データアンパッキング処理とを 1 命令で実行することができるデータ処理装置を提供することを目的として開発された。

[0033] このデータ処理装置 1 0 0 は、命令デコーダとレジスタファイルとビット操作回路とを備え、ビット操作回路が、レジスタファイルに格納されているデータに対して、命令デコーダから出力されるデコード信号に基づいて、シ

フト処理に係るビット操作を行う。

[0034] 以下、データ処理装置100について、図面を用いて詳細に説明する。

[0035] <データ処理装置100の構成>

図1は、データ処理装置100の主要なハードウェア構成を示すブロック図である。

[0036] データ処理装置100は、命令フェッチユニット110と命令デコーダ120とロードストアユニット130とレジスタファイル140とビット操作回路150とクロックジェネレータ160とから構成される、例えば、半導体集積回路である。

[0037] クロックジェネレータ160は、命令フェッチユニット110と命令デコーダ120とロードストアユニット130とレジスタファイル140とビット操作回路150とに供給する、例えば、100MHzのクロック信号を生成して出力する機能を有する。

[0038] 命令フェッチユニット110は、命令デコーダ120と外部メモリとに接続され、以下の2つの機能を有する。

[0039] 機能1：クロックジェネレータ160から供給されるクロック信号に同期して、外部メモリに記憶されている4バイトの1命令を読み出し、読み出した命令を命令デコーダ120に出力する機能。

[0040] 機能2：外部メモリから命令を読み出す場合に用いるアドレスを、前回外部メモリから命令を読み出す際に用いたアドレス対して4バイトインクリメントしたものとする機能。

[0041] 図2は、外部メモリに記憶されている、4バイトからなる1命令のビットフィールド構成図である。

[0042] 同図に示される通り、外部メモリに記憶されている、4バイトからなる1命令は、1ビットのsignフィールド201と、1ビットのmask_dirフィールド202と、1ビットのoperationフィールド203と、6ビットのNフィールド204と、5ビットのwidthフィールド205と、5ビットのshift_lenフィールド206と4ビットのread_addrフィールド207と、4ビットのwr

ite_addrフィールド208と、5ビットのその他フィールド209とから構成されている。

- [0043] 再び図1に戻って、データ処理装置100の構成の説明を続ける。
- [0044] 命令デコーダ120は、命令フェッチユニット110とロードストアユニット130とレジスタファイル140とビット操作回路150とに接続され、命令フェッチユニット110から入力される、予め定められたフォーマットで記述された命令を、予め定められたアルゴリズムに従ってデコードする機能を有する。
- [0045] 命令デコーダ120のデコード対象となる命令には、ロードストアユニット130とレジスタファイル140とを用いて行うロードストア処理に係るロードストア種命令と、レジスタファイル140とビット操作回路150とを用いて行うビット操作処理に係るビット操作種命令との2種類ある。
- [0046] ここでは、命令デコーダ120の行うビット操作種命令のデコードについて、以下、図面を用いて説明する。
- [0047] 図3は、命令デコーダ120がビット操作種命令をデコードする場合における、デコード対象となるビット操作種命令とデコード信号との対応関係を示す、デコード信号対応表200である。
- [0048] 同図において、命令形式210は、デコード対象となるビット操作種命令の、ニーモニック形式における命令形式を示すものである。
- [0049] ビット操作種命令には、オペコードがshrn hN （ N は8又は16又は32のいずれか。以下同様）となるshrn hN 命令と、オペコードがshrn lN となるshrn lN 命令と、オペコードがshrn lsN となるshrn lsN 命令と、オペコードがextr hN となるextr hN 命令と、オペコードがextr lN となるextr lN 命令と、オペコードがextr lsN となるextr lsN 命令との6種類ある。
- [0050] そして、これらの命令は、レジスタファイル140内のデスティネーションレジスタを指定するレジスタ名 B と、レジスタファイル140内のソースレジスタを指定するレジスタ名 A と、 N 未満となる第1のビット長を示すshift_lenと、 N 未満となる第2のビット長を示すwidthとをオペランドとする。

- [0051] sign220と、mask_dir230と、operation240と、N250と、width260と、shift_len270と、read_addr280と、write_addr290とのそれぞれは、命令デコーダ120がビット操作種命令をデコードする際に出力するデコード信号である。
- [0052] sign220は、オペコードの6文字目が“s”である場合には論理値“1”となり、オペコードの6文字目が“s”でない場合には論理値“0”となる信号であって、ビット操作回路150に出力される。このsign220は、ビット操作回路150の行うビット操作処理が、符号拡張を伴う符号拡張処理であるか、符号拡張を伴わないゼロ拡張処理であるかを示す信号である。
- [0053] また、sign220は、図2におけるsignフィールド201に基づいてデコードされたものである。
- [0054] mask_dir230は、オペコードの5文字目が“h”である場合には論理値“0”となり、オペコードの5文字目が“h”でない場合、すなわち、“l”である場合には論理値“1”となる信号であって、ビット操作回路150に出力される。このmask_dir230は、ビット操作回路150の行うビット操作処理において、シフト処理対象とならないビット列を特定するための信号である。シフト処理対象とならないビット列については、後述する。
- [0055] また、mask_dir230は、図2におけるmask_dirフィールド202に基づいてデコードされたものである。
- [0056] operation240は、オペコードの1文字目～4文字目が“shrn”である場合には論理値“0”となり、オペコードの1文字目～4文字目が“shrn”でない場合、すなわち、“extr”である場合には論理値“1”となる信号であって、ビット操作回路150に出力される。このoperation240は、ビット操作回路150の行うビット操作処理において、シフト処理対象となるビット列の、シフト処理対象とならないビット列に対するシフト方向を特定するための信号である。シフト処理対象となるビット列については、後述する。
- [0057] また、operation240は、図2におけるoperationフィールド203に基づいてデコードされたものである。

- [0058] N250は、オペコードに含まれる数字Nを2進数で示す6ビットの信号であって、ビット操作回路150に出力される。このN250は、ビット操作回路150の行うビット操作処理対象となるビット列のビット長を示す信号である。
- [0059] また、N250は、図2におけるNフィールド204に基づいてデコードされたものである。
- [0060] width260は、オペランドwidthの数値を2進数で示す5ビットの信号であって、ビット操作回路150に出力される。このwidth260は、ビット操作回路150の行うビット操作処理において、シフト対象となるビット列のビット長を示す信号である。
- [0061] また、width260は、図2におけるwidthフィールド205に基づいてデコードされたものである。
- [0062] shift_len270は、オペランドshift_lenの数値を2進数で示す5ビットの信号であって、ビット操作回路150に出力される。このshift_len270は、ビット操作回路150の行うビット操作処理対象となるビット列のシフト量を示す信号である。
- [0063] また、shift_len270は、図2におけるshift_lenフィールド206に基づいてデコードされたものである。
- [0064] read_addr280は、オペランドAで指定される、レジスタファイル140内のソースレジスタの4ビットのアドレスを示す信号(addrA)であって、レジスタファイル140に出力される。このread_addr280は、図2におけるread_addrフィールド207に基づいてデコードされたものである。
- [0065] write_addr290は、オペランドBで指定される、レジスタファイル140内のソースレジスタの4ビットのアドレスを示す信号(addrB)であって、レジスタファイル140に出力される。このwrite_addr290は、図2におけるwrite_addrフィールド208に基づいてデコードされたものである。
- [0066] 再び図1に戻って、データ処理装置100の構成の説明を続ける。
- [0067] ロードストアユニット130は、命令デコーダ120とレジスタファイル

140と外部メモリとに接続され、以下の2つの機能を有する。

- [0068] 機能1：命令デコーダ120から、ロードストア種命令をデコードした際に出力される、外部メモリのアドレスを示す信号ld_addrが入力されると、クロックジェネレータ160から供給されるクロック信号に同期して、ld_addrによって示されるアドレスで指定される、外部メモリにおける記憶領域の4バイトのデータを外部メモリから読み出して、レジスタファイル140に出力する機能。
- [0069] 機能2：命令デコーダ120から、ロードストア種命令をデコードした際に出力される、外部メモリのアドレスを示す信号st_addrが入力されると、クロックジェネレータ160から供給されるクロック信号に同期して、レジスタファイル140から入力される4バイトのデータを、st_addrによって示されるアドレスで指定される、外部メモリにおける記憶領域に書き込む機能。
- [0070] レジスタファイル140は、命令デコーダ120とロードストアユニット130とビット操作回路150とに接続され、内部に4バイトからなる汎用レジスタを16本有し、以下の2つの機能を有する。
- [0071] 機能1：命令デコーダ120からread_addrが入力されると、クロックジェネレータ160から供給されるクロック信号に同期して、read_addrによって指定されるソースレジスタに格納されている4バイトのデータ出力する機能。
- [0072] 機能2：命令デコーダ120からwrite_addrが入力されると、クロックジェネレータ160から供給されるクロック信号に同期して、ロードストアユニット130又はビット操作回路150から、同じクロックサイクル内に出力された4バイトのデータを、write_addrによって指定されるデスティネーションレジスタに格納する機能。
- [0073] ビット操作回路150は、命令デコーダ120とレジスタファイル140とに接続され、レジスタファイル140から入力されるデータdataAに対して、命令デコーダ120から入力されるデコード信号に応じた処理を行ってデータdataBを生成して出力する機能を有する。

- [0074] 図4は、ビット操作回路150の回路構成を示す回路構成図である。
- [0075] 同図に示される通り、ビット操作回路150は、排他的論理和ゲート310と左右シフタ320と符号拡張回路330と加算器340とマスクデータ生成回路350と第1ビットセレクタ360とから構成される。
- [0076] 排他的論理和ゲート310は、デコード信号mask_dirの信号線とデコード信号operationの信号線と左右シフタ320とに接続される、2入力排他的論理和回路である。
- [0077] この排他的論理和ゲート310は、mask_dirとoperationとの2つのデコード信号が入力されると、左右シフタ320のシフト方向を示す1ビットの信号shift_dirを、左右シフタ320へ出力する。
- [0078] 左右シフタ320は、デコード信号signの信号線とデコード信号shift_lenの信号線と排他的論理和ゲート310と入力データdataAの信号線と符号拡張回路330とに接続される左右バレルシフタであって、制御信号である、sign、shift_len、及びshift_dirに基づいて、入力データdataAをシフトして出力データdataAAを出力する機能を有する。
- [0079] 以下、入力される制御信号それぞれについて説明する。
- [0080] shift_dir：シフト方向が右方向（下位ビット側）であるか左方向（上位ビット側）であるかを制御する信号。左右シフタ320は、shift_dirが論理値“1”の場合に右方向へのシフトを行い、shift_dirが論理値“0”の場合に左方向へのシフトを行う。
- [0081] shift_len：シフト量を示す制御信号。左右シフタ320は、shift_lenによって示されるシフト量だけ、右又は左にシフトする。
- [0082] sign：右シフト時におけるシフト方式を、符号拡張を伴う右シフト、すなわち算術シフトとするか、符号拡張を伴わない右シフト、すなわち論理シフトとするかを制御する信号。左右シフタ320は、右シフト時においてsignが論理値“1”の場合に、符号拡張を伴う右シフトを行い、右シフト時においてsignが論理値“0”の場合に、符号拡張を伴わない右シフトを行う。
- [0083] 符号拡張回路330は、デコード信号signの信号線とデコード信号mask_dir

rの信号線とデコード信号operationの信号線とデコード信号Nの信号線とデコード信号widthの信号線と左右シフタ320と第1ビットセレクタ360とに接続され、制御信号であるsign、mask_dir、operation、N、及びwidthに基づいて、入力データdataAAにおける所定のビット列の位置に、所定のビットの位置の信号の複製、又は論理値“0”を挿入して出力データdataAAAを出力する機能を有する。

- [0084] 図5は、符号拡張回路330の回路構成を示す回路構成図である。
- [0085] 同図に示される通り、符号拡張回路330は、減算器410と第1セレクタ420と第2セレクタ430と第3セレクタ440とデコーダ450と第2ビットセレクタ460とから構成される。
- [0086] 減算器410は、デコード信号widthの信号線とデコード信号Nの信号線と第1セレクタ420とに接続され、Nの値からwidthの値を減算し、6ビットの信号N-widthを出力する機能を有する。
- [0087] 第1セレクタ420は、デコード信号mask_dirの信号線とデコード信号widthの信号線と減算器410とに接続される、6ビットの2入力1出力セレクタであって、制御信号であるmask_dirが論理値“1”の場合にwidthを選択し、mask_dirが論理値“0”の場合にN-widthを選択して、6ビットの信号sign_offsetを出力する機能を有する。このsign_offsetは、ビット操作処理対象となるNビットのビット列において、上記所定のビットの位置を示す信号となる。
- [0088] 第2セレクタ430は、入力データdataAAの信号線と第1セレクタ420と第3セレクタ440とに接続される、1ビットの32入力1出力セレクタであって、32ビットのビット列である入力データdataAAAから、制御信号であるsign_offsetによって示される位置のビットを選択して、1ビットの信号sign_dataを出力する機能を有する。
- [0089] 第3セレクタ440は、デコード信号signの信号線と第2セレクタ430と第2ビットセレクタ460とに接続される、1ビットの2入力1出力セレクタであって、制御信号であるsignが論理値“1”の場合にsign_dataを選択

し、signが論理値“0”の場合に論理値“0”を選択して、1ビットの信号ext_dataを出力する機能を有する。

[0090] デコーダ450は、デコード信号operationの信号線とデコード信号mask_dirの信号線とデコード信号widthの信号線とデコード信号Nの信号線と第2ビットセクタ460とに接続されるデコーダであって、入力されるoperation、mask_dir、width、及びNから、32ビットのext_mask信号をデコードする機能を有する。

[0091] 図6は、デコーダ450の行うデコードの真理値表である。

[0092] 同図において、operation510とmask_dir520とwidth530とは、デコーダ450に入力される信号であって、ext_mask(Nbit)540は、32ビットのデコード信号ext_maskのうちの下位Nビットからなるビット列である。32ビットからなるext_maskのうちの下位Nビットを除く部分の値は、ドントケア値となっている（図示していない）。

[0093] デコーダ450は、operation、mask_dir、width、及びNが入力されると、同図に示される真理値表に従って、ext_maskを出力する。

[0094] 再び図5に戻って、符号拡張回路330の説明を続ける。

[0095] 第2ビットセクタ460は、入力データdataAAの信号線と第3セクタ440とデコーダ450と第1ビットセクタ360（図4参照）とに接続され、32個の2入力1出力セクタからなり、dataAAに対して、ext_maskで指定されるビット位置に、ext_dataのビットを挿入してdataAAAを生成する機能を有する。

[0096] 第2ビットセクタ460を構成するセクタのそれぞれは、ext_maskの各ビットを制御信号として、対応するdataAAのビットとext_dataとのいずれか一方を選択して出力する機能を有する。

[0097] 再び図4に戻って、ビット操作回路150の説明を続ける。

[0098] 加算器340は、デコード信号widthの信号線とデコード信号shift_lenの信号線とマスクデータ生成回路350とに接続され、widthの値とshift_lenの値とを加算し、6ビットの信号であるmask_widthを出力する機能を有する

