(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 8 月 30 日(30.08.2012)

WIPO | PCT

(10) 国際公開番号
WO 2012/114399 A1

(51) 国際特許分類:
G06F 7/00 (2006.01)

(21) 国際出願番号: PCT/JP2011/005905

(22) 国際出願日: 2011 年 10 月 21 日(21.10.2011)

(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 2011-034906 2011 年 2 月 21 日(21.02.2011) JP
特願 2011-182707 2011 年 8 月 24 日(24.08.2011) JP

(71) 出願人 (米国を除く全ての指定国について): 日本電気株式会社(NEC Corporation) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 井上 浩明(INOUE, Hiroaki) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号日本電気株式会社内 Tokyo (JP). 竹中 崇(TAKENAKA, Takashi) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号日本電気株式会社内 Tokyo (JP).

(74) 代理人: 家入 健(IEIRI, Takeshi); 〒2210835 神奈川県横浜市神奈川区鶴屋町三丁目３３番８アサヒビルヂング１０階響国際特許事務所 Kanagawa (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).

[続葉有]

(54) **Title**: COMPUTATION DEVICE AND COMPUTATION EXECUTION METHOD

(54) 発明の名称 : 演算装置及び演算実行方法

[図1]

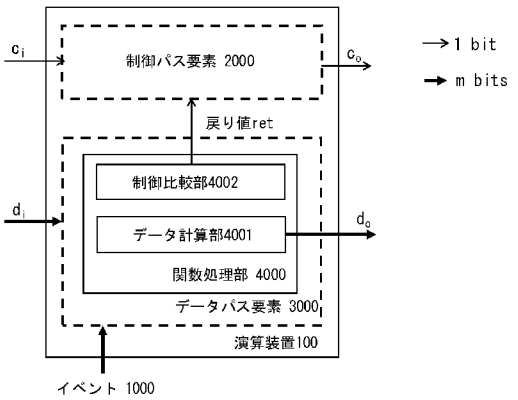


FIG. 1
100 Computation device
1000 Event
2000 Control path element
3000 Data path element
4000 Function processing unit
4001 Data calculation unit
4002 Control comparison unit
ret Return value

(57) **Abstract**: A computation device is provided with: a data path element (3000) having a function processing unit (4000) which executes an operation defined by a function using, as a parameter, input data included in an event (1000); and a control path element (2000) which detects the event (1000) using a return value (ret) of the function. The function processing unit (4000) includes: a data calculation unit (4001) which executes a computation and outputs a first result (d_0); and a control comparison unit (4002) which outputs, as the return value (ret), the result of comparing the input data and the data specifying the event (1000) to the control path element (2000).

(57) 要約: 演算装置は、イベント１０００に含まれる入力データを引数とする関数に規定された演算を実行する関数処理部４０００を有するデータパス要素３０００と、関数の戻り値ｒｅｔを使用して、イベント１０００を検出する制御パス要素２０００とを有する。関数処理部４０００は、演算を実行して第１の結果ｄ₀を出力するデータ計算部４００１と、入力データとイベント１０００を特定するデータとの比較結果を戻り値ｒｅｔとして制御パス要素２０００に出力する制御比較部４００２とを有する。

WO 2012/114399 A1

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 ： 演算装置及び演算実行方法

技術分野

[0001] 本発明は、イベントを処理するのに好適な演算装置及び演算実行方法に関し、特に、正規表現に基づいて、イベント処理関数の実行を制御する演算装置及び演算実行方法に関する。

背景技術

[0002] 図３７Ａ～Ｄは、特許文献１に記載の図であり、与えられた文字列から、正規表現にマッチする文字列を検出するハードウェアの構成方法を示している。図３７Ａ～Ｄには、正規表現の基本的な演算子をハードウェア化するための４つの構成パターンが図示されている。

[0003] 例えば、正規表現で記述された文字列（ａ|ｂ）＊ｃでは、ａとｂの和（ａとｂの少なくとも一方が含まれることを意味する。）が０回以上、繰り返された後にｃが続くような文字列を表す。従って、例えば、文字列ａｂｃｄｅｆｇｈに対しては、文字列ａｂｃが上記の正規表現で記述された文字列にマッチする。

[0004] ここで、この正規表現を回路化する場合、まずａとｂとｃが図３７Ａに示す文字比較、すなわち、それぞれの文字に相当するかを検出する回路で置き換えられる。次に、ａ|ｂは、図３Ｂに示す和、すわなち、ａとの文字比較回路とｂとの文字比較回路を論理和を用いたパターンに従って接続する。そして、（ａ|ｂ）＊は、図３７Ｄに示す繰り返し、すなわち、ａ｜ｂを基礎単位とした回路を所定のパターンに従って接続する。最後に、（ａ|ｂ）＊ｃは、図３７Ｃに示す連接、すなわち、図３７Ｄの（ａ|ｂ）＊の回路とｃとの文字比較回路を基礎単位とした回路を、直列に接続する。以上のように、文字比較回路を基礎単位として、文字比較回路を所定のパターンで接続することにより、どのような正規表現であってもシステマティックに回路に変換することが可能である。

[0005] 上記の文字列照合回路は、一例である。正規表現を用いて、特定の関数が特定の順序で呼ばれているかどうかを判定する装置は、例えば、特許文献２にも記載されている。

[0006] さらに、関数型とは異なる特殊な言語を用いて、多量のイベントから特定の条件を満たすイベント列を検出する、非特許文献１に記載のような回路も提案されている。また、繰り返しを表現する正規表現の演算子を実現する、非特許文献２に記載のような回路も提案されている。

先行技術文献

特許文献

[0007] 特許文献1：特開２００９-０９３５９９

特許文献2：特開平０７-２６２０４４

非特許文献

[0008] 非特許文献1：Ｒ．ミューラー（Mueller）他著、”Streams on Wires ― A Query Compiler for FPGAs”、Proceedings of Very Large Data Base Endowment、米国、２００９年８月、第２巻issue 1、２２９－２４０ページ

非特許文献2：Ｌ．ウッズ（Woods）他著、”Complex Event Detection at Wire Speed with FPGAs”、Proceedings of Very Large Data Base Endowment、米国、２０１０年９月、第３巻issue 1-2、６６０－６６９ページ

発明の概要

発明が解決しようとする課題

[0009] 後述のように、本明細書における「イベント」とは、「発生した事象」ではなく、その事象の発生に伴って生成された各種の情報を意味する。この情報は、１種類には限定されない。すなわち、ある事情の発生に伴って生成された１種類の情報または複数種類の情報の組を「イベント」という。このように、イベントは、１種類または複数種類の情報で構成される。イベントを構成する情報を、「イベント情報」という。

[0010] そして、上記のように定義された、複数個のイベントが時系列的に発生し

た場合、そのイベントの連なりを「イベント列」と呼ぶ。なお、便宜上、イベント列には、単独のイベントも含まれるものとする。

[0011] また、入力されるイベント列に対して、所定のイベント列を検出し、検出したイベント列に応じて所定の処理を行うことがある。イベント列に対するこのような処理を、「イベント処理」という。

[0012] イベント列の処理には、イベント列の検出が含まれる。そのため、イベント処理に特許文献１に記載された文字列照合回路を適用することも考えられる。しかしながら、特許文献１に記載の技術をイベント処理に適用した場合、以下の問題点がある。

[0013] 特許文献１記載の技術は、文字の比較による文字列中の個々の文字の検出と、正規表現された文字列の検出に関しては対応可能である。個々の文字は、各文字を識別するための情報、例えば、文字の種類に関する情報のみを備える。

[0014] 一方、上記のように、イベントは複数のイベント情報で構成される場合がある。そのため、特許文献１記載の技術は、本発明で取り扱うイベントの検出に適用することはできない。なぜなら、イベントを構成するイベント情報が複数種類である場合には、ある１種類のイベント情報についてのみしか、イベント情報の検出およびイベント情報の列の検出に対応することができないからである。従って、一般のイベントおよびイベント列を検出することができない。

[0015] さらに、特許文献１記載の技術では元々文字列の照合のみが想定されており、各文字に対応して何らかの演算を行うことは想定されていない。そのため、イベント列の処理への、特許文献１記載の技術の適用を図った場合、マッチしたイベント列に対応したデータ、すなわちイベント情報（例：温度など）に対して演算を行うことはできない。

[0016] 以上の条件から、特許文献１記載の技術は、イベント処理に必要な２つの条件

１）イベント列を検出できること

２）イベント列に付随したデータに対して演算が実行できること

を満足することができず、結果、システマティックにイベント処理の回路を構成することはできない。

[0017] また、特許文献２に記載の技術においては、関数の呼び出し順序をソフトウェア的に検出するだけである。すなわち、関数の呼び出し順序の検出ができるだけで、マッチした正規表現されたイベント列に対応して定義された計算をすることができない。

[0018] さらに、非特許文献１に記載の技術では、繰り返しを表現するような正規表現の演算子を回路的に実現しておらず、複雑なイベント列の検出が非常に困難である。また、マッチしたイベント列に対応して定義された計算も予め固定された機能しか利用することができない。

[0019] さらにまた、非特許文献２に記載の技術では、マッチしたイベント列に対応して定義された計算手法が開示されておらず、イベント列に対応して定義された計算ができないという問題点がある。

[0020] すなわち、いずれの従来技術においても、イベント処理において、複雑なイベント列の検出と、マッチしたイベント列に対応して定義された計算とを両立することは非常に困難であるという問題点がある。

[0021] 本発明は、このような問題点を解決するためになされたものであり、イベントの検出とイベント列に対応して定義された計算が可能な演算装置及び演算実行方法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

[0022] 上記目的を達成する本発明は、その概略を述べれば以下の通りである。本発明に係る演算装置は、第１のイベントに含まれる第１の入力データを引数とする第１の関数に規定された第１の演算を実行する第１の関数処理手段を有する第１のデータ処理手段と、前記第１の関数の第１の戻り値を使用して、前記第１のイベントを検出する第１の制御処理手段とを有し、前記第１の関数処理手段は、前記第１の演算を実行して第１の結果を出力する第１のデータ計算手段と、前記第１の入力データと前記第１のイベントを特定する第

１のデータとの比較結果を前記第１の戻り値として前記第１の制御処理手段に出力する第１の制御比較手段とを有するものである。

[0023] 本発明に係る演算実行方法は、第１の演算を実行して第１の結果を出力すると共に、第１のイベントに含まれる第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする第１の関数に規定された第１の演算を実行し、前記第１の関数の前記第１の戻り値を使用して、前記第１のイベントを検出する、ものである。

[0024] 本発明にかかるプログラムを格納した非一時的なコンピュータ可読媒体は、第１の演算を実行して第１の結果を出力すると共に、第１のイベントに含まれる第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする第１の関数に規定された第１の演算を実行する第１のデータ処理と、前記第１の関数の前記第１の戻り値を使用して、前記第１のイベントを検出する第１の制御処理とを、コンピュータに実行させるプログラムを格納するものである。

発明の効果

[0025] 本発明によれば、イベント又はイベント列の検出と、検出したイベント又はイベント列に対応して定義された計算を実現することができる演算装置、演算実行方法及びプログラムを格納した非一時的なコンピュータ可読媒体を提供することができる。

図面の簡単な説明

[0026] [図1]本発明の実施の形態１にかかる演算装置を示す図である。

[図2]本発明の実施の形態１にかかる関数演算を実施する演算装置を示す図である。

[図3]本発明の実施の形態１にかかる関数処理部４０００の詳細を示す図である。

[図4]本発明の実施の形態２にかかる演算子を説明する図である。

[図5A]本発明の実施の形態２における演算装置が演算を実施する正規表現の連接を示す図である。

[図5B]本発明の実施の形態２における演算装置が演算を実施する正規表現の連接を示す図である。

[図6A]本発明の実施の形態２における演算装置が演算を実施する正規表現の和を示す図である。

[図6B]本発明の実施の形態２における演算装置が演算を実施する正規表現の和を示す図である。

[図7A]本発明の実施の形態２における演算装置が演算を実施する正規表現の繰り返しを示す図である。

[図7B]本発明の実施の形態２における演算装置が演算を実施する正規表現の繰り返しを示す図である。

[図8]本発明の実施の形態３における正規表現の連接の例を示す図である。

[図9]図８で示した正規表現の連接を演算処理する本発明の実施の形態３にかかる演算装置を示す図である。

[図10]図９で示した正規表現の連接の演算を実施する本発明の実施の形態３にかかる演算装置の動作の一例を示す図である。

[図11]図９で示した正規表現の連接の演算を実施する本発明の実施の形態３にかかる演算装置の動作の一例を示す図である。

[図12]図９で示した正規表現の連接の演算を実施する本発明の実施の形態３にかかる演算装置の動作の一例を示す図である。

[図13]図９で示した正規表現の連接の演算を実施する本発明の実施の形態３にかかる演算装置の動作の一例を示す図である。

[図14]本発明の実施の形態４における正規表現の和の例を示す図である。

[図15]図１４で示した正規表現の和の演算を実施する本発明の実施の形態４にかかる演算装置を示す図である。

[図16]図１５で示した正規表現の和の演算を実施する演算装置の動作の一例を示す図である。

[図17]図１５で示した正規表現の和の演算を実施する演算装置の動作の一例を示す図である。

[図18]図１５で示した正規表現の和の演算を実施する演算装置の動作の一例を示す図である。

[図19]図１５で示した正規表現の和の演算を実施する演算装置の動作の一例を示す図である。

[図20]図１５で示した正規表現の和の演算を実施する演算装置の動作の一例を示す図である。

[図21]本発明の実施の形態５における正規表現の繰り返しの例を示す図である。

[図22]図２１で示した正規表現の繰り返し演算を実施する本発明の実施の形態５にかかる演算装置を示す図である。

[図23]図２２で示した正規表現の繰り返しの演算処理を実施する本発明の実施の形態５にかかる演算装置の動作の一例を示す図である。

[図24]図２２で示した正規表現の繰り返しの演算処理を実施する本発明の実施の形態５にかかる演算装置の動作の一例を示す図である。

[図25]図２２で示した正規表現の繰り返しの演算処理を実施する本発明の実施の形態５にかかる演算装置の動作の一例を示す図である。

[図26]図２２で示した正規表現の繰り返しの演算処理を実施する本発明の実施の形態５にかかる演算装置の動作の一例を示す図である。

[図27]図２２で示した正規表現の繰り返しの演算処理を実施する本発明の実施の形態５にかかる演算装置の動作の一例を示す図である。

[図28]関数の戻り値が複数ビットの場合の関数を実行する本発明の実施の形態６にかかる演算装置を示す図である。

[図29A]複数ビットの戻り値を持つ関数に対する和の演算を行う場合の制御パス要素の接続を説明する図である。

[図29B]複数ビットの戻り値を持つ関数に対する和の演算を行う場合のデータパス要素の接続を説明する図である。

[図30A]複数ビットの戻り値を有する関数に対する繰り返しの演算を行う場合の制御パス要素の接続を説明する図である。

[図30B]複数ビットの戻り値を有する関数に対する繰り返しの演算を行う場合のデータパス要素の接続を説明する図である。

[図31]Ｒｅａｄｙ信号、及びＶａｌｉｄ信号を扱う本発明の実施の形態７にかかる演算装置を示す図である。

[図32]本発明の実施の形態７における正規表現の連接の例を示す図である。

[図33]本発明の実施の形態７における正規表現の和の例を示す図である。

[図34]本発明の実施の形態７における正規表現の繰り返しの例を示す図である。

[図35]本発明の実施の形態８にかかるプログラムを実行する装置の一例を示す図である。

[図36]本発明の実施の形態８にかかるプログラムのイベント検出動作を示すフローチャートである。

[図37A]特許文献１に記載の図であって、正規表現の基本的な演算子（文字比較）をハードウェア化するための構成パターンを示す図である。

[図37B]特許文献１に記載の図であって、正規表現の基本的な演算子（和）をハードウェア化するための構成パターンを示す図である。

[図37C]特許文献１に記載の図であって、正規表現の基本的な演算子（連接）をハードウェア化するための構成パターンを示す図である。

[図37D]特許文献１に記載の図であって、正規表現の基本的な演算子（繰り返し）をハードウェア化するための構成パターンを示す図である。

発明を実施するための形態

[0027] 以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。この実施の形態は、本発明を、「関数を要素とした正規表現の演算」をするものであって、複雑なイベントの検出と、マッチしたイベント列に対応して定義された計算を実現することができる演算装置、演算実行方法及びプログラムに適用したものである。