- 。
- [0099] マスクデータ生成回路350は、デコード信号mask_dirの信号線とデコード信号Nの信号線と加算器340と第1ビットセレクタ360とに接続されるデコーダであって、入力されるmask_dir、N、及びmask_widthから、32ビットのmask信号をデコードする機能を有する。
- [0100] 図7は、マスクデータ生成回路350の行うデコードの真理値表である。
- [0101] 同図において、mask_dir610とmask_width620とは、マスクデータ生成回路350に入力される信号であって、mask(Nbit)630は、32ビットのデコード信号maskのうちの下位Nビットからなるビット列である。32ビットからなるmaskのうちの下位Nビットを除く部分の値は、ドントケア値となっている（図示していない）。
- [0102] マスクデータ生成回路350は、mask_dir、mask_width及びNが入力されると、同図に示される真理値表に従って、maskを出力する。
- [0103] 再び図4に戻って、ビット操作回路150の説明を続ける。
- [0104] 第1ビットセレクタ360は、入力データdataAの信号線と符号拡張回路330とマスクデータ生成回路350と出力データdataBの信号線とに接続され、32個の2入力1出力セレクタからなり、dataAにおける、maskで指定されるビット位置に、dataAAAのデータを挿入してdataBを生成する機能を有する。
- 。
- [0105] これらのセレクタのそれぞれは、maskの各ビットを制御信号として、対応するdataAのビットと、対応するdataAAAのビットとのいずれか一方を選択して出力する機能を有する。
- [0106] 上述の構成を備えるデータ処理装置100について、命令デコーダ120がビット操作種命令をデコードした場合における、ビット操作回路150の動作について、以下図面を用いて説明する。
- [0107] <ビット操作回路150の動作>
- 図8は、命令デコーダ120がshrnN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成

図である。

- [0108] 同図において、dataA700は、命令デコーダ120がデコード信号を出力したクロックサイクルと同じサイクルに、レジスタファイル140から出力された32ビットの入力データのうちの低位Nビットの部分のビット列である。dataAA710は、レジスタファイル140から入力されるdataAに対応して、左右シフタ320から出力される32ビットのデータdataAAのうちの低位Nビットの部分のビット列である。dataAAA720は、左右シフタ320から入力されるdataAAに対応して、符号拡張回路330から出力される32ビットのデータdataAAAのうちの低位Nビットの部分のビット列である。dataB730は、符号拡張回路330から入力されるdataAAAに対応して、第1ビットセクタ360から出力される32ビットのデータdataBのうちの低位Nビットの部分のビット列である。width740は、命令デコーダ120から入力されるデコード信号widthによって指定されるビット長である。shift_len750は、命令デコーダ120から入力されるデコード信号shift_lenによって指定されるシフト量である。
- [0109] 排他的論理和ゲート310（図4参照）は、命令デコーダ120からshrnN命令に対応するデコード信号が入力されると、論理値“0”のshift_dirを左右シフタ320に出力する。すると、左右シフタ320は、dataA700を左方向にshift_len750で指定されるシフト量だけシフトをして、dataAA710を出力する。
- [0110] また、デコーダ450（図5参照）は、命令デコーダ120からshrnN命令に対応するデコード信号が入力されると、低位Nビットが全て論理値“0”のext_maskを第2ビットセクタ460に出力する。すると、第2ビットセクタ460は、dataAA710をそのままdataAAA720として出力する。
- [0111] 一方、加算器340（図4参照）は、width740の値とshift_len750の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路350は、低位Nビットのうちの、mask_widthの値で示される最下位ビットを含むビット列が論理値“1”、低位Nビットのうちのそれら以外のビットが

論理値“0”となるmaskを第1ビットセクタ360に出力する。

[0112] 従って、第1ビットセクタ360は、width740の値とshift_len750の値との和の値のビット幅の最下位ビットを含むビット列をdataAAA720から選択し、それら以外のビットをdataA700から選択して、dataB730として出力する。

[0113] このように、ビット操作回路150は、命令デコーダ120がshrnN命令をデコードした場合に、dataA700から、width740によって指定されるビット長の最下位ビットを含むビット列を、shift_len750によって指定されるシフト量だけ左シフトし、 $N - (\text{width}740 + \text{shift_len}750)$ の値によって指定されるビット長の最上位ビットを含むビット列をシフトしないことで、ビット列dataB730を生成して出力する。

[0114] 図9は、命令デコーダ120がshrnLN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

[0115] 同図において、dataA800、dataAA810、dataAAA820、dataB830、width840、及びshift_len850は、それぞれ、図8におけるdataA700、dataAA710、dataAAA720、dataB730、width740、及びshift_len750と同様のものである。よって、ここではこれらの説明を省略する。

[0116] 排他的論理和ゲート310は、命令デコーダ120からshrnLN命令に対応するデコード信号が入力されると、論理値“1”のshift_dirを左右シフタ320に出力する。すると、左右シフタ320は、dataA800を右方向にshift_len850で指定されるシフト量だけ符号拡張を伴わないシフトをして、dataAA810を出力する。ここで、左右シフタ320の行うシフトは、符号拡張を伴わないものであるため、図9中のゼロ拡張領域860には論理値“0”が挿入されることとなる。

[0117] また、デコーダ450は、命令デコーダ120からshrnLN命令に対応するデコード信号が入力されると、下位Nビットが全て論理値“0”のext_maskを第2ビットセクタ460に出力する。すると、第2ビットセクタ46

0は、dataAA810をそのままdataAAA820として出力する。

[0118] 一方、加算器340は、width840の値とshift_len850の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路350は、下位Nビットのうちの、 $N - (\text{width}840 + \text{shift_len}850)$ の値によって指定されるビット長の最下位ビットを含むビット列が論理値“1”、下位Nビットのうちのそれら以外のビットが論理値“0”となるmaskを第1ビットセレクタ360に出力する。

[0119] 従って、第1ビットセレクタ360は、width840の値とshift_len850の値との和の値のビット幅の最上位ビットを含むビット列をdataAAA820から選択し、それら以外のビットをdataA800から選択して、dataB830として出力する。

[0120] このように、ビット操作回路150は、命令デコーダ120がshrnlN命令をデコードした場合に、dataA800から、width840によって指定されるビット長の最上位ビットを含むビット列を、shift_len850によって指定されるシフト量だけ符号拡張を伴わずに右シフトし、 $N - (\text{width}840 + \text{shift_len}850)$ の値によって指定されるビット長の最下位ビットを含むビット列をシフトしないことで、ビット列dataB830を生成して出力する。

[0121] 図10は、命令デコーダ120がshrnlN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

[0122] 同図において、dataA900、dataAA910、dataAAA920、dataB930、width940、及びshift_len950は、それぞれ、図8におけるdataA700、dataAA710、dataAAA720、dataB730、width740、及びshift_len750と同様のものである。よって、ここではこれらの説明を省略する。

[0123] 排他的論理和ゲート310は、命令デコーダ120からshrnlN命令に対応するデコード信号が入力されると、論理値“1”のshift_dirを左右シフト320に出力する。すると、左右シフト320は、dataA800を右方向にshift_len850で指定されるシフト量だけ符号拡張を伴うシフトをして、dataAA

810を出力する。ここで、左右シフタ320の行うシフトは、符号拡張を伴うものであるため、図10中の符号拡張領域960にはdataA900の最上位ビットが符号拡張されることとなる。

[0124] また、デコーダ450は、命令デコーダ120からshrnlsN命令に対応するデコード信号が入力されると、下位Nビットが全て論理値“0”のext_maskを第2ビットセレクタ460に出力する。すると、第2ビットセレクタ460は、dataAA910をそのままdataAAA920として出力する。

[0125] 一方、加算器340は、width940の値とshift_len950の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路350は、下位Nビットのうちの、 $N - (\text{width}940 + \text{shift_len}950)$ の値によって指定されるビット長の最下位ビットを含むビット列が論理値“1”、下位Nビットのうちのそれら以外のビットが論理値“0”となるmaskを第1ビットセレクタ360に出力する。

[0126] 従って、第1ビットセレクタ360は、width940の値とshift_len950の値との和の値のビット幅の最上位ビットを含むビット列をdataAAA920から選択し、それら以外のビットをdataA900から選択して、dataB930として出力する。

[0127] このように、ビット操作回路150は、命令デコーダ120がshrnln命令をデコードした場合に、dataA900から、width940によって指定されるビット長の最上位ビットを含むビット列を、shift_len950によって指定されるシフト量だけ符号拡張を伴って右シフトし、 $N - (\text{width}940 + \text{shift_len}950)$ の値によって指定されるビット長の最下位ビットを含むビット列をシフトしないことで、ビット列dataB830を生成して出力する。

[0128] 図11は、命令デコーダ120がextrhN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

[0129] 同図において、dataA1000、dataAA1010、dataAAA1020、dataB1030、width1040、及びshift_len1050は、それぞれ、図8にお

けるdataA700、dataAA710、dataAAA720、dataB730、width740、及びshift_len750と同様のものである。よって、ここではこれらの説明を省略する。

[0130] 排他的論理和ゲート310は、命令デコーダ120からextrhN命令に対応するデコード信号が入力されると、論理値“1”のshift_dirを左右シフト320に出力する。すると、左右シフト320は、dataA1000を右方向にshift_len1050で指定されるシフト量だけ符号拡張を伴わないシフトをして、dataAA1010を出力する。ここで、左右シフト320の行うシフトは、符号拡張を伴わないものであるため、図11中の第1ゼロ拡張領域1060には論理値“0”が挿入されることとなる。

[0131] また、第3セクタ440は、命令デコーダ120から入力されるsignが論理値“0”であるため、論理値“0”をext_dataとして出力する。そして、デコーダ450は、命令デコーダ120からextrhN命令に対応するデコード信号が入力されると、下位Nビットのうちの、width1040の値によって指定されるビット長の最下位ビットを含むビット列が論理値“1”となり、下位Nビットのうちのそれら以外のビットが論理値“0”となるext_maskを出力する。すると、第2ビットセクタ460は、dataAA1010から、width1040によって指定されるビット長の最下位ビットを含むビット列を選択し、それら以外のビットに論理値“0”を選択することで、ビット列dataAA1020を生成して出力する。よって、図11中の第2ゼロ拡張領域1070は、論理値“0”となる。

[0132] 一方、加算器340は、width1040の値とshift_len1050の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路350は、下位Nビットのうちの、width1040+shift_len1050の値によって指定されるビット長の最下位ビットを含むビット列が論理値“1”、下位Nビットのうちのそれら以外のビットが論理値“0”となるmaskを第1ビットセクタ360に出力する。

[0133] 従って、第1ビットセクタ360は、width1040の値とshift_len1

050の値との和の値のビット幅の最下位ビットを含むビット列をdataAAA1020から選択し、それら以外のビットをdataA1000から選択して、dataB1030として出力する。

[0134] このように、ビット操作回路150は、命令デコーダ120がextrhN命令をデコードした場合に、dataA1000から、width1040+shift_len1050の値によって指定されるビット長の最下位ビットを含むビット列を、shift_len1050によって指定されるシフト量だけ符号拡張を伴わずに右シフトし、 $N - (\text{width}1040 + \text{shift_len}1050)$ の値によって指定されるビット長の最上位ビットを含むビット列をシフトしないことで、ビット列dataB1030を生成して出力する。

[0135] 図12は、命令デコーダ120がextrLN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

[0136] 同図において、dataA1100、dataAA1110、dataAAA1120、dataB1130、width1140、及びshift_len1150は、それぞれ、図8におけるdataA700、dataAA710、dataAAA720、dataB730、width740、及びshift_len750と同様のものである。よって、ここではこれらの説明を省略する。

[0137] 排他的論理和ゲート310（図4参照）は、命令デコーダ120からextrLN命令に対応するデコード信号が入力されると、論理値“0”のshift_dirを左右シフタ320に出力する。すると、左右シフタ320は、dataA1100を左方向にshift_len1150で指定されるシフト量だけシフトをして、dataAA1110を出力する。

[0138] また、第3セレクタ440は、命令デコーダ120から入力されるsignが論理値“0”であるため、論理値“0”をext_dataとして出力する。そして、デコーダ450は、命令デコーダ120からextrLN命令に対応するデコード信号が入力されると、下位Nビットのうちの、width1040の値によって指定されるビット長の最上位ビットを含むビット列が論理値“1”となり、

下位Nビットのうちのそれら以外のビットが論理値“0”となるext_maskを出力する。すると、第2ビットセクタ460は、dataAA1110から、width1140によって指定されるビット長の最上位ビットを含むビット列を選択し、それら以外のビットに論理値“0”を選択することで、ビット列dataAA1120を生成して出力する。よって、図12中の第1ゼロ拡張領域1160は、論理値“0”となる。

[0139] 一方、加算器340は、width1140の値とshift_len1150の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路350は、下位Nビットのうちの、width1140+shift_len1150の値によって指定されるビット長の最上位ビットを含むビット列が論理値“1”、下位Nビットのうちのそれら以外のビットが論理値“0”となるmaskを第1ビットセクタ360に出力する。

[0140] 従って、第1ビットセクタ360は、width1140の値とshift_len1150の値との和の値のビット幅の最上位ビットを含むビット列をdataAAA1120から選択し、それら以外のビットをdataA1100から選択して、dataB1130として出力する。

[0141] このように、ビット操作回路150は、命令デコーダ120がextrLN命令をデコードした場合に、dataA1100から、width1140+shift_len1150の値によって指定されるビット長の最上位ビットを含むビット列を、shift_len1150によって指定されるシフト量だけ左シフトして、N-(width1140+shift_len1150)の値によって指定されるビット長の最下位ビットを含むビット列をシフトせず、図12中の第2ゼロ拡張領域1170の部分の論理値を0とすることで、ビット列dataB1130を生成して出力する。

[0142] 図13は、命令デコーダ120がextrlsN命令をデコードした場合における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

[0143] 同図において、dataA1200、dataAA1210、dataAAA1220、dataB

1 2 3 0、width 1 2 4 0、及びshift_len 1 2 5 0は、それぞれ、図 8 におけるdataA 7 0 0、dataAA 7 1 0、dataAAA 7 2 0、dataB 7 3 0、width 7 4 0、及びshift_len 7 5 0と同様のものである。よって、ここではこれらの説明を省略する。

[0144] 排他的論理和ゲート 3 1 0 (図 4 参照) は、命令デコーダ 1 2 0 からextrlsN命令に対応するデコード信号が入力されると、論理値 “0” のshift_dirを左右シフタ 3 2 0 に出力する。すると、左右シフタ 3 2 0 は、dataA 1 1 0 0 を左方向にshift_len 1 1 5 0 で指定されるシフト量だけシフトをして、data AA 1 1 1 0 を出力する。

[0145] また、第 1 セレクタ 4 2 0 は、命令デコーダ 1 2 0 から出力されるmask_dirが論理値 “1” であるため、減算器 4 1 0 から出力されるN-widthを選択して、sign_offsetとして出力する。すると、第 2 セレクタ 4 3 0 は、dataAA 1 2 1 0 のうちのN-widthビット目の信号をsign_dataとして出力する。さらに、第 3 セレクタ 4 4 0 は、命令デコーダ 1 2 0 から入力されるsignが論理値 “1” であるため、dataAA 1 2 1 0 のうちのN-widthビット目の信号をext_dataとして出力する。そして、デコーダ 4 5 0 は、命令デコーダ 1 2 0 からextrlsN命令に対応するデコード信号が入力されると、下位Nビットのうちの、width 1 0 4 0 の値によって指定されるビット長の最上位ビットを含むビット列が論理値 “1” となり、下位Nビットのうちのそれら以外のビットが、論理値 “0” となるext_maskを出力する。すると、第 2 ビットセレクタ 4 6 0 は、dataAA 1 2 1 0 から、width 1 1 4 0 によって指定されるビット長の最上位ビットを含むビット列を選択し、それら以外のビットにdataAA 1 2 1 0 のうちのN-widthビット目の信号を選択することで、ビット列dataAAA 1 1 2 0 を生成して出力する。よって、図 1 3 中の第 1 符号拡張領域 1 2 6 0 は、dataAA 1 2 1 0 のうちのN-widthビット目の論理値となる。

[0146] 一方、加算器 3 4 0 は、width 1 2 4 0 の値とshift_len 1 2 5 0 の値とを加算し、mask_widthを出力する。そして、マスクデータ生成回路 3 5 0 は、下位Nビットのうちの、width 1 2 4 0 +shift_len 1 2 5 0 の値によって指

定されるビット長の最上位ビットを含むビット列が論理値“1”、下位Nビットのうちのそれら以外のビットがdataAA1210のうちのN-widthビット目の論理値となるmaskを第1ビットセクタ360に出力する。

[0147] 従って、第1ビットセクタ360は、width1240の値とshift_len1250の値との和の値のビット幅の最上位ビットを含むビット列をdataAAA1220から選択し、それら以外のビットをdataA1200から選択して、dataB1230として出力する。

[0148] このように、ビット操作回路150は、命令デコーダ120がextrlsN命令をデコードした場合に、dataA1200から、width1240+shift_len1250の値によって指定されるビット長の最上位ビットを含むビット列を、shift_len1250によって指定されるシフト量だけ左シフトして、N-(width1240+shift_len1250)の値によって指定されるビット長の最下位ビットを含むビット列をシフトせず、図13中の第2符号拡張領域1270の部分に、dataAA1210のうちのN-widthビット目の信号を複製することで、ビット列dataB1230を生成して出力する。

[0149] <動作>

データ処理装置100の行う特徴的な動作に、ビット操作処理がある。以下、このビット操作処理について、図面を用いて説明する。

[0150] <ビット操作処理>

ビット操作処理は、命令フェッチユニット110（図1参照）が外部メモリからビット操作種命令を読み出した場合に、命令デコーダ120とレジスタファイル140とビット操作回路150とが協働して行う処理である。

[0151] 図14は、データ処理装置100の行うビット操作処理のフローチャートである。

[0152] ビット操作処理は、命令フェッチユニット110がビット操作種命令を外部メモリから読み出すことで開始される。

[0153] 命令フェッチユニット110は、ビット操作種命令を外部メモリから読み出すと、読み出したビット操作種命令を、命令デコーダ120へ出力する。

すると、命令デコーダ120は、入力されたビット操作種命令のデコードを開始する（ステップS1310）。

[0154] 命令デコーダ120は、ビット操作種命令をデコードすると、デコード信号read_addrとデコード信号write_addrとをレジスタファイル140へ出力し、デコード信号signとデコード信号mask_dirとデコード信号operationとデコード信号Nとデコード信号widthとデコード信号shift_lenとをビット操作回路150へ出力する（ステップS1320）。

[0155] レジスタファイル140は、命令デコーダ120からread_addrが入力されると、read_addrによって指定されるレジスタに格納されているデータを出力する（ステップS1330）。

[0156] ビット操作回路150は、レジスタファイル140から入力されるデータに対して、命令デコーダ120から入力される、sign、mask_dir、operation、N、width、及びshift_lenに基づくビット操作を実行して、ビット操作を実行するによって生成されたデータを、レジスタファイル140における、write_addrによって指定されるレジスタに書き込む（ステップS1340）。

[0157] ステップS1340の処理が終了すると、データ処理装置100は、そのビット操作処理を終了する。

[0158] <考察>

データ処理装置100は、1命令で上詰めアンパッキング処理を行うことができる。

[0159] 以下、データ処理装置100が行う上詰めアンパッキング処理の具体例について、図面を用いながら説明する。

[0160] ここで例示する具体例は、レジスタファイル140の中の第1レジスタに格納されている32ビットのデータに対して、上詰めアンパッキング処理を実行して、実行結果を、レジスタファイル140の中の第2レジスタに格納する場合の例である。

[0161] 図15は、例示する具体例における、ビット操作回路150が利用、及び生成するデータの構成を示すデータ構成図である。

- [0162] 同図において、dataA 1 4 0 0は、レジスタファイル1 4 0の中の第1レジスタに格納されている3 2ビットのデータであって、7ビットの有効ビット列1 4 0 1と9ビットの有効ビット列1 4 0 2とからなる1 6ビットのビット列と、1 6ビットの論理値“0”となる不要ビット列1 4 0 3とから構成されている。
- [0163] dataAA 1 4 1 0は、レジスタファイル1 4 0から入力されるdataA 1 4 0 0に対応して、左右シフタ3 2 0（図4参照）から出力される3 2ビットのビット列である。dataAAA 1 4 2 0は、左右シフタ3 2 0から入力されるdataAA 1 4 1 0に対応して、符号拡張回路3 3 0から出力される3 2ビットのビット列である。dataB 1 4 3 0は、符号拡張回路3 3 0から入力されるdataAAAに対応して、第1ビットセクタ3 6 0から出力される3 2ビットのビット列である。width 1 4 4 0は、命令デコーダ1 2 0から入力されるデコード信号widthによって指定されるビット長であり、後述するようにここではその値が1 6となっている。shift_len 1 4 5 0は、命令デコーダ1 2 0から入力されるデコード信号shift_lenによって指定されるシフト量であり、後述するようにここではその値が9となっている。
- [0164] レジスタファイル1 4 0の中の第1レジスタにdataA 1 4 0 0が格納されている場合において、命令デコーダ1 2 0に、オペコードがextrh32で、オペランドBが第2レジスタのレジスタ名で、オペランドAが第1レジスタのレジスタ名で、オペランドshift_lenが有効ビット列1 4 0 2のビット長となる9で、オペランドwidthがアンパック処理後のデータの単位ビット長となる1 6である命令が入力されることで、データ処理装置1 0 0は、本具体例における上詰めアンパッキング処理となるビット操作処理を開始する。
- [0165] ビット操作処理が開始されると、命令デコーダ1 2 0はデコード信号をビット操作回路1 5 0へ出力し、レジスタファイル1 4 0は、第1レジスタに格納されているdataA 1 4 0 0をビット操作回路1 5 0へ出力する。
- [0166] ビット操作回路1 5 0にデコード信号とdataA 1 4 0 0とが入力されると、左右シフタ3 2 0（図4参照）は、dataA 1 4 0 0を右方向にshift_len 1 4

50で指定される9ビット分の符号拡張を伴わないシフトをして、dataAA1410を出力する。

[0167] また、デコーダ450は、デコード信号が入力されると、width1040によって指定される16ビットのビット長となる最下位ビットを含むビット列が論理値“1”となり、それら以外のビットが論理値“0”となるext_maskを出力する。すると、第2ビットセクタ460は、dataAA1410から、最下位ビットを含む16ビットのビット列を選択し、それら以外のビットに論理値“0”を選択することで、ビット列dataAAA1420を生成して出力する。

[0168] 一方、マスクデータ生成回路350は、width1440 (=16) + shift_len1450 (=9) の値すなわち、25ビットのビット長となる最下位ビットを含むビット列が論理値“1”、それら以外のビットが論理値“0”となるmaskを第1ビットセクタ360に出力する。

[0169] 従って、第1ビットセクタ360は、25ビットのビット幅の最下位ビットを含むビット列をdataAAA1420から選択し、それら以外のビットをdataA1400から選択して、dataB1430として出力する。

[0170] その後、レジスタファイル140は、ビット操作回路150から出力されるdataB1430を第2レジスタに格納する。

[0171] このように、データ処理装置100は、第1レジスタに格納されているデータのうちの上位側16ビットにおける、7ビットの有効ビット列1401と9ビットの有効ビット列1402とのそれぞれを、16ビットのビット列の最上位ビット側に配置して第2レジスタに格納させるという上詰めアンパッキング処理を、1命令で実行することができる。

<実施の形態2>

以下、本発明に係るデータ処理装置の一実施形態として、実施の形態1に係るデータ処理装置100の一部を変形したデータ処理装置1500について説明する。

[0172] このデータ処理装置1500は、実施の形態1におけるビット操作回路1

50と同様のビット操作回路を複数備え、これら複数のビット操作回路を並列に実行させることで、SIMD (Single Instruction Multi Data) 型データ処理を行うことができる。

[0173] 以下、本実施の形態2に係るデータ処理装置1500の構成について、図面を参照しながら、実施の形態1に係るデータ処理装置100の構成との相違点を中心に説明する。

[0174] <データ処理装置1500の構成>

図16は、データ処理装置1500の主要なハードウェア構成を示すブロック図である。

[0175] 同図に示される通り、データ処理装置1500は、実施の形態1に係るデータ処理装置100から、命令デコーダ120が命令デコーダ1520に変更され、レジスタファイル140がレジスタファイル1540に変更され、アライナ1570が追加され、さらに、備えるビット操作回路が、第1ビット操作回路1551～第8ビット操作回路1558の8つとなるように変更されたものである。

[0176] 命令デコーダ1520は、実施の形態1に係る命令デコーダ120から、SIMD型ビット操作種命令をデコードする機能が追加されるように変形されたものである。

[0177] 図17は、命令デコーダ120がSIMD型ビット操作種命令をデコードする場合における、デコード対象となるSIMD型ビット操作種命令とデコード信号との対応関係を示す、デコード信号対応表1600である。

[0178] 同図において、命令形式1610は、デコード対象となるSIMD型ビット操作種命令の、ニーモニック形式における命令形式を示すものである。

[0179] SIMD型ビット操作種命令には、オペコードがshrn $N \times M$ (Nは8又は16又は32のいずれか。Mは2又は4又は8のいずれか。以下同様)となるshrn $N \times M$ 命令と、オペコードがshrnln $N \times M$ となるshrnln $N \times M$ 命令と、オペコードがshrnls $N \times M$ となるshrnls $N \times M$ 命令と、オペコードがextrh $N \times M$ となるextrh $N \times M$ 命令と、オペコードがextrln $N \times M$ となるextrln $N \times M$ 命令と、オペコードがextrls $N \times M$

となるextrlsNxM命令との6種類ある。

- [0180] そして、これらの命令は、レジスタファイル1540内のデスティネーションレジスタを指定するレジスタ名Bと、レジスタファイル1540内のソースレジスタを指定するレジスタ名Aと、N未満となる第1のビット長を示すshift_lenと、N未満となる第2のビット長を示すwidthとをオペランドとする。
- [0181] sign1620と、mask_dir1630と、operation1640と、N1650と、M1655と、width1660と、shift_len1670と、read_addr1680と、write_addr1690とのそれぞれは、命令デコーダ1520がビット操作種命令をデコードする際に出力するデコード信号である。
- [0182] これらのデコード信号のうち、sign1620と、mask_dir1630と、operation1640と、N1650と、width1660と、shift_len1670と、read_addr1680と、write_addr1690とは、それぞれ、実施の形態1における、sign220と、mask_dir230と、operation240と、N250と、width260と、shift_len270と、read_addr280と、write_addr290と同様のものである。よって、ここではこれらの説明を省略する。
- [0183] M1655は、オペコードに含まれる数字Mを2進数で示す4ビットの信号であって、レジスタファイル1540に出力される。このM1655は、レジスタファイル1540がデータの読み出し、又は書き込みを行う場合における、読み出し対象、又は書き込み対象となるレジスタの本数を示す信号である。
- [0184] 再び図16に戻って、データ処理装置1500の構成の説明を続ける。
- [0185] レジスタファイル1540は、実施の形態1に係るレジスタファイル140から、16本の汎用レジスタそれぞれのビット長が、4バイトから8バイトとなるように変形されたものである。
- [0186] アライナ1570は、命令デコーダ120とレジスタファイル1540と第1ビット操作回路1551～第8ビット操作回路1558とに接続され、以下の6つの機能を有する。

- [0187] 機能1：命令デコーダ1520から、M1655が入力された場合において、M1655が“2”を示すときに、レジスタファイル1540から64ビットのビット列が入力されると、入力ビット列のうちの上位32ビットからなるビット列を第1ビット操作回路1551に出力し、入力ビット列のうちの下位32ビットからなるビット列を第2ビット操作回路1552に出力する機能。
- [0188] 機能2：命令デコーダ1520から、M1655が入力された場合において、M1655が“2”を示すときに、第1ビット操作回路1551から32ビットのビット列（以後、ビット列Aと呼ぶ。）と第2ビット操作回路1552から32ビットのビット列（以後、ビット列Bと呼ぶ。）とが入力されると、ビット列Aが上位側となりビット列Bが下位側となる64ビットのビット列を生成して、レジスタファイル1540へ出力する機能。
- [0189] 機能3：命令デコーダ1520から、M1655が入力された場合において、M1655が“4”を示すときに、レジスタファイル1540から64ビットのビット列が入力されると、入力ビット列のうちの上位16ビットからなるビット列を32ビットのビット列となるように上位側にビット拡張して第1ビット操作回路1551に出力し、入力ビット列のうちの下位16ビットからなるビット列を32ビットのビット列となるように上位側にビット拡張して第2ビット操作回路1552に出力し、3番目に上位の16ビットからなるビット列を32ビットのビット列となるように上位側にビット拡張して第3ビット操作回路1553に出力し、最下位16ビットからなるビット列を32ビットのビット列となるように上位側にビット拡張して第4ビット操作回路1554に出力する機能。ここで、ビット拡張は、例えば、拡張する部分に論理値“0”を挿入することで実現される。
- [0190] 機能4：命令デコーダ1520から、M1655が入力された場合において、M1655が“4”を示すときに、第1ビット操作回路1551から16ビットのビット列（以後、ビット列Cと呼ぶ。）と第2ビット操作回路1552から16ビットのビット列（以後、ビット列Dと呼ぶ。）と第3ビット操

作回路 1 5 5 3 から 1 6 ビットのビット列（以後、ビット列 E と呼ぶ。）と第 4 ビット操作回路 1 5 5 4 から 1 6 ビットのビット列（以後、ビット列 F と呼ぶ。）とが入力されると、上位側から順に、ビット列 C、ビット列 D、ビット列 E、ビット列 F となる 6 4 ビットのビット列を生成して、レジスタファイル 1 5 4 0 へ出力する機能。

[0191] 機能 5 : 命令デコーダ 1 5 2 0 から、M 1 6 5 5 が入力された場合において、M 1 6 5 5 が “8” を示すときに、レジスタファイル 1 5 4 0 から 6 4 ビットのビット列が入力されると、入力ビット列のうちの最上位 1 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 1 ビット操作回路 1 5 5 1 へ出力し、入力ビット列のうちの 2 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 2 ビット操作回路 1 5 5 2 へ出力し、3 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 3 ビット操作回路 1 5 5 3 へ出力し、4 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 4 ビット操作回路 1 5 5 4 へ出力し、5 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 5 ビット操作回路 1 5 5 5 へ出力し、6 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 6 ビット操作回路 1 5 5 6 へ出力し、7 番目に上位の 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 7 ビット操作回路 1 5 5 7 へ出力し、最下位 8 ビットからなるビット列を 3 2 ビットのビット列となるように上位側にビット拡張して第 8 ビット操作回路 1 5 5 4 へ出力する機能。ここで、ビット拡張は、例えば、拡張する部分に論理値 “0” を挿入することで実現される。

[0192] 機能 6 : 命令デコーダ 1 5 2 0 から、M 1 6 5 5 が入力された場合において、M 1 6 5 5 が “8” を示すときに、第 1 ビット操作回路 1 5 5 1 から 8 ビットのビット列（以後、ビット列 G と呼ぶ。）と第 2 ビット操作回路 1 5 5 2