[0028] ここで、「関数を要素とした正規表現の演算」について説明する。本実施形態における演算装置は、入力されたイベントに対して、関数ごとに定義された所定の処理を行う。所定の処理には、イベントが所定の条件に合致するか否かの検出、およびイベントを用いた所定の演算が含まれる。ここでの「イベント」とは、前述の「ある事象の発生に伴って生成された各種の情報、あるいは情報の組」を意味する。所定の検出とは、例えば、イベントに含まれる情報（以降、「イベント情報」という。）がある情報に一致するか否かの検出である。所定の演算とは、例えば、イベント情報を用いた演算である。

[0029] 複数のイベントが時系列的に入力される場合、すなわち、イベント列が入力される場合には、イベント列を構成する各イベントに対して行う、連続な処理を定義することができる。すなわち、連続する関数との意味で、「関数列」と定義することができる。関数列は、異なる処理の連続や繰返し、一方の処理のみの実行などが定義可能である。従って、文字列の場合と同様に、関数列も正規表現で表すことができる。

[0030] このように、「関数を要素とした正規表現の演算」とは、関数において定義された、連続して実行される演算の実行順序、実行条件などを、正規表現で表したものである。

[0031] 本実施の形態においては、以下に説明するシステマティックな構成方法により、ＣＡＤ（Computer Aided Design）などを用いて、演算装置の自動合成を行うことができる。さらに、従来のような特殊な言語を用いるアプローチとは異なり、手続き型言語と正規表現という、ユーザが親しみやすいインタフェースを有するため、様々な応用の開発に好適である。

[0032] （１）本発明の実施の形態１．

本実施の形態にかかる演算装置は、１以上のデータ（以下、入力データという。）を含むデータ群（イベント）を入力とし、所定の演算を実行するものである。具体的には、イベントを入力とし、入力データを引数とする関数において規定された演算を実行し結果を出力すると共に、入力データの比較

結果を戻り値として出力し、また、関数の戻り値を使用してイベントを検出する処理を実行する。

[0033] 図１は、本実施の形態にかかる演算装置を示す図である。図１に示すように、演算装置１００は、データ処理手段としてのデータパス要素３０００と、制御処理手段としての制御パス要素２０００とを有している。データパス要素３０００は、イベント１０００を構成するイベント情報を引数とする関数に規定された演算を実行する関数処理部４０００を有する。制御パス要素２０００は、関数の戻り値ｒｅｔを使用して、１以上のイベント情報を含むイベントを検出する。

[0034] ここで、関数処理部４０００は、演算を実行して結果d_oを出力するデータ計算部４００１と、イベント情報又は第１のデータとの比較結果を戻り値ｒｅｔとしての制御パス要素２０００に出力する制御比較部４００２とを有する。以下、本実施の形態にかかる演算装置について更に具体的に説明する。

[0035] 図２は、本実施の形態にかかる演算装置であって、図１に示す演算装置の一具体例を示す図である。図２に示すように、演算装置には、１以上の入力データを有するイベント１０００が入力される。ここでの入力データは、上述のイベント情報に相当する。この演算装置１は、関数の演算処理を実施する関数処理部４０００を有するデータ処理手段としてのデータパス要素３０００と、関数の戻り値ｒｅｔを使用してイベントを検出する制御処理手段としての制御パス要素２０００とを有する。

[0036] 関数処理部４０００は、入力データを引数とする、関数に規定された演算を実行し、出力するデータ計算部（後述）と、入力データの比較結果を前記戻り値として制御パス要素２０００に出力する制御比較部（後述）とを有する。データ計算部は、入力データを格納、比較又は演算して出力する。

[0037] 関数処理部４０００は、ユーザに指定された関数ごとに定義された演算を実行するもので、イベント１０００及び入力引数ｉｎを入力とし、戻り値ｒｅｔを出力し、出力引数ｏｕｔを出力する。ここで、入力ｉｎは、例えば前段の演算結果d_iの一部がＭＵＸ１００で選択された値であり、演算結果d_i

のうち不要なものは関数処理部４０００に入力されず、レジスタ１０２を介して、関数処理部４０００の演算結果ｏｕｔと共に出力d_oとして出力される。

[0038] また、制御パス要素２０００は、関数処理部４０００の戻り値ｒｅｔと、直前の要素の真偽値との論理積をとる回路ＡＮＤ１０１と、ＡＮＤ１０１の結果を格納するレジスタ１０３とから構成される。データパス要素３０００は、関数処理部４０００に加え、直前の計算結果のうち、関数処理部４０００に必要とされる引数を与え、必要でない引数をバイパスするマルチプレクサＭＵＸ１００と、関数処理部４０００の出力結果及びバイパスされたデータを格納するレジスタ１０２とから構成される。

[0039] ここで、関数の戻り値ｒｅｔは、入力されたイベント１０００をチェックした結果、イベントに含まれる入力データがイベントを特定する所定のデータであるか、すなわちある特定のイベントが発生したか否かの真偽値を示す。一方、個々の引数に対しては、イベント１０００に含まれるイベント情報を用いた所定の演算の結果の参照ないし格納がなされる。

[0040] 図３は、本実施の形態にかかる関数処理部４０００の詳細を示す図である。図３に示すように、関数処理部４０００は、制御比較部４００１とデータ計算部４００２とを有する。制御比較部４００１は、データ計算部４００２で行われた演算の結果又は演算の途中の計算値と、イベント１０００とから、当該イベントが発生したか否かを戻り値ｒｅｔとして出力する。一方、データ計算部４００２は、直前の計算結果を引数ｉｎとして持ち、イベント１０００に基づいて関数に規定される所定の演算を実施し、引数ｏｕｔに演算結果を出力する。なお、制御比較部４００１には、少なくともイベント１０００が入力されればよい。また、データ計算部４００２は、関数に規定される演算によっては、イベント１０００のみを引数としてもよい。

また、関数に規定される演算の内容は特に限定されない。例えば、演算は、所定の文字又は文字列との比較であってもよい。さらに、演算は、何も処理をしないものであってもよい。すなわち、演算は、直前の計算結果を引数

ｉｎ、または入力データを単に格納し、そのまま引数ｏｕｔとして出力するものであってもよい。

[0041] 例えば、関数処理部４０００の制御比較部４００１でイベントが検出されると、戻り値ｒｅｔとして１が出力される。このとき、制御パス要素２０００に入力する信号ｃｉを１に設定しておけば、制御パス要素２０００は、イベントが発生した場合に、イベント検出結果信号c_oにイベントが検出されたことを示す”１”を出力することができる。同時に、イベントに対してデータ計算部４００２で演算を実行して出力ｏｕｔを通して結果信号d_oとして出力することができ、イベントが検出された場合の演算結果を得ることができる。

[0042] ここで、上述の文字列についての正規表現においては、計算に資する引数がない。従って、文字列の照合処理は、単純に内部で該当文字が与えられたどうかを比較して戻り値として返す関数とみなすことができる。

これに対し、本実施の形態においては、関数処理部４０００にて処理される、少なくとも１つの関数は、引数を有する関数として定義される。このため、本実施の形態においては、イベント列（単独のイベントも含む。）を検出するのみではなく、個々のイベントに対して実行すべき演算を定義することができる。

従って、この演算装置を組み合わせることによって、複雑なイベント列を検出し、必要に応じて個々のイベントに所定の演算を実行することができる演算装置を構成することができる。以下の実施の形態においてその詳細を説明する。

[0043] （２）本発明の実施の形態２．

次に、本発明の実施の形態２について説明する。本実施の形態においては、イベントからなるイベント列を入力とし、１以上の関数と演算子とを含む正規表現の演算を実施する演算装置について説明する。

[0044] 図４は、本実施の形態にかかる演算子を説明する図である。図４に示すように、演算子としては、一の関数の後に次の関数の演算の実行を記述する連

接、２つの関数のいずれの演算の結果を選択するかを記述する和、又は１回以上の関数の演算の連続した実行を記述する繰り返し等がある。

[0045] ここで、「グルーピング」は、関数と演算子の曖昧性解決のために便宜的に設けられた記法である。つまり、関数列がｅ１ｅ２|ｅ３ｅ４と書かれた場合、この関数列がｅ１の連接として、ｅ２とｅ３の和があり、さらにその連接としてｅ４があるのか、あるいは、ｅ１とｅ２の連接と、ｅ３とｅ４の連接と、両者の和であるのかを判別するために用いられる。関数列が前者を意味する場合は、関数列はｅ１（ｅ２|ｅ３）ｅ４と記述される。関数列が後者を意味する場合は、関数列は（ｅ１ｅ２）|（ｅ３,ｅ４）として記述される。すなわち、グルーピングは、関数列の意味の解釈に必要な表現記法であり、本質的に特定の回路要素にマップされるわけではない。

[0046] 本実施の形態では、関数列は正規表現を用いて表現される。そして、関数列を構成する個々の関数は、イベントを入力とし、すなわち、正規表現で表現される関数列の個々の関数のうち、少なくとも１つの関数は１以上の引数を有する。イベントに含まれる入力データの一部又は全部が関数の引数となるが、関数が前段の演算装置における１以上の演算結果を参照する場合はこれも引数となる。そして、関数は、戻り値としてある特定の入力データが検出されたか否かを示す真偽値を出力する。

[0047] ここで、本実施の形態にかかる演算装置おいては、複数の入力データを含むイベントがイベント列として入力される。例えば、演算装置は、入力されるイベント列から、ある特定のイベント列を検出すると共にその際の演算結果を出力する。ここで、イベントとは、前述のように、ある一事象に付随して発生したデータ（イベント情報）からなることを意味する。イベントの具体例には、イベント情報として、例えば、患者の名前、薬の名前、薬の量、薬を飲む時間、体温等がある。この場合、演算装置は、ある患者Ａが薬Ｂ１、Ｂ２、Ｂ３をこの順に飲んだ場合を検出対象のイベント列として検出する。そして、演算装置は、個々のイベントに対して実行される関数からなる関数列で定義された演算の結果として、例えば平均体温を出力することができ

る。

[0048] イベント列には、イベント情報の列、すなわち、連続する情報が含まれる。従って、イベント列の具体例には、例えば、センサーや端末等から入力されるリアルタイム情報がある。なお、本実施の形態については、イベントには複数のイベント情報（入力データ）が含まれるものとして説明するが、入力データは１つであってもよい。

[0049] 演算装置として、上述のようにイベント列を検出し、各イベントに対応した演算を実行する場合は、実施の形態１に記載の演算装置１を、検出するイベントの数だけ設けてもよい。また、複数の演算装置により一の演算装置を構成する場合、少なくとも実施の形態１に記載の演算装置１を含めばよく、他の演算装置は別の構成であってもよいことは勿論である。

[0050] 次に、上述の実施の形態１にかかる演算装置を１又は２つ使用して本実施の形態にかかる演算装置を構成する場合について説明する。

[0051]（２－１）関数の連接

図５Ａ及び図５Ｂは、本実施の形態における演算装置によって実行される演算の、正規表現における連接を示す。本例における演算装置は、実施の形態１にかかる演算装置を２つ用いて構成されてもよい。あるいは、本例における演算装置は、制御パス要素及びデータパス要素を有する２つの演算装置からなり、少なくとも一方が実施の形態１にかかる演算装置であってもよい。

[0052] 図５Ａ及び図５Ｂに、制御パスとデータパスの各々の接続方法が示されている。図５Ａには、制御パスの接続方法が示されている。制御パスに関しては、制御パス要素２１００と制御パス要素２１０１が直列に接続される。図５Ｂには、データパスの接続方法が示されている。データパス要素に関しては、データパス要素３１００とデータパス要素３１０１が直列に接続される。なお、各々の制御パス要素とデータパス要素は、図４で示した正規表現の規則に従い、関数だけでなく、関数列から構成されてもよい。このように、データパス要素３１０１は、データパス要素３１００の演算処理結果に基づ

き演算処理結果を出力し、制御パス要素２１０１は、制御パス要素２１００の結果に基づき検出結果を出力する。

[0053] （２－２）関数の和

図６Ａ及び図６Ｂは、本実施の形態における演算装置によって実行される演算の、正規表現における和を示す。本例における演算装置も、実施の形態１にかかる演算装置を２つ用いて構成されてもよい。あるいは、本例における演算装置は、制御パス要素及びデータパス要素を有する２つの演算装置からなり、少なくとも一方が実施の形態１にかかる演算装置であってもよい。

[0054] 図６Ａ及び図６Ｂに、制御パスとデータパスの各々の接続方法が示されている。図６Ａには、制御パスの接続方法が示されている。す制御パスに関しては、直前の制御パスの出力結果c_iが、制御パス要素２１００と制御パス要素２１０１とに並列に入力される。さらに、制御パス要素２１００と制御パス要素２１０１のそれぞれの出力c_1、c_2が、論理和（ＯＲ）２０２で接続される。ＯＲ２０２の出力C_0がイベント検出結果となる。つまり、ＯＲ２０２は、制御パス要素２１００及び制御パス要素２１０１のいずれか一方以上がイベントを検出した場合にイベント列を検出したと判断する判定手段として機能する。

[0055] 図６Ｂには、データパスの接続方法が示されている。データパス要素に関しては、直前のデータパスの出力結果d_iが、データパス要素３２００とデータパス要素３２０１とに並列に入力される。そして、データパス要素３２００の結果d_1とデータパス要素３２０１の結果d_2は、マルチプレクサＭＵＸ２０１に接続される。マルチプレクサＭＵＸ２０１は、制御パス２２００の出力結果c_1が１であれば、データパス要素３２００の結果d_1を、制御パス２２０１の出力結果c_2が１であれば、データパス要素３２０１の結果d_2を出力する。すなわち、マルチプレクサＭＵＸ２０１の出力結果d_0がイベント列に対応する演算の演算結果となる。

[0056] ここで、両制御パスの結果c_1、c_2が１である場合は、データパス要素d_1、d_2のいずれかの結果を出力する。また、両制御パスの結果c_1、c_2が０で

ある場合は、データパス要素d_1、d_2のいずれか結果を出力してもよいし、任意の値を出力してもよい。つまり、マルチプレクサＭＵＸ２０１は、制御パス要素２１００及び制御パス要素２１０１の出力結果c_1、c_2に応じてデータパス要素３２００及びデータパス要素３２０１の演算結果d_1、d_2のいずれか一方を選択して出力する第１の選択手段として機能する。

[0057] なお、各々の制御パス要素とデータパス要素は、図４で示した正規表現の規則に従い、関数だけでなく、関数列から構成されてもよい。

[0058] （２－３）関数の繰り返し

図７Ａ及び図７Ｂは、本実施の形態における演算装置によって実行される演算の、正規表現における繰り返しを示す。本例における演算装置は、実施の形態１にかかる演算装置１つを構成に含んでもよい。

[0059] 図７に、制御パスとデータパスの各々の接続方法が示されている。図７Ａには、制御パスの接続方法が示されている。制御パスに関しては、直前の制御パスの結果c_i、または１を、制御パス要素の２３００に入力するマルチプレクサＭＵＸ３０１を備える。マルチプレクサＭＵＸ３０１は、制御パス要素２３００の出力結果c_0が０の場合は、直前の制御パスの結果c_iを、制御パス要素２３００の出力結果c_0が１の場合は、１を制御パス要素２３００に入力する。つまり、マルチプレクサＭＵＸ３０１は、制御パス要素の２３００の出力結果c_0に応じて、前段の制御パス要素の出力c_iを当該制御パス要素２３００に入力するか否かを決定する第２の判定手段として機能する。

[0060] 図７Ｂには、データパスの接続方法が示されている。データパス要素に関しては、直前のデータパス要素の結果d_i、またはデータパス要素３３００の出力結果d_0を、データパス要素３３００に入力するマルチプレクサＭＵＸ３０２を備える。マルチプレクサＭＵＸ３０２は、制御パス要素２３００の出力結果c_0が０の場合は、直前のデータパス要素の結果d_iを、制御パス要素２３００の出力結果c_0が１の場合は、データパス要素３３００の出力結果d_0を、データパス要素３３００に入力する。つまり、マルチプレクサＭＵＸ３０２は、データパス要素３３００及び前段のデータパス要素の出力結果d_0、

d_iのいずれか一方を、制御パス要素２３００の出力結果c_0に応じて、当該データパス要素３３００に選択入力する第３の判定手段として機能する。

[0061] なお、上述した図２、図６、及び図７に示した例では、マルチプレクサ、レジスタ、論理積、又は論理和等を構成要素として説明したが、同じ論理動作を実現するものであれば、どのような回路構成であってもかまわない。

[0062] ところで、近年、センサーや端末からのリアルタイム情報（イベント列）の処理を行う新しいサービスが提供されている。本明細書においては、イベントは、複数のイベント情報（データ）かなる構造体を用いて表現されてもよい。イベントが大量に発生した場合、いわゆる、実時間で処理するためには、通常のＣＰＵ（Central Processing Unit）だけではなく、ＡＳＩＣ（Application Specific Integrated Circuit）やＦＰＧＡ（Field-Programmable Gate Array）といったハードウェアで、イベント処理を代替することが重要となる。ここで、上記のイベント処理を実現するには、

１）イベントがどのような順序で発生したかを検出すること（例：イベントＡの後にイベントＢがきて、最後にイベントＣがくる）

２）所定の条件にマッチした各イベントに対して、定義された計算ができること（例：イベントＡで渡された温度とイベントＢで渡された温度の平均をとる）

の両者を満足することが重要である。

[0063] 上述したように、ハードウェアでこれら２つを実現することは、困難であった。これに対し、本実施の形態においては、演算装置がイベントに含まれる入力データを引数とする関数に付随する演算を実行し結果を出力すると共に、入力データの比較結果を戻り値として出力する関数処理部を備える。そのため、当該戻り値を使用してイベント列を検出できると共に、関数処理部によりイベントに対応して定義された演算を実行することができる。従って、イベント列の処理をハードウェアにより構成された演算装置で実施することで、高速に処理することができる。

なお、一方で、本願発明は、ソフトウェアで構成することも可能である。

詳細は、後述する。ソフトウェアで実現することによって、演算装置にイベント処理用の専用回路を備える必要がなくなる。従って、演算装置の回路規模を削減することができる。

[0064] （３）本発明の実施の形態３．

次に、本発明の実施の形態３について説明する。本実施の形態における演算装置は、実施の形態１にかかる演算装置２つを上述の実施の形態２における連接を用いて接続したものである。

[0065] 先ず、本実施の形態にかかる演算装置により演算する関数について説明する。図８は、本発明の実施の形態３における正規表現の連接の例を示す。図８に示すように、正規表現としては、ｓｕｍを引数に持つＡとＢという関数の連接として記述されている（「ＲＵＬＥ」の行を参照）。すなわち、関数Ａ及び関数Ｂは正規表現で表現される関数列を構成している。そして、イベントは、そのイベント情報であるセンサーのＩＤ及びそのセンサーの測定した温度ｔｅｍｐから構成される。

[0066] ここで、関数Ａは、イベント情報のうちの温度を引数ｓｕｍに出力し、入力イベントのセンサーのＩＤが１００であったら１を、異なれば０を返す。一方、関数Ｂは、関数Ａの引数ｓｕｍの値と、入力イベントの温度との和を求め、２で割ることで、２つのイベントの温度の平均値を計算する。そして、入力イベントのＩＤが２００であれば１を、異なれば０を返す。

[0067] 図９は、図８で示した正規表現の連接を演算処理する本実施の形態にかかる演算装置を示す図である。図９に示すように、図２、図３、及び図５の規則に従い、本実施の形態にかかる演算装置は、制御パス要素２１００Ａ、２１０１Ｂと、データパス要素３１００Ａ、３０１Ｂとを有する。データパス要素３１００Ａは、関数処理部４０００Ａ、及びレジスタ１０２Ａを含み、データパス要素３１０１Ｂは、関数処理部４０００Ｂ及びレジスタ１０２Ｂを含む。制御パス要素２１００Ａは、論理積（ＡＮＤ）１０１Ａ、及びレジスタ１０３Ａを含み、制御パス要素２１０１Ｂは、ＡＮＤ１０ＢＡ、及びレジスタ１０３Ｖを含む。関数処理部４０００Ａは、制御比較部４００１Ａ及

びデータ計算部４００２Ａを有し、関数４００１Ｂは、制御比較部４００１Ｂ及びデータ計算部４００２Ｂを有する。

そして、図２、図３、及び図５の規則に従い、制御パス要素２１００Ａと制御パス要素２１０１Ｂ、データパス要素３１００Ａと３１０１Ｂとが、それぞれ連接として接続されている。

[0068] ここで、関数処理部４０００Ａの制御比較部４００１Ａは入力イベントのＩＤが１００と一致するか否かを判定し、戻り値ｒｅｔを出力する。データ計算部４００２Ａは入力イベントの温度ｔｅｍｐを出力引数ｓｕｍに格納する。一方、関数処理部４０００Ｂの制御比較部４００１Ｂは、入力イベントのＩＤが２００と一致するか否かを判定し、戻り値ｒｅｔを出力する。データ計算部４００２Ｂは入力イベントの温度ｔｅｍｐと関数Ａの温度、具体的にはデータパス要素３１００Ａのレジスタ１０２Ａに格納された値との平均をとる。なお、本例においては、制御パス要素２１００Ａの入力は、直前の結果がないため１に固定され、データパス要素３１００Ａは、前段からの入力を必要としないので、何も値が与えられていない。

[0069] （３－１）動作１（イベント１０００＿１｛ＩＤ＝１００，ｔｅｍｐ＝５０｝）

次に、本実施の形態にかかる演算装置のイベント列の処理方法について説明する。図１０は、図９で示した正規表現における連接の演算を実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図１０において、入力としてＩＤが１００、温度が５０のイベント１０００＿１が関数処理部４０００Ａ及び４０００Ｂに与えられる例を示している。この結果、制御比較部４００１Ａの出力結果は１、データ計算部４００２Ａの出力結果は５０、ＡＮＤ１０１Ａの出力結果は１、制御比較部４００１Ｂの出力結果は０、ＡＮＤ１０１Ｂの出力結果は０、データ計算部４００２Ｂの出力結果は２５となる。

[0070] （３－２）動作２（イベント１０００＿２｛ＩＤ＝２００，ｔｅｍｐ＝１５０｝）

図１１は、図９で示した演算装置の動作の一例を示す図である。図１１は、図１０に示すイベント１０００＿１入力の後に、入力としてＩＤが２００、温度が１５０のイベント１０００＿２が関数処理部４０００Ａ及び４０００Ｂに与えられた例を示している。この結果、制御比較部４００１Ａの出力結果は０、データ計算部４００２Ａの出力結果は１５０、論理積ＡＮＤ１０１Ａの出力結果は０、制御比較部４００１Ｂの出力結果は１、論理積ＡＮＤ１０１Ｂの出力結果は１、データ計算部４００２Ｂの出力結果は１００となる。

[0071] （３－３）動作３（イベント１０００＿３{ＩＤ＝３００，ｔｅｍｐ＝０}）

図１２は、図９で示した演算装置の動作の一例を示す図である。図１２は、図１１に示すイベント１０００＿２の入力後に、入力としてＩＤが３００、温度が０のイベント１０００＿３が関数処理部４０００Ａ及び４０００Ｂに与えられる例を示している。この結果、制御比較部４００１Ａの出力結果は０、データ計算部４００２Ａの出力結果は０、ＡＮＤ１０１Ａの出力結果は０、制御比較部４００１Ｂの出力結果は０、ＡＮＤ１０１Ｂの出力結果は０、データ計算部４００２Ｂの出力結果は７５となる。ここで、本例においては、制御パス要素２１０１Ｂのレジスタ１０３Ｂから出力されるイベント検出結果が１となり、データパス要素３１０１Ｂのレジスタ１０２から出力される関数Ａ（ｓｕｍ）Ｂ（ｓｕｍ）の演算結果が１００となる。すなわち、レジスタ１０３Ｂの出力が１となることから、このイベント列においてマッチが発生したことがわかり、すなわち対象のイベント列が検出されたことを示し、そのときの温度の平均値（演算結果）は、レジスタ１０２Ｂの出力結果から１００とわかる。

[0072] （３－４）動作４（イベント１０００＿４{ＩＤ＝３００，ｔｅｍｐ＝０}）

図１３は、図９で示した演算装置の連接の動作の一例を示す図である。図１３は、図１０に示すイベント１０００＿１の入力の後に、入力としてＩＤが３００、温度が０のイベント１０００＿４が関数処理部４０００Ａ及び４０００Ｂに与えられる例を示している。この結果、制御比較部４００１Ａの

出力結果は０、データ計算部４００２Ａの出力結果は０、ＡＮＤ１０１Ａの出力結果は０、制御比較部４００１Ｂの出力結果は０、ＡＮＤ１０１Ｂの出力結果は０、データ計算部４００２Ｂの出力結果は２５となる。この場合は、入力イベント列は関数処理部４０００Ａ、４０００Ｂにマッチしないため、レジスタ１０３Ｂの出力は０となり、イベント列は検出されない。この場合、データ計算部４００２Ｂの出力結果は、無効なデータとして扱われ、本演算装置の正規の出力結果として採用されない。

[0073] 以上のように、本実施の形態にかかる演算装置は、従って、関数Ａ（ｓｕｍ）Ｂ（ｓｕｍ）で記述される正規表現の演算を行い、関数に規定される比較結果にマッチしたイベント列を検出し、その際の演算結果を出力することができる。

本実施の形態にかかる演算装置は、図２、図３、及び図５の規則に従い、制御パス要素２１００Ａと制御パス要素２１０１Ｂ、データパス要素３１００Ａと３１０１Ｂとが、それぞれ連接として接続されている。この構成により、関数Ａ（ｓｕｍ）Ｂ（ｓｕｍ）で記述される正規表現の演算を行うことができる。また、各データパス要素３１００Ａと３１０１Ｂは、制御比較部４００１Ａ、４００１Ｂを有するため、関数に規定される比較結果にマッチしたイベント列を検出することができる。更に、データ計算部４００２Ａ、４００２Ｂを有するため、各関数に規定された演算を実行し演算結果を出力することができる。

[0074] （４）本発明の実施の形態４．

次に本発明の実施の形態４について説明する。本実施の形態における演算装置は、実施の形態１にかかる演算装置２つを上述の実施の形態２における和を用いて接続したものである。

[0075] 図１４は、本発明の実施の形態４で示した正規表現における和の例を示す図である。図１４に示すように、正規表現としては、ｙｅｎを引数にもつＡとＢという関数の和として記述されている（「ＲＵＬＥ」の行を参照）。そして、イベントは、イベント情報として、通貨の名称ｃｕｒｒｅｎｃｙと通

貨量Ａｍｏｕｎｔから構成される。

[0076] ここで、関数Ａは、イベント情報のうちの通貨量を引数ｙｅｎに出力し、入力イベントの通貨の名称が円であったら１を、異なれば０を返す。一方、関数Ｂは、入力イベントの通貨量に８０をかけたものを引数ｙｅｎに出力し、入力イベントの通過の名称がドルであったら１を、異なれば０を返す。すなわち、この例では、通貨を円に統一する例が示されている。