から8ビットのビット列（以後、ビット列Hと呼ぶ。）と第3ビット操作回路1553から8ビットのビット列（以後、ビット列Iと呼ぶ。）と第4ビット操作回路1554から8ビットのビット列（以後、ビット列Jと呼ぶ。）と第5ビット操作回路1555から8ビットのビット列（以後、ビット列Kと呼ぶ。）と第6ビット操作回路1556から8ビットのビット列（以後、ビット列Lと呼ぶ。）と第7ビット操作回路1557から8ビットのビット列（以後、ビット列Mと呼ぶ。）と第8ビット操作回路1558から8ビットのビット列（以後、ビット列Nと呼ぶ。）とが入力されると、上位側から順に、ビット列G、ビット列H、ビット列I、ビット列J、ビット列K、ビット列L、ビット列M、ビット列Nとなる64ビットのビット列を生成して、レジスタファイル1540へ出力する機能。

[0193] 第1ビット操作回路1551～第8ビット操作回路1558は、それぞれ、実施の形態1におけるビット操作回路150と同様の機能と同様の構成とを有し、それぞれ、命令デコーダ1520とアライナ1570とに接続され、それぞれ、互いに共通の命令デコーダ1520からのデコード信号が入力される。

[0194] <まとめ>

上記構成を備えるデータ処理装置1500は、1命令で、実施の形態1に係るデータ処理装置100の行うビット操作処理と同様の処理を、レジスタファイル1540のレジスタに格納されているデータに対して並列に実行することができる。すなわち、いわゆるSIMD型ビット操作処理を行うことができる。

<実施の形態3>

以下、本発明に係るデータ処理装置の一実施形態として、実施の形態1に係るデータ処理装置100の一部を変形したデータ処理装置1700について説明する。

[0195] このデータ処理装置1700は、命令デコーダが、オペコードがextrhNとextrlnとextrlsNとなる命令（以下、これらの命令を「伸張命令」と呼ぶ。）

のうちのいずれかをデコードした場合に、レジスタファイルから出力されたデータに対して、プレアンパック処理を実施してからビット操作回路に入力し、命令デコーダが、オペコードがshrn n NとshrnLNとshrnLsNとなる命令（以下、これらの命令を「収縮命令」と呼ぶ。）のうちのいずれかをデコードした場合に、ビット操作回路から出力されたデータに対して、ポストパック処理を実施してからレジスタファイルに出力するものである。

[0196] ここで、プレアンパック処理とは、アンパック処理の対象となるビット列において、ビット長を0拡張あるいは符号拡張に拡張する処理であり、ポストパック処理とは、パック処理後のビット列において、不要ビット列が格納されている部分のデータを無効なものとする処理のことをいう。

[0197] 以下、本実施の形態3に係るデータ処理装置1700の構成について、図面を参照しながら、実施の形態1に係るデータ処理装置100の構成との相違点を中心に説明する。

[0198] <データ処理装置1700の構成>

図18は、データ処理装置1700の主要なハードウェア構成を示すブロック図である。

[0199] 同図に示される通り、データ処理装置1700は、実施の形態1に係るデータ処理装置100に対して、プレアンパック回路1710とポストパック回路1720とが追加されるように変形されたものである。

[0200] 同図において、プレアンパック回路1710は、命令デコーダ120とレジスタファイル140とビット操作回路150とに接続され、以下の4つのプレアンパック機能を有する。

[0201] プレアンパック機能1：命令デコーダ120が伸張命令をデコードした場合において、命令デコーダ120から入力される、デコード信号signの論理値が“0”でデコード信号mask_dirの論理値が“0”のときに、レジスタファイル140から入力される32ビットのビット列のうちの下位Nビットのビット列について、ビット長がN/2となる最下位ビットを含むビット列を左にN/2ビットシフトして、シフト後の、ビット長がN/2となる最下位

ビットを含むビット列の部分に、論理値“0”からなるビット列を挿入して、出力する機能。

[0202] プリアンパック機能2：命令デコーダ120が伸張命令をデコードした場合において、命令デコーダ120から入力される、デコード信号signの論理値が“0”でデコード信号mask_dirの論理値が“1”のときに、レジスタファイル140から入力される32ビットのビット列のうちの下位Nビットのビット列について、ビット長が $N/2$ となる最上位ビットを含むビット列に論理値“0”を挿入して、出力する機能。

[0203] プリアンパック機能3：命令デコーダ120が伸張命令をデコードした場合において、命令デコーダ120から入力されるデコード信号signの論理値が“1”のときに、レジスタファイル140から入力される32ビットのビット列のうちの下位Nビットのビット列について、ビット長が $N/2$ ビットとなる最上位ビットを含むビット列に、ビット長が $N/2$ となる最下位ビットを含むビット列の最上位ビットの複製を挿入して、出力する機能。

[0204] プリアンパック機能4：命令デコーダ120が収縮命令をデコードした場合に、レジスタファイル140から入力される32ビットのビット列を、そのまま出力する機能。

[0205] ポストパック回路1720は、命令デコーダ120とビット操作回路150とレジスタファイル140とに接続され、以下の3つのポストパック機能を有する。

[0206] ポストパック機能1：命令デコーダ120が収縮命令をデコードした場合において、命令デコーダから出力されるデコード信号mask_dirが論理値“0”のときに、ビット操作回路150から入力される32ビットのビット列のうちの下位Nビットのビット列について、ビット長が $N/2$ となる最上位ビットを含むビット列を右に $N/2$ ビットシフトして、シフト後の、ビット長が $N/2$ となる最上位ビットを含むビット列の部分に、論理値“0”からなる無効ビット列を挿入して、出力する機能。

[0207] ポストパック機能2：命令デコーダ120が収縮命令をデコードした場合

において、命令デコーダから出力されるデコード信号mask_dirが論理値“1”のときに、ビット操作回路150から入力される32ビットのビット列のうちの下位Nビットのビット列について、ビット長が $N/2$ となる最上位ビットを含むビット列に論理値“0”からなる無効ビット列を挿入して出力する機能。

[0208] ポストパック機能3：命令デコーダ120が伸張命令をデコードした場合に、ビット操作回路150から入力される32ビットのビット列を、そのまま出力する機能。

[0209] <まとめ>

上記構成を備えるデータ処理装置1700は、命令デコーダ120が伸張命令をデコードした場合に、プレアンパック回路1710がプレアンパック処理を実施した後に、ビット操作回路150がビット操作を行い、命令デコーダ120が収縮命令をデコードした場合に、ビット操作回路150がビット操作を行った後に、ポストパック回路1720がポストパック処理を行うことができる。

<補足>

以上、本発明に係るデータ処理装置の一実施形態として、実施の形態1、実施の形態2において、データ処理装置の例について説明したが、以下のように変形することも可能であり、本発明は上述した実施の形態で示した通りのデータ処理装置に限られないことはもちろんである。

(1) 実施の形態1において、命令デコーダ120がデコード対象とする命令は、4バイト、すなわち32ビットのビット長となる構成の例について説明したが、ビット操作種命令を表現することができるビット数であれば、必ずしも32ビットのビット長となる構成に限られず、例えば64ビットのビット長となる構成であっても構わない。さらには、命令デコーダがデコード対象とする命令は、可変長命令となる構成であっても構わない。

(2) 実施の形態1において、ビット操作回路150は、32ビットのビット長のデータに対してビット操作処理を行う構成の例について説明したが、

命令デコーダ120からのデコード信号に基づいて、ビット操作処理を行うことができれば、必ずしも32ビットのビット長のデータに対してビット操作処理を行う構成に限られず、例えば、128ビットや16ビットのビット長のデータに対してビット操作処理を行う構成であっても構わない。

(3) 実施の形態1において、オペコードに含まれる“N”は、8又は16又は32のいずれかである構成の例について説明したが、ビット操作回路150のビット操作処理対象となるデータのビット数以下であれば、必ずしも、8又は16又は32のいずれかである構成に限られず、例えば、12となる構成であっても構わない。

(4) 実施の形態1において、レジスタファイル140は、レジスタを16本で有する構成の例について説明したが、命令デコーダ120からのデコード信号に基づいて、レジスタの値を読み出し、又は書き込みを行うことができれば、必ずしもレジスタを16本有する構成に限られず、例えば、レジスタを32本有する構成であっても構わない。

(5) 実施の形態1において、ビット操作回路150は、図4、図5で示される構成の例について説明したが、図4、図5で示される構成の回路と同様の機能を有する回路であれば、必ずしも図4、図5で示される通りの構成に限られない。一例として、図4、図5で示されるビット操作回路150の有する機能と同様の機能を記述するRTL (Resister Transfer Language) 記述に対して、論理合成ツールを利用して得られる論理合成回路が考えられる。

(6) 実施の形態1において、mask_dirとoperationとshift_dirとは、これら3つの信号のうちの任意の2つの信号によって、他の1つの信号が一意に決定されるという関係にある。

[0210] よって、実施の形態1において、データ処理装置100は、命令デコーダ120がmask_dirとoperationとをデコード信号として出力し、ビット操作回路150がこれらの信号からshift_dirを生成して利用する構成の例について説明したが、命令デコーダ120が、mask_dirとoperationとshift_dirとの

うちのいずれか2つの信号をデコード信号として出力し、ビット操作回路150がそれら2つの信号から他の1つの信号を生成して利用する構成であれば、必ずしも、命令デコーダ120がmask_dirとoperationとをデコード信号として出力し、ビット操作回路150がこれらの信号からshift_dirを生成して利用する構成に限られない。

[0211] さらに、命令デコーダ120から出力されるデコード信号によって、(1) デコード対象の命令が、収縮命令と伸張命令とのいずれであるかと、(2) シフト対象とならないビット列が、最上位ビットを含むビット列と最下位ビットを含むビット列とのいずれであるかとを特定することができれば、必ずしも、命令デコーダ120が、mask_dirとoperationとshift_dirとのうちのいずれか2つの信号をデコード信号として出力する構成に限られない。例えば、デコード対象の命令が、収縮命令と伸張命令とのいずれであることを示す信号(operation)と、シフト対象となるビット列のシフト方向が、上位ビット側と下位ビット側とのいずれの方向であることを示す信号との2つの信号をデコード信号として出力する構成の例等が考えられる。

(7) 実施の形態1において、widthとshift_lenとmask_widthとは、これら3つの信号のうちの任意の2つの信号によって、他の1つの信号が一意に決定されるという関係にある。

[0212] よって、実施の形態1において、データ処理装置100は、命令デコーダ120がwidthとshift_lenとをデコード信号として出力し、ビット操作回路150がこれらの信号からmask_widthを生成して利用する構成の例について説明したが、命令デコーダ120が、widthとshift_lenとmask_widthとのうちのいずれか2つの信号をデコード信号として出力し、ビット操作回路150がそれら2つの信号から他の1つの信号を生成して利用する構成であれば、必ずしも、命令デコーダ120がwidthとshift_lenとをデコード信号として出力し、ビット操作回路150がこれらの信号からmask_widthを生成して利用する構成に限られない。

[0213] さらに、命令デコーダ120から出力されるデコード信号によって、シ

フト対象となるビット列のビット長と、シフト対象とならないビット列のビット長とを特定することができれば、必ずしも、命令デコーダ120が、widthとshift_lenとmask_widthとのうちのいずれか2つをデコード信号として出力する構成に限られない。例えば、シフト対象となるビット列のビット長を示す信号（width）と、シフト対象とならないビット列のビット長を示す信号との2つの信号をデコード信号として出力する構成の例が考えられる。また、例えば、シフト対象とならないビット列のビット長を示す信号と、シフト対象とならないビット列のビット長とシフト対象となるビット列のビット長との差を示す信号との2つの信号をデコード信号として出力する構成の例等が考えられる。

（8）実施の形態1において、命令デコーダ120のデコード対象となるビット操作種命令は、ソースレジスタとデスティネーションレジスタとを互いに独立に指定する構成の例について説明したが、ソースレジスタとデスティネーションレジスタとが同一のレジスタとなるように、ソースレジスタとデスティネーションレジスタとを単一のレジスタとして指定する構成であっても構わない。

（9）実施の形態1において、データ処理装置100は、ビット操作処理の対象となるビット列がレジスタファイル140のレジスタに格納されているものである構成の例について説明したが、一意に特定することができる記憶領域に格納されているビット列であれば、必ずしもビット操作処理の対象となるビット列がレジスタファイル140のレジスタに格納されているものである構成に限られず、例えば、メモリ等の外部記憶装置に格納されているビット列をビット操作処理の対象とする構成であっても構わない。

（10）実施の形態2において、データ処理装置1500は、8つのビット操作回路を備える構成の例について説明したが、それぞれのビット操作回路が、互いに共通の命令デコーダ120からのデコード信号に基づいて動作する構成であれば、必ずしも8つのビット操作回路を備える構成に限られず、例えば、16のビット操作回路を備える構成であっても構わない。

(11) 実施の形態3において、命令デコーダ120のデコード対象となるビット操作種命令は、図3に示されるものである構成の例について説明したが、命令デコーダ120がレジスタファイル140とビット操作回路150とプレアンパック回路1710とを制御する信号をデコードすることができれば、プレアンパック回路1710を動作させるための部分と、ビット操作回路150を動作させるための部分とを1つの命令に含む複合命令となる構成であっても構わない。また、命令デコーダ120のデコード対象となるビット操作種命令は、図3に示されるものである構成の例について説明したが、命令デコーダ120がレジスタファイル140とビット操作回路150とポストパック回路1720とを制御する信号をデコードすることができれば、ポストパック回路1720を動作させるための部分と、ビット操作回路150を動作させるための部分とを1つの命令に含む複合命令となる構成であっても構わない。

(12) 実施の形態1において、データ処理装置100は、命令デコーダ120が、図3に示される命令形式210からなるビット操作種命令をデコード対象とする構成の例であった。しかしながら、命令デコーダ120が、ビット操作回路150が必要とする全ての制御信号を含むデコード信号を出力することができれば、デコード対象とするビット操作種命令の命令形式は、必ずしも図3に示される命令形式210に限られる必要はない。一例として、図19に例示される命令形式の例等が考えられる。

[0214] 図19(a)に例示される命令形式は、デコード対象となるビット操作種命令が伸張命令である場合における命令形式をニーモニック形式で模式的に示したものであり、図19(b)に例示される命令形式は、デコード対象となるビット操作種命令が収縮命令である場合における命令形式をニーモニック形式で模式的に示したものである。

[0215] 図19(a)、(b)に示されるように、これらの命令形式は、1つのオペコードと3つのオペランドとから命令が構成される。

[0216] ここで、オペランドsh15、オペランドsh25で示されるシフト量は

、任意の値を指定し得る形式であっても構わないし、複数の固定シフト量のうちのいずれか1つを指定する形式であっても構わない。また、オペランド $m s k 1 6$ 、オペランド $m s k 2 6$ で示されるビット長は、任意の値を指定し得る形式であっても構わないし、複数の固定ビット長のうちのいずれか1つを指定する形式であっても構わない。

(13) 以下、さらに本発明の一実施形態に係るデータ処理装置の構成及びその変形例と各効果について説明する。

[0217] (a) 本発明の一実施形態に係るデータ処理装置は、1命令から、 N (N は2以上の整数) ビットの対象ビット列を格納するビット列格納領域を指定する情報と、 N ビットのビット列に含まれる、当該 N ビットのビット列における第1の端ビットを含む第1ビット範囲を示す情報と、当該 N ビットのビット列に含まれる、当該第1ビット範囲に連続するビット範囲であって、当該 N ビットのビット列における第2の端ビットを含まない第2ビット範囲を示す情報とを読み取って、読み取った情報に応じたデコード信号を出力するデコーダと、前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に格納される前記対象ビット列に基づく、 N ビットからなる出力ビット列を生成して出力するビット操作回路とを備え、前記ビット操作回路は、前記出力ビット列における前記第1ビット範囲に、前記対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第2の端ビットを含む、前記第2ビット範囲のビット長からなる第3ビット範囲に、前記対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の所定値を配置することで、前記出力ビット列の生成を行うことを特徴とする。