[0077] 図１５は、図１４で示した正規表現における和の演算を実施する本実施の形態にかかる演算装置を示す図である。図１４に示すように、図２、図３、及び図６の規則に従い、本実施の形態にかかる演算装置は、制御パス要素２１１０Ａと制御パス要素２１１１Ｂが論理和（ＯＲ）２１２を介して、データパス要素３１１０Ａと３１１１ＢとがマルチプレクサＭＵＸ２１１を介して接続されている。

[0078] データパス要素３１１０Ａは、関数４０１０Ａ、及びレジスタ１１２Ａを含み、データパス要素３１１１Ｂは、関数４０１０Ｂ、及びレジスタ１１２Ｂを含む。制御パス要素２１１０Ａは、ＡＮＤ１１１Ａ、及びレジスタ１１３Ａを含み、制御パス要素２１１１Ｂは、ＡＮＤ１１１Ｂ、及びレジスタ１１３Ｂを含む。また、関数４０１０Ａは、制御比較部４０１１Ａ及びデータ計算部４０１２Ａを有し、関数４０１０Ｂは、制御比較部４０１１Ｂ及びデータ計算部４０１２Ｂを有する。

[0079] ここで、関数４０１０Ａの制御比較部４０１１Ａは入力イベントの通貨の名称が円と一致するか否かを検出し、戻り値ｒｅｔを出力する。データ計算部４０１２Ａは入力イベントの通貨量ａｍｏｕｎｔを出力引数ｙｅｎに格納する。一方、関数４０１０Ｂの制御比較部４０１１Ｂは、入力イベントの通貨の名称がドルと一致するか否かを検出し、戻り値ｒｅｔを出力する。データ計算部４０１２Ｂは入力イベントの通貨量ａｍｏｕｎｔに８０を掛けてｙｅｎとして出力する。なお、制御パス要素２１１０Ａと２１１１Ｂの入力は、直前の結果がないので１に固定され、データパス要素３１１０Ａと３１１１Ｂは前段からの入力を必要としないので、何も値が与えられない。

[0080] （４－１）動作１（イベント１０１０＿１{ｃｕｒｒｅｎｃｙ＝円，ａｍｏｕｎｔ＝１００}）

次に、本実施の形態にかかる演算装置のイベント列の処理方法について説明する。図１６は、図１５で示した正規表現における和の演算を実施する演算装置の動作の一例を示す図である。図１６は、入力として通貨の名称が円、通過量が１００のイベント１０１０＿１が関数４０１０Ａ及び４０１０Ｂに与えられる例を示している。この結果、制御比較部４０１１Ａの出力結果は１、データ計算部４０１２Ａの出力結果は１００、ＡＮＤ１１１Ａの出力結果は１、制御比較部４０１１Ｂの出力結果は０、ＡＮＤ１１１Ｂの出力結果は０、データ計算部４０１２Ｂの出力結果は８０００となる。

[0081] （４－２）動作２（イベント１０１０＿２{ｃｕｒｒｅｎｃｙ＝元，ａｍｏｕｎｔ＝５}）

図１７は、図１５で示した正規表現における和の演算を実施する演算装置の動作の一例を示す図である。図１７は、図１６に示したイベント１０１０＿１の入力に続き、入力として通貨の名称が元、通過量が５のイベント１０１０＿２が関数４０１０Ａ及び４０１０Ｂに与えられる例を示している。この結果、制御比較部４０１１Ａの出力結果は０、データ計算部４０１２Ａの出力結果は５、ＡＮＤ１１１Ａの出力結果は０、制御比較部４０１１Ｂの出力結果は０、ＡＮＤ１１１Ｂの出力結果は０、データ計算部４０１２Ｂの出力結果は４００となる。ここで、イベント検出結果を出力するＯＲ２１２の出力が１となることから、入力されたイベント列においてマッチが発生、つまりイベントが検出されたことがわかる。そして、そのときの当該正規表現における演算結果、すなわち円建て通貨量は、マルチプレクサＭＵＸ２１１の出力結果から１００とわかる。

[0082] （４－３）動作３（イベント１０１０＿３{ｃｕｒｒｅｎｃｙ＝ドル，ａｍｏｕｎｔ＝１０}）

図１８は、図１５で示した正規表現における和の演算を実施する演算装置の動作の一例を示す図である。図１８は、入力として通貨の名称がドル、通

過量が１０のイベント１０１０＿３が関数４０１０Ａ及び４０１０Ｂに与えられる例を示している。この結果、制御比較部４０１１Ａの出力結果は０、データ計算部４０１２Ａの出力結果は１０、ＡＮＤ１１１Ａの出力結果は０、制御比較部４０１１Ｂの出力結果は１、ＡＮＤ１１１Ｂの出力結果は１、データ計算部４０１２Ｂの出力結果は８００となる。

[0083] （４－４）動作４（イベント１０１０＿４{ｃｕｒｒｅｎｃｙ＝元，ａｍｏｕｎｔ＝５}）

図１９は、図１５で示した正規表現における和の演算を実施する演算装置の動作の一例を示す図である。図１９は、図１８に示したイベント１０１０＿３の入力続き、入力として通貨の名称が元、通過量が５のイベント１０１０＿４が関数４０１０Ａ及び４０１０Ｂに与えられる例を示している。この結果、制御比較部４０１１Ａの出力結果は０、データ計算部４０１２Ａの出力結果は５、ＡＮＤ１１１Ａの出力結果は０、制御比較部４０１１Ｂの出力結果は０、ＡＮＤ１１１Ｂの出力結果は０、データ計算部４０１２Ｂの出力結果は０となる。ここで、本例においても、イベント検出結果を示すＯＲ２１２の出力が１となることから、入力イベント列においてマッチが発生したことがわかり、そのときの演算結果である円建て通貨量は、マルチプレクサＭＵＸ２１１の出力結果から８００とわかる。

[0084] （４－５）動作５（イベント１０１０＿５{ｃｕｒｒｅｎｃｙ＝元，ａｍｏｕｎｔ＝５}）

図２０は図１５で示した正規表現における和の演算を実施する演算装置の動作の一例を示す図である。図２０は、入力として通貨の名称が元、通過量が５のイベント１０１０＿５が関数４０１０Ａと４０１０Ｂに与えられる。この結果、制御比較部４０１１Ａの出力結果は０、データ計算部４０１２Ａの出力結果は５、ＡＮＤ１１１Ａの出力結果は０、制御比較部４０１１Ｂの出力結果は０、ＡＮＤ１１１Ｂの出力結果は０、データ計算部４０１２Ｂの出力結果は４００となる。ここで、イベントは関数４０１０Ａと４０１０Ｂにマッチしないため、検出結果であるＯＲ２１２の出力が０となり、イベン

ト列は検出されない。この場合、データ計算部４０１２Ａ、４０１２Ｂの出力結果は、共に無効なデータとして扱われ、本演算装置の正規の出力結果として採用されない。

[0085] 本実施の形態にかかる演算装置は、図２、図３、及び図６の規則に従い、制御パス要素２１１０Ａと制御パス要素２１１１Ｂ、データパス要素３１１０Ａと３１１１Ｂとが、それぞれ和として接続されている。この構成により、関数Ａ（ｓｕｍ）｜Ｂ（ｓｕｍ）で記述される正規表現の演算を行うことができる。また、各データパス要素３１１０Ａと３１１１Ｂは、制御比較部４０１１Ａ、４０１１Ｂを有するため、関数に規定される比較結果にマッチしたイベント列を検出することができる。更に、データ計算部４０１２Ａ、４０１２Ｂを有するため、各関数に規定された演算を実行し演算結果を出力することができる。

[0086] （５）本発明の実施の形態５．

次に本発明の実施の形態５について説明する。本実施の形態における演算装置は、実施の形態１にかかる演算装置を２つとし上述の実施の形態２で説明した和を用いて接続したものである。

[0087] 図２１は、本発明の実施の形態５で示した正規表現における繰り返しの例を示す。図２１に示すように、正規表現としては、毎秒降水量の合計ｔｏｔａｌを引数に持つ関数Ａの繰り返しと、毎秒降水量の合計ｔｏｔａｌを入力とし、雨天中の積算降水量の毎秒平均を出す関数Ｂとの連接として記述されている（「ＲＵＬＥ」の行を参照）。そして、イベントは、イベント情報として、天気状況ｗｅａｔｈｅｒと毎秒降水量ｒｅｉｎｆａｌｌ＿ａｍｏｕｎｔから構成される。なお、イベントは毎秒ごとに送られると仮定する。

[0088] ここで、関数Ａは、イベント情報のうちの毎秒降水量をｔｏｔａｌに積算して出力し、入力イベントの天気状況が雨であったら１を、異なれば０を返す。一方、関数Ｂは、関数Ａの毎秒降水量の積算を入力とし、ｇｅｔ＿ｎｕｍ＿ｅｖｅｎｔという組込み関数により、今までのイベント数（積算秒）による平均値を出力し、入力イベントの天気状況が晴れであったら１を、異な

れば０を返す。すなわち、この例では、晴れるまでの雨天中の毎秒降水量を積算し、その平均を出力する。

[0089] 図２２は、図２１で示した正規表現における繰り返し演算を実施する本実施の形態にかかる演算装置を示す図である。図２２に示すように、図２、図３、図５、及び図７の規則に従い、本実施の形態にかかる演算装置は、制御パス要素２１２０Ａと制御パス要素２１２１Ｂ、データパス要素３１２０Ａと３１２１Ｂとが、上述した実施の形態２における繰り返しとして接続されている。

[0090] データパス要素３１２０Ａは、関数４０２０Ａ、及びレジスタ１２２Ａを含み、データパス要素３１２１Ｂは、関数４０２０Ｂ、及びレジスタ１２２Ｂを含む。制御パス要素２１２０Ａは、ＡＮＤ１２１Ａ、及びレジスタ１２３Ａを含み、制御パス要素２１２１Ｂは、ＡＮＤ１２１Ｂ、及びレジスタ１２３Ｂを含む。また、関数４０２０Ａは、制御比較部４０２１Ａ及びデータ計算部４０２２Ａを有し、関数４０２０Ｂは、制御比較部４０２１Ｂ及びデータ計算部４０２２Ｂを有する。

[0091] ここで、関数４０２０Ａの制御比較部４０２１Ａは入力イベントの天気状況が雨と一致するか否かを判定し、戻り値ｒｅｔを出力する。データ計算部４０２２Ａは入力イベントの毎秒降水量を積算する。一方、関数４０２０Ｂの制御比較部４０２１Ｂは入力イベントの天気状況が晴れと一致するか否かを判定し、戻り値ｒｅｔを出力する。データ計算部４０２２Ｂは関数Ａの出力である積算毎秒降水量を積算秒数で割った、雨天時の平均毎秒降水量を出力する。なお、マルチプレクサＭＵＸ３２１Ａへの直前値の入力は、直前の結果がないので１に固定され、マルチプレクサＭＵＸ３２２Ａへの直前値の入力は、積算のために０に固定される。

[0092] （５－１）動作１（イベント１０２０＿１{weather＝雨，rainfall_amount＝１０}）

次に、本実施の形態にかかる演算装置のイベント列の処理方法について説明する。図２３は、図２２で示した正規表現における繰り返しの演算処理を

実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図２３は、入力として天気状況が雨、毎秒降水量が１０のイベント１０２０＿１が関数４０２０Ａ及び４０２０Ｂに与えられる例を示している。この結果、制御比較部４０２１Ａの出力結果は１、データ計算部４０２２Ａの入力は０、データ計算部４０２２Ａの出力結果は１０、ＡＮＤ１２１Ａの出力結果は１、制御比較部４０２１Ｂの出力結果は０、ＡＮＤ１２１Ｂの出力結果は０、データ計算部４０２２Ｂの出力結果は０となる。

[0093] （５－２）動作２（イベント１０２０＿２{weather＝雨，rainfall_amount＝２０}）

図２４は、図２２で示した正規表現における繰り返しの演算処理を実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図２４は、図２３に示すイベント１０２０＿１に続き、入力として天気状況が雨、毎秒降水量が２０のイベント１０２０＿２が関数４０２０Ａ及び４０２０Ｂに与えられる例を示している。この結果、制御比較部４０２１Ａの出力結果は１、データ計算部４０２２Ａの入力は１０、データ計算部４０２２Ａの出力結果は３０、ＡＮＤ１２１Ａの出力結果は１、制御比較部４０２１Ｂの出力結果は０、ＡＮＤ１２１Ｂの出力結果は０、データ計算部４０１２Ｂの出力結果は、積算秒数は１のため、３０となる。

[0094] （５－３）動作３（イベント１０２０＿３{weather＝晴れ，rainfall_amount＝０}）

図２５は、図２２で示した正規表現における繰り返しの演算処理を実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図２５は、図２４に示すイベント１０２０＿２に続き、入力として天気状況が晴れ、毎秒降水量が０のイベントが関数４０２０Ａ及び４０２０Ｂに与えられる。この結果、制御比較部４０２１Ａの出力結果は０、データ計算部４０２２Ａの入力は３０、データ計算部４０２２Ａの出力結果は３０、ＡＮＤ１２１Ａの出力結果は０、制御比較部４０２１Ｂの出力結果は１、ＡＮＤ１２１Ｂの出力結果は１、データ計算部４０１２Ｂの出力結果は、積算秒数は２のため、

１５となる。

[0095] （５－４）動作４（イベント１０＿４{weather＝曇り，rainfall_amount＝０})

図２６は、図２２で示した正規表現における繰り返しの演算処理を実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図２６は、図２５に示すイベント１０２０＿３の入力に続き、入力として天気状況が曇り、毎秒降水量が０のイベント１０２０＿４が関数４０２０Ａ及び４０２０Ｂに与えられる例を示している。この結果、制御比較部４０２１Ａの出力結果は０、データ計算部４０２２Ａの入力は０、データ計算部４０２２Ａの出力結果は０、ＡＮＤ１２１Ａの出力結果は０、制御比較部４０２１Ｂの出力結果は０、ＡＮＤ１２１Ｂの出力結果は０、データ計算部４０２２Ｂの出力結果は、積算秒数は１のため、１０となる。ただし、イベント検出結果を示すレジスタ１２３Ｂの出力が１となることから、このイベント列においてマッチが発生、つまりイベントが検出されたことがわかり、そのとき平均積算毎秒降水量は、レジスタ１２２Ｂの出力結果から１５とわかる。

[0096] （５－５）動作５（イベント１０１０{weather＝曇り，rainfall_amount＝０})

図２７は、図２２で示した正規表現における繰り返しの演算処理を実施する本実施の形態にかかる演算装置の動作の一例を示す図である。図２７は、図２４に示すイベント１０２０＿１の入力に続き、入力として天気状況が曇り、毎秒降水量が０のイベント１０１０＿５が関数４０２０Ａ及び４０２０Ｂに与えられる例を示す。この結果、制御比較部４０２１Ａの出力結果は０、データ計算部４０２２Ａの入力は３０、データ計算部４０２２Ａの出力結果は３０、ＡＮＤ１２１Ａの出力結果は０、制御比較部４０２１Ｂの出力結果は０、ＡＮＤ１２１Ｂの出力結果は０、データ計算部４０２２Ｂの出力結果は、積算秒数は２のため、１５となる。しかしながら、このイベントは関数４０２０Ａと４０２０Ｂにマッチしないため、イベント検出結果を示すレジスタ１２３Ｂの出力は０となり、イベント列は検出されない。この場合、

データ計算部４０２２Ｂの出力結果は、無効なデータとして扱われ、本演算装置の正規の出力結果として採用されない。

[0097] 本実施の形態にかかる演算装置は、図２、図３、図５、及び図７の規則に従い、制御パス要素２１２０Ａの繰り返しと制御パス要素２１２１Ｂとが連接として接続され、データパス要素３１２０Ａの繰り返しとデータパス要素３１２１Ｂとが連接として接続されている。この構成により、関数Ａ（ｓｕｍ）＋Ｂ（ｓｕｍ）で記述される正規表現の演算を行うことができる。また、各データパス要素３１２０Ａと３１２１Ｂは、制御比較部４０２１Ａ、４０２１Ｂを有するため、関数に規定される比較結果にマッチしたイベント列を検出することができる。更に、データ計算部４０２２Ａ、４０２２Ｂを有するため、各関数に規定された演算を実行し演算結果を出力することができる。

[0098] （６）本発明の実施の形態６．

以上の実施の形態においては、関数の戻り値は、真偽値を表す１ビットの情報である。関数の戻り値は、複数ビットに拡張されてもよい。本実施の形態においては、演算装置で実施する関数の戻り値が複数ビットである場合の例を示す。

[0099] 関数の戻り値が複数ビットで表示される具体例としては、戻り値がイベントの発生確率を意味する場合がある。すなわち、関数の戻り値は、イベントの発生の有無ではなく、イベントの発生確率を示す。イベントの発生確率は、イベントが発生した可能性と解釈することもできる。このように、各イベントの発生確率を定義することができるので、イベント列の発生確率を定義することもできる。

[0100] イベントの発生確率を示す戻り値は、以下のようにして求めることができる。イベントに含まれるデータ（イベント情報）が物理的なゆらぎに従う値の場合、例えば、熱源での温度を表すのであれば、熱源に関する物理法則から予め計算される確率分布に従い、当該温度の発生確率を求めることができる。また、イベント情報がそのような物理法則に従わないようなデータの場

合、例えば、テストの点などにおいては、ロジスティクス分布や正規分布などの、よく知られた分布を仮定し、その分布から発生確率を計算することができる。さらに、過去のイベント列で発生したデータ頻度に従って現在のイベント発生確率を計算してもよい。他にも、イベント発生確率の計算には、ベイズ法や最尤法など、統計計算・データマイニング分野で使われる手法であれば、どのような手法を用いてもかまわない。

[0101] また、戻り値が複数ビットである場合、実施形態１乃至５のように、戻り値同士の論理演算は定義できない。複数ビットの戻り値同士の演算方法については、以下に具体例を示す。

[0102] 本実施の形態においては、ある確率以上で発生するイベント列を検出するものとして説明する。図２８は、関数の戻り値が複数ビットの場合の関数を実行する本実施の形態にかかる演算装置を示す図である。ここで、この複数ビットは、この関数のイベントの発生確率を表すものとする。イベントの発生確率を戻り値として定義した場合、図２とは異なり、前段の制御パス要素の結果c_iと関数演算部４２００からの結果の論理和ＡＮＤ１０１の代わりに、図２８に示すように、乗算器１１０を備える。これにより、前段の制御パス要素のイベント発生確率c_iと、本関数演算部４２００からのイベント発生確率ｒｅｔの積を出力することで、レジスタ１０３は、イベント列自体の発生確率c_0を出力する。ここで、関数の戻り値ｒｅｔはイベントの発生確率だけにとどまらず、他の状態を表現してもよい。その場合、乗算器ではなく、異なる演算器を備えてもよい。

[0103] このように関数の戻り値を複数ビットに定義した場合、正規表現の規則の中で、和と繰り返し規則の構成に違いが生じる。図２９Ａ及び図２９Ｂは、複数ビットの戻り値を持つ関数に対する和の演算を行う場合の制御パス要素、データパス要素の接続を説明する図である。図６Ａ及び図６Ｂとは異なり、制御パス要素２２００と制御パス要素２２０１の結果の論理和２０２を行うのではなく、図２９Ａに示すように、制御パス要素２２００と制御パス要素２２０１のうち、いずれか大きい方の値を選択して出力する最大値選択演

算部２３２を備える。これは、より大きなイベントの発生確率を出力した関数が選ばれることを意味する。

[0104] この構成に伴い、データパス要素３２００の出力d_1とデータパス要素３２０１の出力d_2は、制御パス要素２２００、２２０１の出力結果c_1、c_2の大きさによって選択される（図２９Ｂ）。すなわち、出力結果c_1が大きいときは出力d_1が、出力結果c_2が大きいときは出力d_2が、最大値選択ＭＵＸ２３１によって選択され、出力d_0として出力される。

[0105] なお、最大値選択演算部２３２および最大値選択ＭＵＸ２３１における選択条件は一例であり、上記以外の条件判断に基づいて、関数または出力結果を選択してもよい。

[0106] 図３０Ａ及び図３０Ｂは、複数ビットの戻り値を有する関数に対する繰り返しの演算を行う場合の制御パス要素、データパス要素の接続を説明する図である。図７Ａ及び図７Ｂとは異なり、制御パス要素２３００の入力と、データパス要素３３００の入力には、ＭＵＸを用いるのではなく、イベントの発生確率がある閾値を超えているかどうかで入力を選択する閾値付ＭＵＸ３１１、３１２を備える。すなわち、繰り返し発生するイベントの連続発生確率がある閾値を下回ると、例え、単体イベントとして十分の発生確率があったとしても、以降は前段からの制御パス要素及びデータパス要素の結果c_i、d_iが選択され、入力されることになる。すなわち、単体イベントの発生確率が８０％であったとしても例えば１０回繰り返されれば１０％の確率になることを示す。

[0107] 本実施の形態においては、関数の正規表現を演算処理する演算装置において、関数の戻り値が複数ビットで表現された場合にも対応することができる。従って、イベント列の発生の有無だけでなく、イベント列の発生確率に基づく処理を行うことができる。すなわち、イベント列を構成する各イベントに対応した演算を実行し、イベント列の発生確率に基づいて、適切な演算結果を出力することができる。

[0108] （７）本発明の実施の形態７．

次に、本発明の実施の形態７について説明する。本実施の形態にかかる演算装置は、後段に位置する演算装置から入力される動作停止要求を扱い、各演算装置における関数演算のサイクル数の違いに対応するものである。

[0109] 図３１は、本実施の形態にかかる演算装置の構成を示す図である。なお、図３１において、同一名及び同一符号を付した処理部であって、以下に言及しない処理部は、実施の形態１、２と基本的に同じ処理を行う。

[0110] 本実施の形態にかかる演算装置１は、図２に示す構成と同様に、制御パス要素２０００と、データパス要素３０００を有している。また、データパス要素３０００は、関数処理部４０００を有しており、関数処理部４０００は、制御比較部４００１と、データ計算部４００２とを有している。

[0111] 演算装置１は、図２の構成に加え、データパス要素３０００内にレジスタ１０４を有する。さらに、演算装置１は、論理積ＡＮＤ１０５と、キュー１０６と、を有する。

[0112] 制御パス要素２０００には、直前の制御パスの出力結果c_iが入力される。関数処理部４０００には、直前のデータパスの出力結果d_iに加え、直前のデータパスの出力結果の有効性を示す有効性情報（以下、当該有効性情報は、０または１の値をとるデジタル信号形式で入力されるものとし、Ｖａｌｉｄ信号と記載する。）v_iが入力される。Ｖａｌｉｄ信号v_iは、０の場合に無効を示し、１の場合に有効を示す。

さらに、論理積ＡＮＤ１０５には、後段に位置する演算装置から出力された動作停止要求r_i（以下、当該要求は、０または１の値をとるデジタル信号形式で入力されるものとし、Ｒｅａｄｙ信号と記載する。）が入力される。Ｒｅａｄｙ信号r_iは、後段に位置する演算装置が動作停止を要求するか否かを示す信号であり、１の場合に動作停止を要求しないことを示し、０の場合に動作停止を要求することを示す。

演算装置１は、Ｒｅａｄｙ信号r_iが０のとき、データ計算部４００２の処理を停止させることができる。例えば、処理速度が遅い後段の演算装置が、演算処理の実行中に、前段の演算装置に対して０の値を持つＲｅａｄｙ信号

r_iを出力する。このとき、演算装置１は、データ計算部４００２の処理を停止させてもよい。そして、演算装置１は、後段の演算装置の演算処理が完了しＲｅａｄｙ信号r_iが１になるのを待って、データ計算部４００２の処理を再開させてもよい。

なお、データ計算部４００２の処理が完了していないときは、Ｒｅａｄｙ信号r_iが０であっても、演算装置１は、データ計算部４００２の処理を直ちに停止させる必要はない。

このように、演算装置１は、Ｒｅａｄｙ信号r_iの値に応じて、データ計算部４００２の処理の停止・許可の制御を行うことができる。

[0113] キュー１０６には、イベント１０００が入力される。キュー１０６は、イベント１０００を順次保持する保持手段として機能する。Ｖａｌｉｄ信号v_iは、キュー１０６にも入力される。キュー１０６は、入力されたＶａｌｉｄ信号v_iの値が１（有効）の場合、先頭に位置するイベントをデータ計算部４００２に入力する。また、キュー１０６は、イベントを保持しているかいなかを示す情報（イベント保持情報）であるｅｍｐｔｙ信号（１：イベントを保持している、０：イベントを保持していない）を論理積ＡＮＤ１０５に入力する。

[0114] 論理積ＡＮＤ１０５は、ｅｍｐｔｙ信号またはＲｅａｄｙ信号r_iのいずれかが０である場合、前段の演算装置に０の値をもつＲｅａｄｙ信号r_0を入力する。すなわち、論理積ＡＮＤ１０５は、前段の演算装置に入力するＲｅａｄｙ信号を生成する動作停止要求生成手段として動作する。ｅｍｐｔｙ信号またはＲｅａｄｙ信号r_iのいずれかが０である場合、データ計算部４００２は、演算の実行が不可能な状態である。それ以外の場合（ｅｍｐｔｙ信号とＲｅａｄｙ信号r_iの双方とも１の場合）、データ計算部４００２は、演算の実行が可能な状態である。このとき、論理積ＡＮＤ１０５は、前段の演算装置に対し、データ計算部４００２が演算の実行が可能であるか否かを示す実行可否状態を伝達する。

[0115] 上述の構成にすることにより、演算装置１は、後段に位置する演算装置の

処理状況をＲｅａｄｙ信号により把握することができる。これにより、例えば前段の演算装置として処理速度の速い演算装置を用い、後段の演算装置として処理速度の遅い演算装置を用いて演算を行う場合であっても、前段の演算装置は、後段の演算装置に確実に演算結果を伝達することができる。従って、正常に演算を行うことができる。

[0116] なお、図３１の構成は、前段へのＲｅａｄｙ信号の伝搬が１サイクルで終了すると仮定した構成である。ＡＮＤ１０５やキュー１０６の周辺の回路の構成等によってＲｅａｄｙ信号の伝搬に複数サイクル要する場合、キュー１０６が空になること、または後段の演算装置の演算処理が遅れることを予測する予測機構を新たに設ければ良い。当該予測機構による予測に応じて、Ｒｅａｄｙ信号の値を設定することにより、Ｒｅａｄｙ信号を正確に伝搬できる。

たとえば、予測機構は、自身のキュー１０６の構成から何クロックか後に空になること、または後段の演算装置の演算処理に何クロックか必要なことがわかっているため後段からのＲｅａｄｙ信号受信が何クロックか遅れることが予測される場合、先にＲｅａｄｙ信号を０にする

[0117] 次に、図３１に示す演算装置２つを用い、上述の実施の形態２で説明した連接を用いて接続する例について説明する。図３２は、正規表現の連接を図３１に示す演算装置２つを用いて構成した図を示す。第１の演算装置Ａは、制御パス要素２１３０Ａと、データパス要素３１３０Ａとを有する。第２の演算装置Ｂは、制御パス要素２１３１Ｂと、データパス要素３１３１Ｂとを有する。図２、図３、及び図５の規則に従い、制御パス要素２１３０Ａと制御パス要素２１３１Ｂ、データパス要素３１３０Ａとデータパス要素３１３１Ｂとが、それぞれ連接として接続されている。

[0118] この際、データパス要素３１３０Ａには、Ｖａｌｉｄ信号v_iが入力され、データパス要素３１３０Ａは、演算結果に応じてＶａｌｉｄ信号ｖをデータパス要素３１３１Ｂに入力する。データパス要素３１３１Ｂは、Ｖａｌｉｄ信号v_oを後段の演算装置に入力する。

同様に、第２の演算装置Ｂには、Ｒｅａｄｙ信号$ｒ_{i}$が入力され、第２の演算装置Ｂは、Ｒｅａｄｙ信号ｒを第１の演算装置Ａに入力する。

[0119] 図３２に示す演算装置の構成は、図２、図３、及び図５の規則に従い、制御パス要素２１３０Ａと制御パス要素２１３１Ｂ、データパス要素３１３０Ａとデータパス要素３１３１Ｂとが、それぞれ連接として接続されている。この構成により、連接により記述される正規表現の演算を行うことができる。さらに、実施の形態３と同様に、関数に規定される比較結果にマッチしたイベント列の検出、及び各関数に規定されて演算の実行を行うことができる。

[0120] 次に、図３１に示す演算装置２つを用い、上述の実施の形態２で説明した和を用いて接続する例について説明する。図３３は、正規表現の和を図３１に示す演算装置２つを用いて構成した図を示す。

[0121] 第１の演算装置Ａは、制御パス要素２１４０Ａと、データパス要素３１４０Ａとを有する。第２の演算装置Ｂは、制御パス要素２１４１Ｂと、データパス要素３１４１Ｂとを有する。図２、図３、及び図６の規則に従い、制御パス要素２１４０Ａと制御パス要素２１４１Ｂ、データパス要素３１４０Ａとデータパス要素３１４１Ｂとが、それぞれ和として接続されている。

[0122] ここで、図１５に示す構成に加え、第１の演算装置Ａ及び第２の演算装置Ｂには、Ｖａｌｉｄ信号$ｖ_{i}$及びＲｅａｄｙ信号$ｒ_{i}$が入力される。