[0218] 上述の構成を備える本実施形態に係るデータ処理装置によると、1命令からデコードされたデコード信号に応じて動作するビット操作回路が、ビット列格納領域に格納されるビット列のうちの第1ビット範囲と第2ビット範囲

とにパッキングされているデータを、第1ビット範囲と第2ビット範囲との間に同一の所定値からなるビット列を挿入するようにアンパッキングして出力することとなる。このことにより、このデータ処理装置は、パッキングされたデータのアンパッキングを比較的効率良く実行することができるようになる。

[0219] 図20は、上記変形例におけるデータ処理装置2200の構成を示すブロック図である。

[0220] 同図に示されるように、データ処理装置2200は、デコーダ2210とビット操作回路2220とから構成される。

[0221] デコーダ2210は、ビット操作回路2220に接続され、1命令から、N（Nは2以上の整数）ビットの対象ビット列を格納するビット列格納領域を指定する情報と、Nビットのビット列に含まれる、当該Nビットのビット列における第1の端ビットを含む第1ビット範囲を示す情報と、当該Nビットのビット列に含まれる、当該第1ビット範囲に連続するビット範囲であって、当該Nビットのビット列における第2の端ビットを含まない第2ビット範囲を示す情報とを読み取って、読み取った情報に応じたデコード信号を出力する機能を有する。一例として、実施の形態1における命令デコーダ120として実現される。

[0222] ビット操作回路2220は、デコーダ2210に接続される、デコーダ2210からデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に格納される前記対象ビット列に基づく、Nビットからなる出力ビット列を生成して出力するビット操作回路であって、前記出力ビット列における前記第1ビット範囲に、前記対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第2の端ビットを含む、前記第2ビット範囲のビット長からなる第3ビット範囲に、前記対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の所定値を配置

することで、前記出力ビット列の生成を行う機能を有する。一例として、実施の形態1におけるビット操作回路150として実現される。

[0223] (b) また、前記デコーダは、さらに、1命令から、符号拡張を行わせるか否かを示す符号拡張情報を読み取って、当該符号拡張情報に応じた、符号拡張を行わせるか否かを示す符号拡張信号を、出力するデコード信号に含ませ、前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれる符号拡張信号が符号拡張を行わせる旨を示す場合に、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのうち、より下位側となるビット範囲に配置されることとなるビット列における最上位ビットの値を前記所定値とし、前記デコーダから入力されるデコード信号に含まれる前記符号拡張信号が符号拡張を行わせる旨を示さない場合に、数値ゼロを前記所定値とするとしてもよい。

[0224] このような構成にすることによって、デコーダのデコード対象となる1命令に符号拡張情報を含ませることで、データ処理装置に対して、符号拡張をする処理とゼロ挿入をする処理とのいずれの処理をさせるかを指定することができるようになる。

[0225] (c) また、前記デコーダによるデコード対象となる1命令は、前記第1ビット範囲のビット長と前記第2ビット範囲のビット長とを指定するためのビット長信号を含むビット列であり、前記デコーダは、前記デコード信号の出力を、当該デコード信号に、前記ビット長信号に応じた、前記第1ビット範囲のビット長と前記第2ビット範囲のビット長とを指定するためのビット長デコード信号を含ませて行い、前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれるビット長デコード信号に応じて、前記出力ビット列の生成を行うとしてもよい。

[0226] このような構成にすることによって、第1ビット範囲のビット長と第2ビット範囲のビット長との指定を、ビット長信号を用いて行うことができるようになる。

[0227] (d) また、前記ビット長信号は、前記第1ビット範囲のビット長を示す

信号と、前記第2ビット範囲のビット長を示す信号と、Nビットからなるビット列における、前記第2ビット範囲と前記第3ビット範囲との差分ビット数を示す信号とのうちのいずれか2つを含むとしてもよい。

[0228] このような構成にすることによって、第1ビット範囲のビット長と第2ビット範囲のビット長との指定を、第1ビット範囲のビット長を示す信号と、第2ビット範囲のビット長を示す信号と、Nビットからなるビット列における、第2ビット範囲と第3ビット範囲との差分ビット数を示す信号とのうちのいずれか2つを用いて行うことができるようになる。

[0229] (e) また、前記デコーダによるデコード対象となる1命令は、さらに、前記第1の端ビットが、最上位ビットと最下位ビットとのいずれであるかを示す端ビット信号を含み、前記デコーダは、前記デコード信号の出力を、さらに、当該デコード信号に、前記端ビット信号に応じた、最上位ビットと最下位ビットとのいずれであるかを示す端ビットデコード信号を含ませて行い、前記ビット操作回路は、さらに、前記デコーダから出力されるデコード信号に含まれる端ビットデコード信号を利用して、前記出力ビット列の生成を行うとしてもよい。

[0230] このような構成にすることによって、第1の端ビットが最上位ビットと最下位ビットとのいずれであるかの指定を、端ビット信号を用いて行うことができるようになる。

[0231] (f) また、前記デコーダは、さらに、1命令から、プレアンパック処理を行わせるか否かを示すプレアンパック情報を読み取って、当該プレアンパック情報に応じた、プレアンパック処理を行わせるか否かを示すプレアンパック信号を、出力するデコード信号に含ませ、前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれるプレアンパック信号がプレアンパック処理を行わせる旨を示す場合に、プレアンパック処理を行うとしてもよい。

[0232] このような構成にすることによって、デコーダのデコード対象となる1命令にプレアンパック情報を含ませることで、データ処理装置にプレアンパッ

ク処理をさせることができるようになる。

[0233] (g) また、前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に対応付けられた追加ビット列格納領域に格納されるNビットからなる追加対象ビット列に基づく、Nビットからなる追加出力ビット列を生成して出力する追加ビット操作回路を備え、前記追加ビット操作回路は、前記追加出力ビット列における前記第1ビット範囲に、前記追加対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記追加出力ビット列における前記第3ビット範囲に、前記追加対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記追加出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の追加所定値を配置することで、前記追加出力ビット列の生成を行うとしてもよい。

[0234] このような構成にすることによって、1命令で、ビット列格納領域に格納される対象ビット列と追加ビット列格納領域に格納される追加対象ビット列とから、これらのビット列のそれぞれに基づく、出力ビット列と追加出力ビット列とを生成して出力することができるようになる。

産業上の利用可能性

[0235] 本発明に係るデータ処理装置は、複数ビットからなるデジタル信号を取り扱う機器に広く利用することができる。

符号の説明

- [0236] 100 データ処理装置
110 命令フェッチユニット
120 命令デコーダ
130 ロードストアユニット
140 レジスタファイル
150 ビット操作回路
160 クロックジェネレータ

請求の範囲

[請求項1]

1 命令から、 N (N は2以上の整数) ビットの対象ビット列を格納するビット列格納領域を指定する情報と、 N ビットのビット列に含まれる、当該 N ビットのビット列における第1の端ビットを含む第1ビット範囲を示す情報と、当該 N ビットのビット列に含まれる、当該第1ビット範囲に連続するビット範囲であって、当該 N ビットのビット列における第2の端ビットを含まない第2ビット範囲を示す情報とを読み取って、読み取った情報に応じたデコード信号を出力するデコーダと、

前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に格納される前記対象ビット列に基づき、 N ビットからなる出力ビット列を生成して出力するビット操作回路とを備え、

前記ビット操作回路は、前記出力ビット列における前記第1ビット範囲に、前記対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第2の端ビットを含む、前記第2ビット範囲のビット長からなる第3ビット範囲に、前記対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の所定値を配置することで、前記出力ビット列の生成を行う

ことを特徴とするデータ処理装置。

[請求項2]

前記デコーダは、さらに、1命令から、符号拡張を行わせるか否かを示す符号拡張情報を読み取って、当該符号拡張情報に応じた、符号拡張を行わせるか否かを示す符号拡張信号を、出力するデコード信号に含ませ、

前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれる符号拡張信号が符号拡張を行わせる旨を示す場合に、前記

出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのうち、より下位側となるビット範囲に配置されることとなるビット列における最上位ビットの値を前記所定値とし、前記デコーダから入力されるデコード信号に含まれる前記符号拡張信号が符号拡張を行わせる旨を示さない場合に、数値ゼロを前記所定値とする

ことを特徴とする請求項1記載のデータ処理装置。

[請求項3]

前記デコーダによるデコード対象となる1命令は、前記第1ビット範囲のビット長と前記第2ビット範囲のビット長とを指定するためのビット長信号を含むビット列であり、

前記デコーダは、前記デコード信号の出力を、当該デコード信号に、前記ビット長信号に応じた、前記第1ビット範囲のビット長と前記第2ビット範囲のビット長とを指定するためのビット長デコード信号を含ませて行い、

前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれるビット長デコード信号に応じて、前記出力ビット列の生成を行う

ことを特徴とする請求項2記載のデータ処理装置。

[請求項4]

前記ビット長信号は、前記第1ビット範囲のビット長を示す信号と、前記第2ビット範囲のビット長を示す信号と、Nビットからなるビット列における、前記第2ビット範囲と前記第3ビット範囲との差分ビット数を示す信号とのうちのいずれか2つを含む

ことを特徴とする請求項3記載のデータ処理装置。

[請求項5]

前記デコーダによるデコード対象となる1命令は、さらに、前記第1の端ビットが、最上位ビットと最下位ビットとのいずれであるかを示す端ビット信号を含み、

前記デコーダは、前記デコード信号の出力を、さらに、当該デコード信号に、前記端ビット信号に応じた、最上位ビットと最下位ビットとのいずれであるかを示す端ビットデコード信号を含ませて行い、

前記ビット操作回路は、さらに、前記デコーダから出力されるデコード信号に含まれる端ビットデコード信号を利用して、前記出力ビット列の生成を行う

ことを特徴とする請求項3記載のデータ処理装置。

[請求項6]

前記デコーダは、さらに、1命令から、プレアンパック処理を行わせるか否かを示すプレアンパック情報を読み取って、当該プレアンパック情報に応じた、プレアンパック処理を行わせるか否かを示すプレアンパック信号を、出力するデコード信号に含ませ、

前記ビット操作回路は、前記デコーダから出力されるデコード信号に含まれるプレアンパック信号がプレアンパック処理を行わせる旨を示す場合に、プレアンパック処理を行う

ことを特徴とする請求項2記載のデータ処理装置。

[請求項7]

前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に対応付けられた追加ビット列格納領域に格納されるNビットからなる追加対象ビット列に基づく、Nビットからなる追加出力ビット列を生成して出力する追加ビット操作回路を備え、

前記追加ビット操作回路は、前記追加出力ビット列における前記第1ビット範囲に、前記追加対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記追加出力ビット列における前記第3ビット範囲に、前記追加対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記追加出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の追加所定値を配置することで、前記追加出力ビット列の生成を行う

ことを特徴とする請求項1記載のデータ処理装置。

[請求項8]

命令をデコードするデコーダと、当該デコーダから出力されるデコード信号に応じた処理を行うビット操作回路とを含むデータ処理装置

が行うデータ処理方法であって、

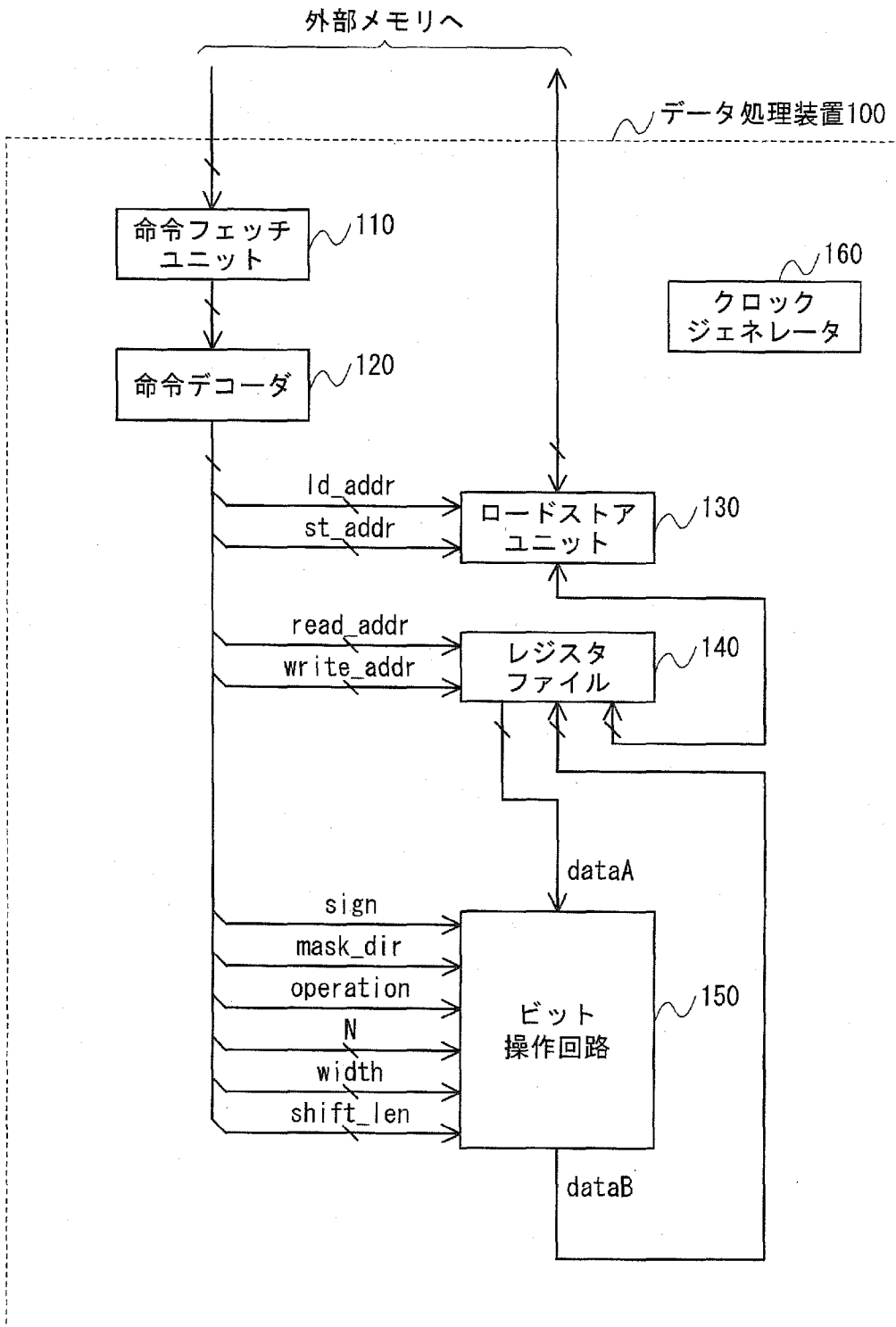
前記デコーダが、1命令から、 N (N は2以上の整数)ビットの対象ビット列を格納するビット列格納領域を指定する情報と、 N ビットのビット列に含まれる、当該 N ビットのビット列における第1の端ビットを含む第1ビット範囲を示す情報と、当該 N ビットのビット列に含まれる、当該第1ビット範囲に連続するビット範囲であって、当該 N ビットのビット列における第2の端ビットを含まない第2ビット範囲を示す情報とを読み取って、読み取った情報に応じたデコード信号を出力するデコードステップと、

前記ビット操作回路が、前記デコードステップにおいて前記デコーダからデコード信号が出力されると、当該デコード信号に応じて、前記ビット列格納領域に格納される前記対象ビット列に基づく、 N ビットからなる出力ビット列を生成して出力するビット操作ステップとを含み、

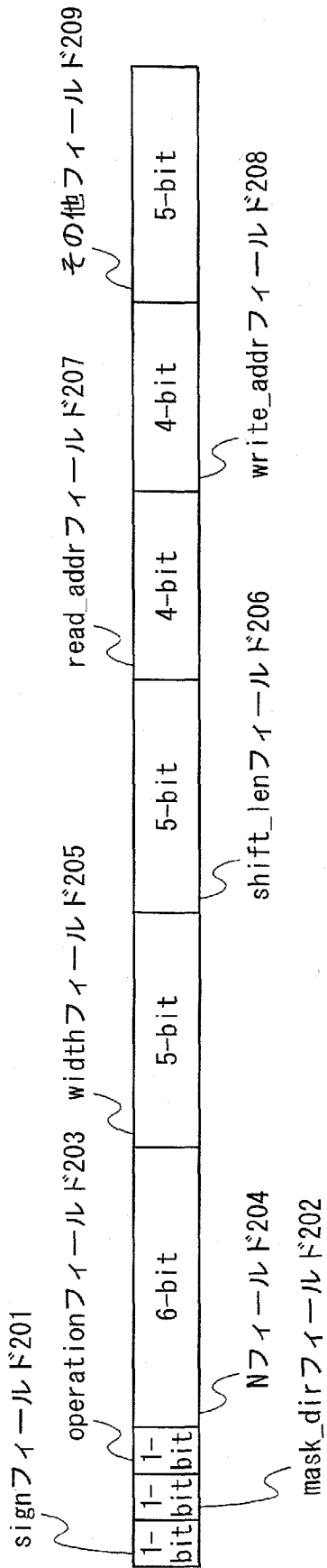
前記ビット操作ステップは、前記出力ビット列における前記第1ビット範囲に、前記対象ビット列における前記第1ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第2の端ビットを含む、前記第2のビット範囲のビット長からなる第3ビット範囲に、前記対象ビット列における前記第2ビット範囲のビット列と同値のビット列を配置し、前記出力ビット列における、前記第1ビット範囲と前記第3ビット範囲とのいずれにも属さない部分に、同一の所定値を配置することで、前記出力ビット列の生成を行う

ことを特徴とするデータ処理方法。

[図1]



[図2]

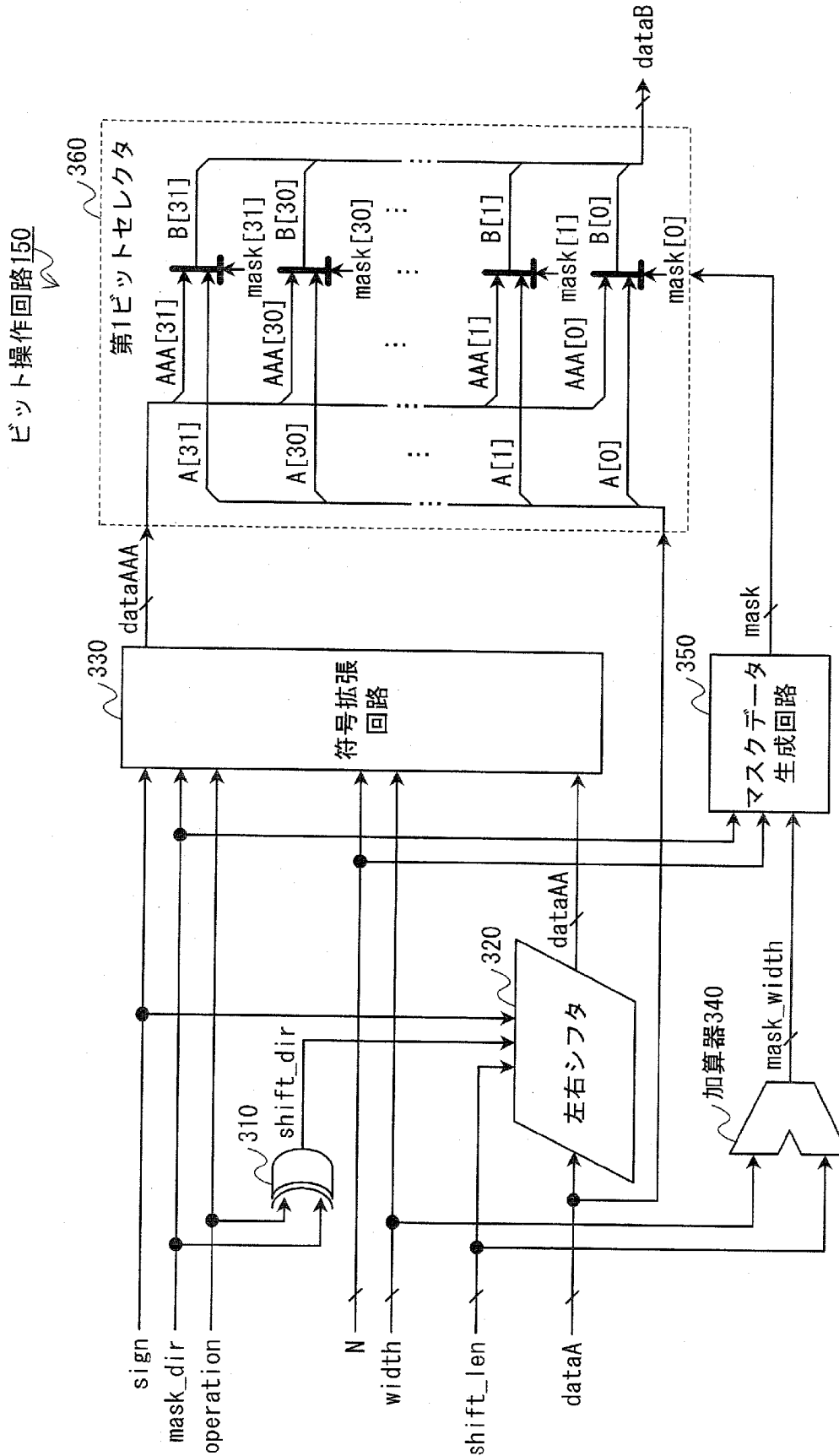


[図3]

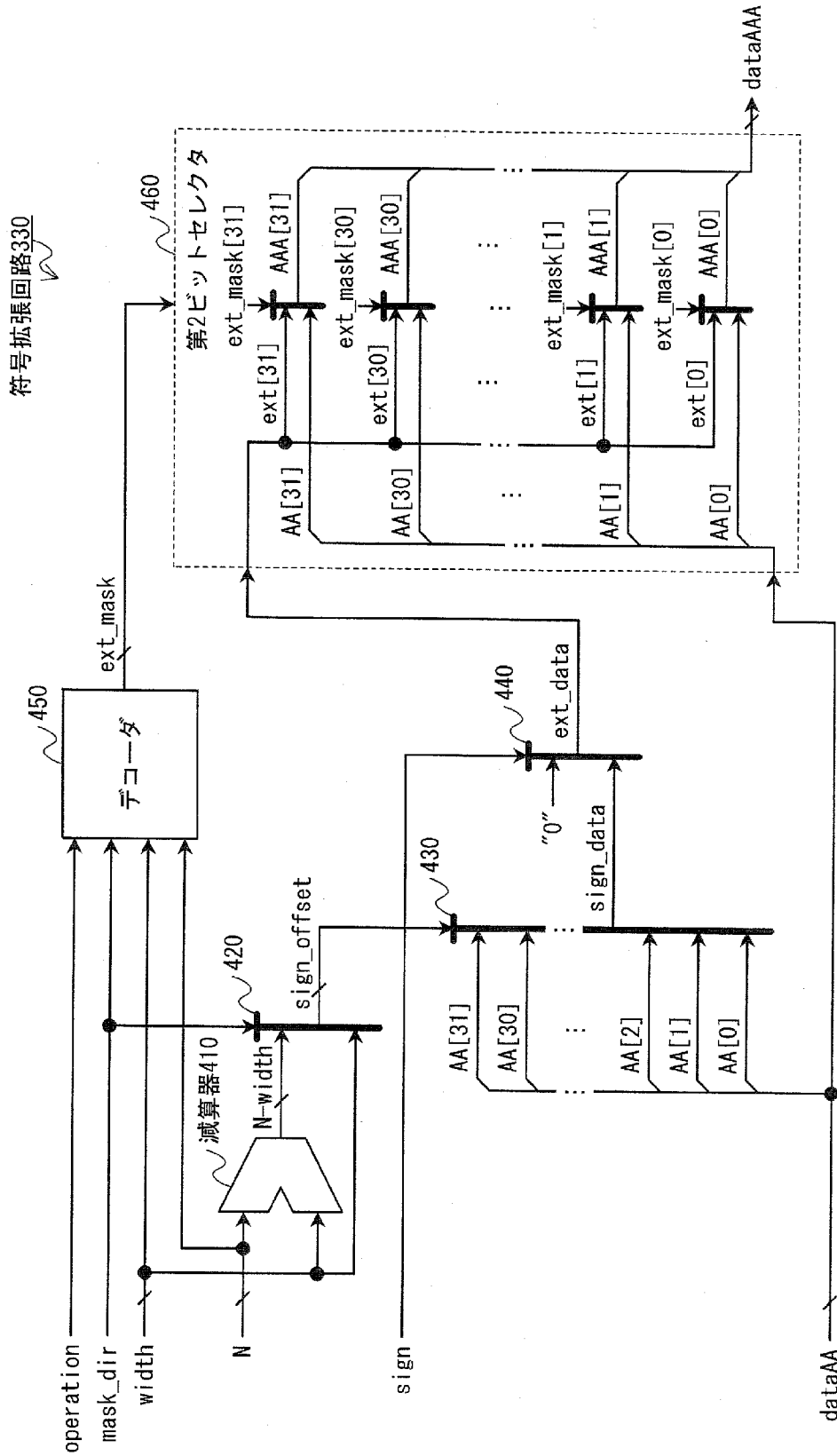
デコード信号対応表200

命令形式	220	230	240	250	260	270	280	290
	sign	mask_dir	operation	N	width	shift_len	read_addr	write_addr
shrnhN B, A, shift_len, width	0 (ゼロ拡張)	0 (MSB固定)	0 (収縮命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB
shrnln B, A, shift_len, width	0 (ゼロ拡張)	1 (LSB固定)	0 (収縮命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB
shrnlsN B, A, shift_len, width	1 (符号拡張)	1 (LSB固定)	0 (収縮命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB
extrhN B, A, shift_len, width	0 (ゼロ拡張)	0 (MSB固定)	1 (伸張命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB
extrln B, A, shift_len, width	0 (ゼロ拡張)	1 (LSB固定)	1 (伸張命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB
extrlsN B, A, shift_len, width	1 (符号拡張)	1 (LSB固定)	1 (伸張命令)	8or16or32 (Nの値)	width	shift_len	addrA	addrB

[図4]



[図5]



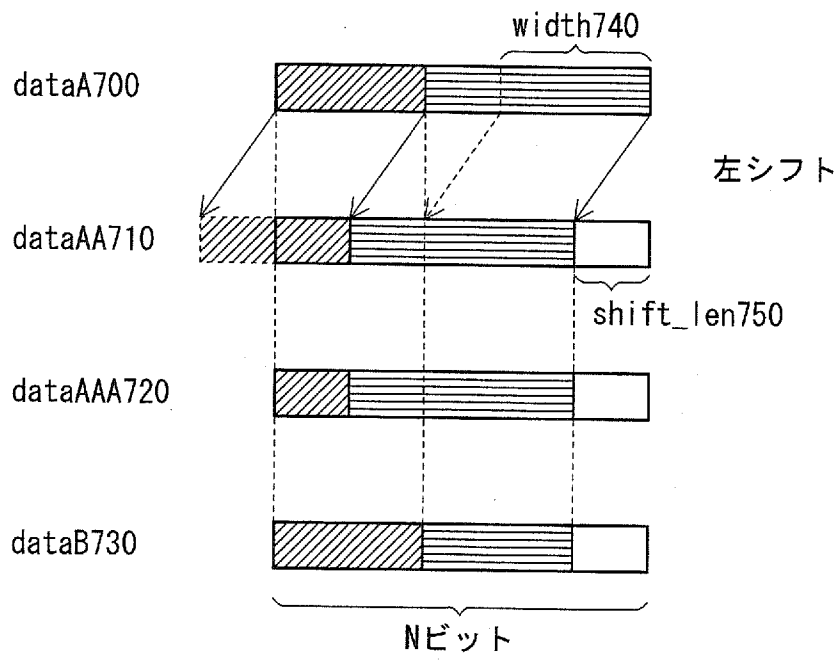
[図6]

510	520	530	540
operation	mask_dir	width	ext_mask (N bit)
0	X	X	0000.... 0000
1	0	0	0000.... 0000
1	0	1	0000.... 0001
1	0	2	0000.... 0011
...
1	0	N-2	0111.... 1111
1	0	N-1	1111.... 1111
1	1	0	0000.... 0000
1	1	1	1000.... 0000
1	1	2	1100.... 0000
...
1	1	N-2	1111.... 1110
1	1	N-1	1111.... 1111

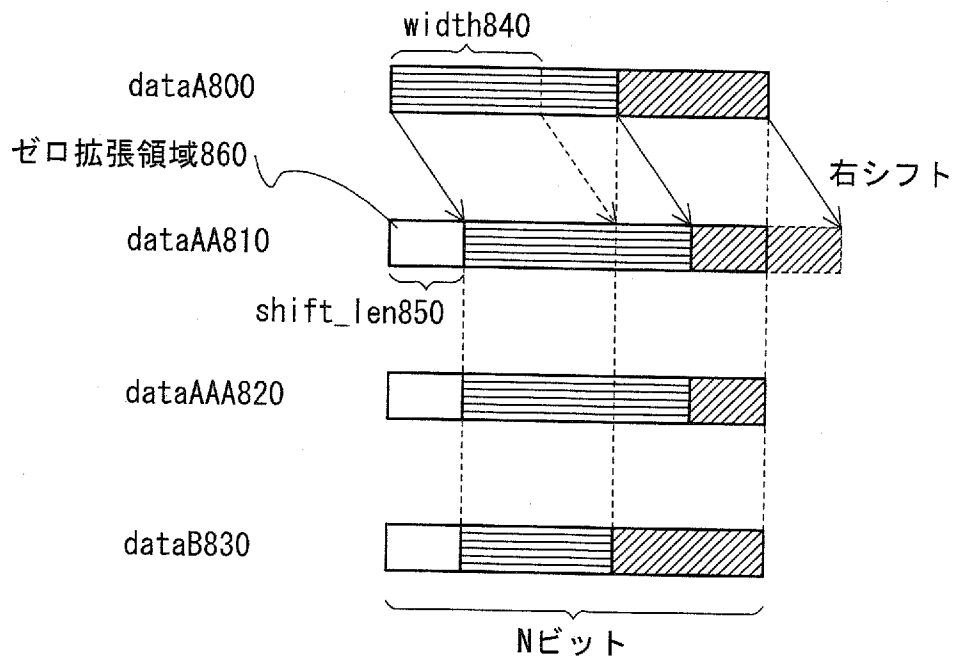
[図7]

mask_dir	mask_width	mask (N bit)
0	0	0000...0000
0	1	0000...0001
0	2	0000...0011
...
0	N-2	0111...1111
0	N-1	1111...1111
1	0	0000...0000
1	1	1000...0000
1	2	1100...0000
...
1	N-2	1111...1110
1	N-1	1111...1111