第１の演算装置Ａは、シフタ２０４に接続される。同様に、第２の演算装置Ｂは、シフタ２０５に接続される。シフタ２０４には、制御パス要素２１４０Ａの制御パス出力結果$ｃ_{1}$、データパス要素３１４０Ａのデータパスの出力結果$ｄ_{1}$、及びデータパス要素３１４０ＡのＶａｌｉｄ信号$ｖ_{1}$が入力される。シフタ２０５には、制御パス要素２１４１Ｂの制御パス出力結果$ｃ_{2}$、データパス要素３１４１Ｂのデータパスの出力結果$ｄ_{2}$、及びデータパス要素３１４１ＢのＶａｌｉｄ信号$ｖ_{2}$が入力される。

[0123] シフタ２０４及びシフタ２０５は、第１の演算装置Ａと第２の演算装置Ｂの演算サイクル数の違いをなくすために遅延を与える遅延手段として動作す

る。例えば、第１の演算装置Ａの演算サイクルが第２の演算装置Ｂの演算サイクルよりも３サイクル少ない場合、シフタ２０４は、各信号に３サイクル分の遅延を与える。なお、第１の演算装置Ａの演算サイクルと、第２の演算装置Ｂの演算サイクルが同じ場合、シフタ２０４及び２０５を備える必要はない。

[0124] 論理積ＡＮＤ２０３には、データパス要素３１４０Ａの Ｖａｌｉｄ信号v_1、及びデータパス要素３１４１Ｂの Ｖａｌｉｄ信号v_2が入力される。論理積ＡＮＤ２０３は、両者（Ｖａｌｉｄ信号v_1、及びＶａｌｉｄ信号v_2）が１である場合にのみ、後段の演算装置にＶａｌｉｄ信号として１を入力する。

[0125] 論理積ＡＮＤ２０６には、第１の演算装置ＡからのＲｅａｄｙ信号r_1、第２の演算装置ＢからのＲｅａｄｙ信号r_2が入力される。論理積ＡＮＤ２０３は、どちらかのＲｅａｄｙ信号が０である場合に、前段に位置する演算装置に、０の値を持つＲｅａｄｙ信号を入力する。

[0126] 図３３に示す演算装置の構成は、図２、図３、及び図６の規則に従い、制御パス要素２１４０Ａと制御パス要素２１４１Ｂ、データパス要素３１４０Ａとデータパス要素３１４１Ｂとが、それぞれ和として接続されている。この構成により、和により記述される正規表現の演算を行うことができる。さらに、実施の形態４と同様に、関数に規定される比較結果にマッチしたイベント列の検出、及び各関数に規定されて演算の実行を行うことができる。

[0127] 次に、図３１に示す演算装置を用いて、上述の実施の形態２で説明した正規表現における繰り返しを実現する例について説明する。図３４は、正規表現の繰り返しを図３１に示す演算装置を用いて構成した図を示す。当該演算装置は、制御パス要素２１５０Ａと、データパス要素３１５０Ａを有する。

[0128] 図３４に示すように、演算装置に繰り返し論理４００が接続されている。繰り返し論理４００は、前述のマルチプレクサＭＵＸ３０１等を含む処理部であり、正規表現における繰り返しを制御する処理を行う。

[0129] 繰り返し論理４００には、直前の制御パスの出力結果c_i、直前のデータパス要素の結果d_i、及び直前のＶａｌｉｄ信号v_iに加え、制御パス要素２１

５０Ａの出力結果c_0、データパス要素３１５０Ａの結果d_0、及びデータパス要素３１５０Ａの出力するＶａｌｉｄ信号v_0が入力される。

[0130] 制御パス要素２１５０Ａの出力結果c_0が有効であり、かつ０である場合、繰り返し論理４００は、前段に位置する演算装置の出力であるc_i、d_i、及びv_iを選択し、制御パス要素２１５０Ａ、データパス要素３１５０Ａに入力する。

[0131] 一方、制御パス要素２１５０Ａの出力結果c_0が有効であり、かつ１である場合、繰り返し論理４００は、制御パス要素２１５０Ａの出力であるc_0、データパス要素３１５０Ａの出力である、d_0、及びv_0を選択し、制御パス要素２１５０Ａ、データパス要素３１５０Ａに入力する。

[0132] ここで、演算装置（詳細にはデータパス要素３１５０Ａ）の演算処理中に、前段の演算装置から入力があると、演算装置は、その入力を処理できない。そのため、繰り返し論理４００は、データパス要素３１５０Ａによる演算の処理中には、前段の演算装置に対して供給するＲｅａｄｙ信号r_0が０になるように制御する。なお、繰り返し論理４００は、後段演算装置から０の値を持つＲｅａｄｙ信号r_iが入力された場合も、前段演算装置に対して供給するＲｅａｄｙ信号r_0が０になるように制御する。

[0133] なお、繰り返し論理４００は、演算装置内に設けられても良く、演算装置外に設けられても良い。

[0134] 図３４に示す構成により、繰り返しを示す正規表現の演算を行うことができる。さらに、実施の形態５と同様に、関数に規定される比較結果にマッチしたイベント列の検出、及び各関数に規定された演算の実行を行うことができる。

[0135] （８）本発明の実施の形態８．

次に、本発明の実施の形態８について説明する。上述の説明においては、演算装置は、基本的にハードウェアであるものとして説明したが、ソフトウェア、すなわち、１以上の入力データを含むイベントを入力とする演算処理を、コンピュータに実行させるためのプログラムとすることも可能である。

[0136] すなわち、本実施の形態にかかるプログラムは、入力データを引数とする関数に規定された演算を実行し結果を出力すると共に、入力データの比較結果を戻り値として出力する処理を関数処理部に実施させるデータ処理と、関数の戻り値を使用してイベントを検出する制御処理とを実行する。

[0137] 図３５は、本実施の形態にかかるプログラムの動作を示すフローチャートの例である。図３５に示すように、イベントが入力されると(ステップＳ１)、プログラムはイベントに含まれる入力データを引き数とし、関数に規定された演算処理を実施する(ステップＳ２)。一方で、イベントに含まれる入力データが所定のデータか否かを比較する(ステップＳ３)。上記制御処理として、ステップＳ２及びＳ３は、第１乃至第７の実施形態では関数処理部が実行するデータ処理である。また、ステップＳ３における戻り値を、入力されたイベントが検出対象のイベントであるか否かを示す比較結果とし、必要であれば前段の比較結果と合わせて、所定のイベント又はイベント列が発生したか否かを検出する(ステップＳ４)。

[0138] 以上のようなイベント検出処理及び関数に規定された演算処理は、専用コンピュータ、又はパーソナルコンピュータ（ＰＣ）などのコンピュータにより実現可能である。但し、コンピュータは、物理的に単一である必要はなく、分散処理を実行する場合には、複数であってもよい。図３６は、本実施の形態にかかるプログラムを実行する装置の構成例を示す図である。

[0139] 図３６に示すように、コンピュータ５１００は、ＣＰＵ５１０１（Central Processing Unit）と、ＲＯＭ（Read Only Memory）又はＲＡＭ５１０３（Random Access Memory）等からなるメモリ５１０２を有する。尚、コンピュータを動作させるためのＯＳ（Operation System）などが、この情報処理装置を構築するコンピュータに備えられているもよい。

[0140] ＣＰＵ５１０１は、メモリ５１０２に記憶されている各種プログラム５１０３をロードし、このプログラム５１０３に従って各種の処理を実行する。本実施の形態においては、例えば、上述のデータ処理及び制御処理等の処理を実行する。メモリ５１０２にはまた、ＣＰＵ５１０１が各種の処理を実行

する上において必要なデータなども適宜記憶される。

[0141] ここで、本実施の形態にかかるプログラムは、様々なタイプの非一時的なコンピュータ可読媒体（non-transitory computer readable medium）を用いて格納され、コンピュータに供給することができる。非一時的なコンピュータ可読媒体は、様々なタイプの実体のある記録媒体（tangible storage medium）を含む。非一時的なコンピュータ可読媒体の例は、磁気記録媒体（例えばフレキシブルディスク、磁気テープ、ハードディスクドライブ）、光磁気記録媒体（例えば光磁気ディスク）、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、ＣＤ－Ｒ／Ｗ、半導体メモリ（例えば、マスクＲＯＭ、ＰＲＯＭ（Programmable ROM）、ＥＰＲＯＭ（Erasable PROM）、フラッシュＲＯＭ、ＲＡＭ）を含む。また、プログラムは、様々なタイプの一時的なコンピュータ可読媒体（transitory computer readable medium）によってコンピュータに供給されてもよい。一時的なコンピュータ可読媒体の例は、電気信号、光信号、及び電磁波を含む。一時的なコンピュータ可読媒体は、電線及び光ファイバ等の有線通信路、又は無線通信路を介して、プログラムをコンピュータに供給できる。

[0142] 以上のように、本実施形態のプログラムは、すべての演算処理をソフトウェアによって処理する。このように、本発明の演算処理は、ハードウェアとしての演算装置を用いずに、ソフトウェアによって処理することも可能である。

[0143] なお、本発明は上述した実施の形態のみに限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能であることは勿論である。

[0144] この出願は、２０１１年２月２１日に出願された日本出願特願２０１１－０３４９０６、及び２０１１年８月２４日に出願された日本出願特願２０１１－１８２７０７を基礎とする優先権を主張し、その開示の全てをここに取り込む。

符号の説明

[0145] １００、２０１、３０１、３０２マルチプレクサ（ＭＵＸ）

１０１、１０５論理積回路（ＡＮＤ）

１０２、１０３、１０４レジスタ

１０６キュー

２０２論理和（ＯＲ）

２０３、２０６論理積（ＡＮＤ）

２０４、２０５シフタ

４００繰り返し論理

１０００、１０１０、１０２０、１０３０イベント

２１００、２１０１、２２００、２２０１，２３００、

２１１０、２１１１、２２１０、２２１１，２３１０、

２１２０、２１２１、２２２０、２２２１，２３２０、

２１３０、２１３１、２１４０、２１４１、２１５０、制御パス要素

３１００、３１０１、３２００、３２０１、３３００

３１１０、３１１１、３２１０、３２１１、３３１０

３１２０、３１２１、３２２０、３２２１、３３２０、

３１３０、３１３１、３１４０、３１４１、３１５０、データパス要素

４０００関数

４００１制御比較部

４００２データ計算部

請求の範囲

[請求項1] 第１のイベントに含まれる第１の入力データを引数とする第１の関数に規定された第１の演算を実行する第１の関数処理手段を有する第１のデータ処理手段と、

前記第１の関数の第１の戻り値を使用して、前記第１のイベントを検出する第１の制御処理手段とを有し、

前記第１の関数処理手段は、前記第１の演算を実行して第１の結果を出力する第１のデータ計算手段と、前記第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を前記第１の戻り値として前記第１の制御処理手段に出力する第１の制御比較手段とを有する、演算装置。

[請求項2] 第２のイベントに含まれる第２の入力データを引数とする第２の関数に規定された演算を実行する第２の関数処理手段を有する第２のデータ処理手段と、

前記第２の関数の第２の戻り値を使用して、第２のイベントを検出する第２の制御処理手段とを有し、

前記第２の関数処理手段は、前記第２の演算を実行して第２の結果を出力する第２のデータ計算手段と、前記第２の入力データと前記第２のイベントを特定する第２のデータとの比較結果を前記第２の戻り値として前記第２の制御処理手段に出力する第２の制御比較手段を有し、

前記第１の戻り値および前記第２の戻り値に基づいて、前記第１のイベント及び第２のイベントを含むイベント列を検出する、請求項１記載の演算装置。

[請求項3] 前記第１の関数及び前記第２の関数は正規表現で表現される関数列を構成し、

前記第１のデータ計算手段は、前記第１の入力データ及び前記第１の結果のうちの少なくとも一つを引数として前記第１の演算を実行し

て前記第１の結果を出力し、

前記第２のデータ計算手段は、前記第２の入力データ、前記第１の結果および前記第２の結果のうちの少なくとも一つを引数として前記第２の演算を実行し前記第２の結果を出力し、

前記正規表現に基づいて、前記第１の結果および前記第２の結果の少なくとも一方を用いて第３の結果を出力する、請求項２記載の演算装置。

[請求項4] 前記第１の関数処理手段は、入力された第１の動作停止要求の値に応じて、前記第１の演算を行うか否かを決定する、請求項１乃至請求項３のいずれか１項記載の演算装置。

[請求項5] 前記第２の関数処理手段は、入力された第２の動作停止要求の値に応じて、前記第２の演算を行うか否かを決定する、請求項２または請求項３のいずれか１項記載の演算装置。

[請求項6] 前記第１のイベントを保持するとともに、前記第１のイベントの保持数が０であるか否かを示す第１のイベント保持情報を出力し、前記第１のイベントを入力した順に前記第１の関数処理手段に供給する第１の保持手段と、

前記第１のイベント保持情報、及び前記第１の動作停止要求の値に基づいて前段の演算装置の演算実行を制御する第３の動作停止要求を生成する第１の動作停止要求生成手段と、を備える請求項４に記載の演算装置。

[請求項7] 前記第２のイベントを保持するとともに、前記第２のイベント保持数が０であるか否かを示す第２のイベント保持情報を出力し、前記第２のイベントを入力した順に前記第２の関数処理手段に供給する第２の保持手段と、

前記第２のイベント保持情報、及び前記第２の動作停止要求の値に基づいて前段の演算装置の演算実行を制御する第４の動作停止要求を生成する第２の動作停止要求生成手段と、を備える請求項５に記載の演

算装置。

[請求項8] 前記第１の保持手段は、前段の演算装置の演算実行結果が有効であるか否かを示す第１の有効性情報に基づいて、前記第１のイベントを前記第１の関数処理手段に供給するか否かを決定する、請求項６に記載の演算装置。

[請求項9] 前記第２の保持手段は、前段の演算装置の演算実行結果が有効であるか否かを示す第２の有効性情報に基づいて、前記第２のイベントを前記第２の関数処理手段に供給するか否かを決定する、請求項７に記載の演算装置。

[請求項10] 第１の関数の演算処理を実施する第１の演算装置と、

第２の関数の演算処理を実施する第２の演算装置とを備え、

前記第１の演算装置は、

第１のイベントに含まれる第１の入力データを引数とする第１の関数に規定された第１の演算を実行する第１の関数処理手段を有する第１のデータ処理手段と、

前記第１の関数の第１の戻り値を使用して、前記第１のイベントを検出する第１の制御処理手段とを有し、

前記第１の関数処理手段は、前記第１の演算を実行して第１の結果を出力する第１のデータ計算手段と、前記第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を前記第１の戻り値として前記第１の制御処理手段に出力する第１の制御比較手段とを有する、演算装置。

[請求項11] 前記第２の演算装置は、

第２のイベントに含まれる第２の入力データを引数とする第２の関数に規定された演算を実行する第２の関数処理手段を有する第２のデータ処理手段と、

前記第２の関数の第２の戻り値を使用して、第２のイベントを検出する第２の制御処理手段とを有し、

前記第２の関数処理手段は、前記第２の演算を実行して第２の結果を出力する第２のデータ計算手段と、前記第２の入力データと前記第２のイベントを特定する第２のデータとの比較結果を前記第２の戻り値として前記第２の制御処理手段に出力する第２の制御比較手段を有し、

前記第１の戻り値および前記第２の戻り値に基づいて、前記第１のイベント及び第２のイベントを含むイベント列を検出する、請求項１０に記載の演算装置。

[請求項12] 前記演算装置には、リアルタイム情報であって、２以上の前記イベントからなるイベント列が入力される、請求項１乃至１１のいずれか１項記載の演算装置。

[請求項13] 前記データ計算手段は、前記入力データを格納、比較又は演算して出力する、請求項１乃至１２のいずれか１項記載の演算装置。

[請求項14] 前記第１の関数及び前記第２の関数は正規表現で表現される関数列を構成し、

前記正規表現は、前記関数と演算子とを含み、

前記演算子は、一の関数の後に次の関数の演算の実施を記述する連接、２つの関数のいずれの演算を選択するかを記述する和、及び１回以上の関数の演算を記述する繰り返し、のいずれか１以上を有する、請求項２乃至１３のいずれか１項記載の演算装置。

[請求項15] 前記正規表現が、前記第１及び第２の関数と、前記第１の関数の後に前記第２の関数の演算の実施を記述する連接とを含むものである場合、

前記第２のデータ処理手段は、前記第１のデータ処理手段の前記演算処理結果に基づき前記演算処理結果を出力し、

前記第２の制御処理手段は、前記第２の制御処理手段の結果に基づき検出結果を出力する、請求項１４記載の演算装置。

[請求項16] 前記正規表現が、前記第１及び第２の関数と、前記第１の関数の後

に前記第２の関数の演算の実施を記述する連接とを含むものである場合、

前記第１及び第２のデータ処理手段は直列に接続され、前記第１及び第２の制御処理手段は直列接続される、請求項１４記載の演算装置。

［請求項17］ 前記正規表現が、前記第１及び第２の関数と、前記第１及び第２の関数のいずれの演算を選択するかを記述する和とを含むものである場合、

前記第１及び第２の制御処理手段のいずれか一方以上がイベントを検出した場合にイベント列を検出したと判断する判定手段と、

前記第１及び第２の制御処理手段の出力結果に応じて前記第１及び第２のデータ処理手段の演算結果のいずれか一方を選択出力する第１の選択手段とを有する、請求項１４記載の演算装置。

［請求項18］ 前記正規表現が、前記第１及び第２の関数と、前記第１及び第２の関数のいずれの演算を選択するかを記述する和とを含むものである場合、

論理和を演算する論理回路及び第１のマルチプレクサを有し、

前記第１及び第２の制御処理手段の出力は前記論理回路に入力され、その論理和がイベント列検出結果として出力され、

前記第１及び第２のデータ処理手段は、各演算結果を前記第１のマルチプレクサの入力に入力し、

当該第１のマルチプレクサは、前記第１及び第２の制御処理手段の出力結果に応じて前記第１及び第２のデータ処理手段の演算結果のいずれか一方を出力する、請求項１４記載の演算装置。

［請求項19］ 前記第１の演算結果が有効値であるか否かを示す情報と、前記第２の演算結果が有効値であるか否かを示す情報と、の論理積を演算結果の有効性を示す情報として後段の演算装置に供給し、

前記第１のデータ計算手段の実行可否状態と、前記第２のデータ計

算手段の実行可否状態と、の論理積を後段の演算装置に供給する動作停止要求として出力する請求項１８記載の演算装置。

[請求項20] 前記第１及び第２のデータ処理手段の演算に要するサイクル数に基づいて、前記第１及び第２のデータ処理手段のいずれか一方の出力を遅延させる遅延手段を有する請求項１８または請求項１９に記載の演算装置。

[請求項21] 前記第１の制御処理手段の出力結果に応じて前段の制御処理手段の出力を当該第１の制御処理手段に入力するか否かを決定する第２の判定手段と、

前記第１のデータ処理手段及び前段のデータ処理手段の出力結果のいずれか一方を前記第１の制御処理手段の出力結果に応じて、当該第１のデータ処理手段に選択入力する第３の判定手段とを有する、請求項１０記載の演算装置。

[請求項22] 前記第１の制御処理手段の出力結果に応じて、前記第１の制御処理手段及び前記第１のデータ処理手段への入力を繰り返す繰り返し論理を備え、

前記繰り返し論理は、前記第１のデータ処理手段が演算中である場合に演算処理の停止を要求する固定値を前段の演算装置に入力する、請求項１０に記載の演算装置。

[請求項23] 第２及び第３のマルチプレクサを有し、

前記第２のマルチプレクサは、前記第１の制御処理手段の出力結果に応じて前段の制御処理手段の出力を当該第１の制御処理手段に入力するか否かを決定し、

前記第３のマルチプレクサは、前記第１のデータ処理手段及び前段のデータ処理手段の出力結果のいずれか一方を前記第１の制御処理手段の出力結果に応じて、当該第１のデータ処理手段に選択入力する、請求項１０項記載の演算装置。

[請求項24] 第１の演算を実行して第１の結果を出力すると共に、第１のイベン

トに含まれる第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする第１の関数に規定された第１の演算を実行し、

前記第１の関数の前記第１の戻り値を使用して、前記第１のイベントを検出する、演算実行方法。

［請求項25］ 前記イベントからなるイベント列を入力とし、前記第１の関数及び１回以上の前記第１の関数の演算を記述する繰り返しを記述する演算子を含む正規表現の演算を実施する請求項２４記載の演算実行方法。

［請求項26］ １以上の第１の入力データを含む第１のイベント及び１以上の第２の入力データを含む第２のイベンを有するイベント列を入力とし、第１の関数の演算処理を実施し、

前記イベント列を入力とし、第２の関数の演算処理を実施するものであり、

前記第１の関数の演算処理は、

前記第１の関数に規定された第１の演算を実行して第１の結果を出力すると共に、前記第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする前記第１の演算を実行し、

前記第１の関数の前記第１の戻り値を使用して、前記第１のイベントを検出するものである、演算実行方法。

［請求項27］ 前記第２の関数の演算処理は、

前記第２の関数に規定された第２の演算を実行して第２の結果を出力すると共に、前記第２の入力データと前記第２のイベントを特定する第２のデータとの比較結果を第２の戻り値として出力することで、前記第２の入力データを引数とする前記第２の演算を実行し、

前記第２の関数の前記第２の戻り値を使用して、前記第２のイベントを検出するものであり、

前記第１の戻り値および前記第２の戻り値に基づいて、前記イベント列が検出される、請求項２６記載の演算実行方法。

[請求項28] 前記イベント列は、リアルタイム情報である、請求項２４乃至２７のいずれか１項記載の演算実行方法。

[請求項29] 前記入力データを引数とする前記第１の関数に規定された演算は、前記入力データを格納、比較又は演算する処理である、請求項２４乃至２８のいずれか１項記載の演算実行方法。

[請求項30] 第１の演算を実行して第１の結果を出力すると共に、第１の入力データと第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする第１の関数に規定された第１の演算を実行する第１のデータ処理と、

前記第１の関数の前記第１の戻り値を使用して、１以上の前記第１の入力データを含む第１のイベントを検出する第１の制御処理とを、コンピュータに実行させるプログラムを格納した非一時的なコンピュータ可読媒体。

[請求項31] 前記イベントからなるイベント列を入力とし、前記第１の関数及び１回以上の前記第１の関数の演算を記述する繰り返しを記述する演算子を含む正規表現の演算を実施する請求項３０記載のプログラムを格納した非一時的なコンピュータ可読媒体。

[請求項32] 所定の動作をコンピュータに実行させるためのプログラムであって、

１以上の第１の入力データを含む第１のイベント及び１以上の第２の入力データを含む第２のイベンを有するイベント列を入力とし、第１の関数の演算処理を実施する第１の演算処理と、

前記イベント列を入力とし、第２の関数の演算処理を実施する第２の演算処理とを有し、

前記第１の演算処理は、

前記第１の関数に規定された第１の演算を実行して第１の結果を出

力すると共に、前記第１の入力データと前記第１のイベントを特定する第１のデータとの比較結果を第１の戻り値として出力することで、前記第１の入力データを引数とする前記第１の演算を実行する第１のデータ処理と、

前記第１の関数の前記第１の戻り値を使用して、前記第１のイベントを検出する第１の制御処理とを有するプログラムを格納した非一時的なコンピュータ可読媒体。

［請求項33］ 前記第２の演算処理は、

前記第２の関数に規定された第２の演算を実行して第２の結果を出力すると共に、前記第２の入力データと前記第２のイベントを特定する第２のデータとの比較結果を第２の戻り値として出力することで、前記第２の入力データを引数とする前記第２の演算を実行する第２のデータ処理と、

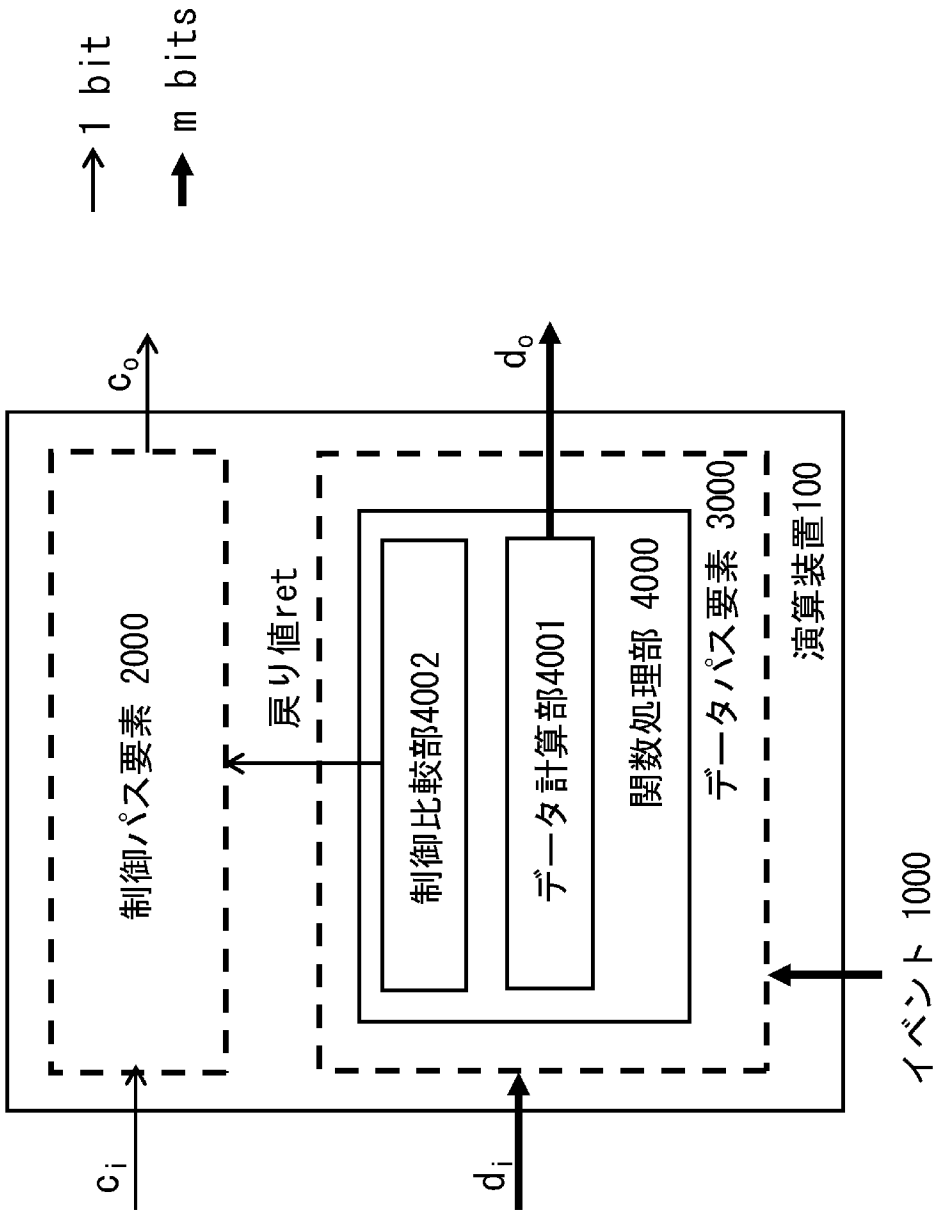
前記第２の関数の前記第２の戻り値を使用して、前記第２のイベントを検出する第２の制御処理とを有し、