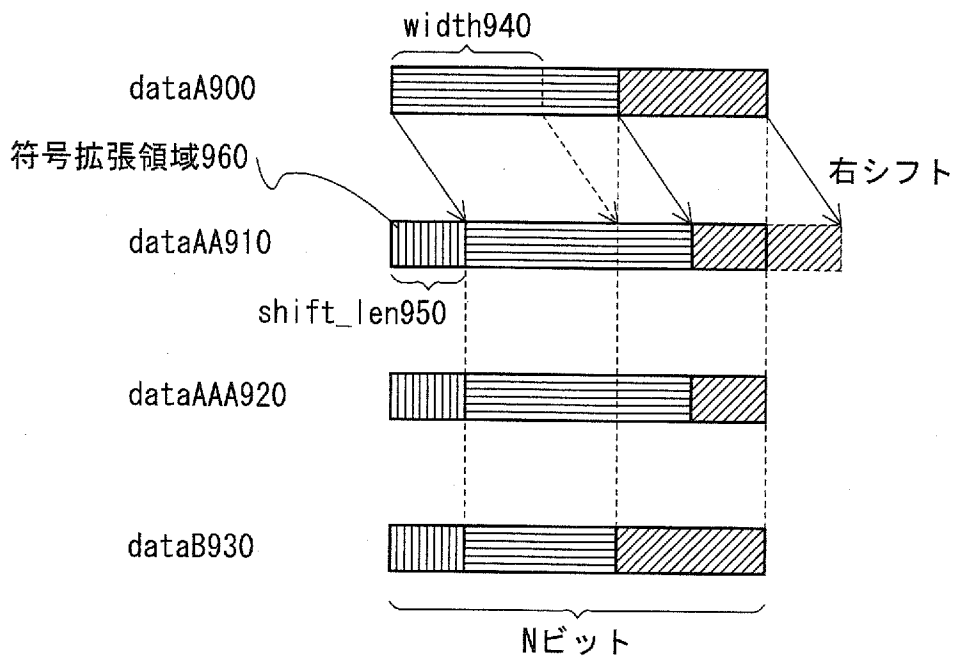
[図8]



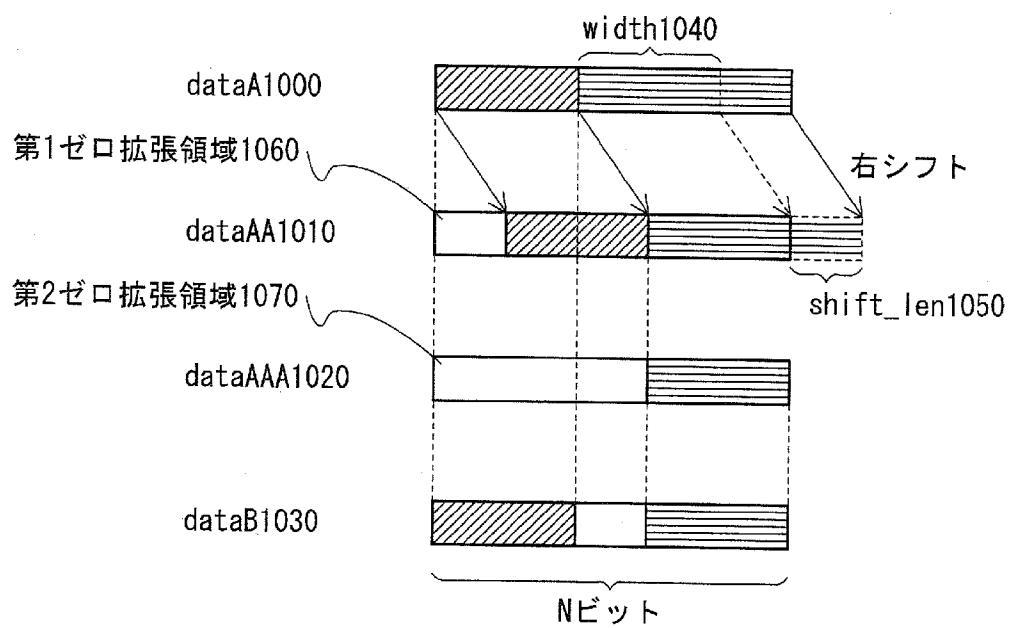
[図9]



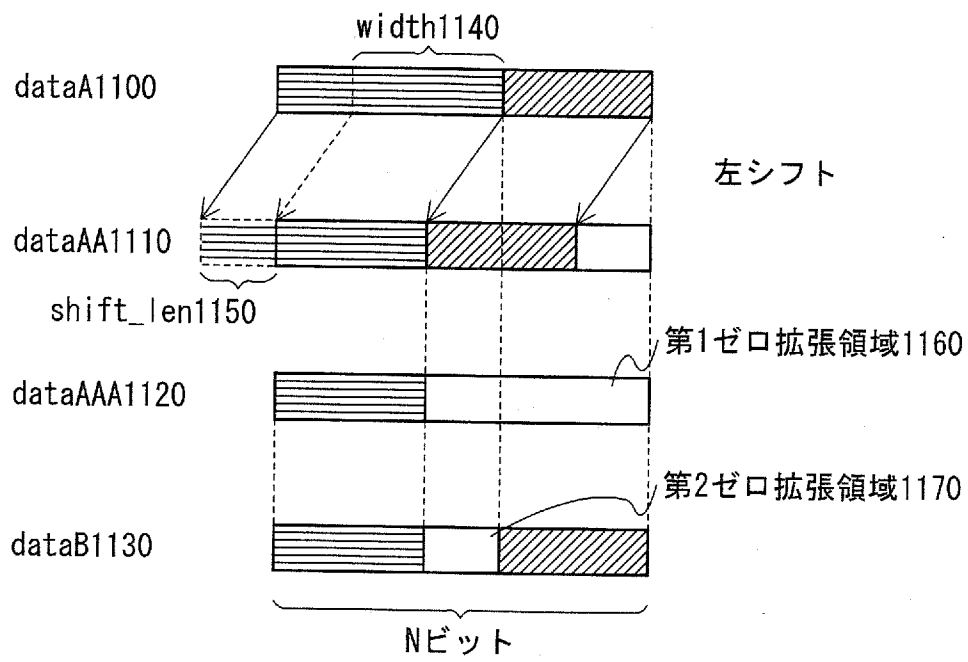
[図10]



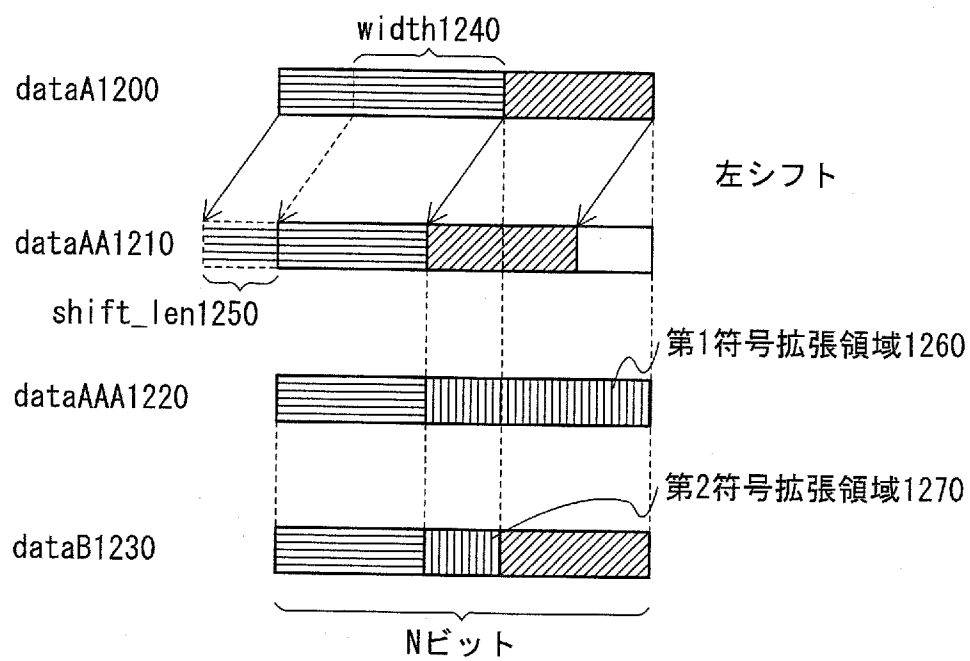
[図11]



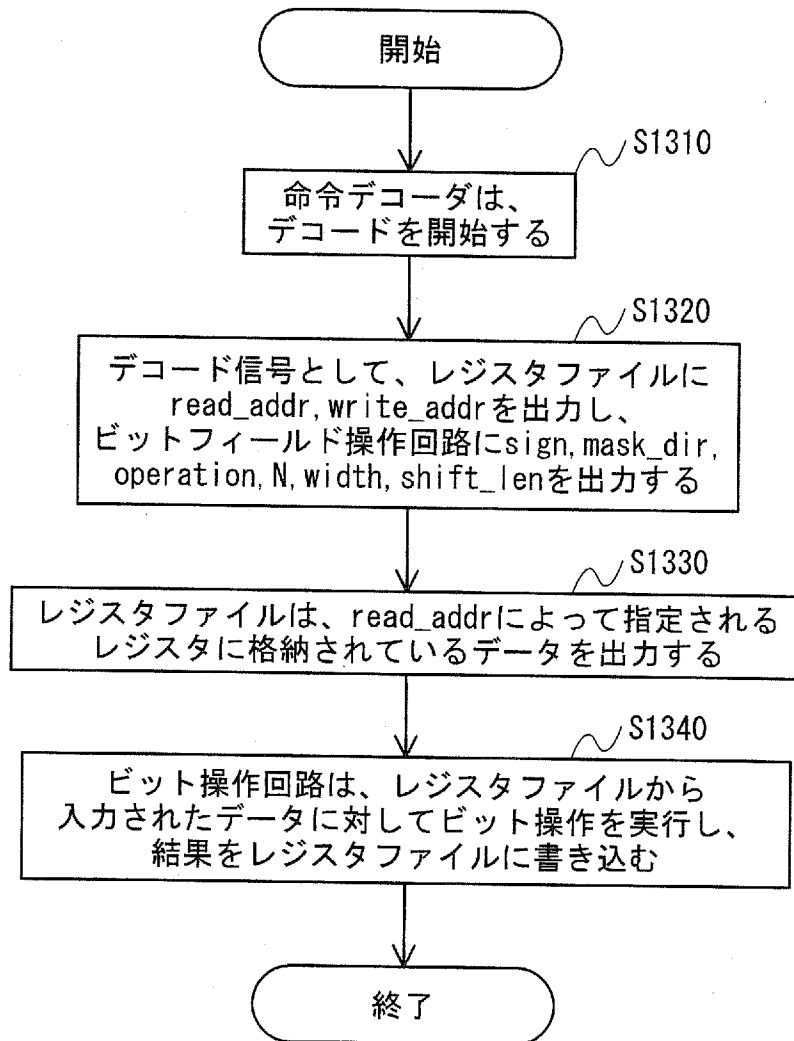
[図12]



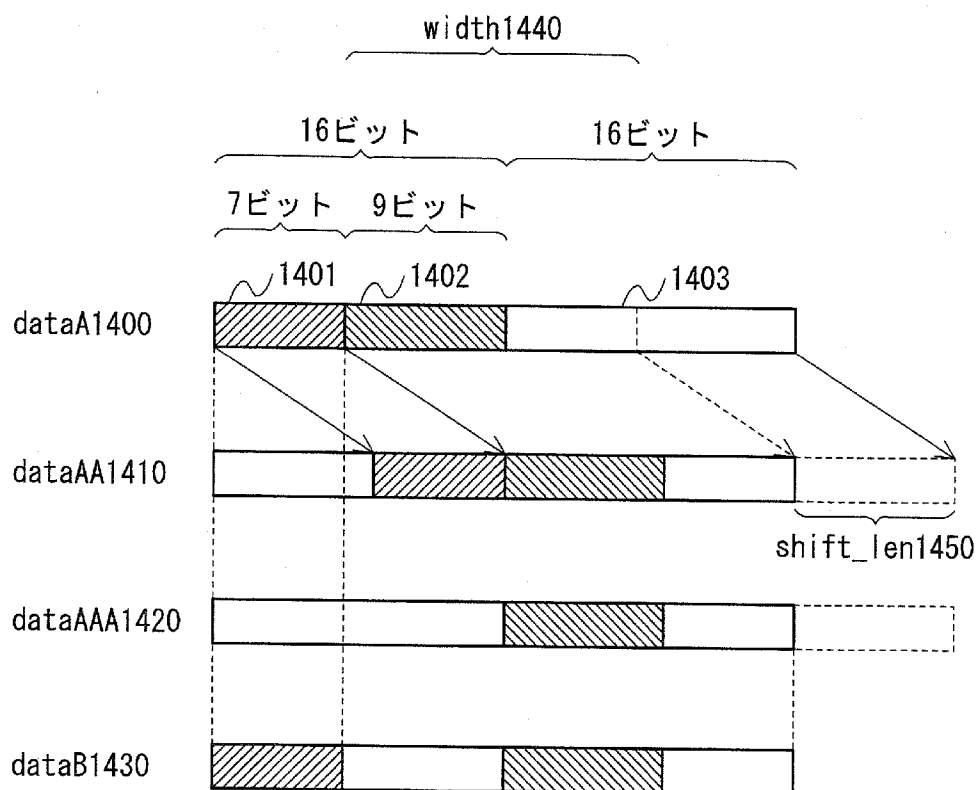
[図13]



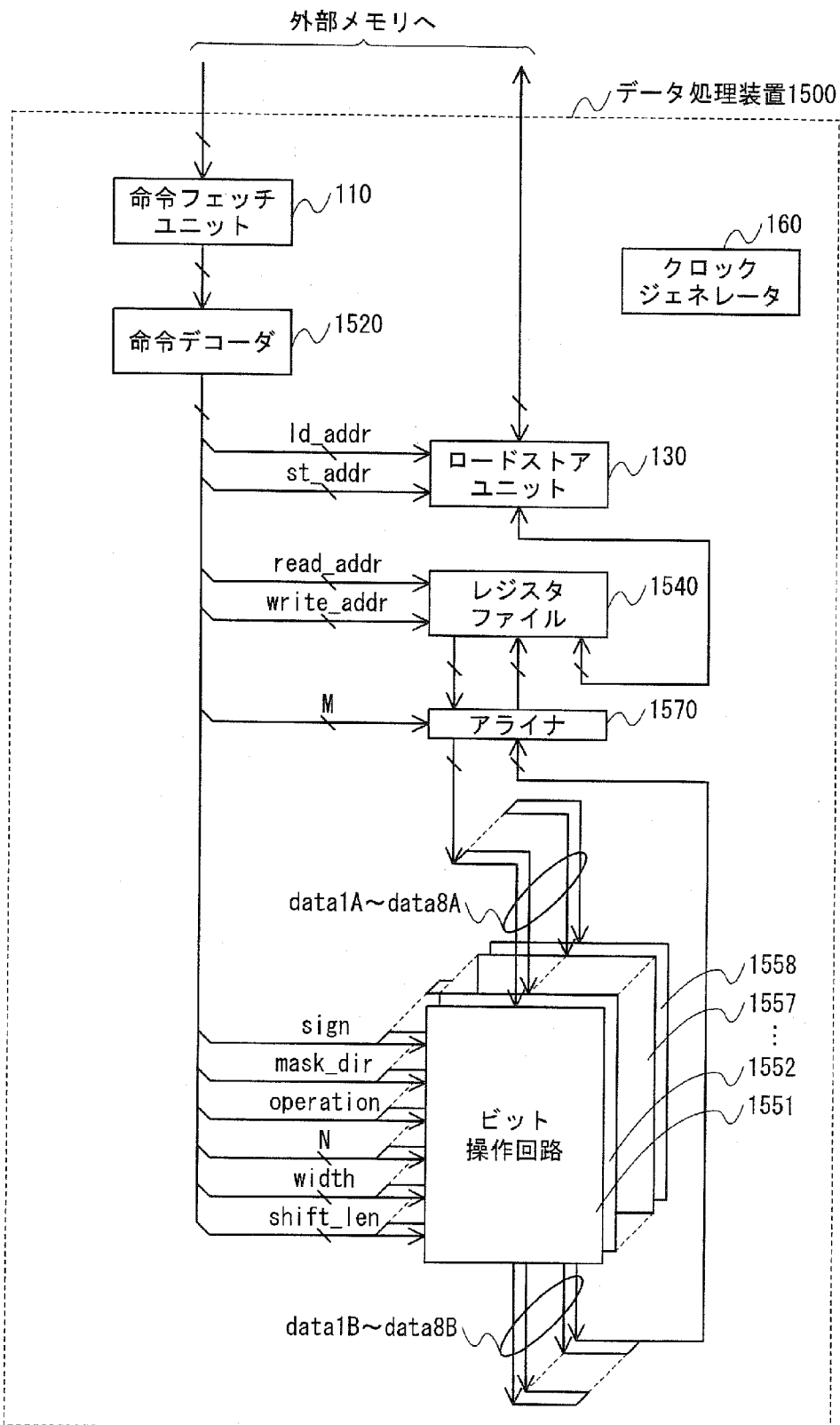
[図14]



[図15]



[図16]

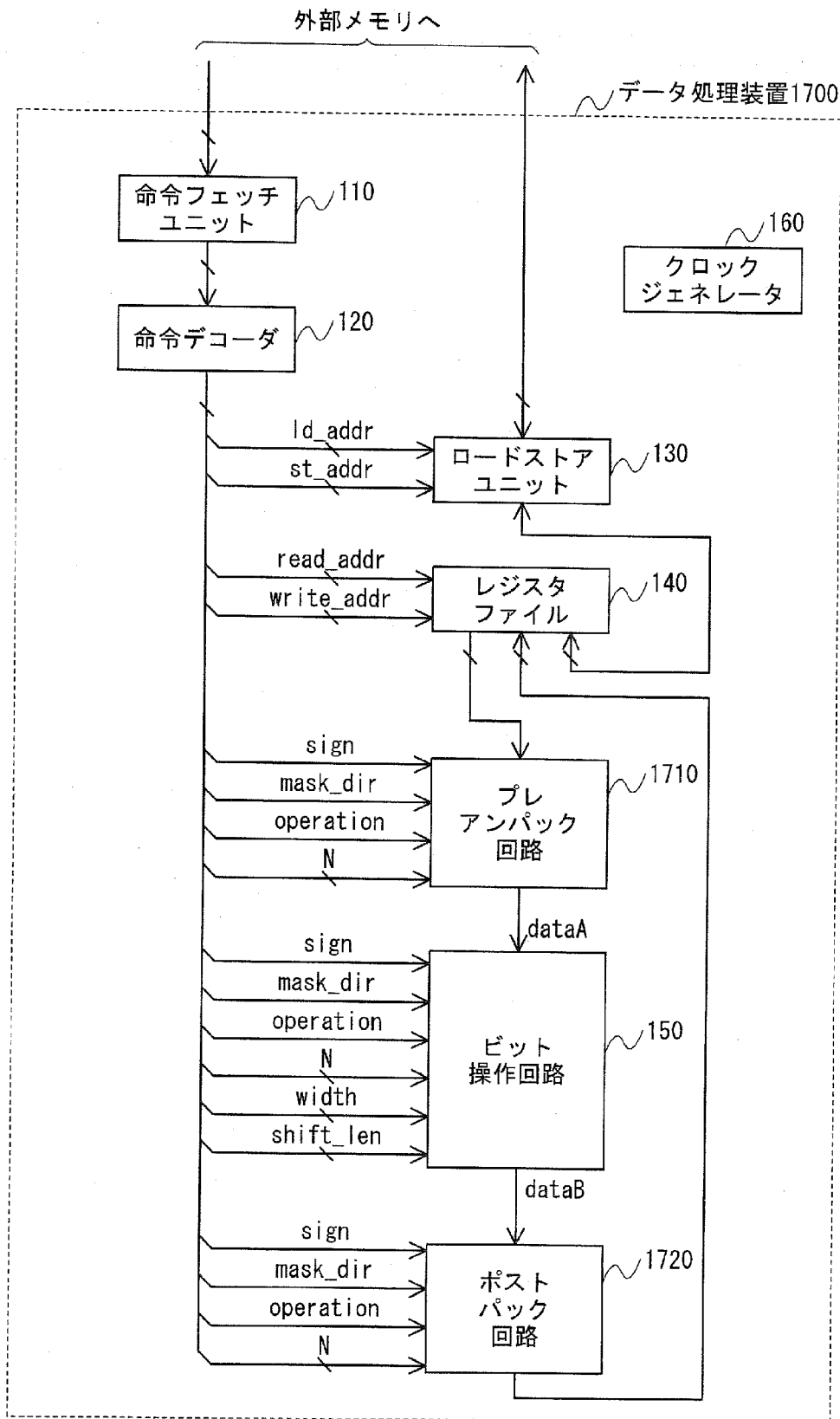


[図17]

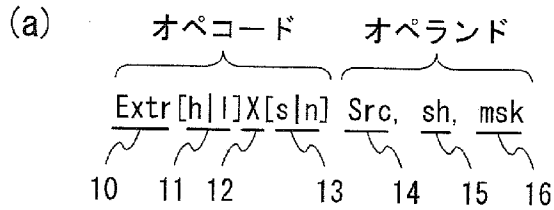
デコード信号対応表1600

命令形式	1610	1620	1630	1640	1650	1655	1660	1670	1680	1690
	sign	mask_dir	operation	N	M	width	shift_len	read_addr	write_addr	
shrnNxM B, A, shift_len, width	0 (ゼロ拡張)	0 (MSB固定)	0 (収縮命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	
shrnINxM B, A, shift_len, width	0 (ゼロ拡張)	1 (LSB固定)	0 (収縮命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	
shrnIsNxM B, A, shift_len, width	1 (符号拡張)	1 (LSB固定)	0 (収縮命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	
extrhNxM B, A, shift_len, width	0 (ゼロ拡張)	0 (MSB固定)	1 (伸張命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	
extrINxM B, A, shift_len, width	0 (ゼロ拡張)	1 (LSB固定)	1 (伸張命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	
extrIsNxM B, A, shift_len, width	1 (符号拡張)	1 (LSB固定)	1 (伸張命令)	8or16or32 (Nの値)	2or4or8 (Mの値)	width	shift_len	addrA	addrB	

[図18]



[図19]



10: 伸張命令であることを示す。

11: "h"の場合には、シフト方向が下位ビット側となり、
"l"の場合には、シフト方向が上位ビット側となることを示す。

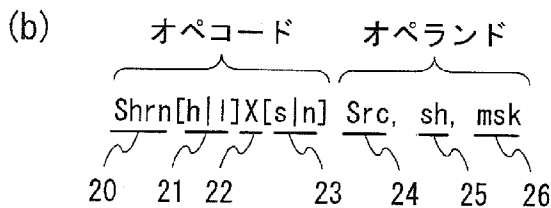
12: データタイプ(8bit or 16bit or 32bit)を示す。

13: プリアンパック処理の有無を示す。

14: ソースレジスタのIDを示すオペランド。
ここでは、処理結果をソースレジスタへ上書きする。

15: シフト量を示す5bitからなるオペランド。

16: シフト対象とならないビット列のビット長を示す
3bitからなるオペランド。



20: 収縮命令であることを示す。

21: "h"の場合には、シフト方向が上位ビット側となり、
"l"の場合には、シフト方向が下位ビット側となることを示す。

22: データタイプ(8bit or 16bit or 32bit)を示す。

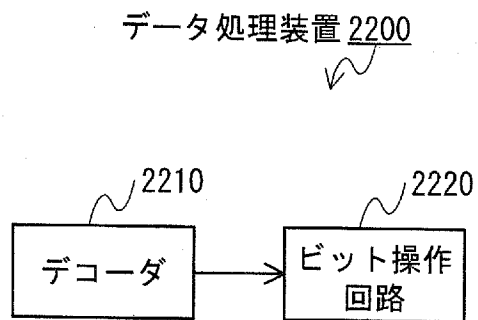
23: ポストパック処理の有無を示す。

24: ソースレジスタのIDを示すオペランド。
ここでは、処理結果をソースレジスタへ上書きする。

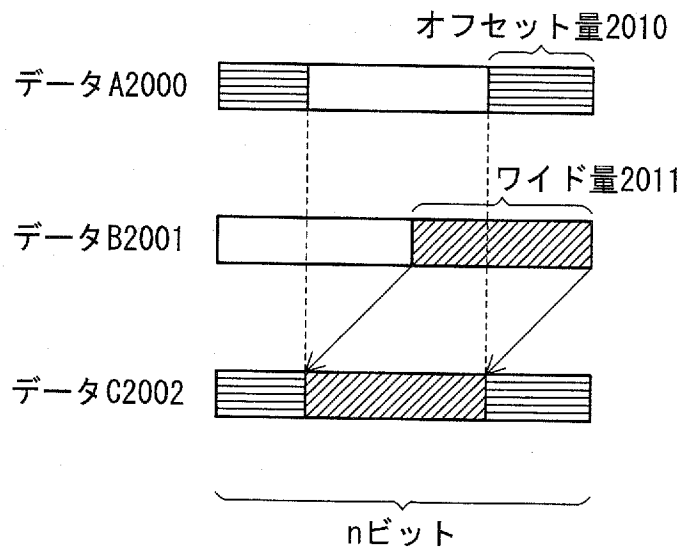
25: シフト量を示す5bitからなるオペランド。

26: シフト対象とならないビット列のビット長を示す
3bitからなるオペランド。

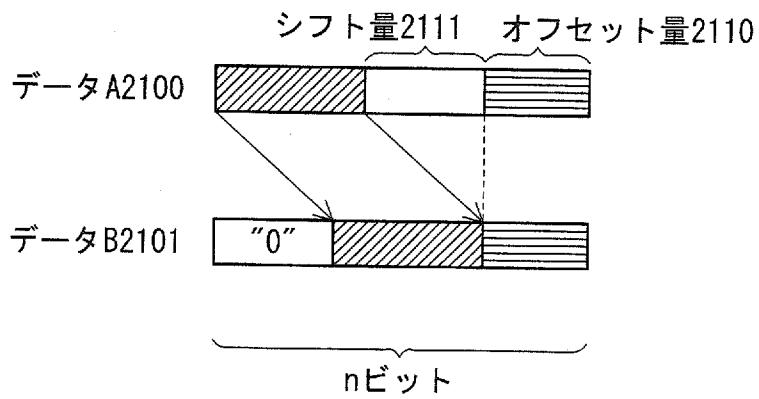
[図20]



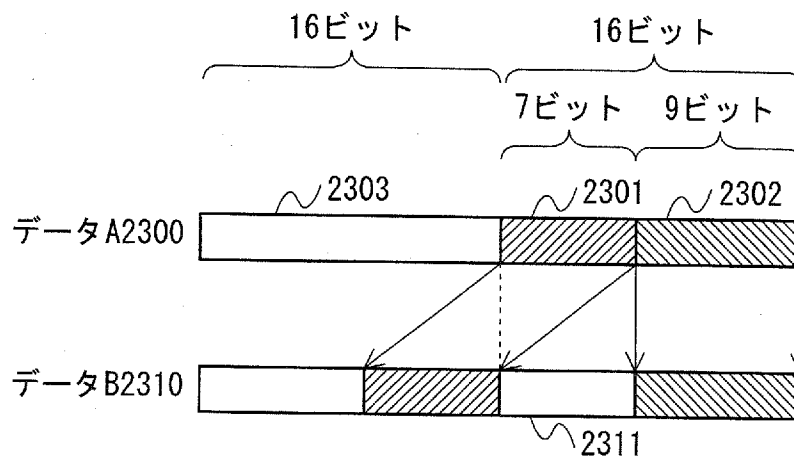
[図21]



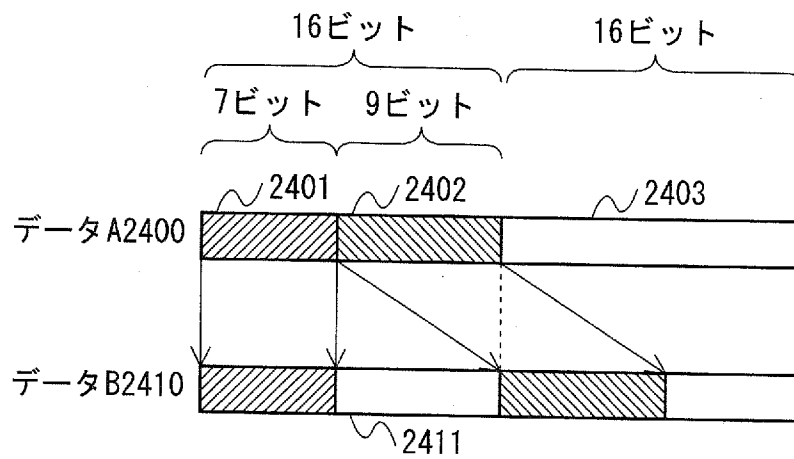
[図22]



[図23]



[図24]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2012/001890

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/305 (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)

G06F9/30-G06F9/315, G06F12/04

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	1996-2012
Kokai Jitsuyo Shinan Koho	1971-2012	Toroku Jitsuyo Shinan Koho	1994-2012

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.
A	JP 2002-522821 A (TriMedia Technologies, Inc.), 23 July 2002 (23.07.2002), paragraphs [0030] to [0032], [0036] to [0039]; fig. 4, 5, 6 & WO 2000/008552 A1 & US 6438676 B1	1-8
A	JP 2005-535966 A (Intel Corp.), 24 November 2005 (24.11.2005), paragraphs [0044] to [0050]; fig. 9, 10A, 10B, 12A, 12B & WO 2004/015563 A1 & US 2006/0149939 A1 & TW I261201 B & AU 2003256870 A1	1-8

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
05 June, 2012 (05.06.12)

Date of mailing of the international search report
19 June, 2012 (19.06.12)

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 (IPC))

Int.Cl. G06F9/305(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 (IPC))

Int.Cl. G06F9/30-G06F9/315
G06F12/04

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	1922-1996年
日本国公開実用新案公報	1971-2012年
日本国実用新案登録公報	1996-2012年
日本国登録実用新案公報	1994-2012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
A	J P 2 0 0 2 - 5 2 2 8 2 1 A (トライメディア テクノロジーズ インク) 2002.07.23, 【0030】 - 【0032】, 【0036】 - 【0039】, 【図4】, 【図5】, 【図6】 &WO 2000/008552 A1 &US 6438676 B1	1-8

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

05.06.2012

国際調査報告の発送日

19.06.2012

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)
郵便番号100-8915
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)

清木 泰

5 B	9 6 4 3
-----	---------

電話番号 03-3581-1101 内線 3545

C (続き) . 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
A	JP 2005-535966 A (インテル・コーポレーション) 2005. 11. 24, 【0044】 - 【0050】, 【図9】 , 【図10A】 , 【図10B】 , 【図12A】 , 【図12B】 &WO 2004/015563 A1 &US 2006/0149939 A1 &TW 1261201 B &AU 2003256870 A1	1-8