前記第１の戻り値および前記第２の戻り値に基づいて、前記イベント列が検出される、請求項３２記載のプログラムを格納した非一時的なコンピュータ可読媒体。

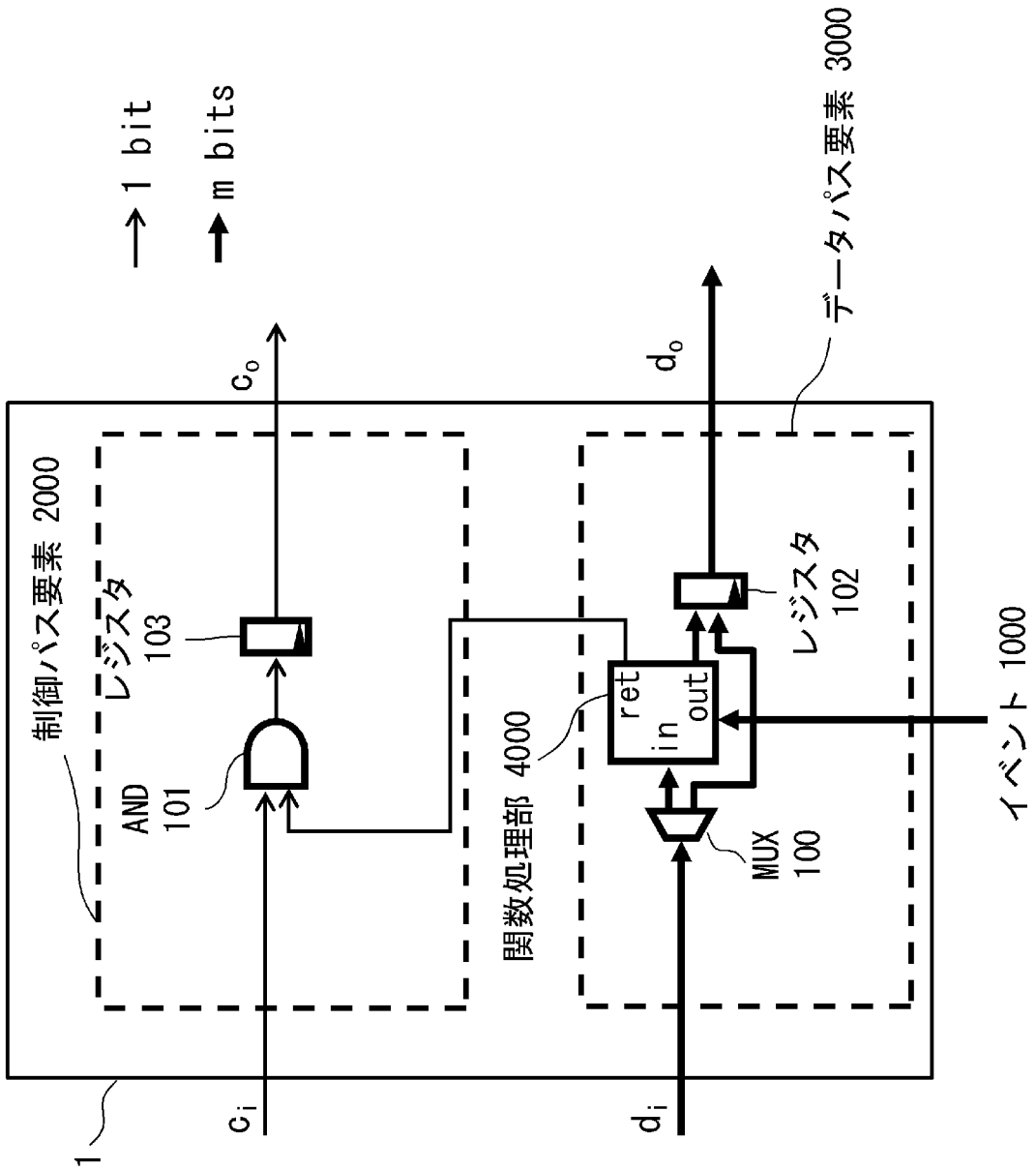
［請求項34］ 前記イベント列は、リアルタイム情報である、請求項３０乃至３３のいずれか１項記載のプログラムを格納した非一時的なコンピュータ可読媒体。

［請求項35］ 前記入力データを引数とする前記第１の関数に規定された演算は、前記入力データを格納、比較又は演算する処理である、請求項３０乃至３４のいずれか１項記載のプログラムを格納した非一時的なコンピュータ可読媒体。

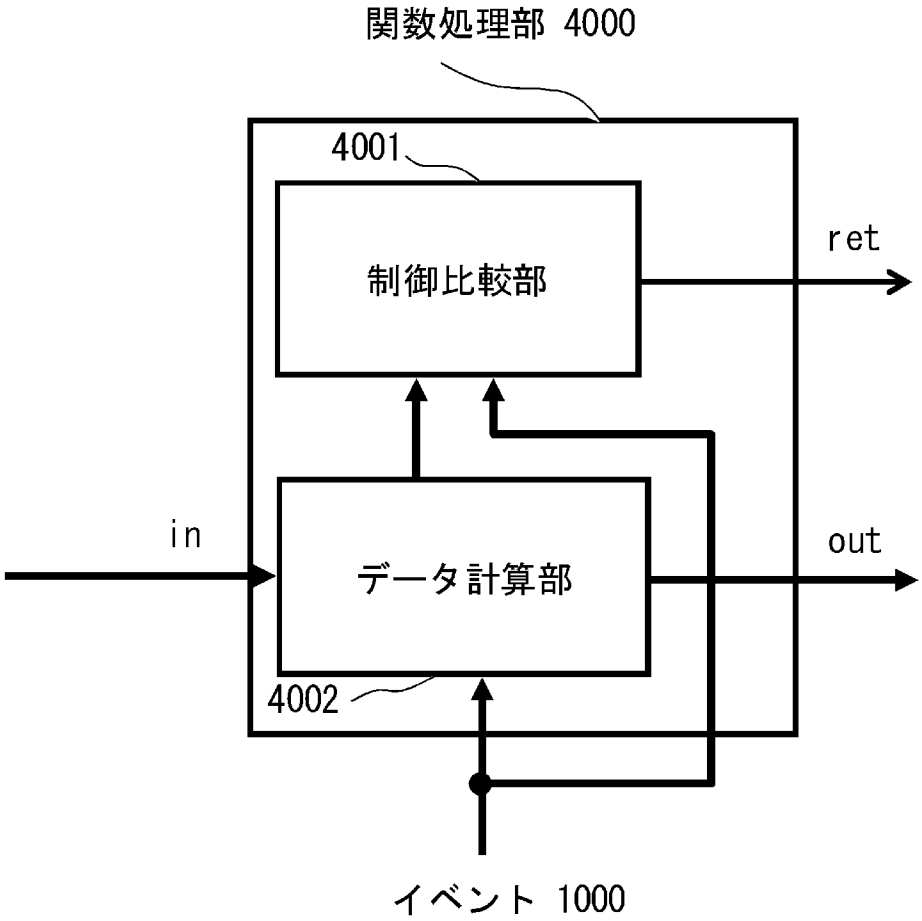
[図1]


1 bit
m bits
c_o
d_o
制御パス要素 2000
戻り値ret
制御比較部4002
データ計算部4001
関数処理部 4000
データパス要素 3000
演算装置100
イベント 1000
c_i
d_i

[図2]


→1 bit
↑m bits
データパス要素 3000
制御パス要素 2000
レジスタ
103
AND
101
関数処理部 4000
ret
in
out
レジスタ
102
MUX
100
イベント 1000
c_o
d_o
c_i
d_i
1

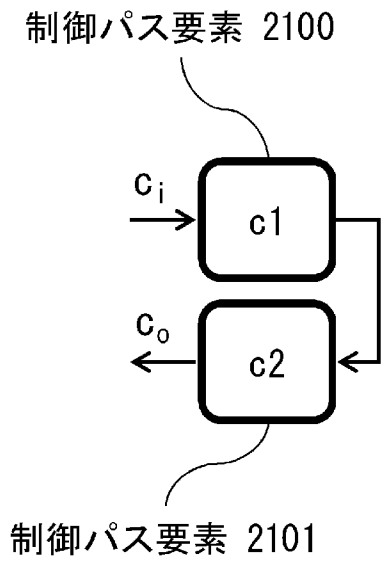
[図3]



[図4]

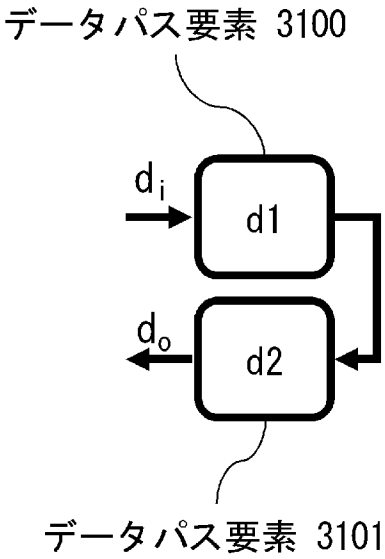
```
関数          = F ( ... )     # 1以上の引数を持つ関数

連接          = E1 E2         # E1の後に E2が実行される

和            = E1 | E2       # E1かE2が実行される

繰り返し      = E+            # Eが1回以上繰り返される

グルーピング  = ( E )         # Eをまとめて実行する

要素 [E, En] = 関数 又は 連接 又は 和 又は
                    繰り返し 又は グルーピング
```

[図5A]



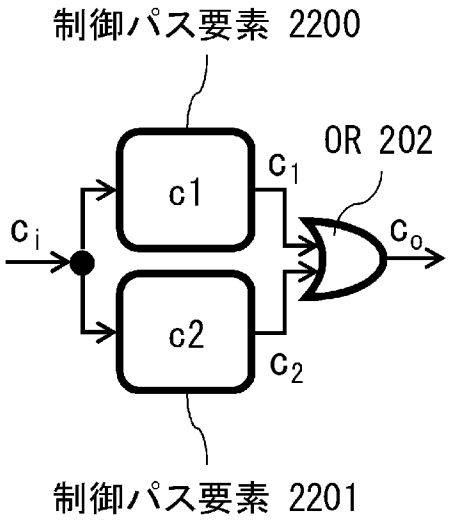
[図5B]

データパスの連接: d1 d2



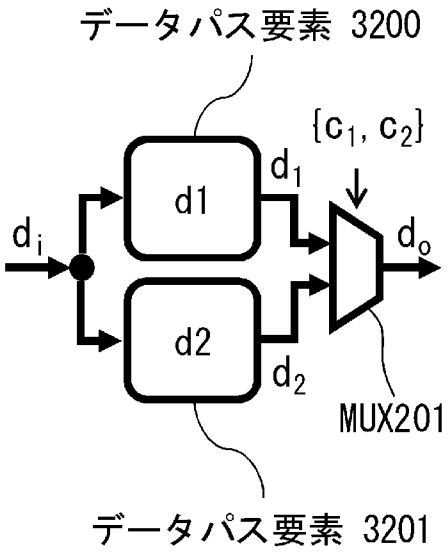
[図6A]

制御パスの和: c1 | c2



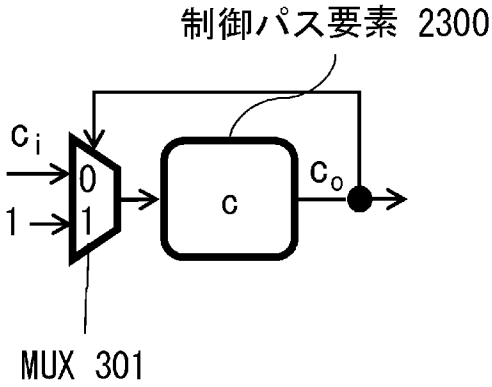
[図6B]

データパスの和: d1 | d2



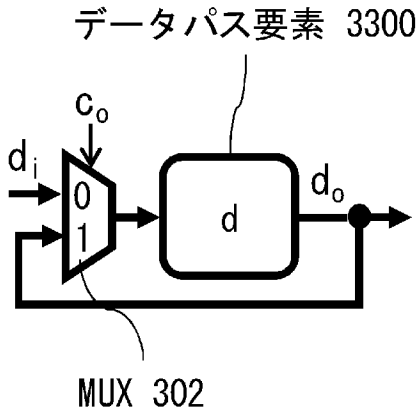
[図7A]

制御パスの繰り返し: c+



[図7B]

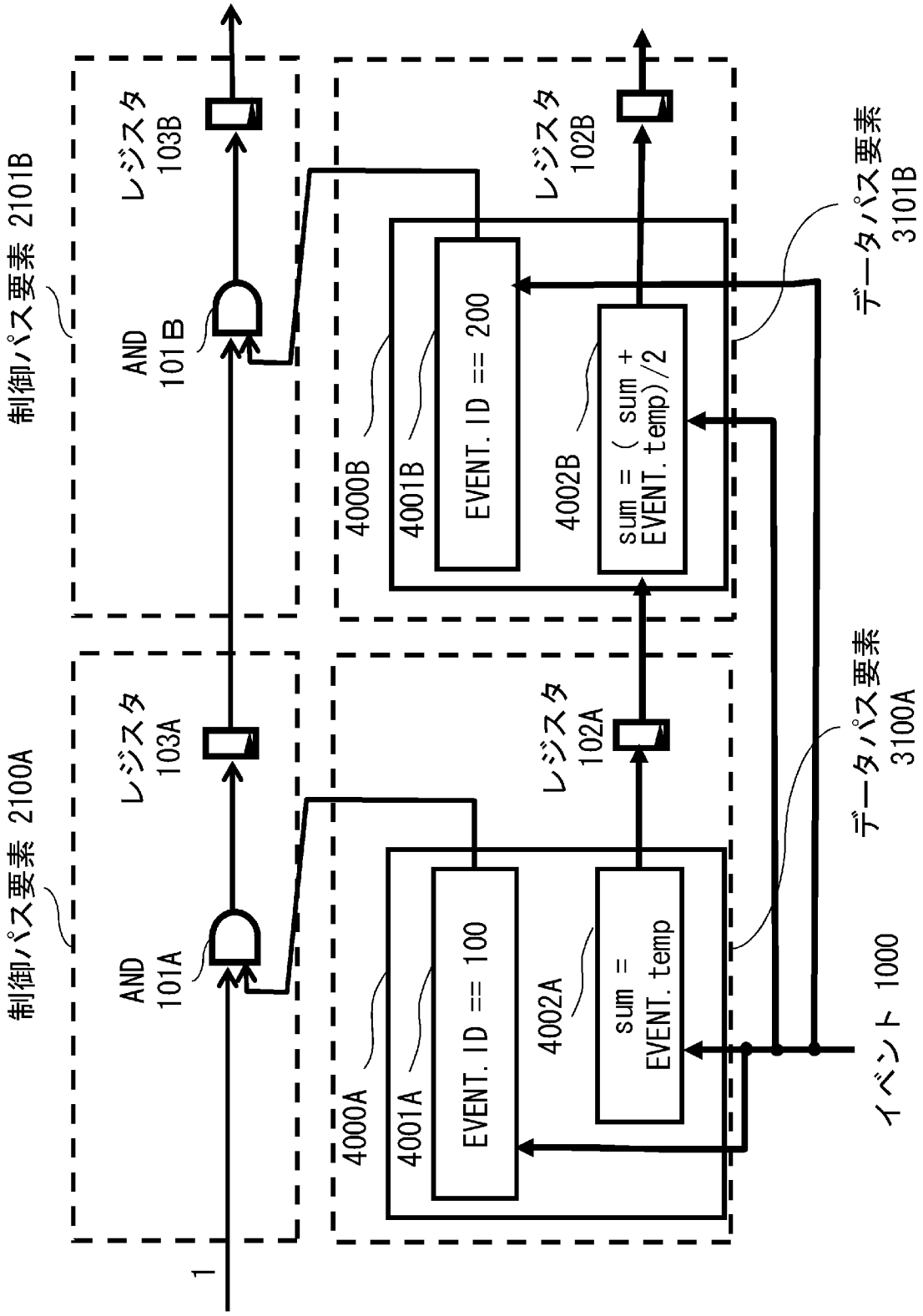
データパスの繰り返し：d+



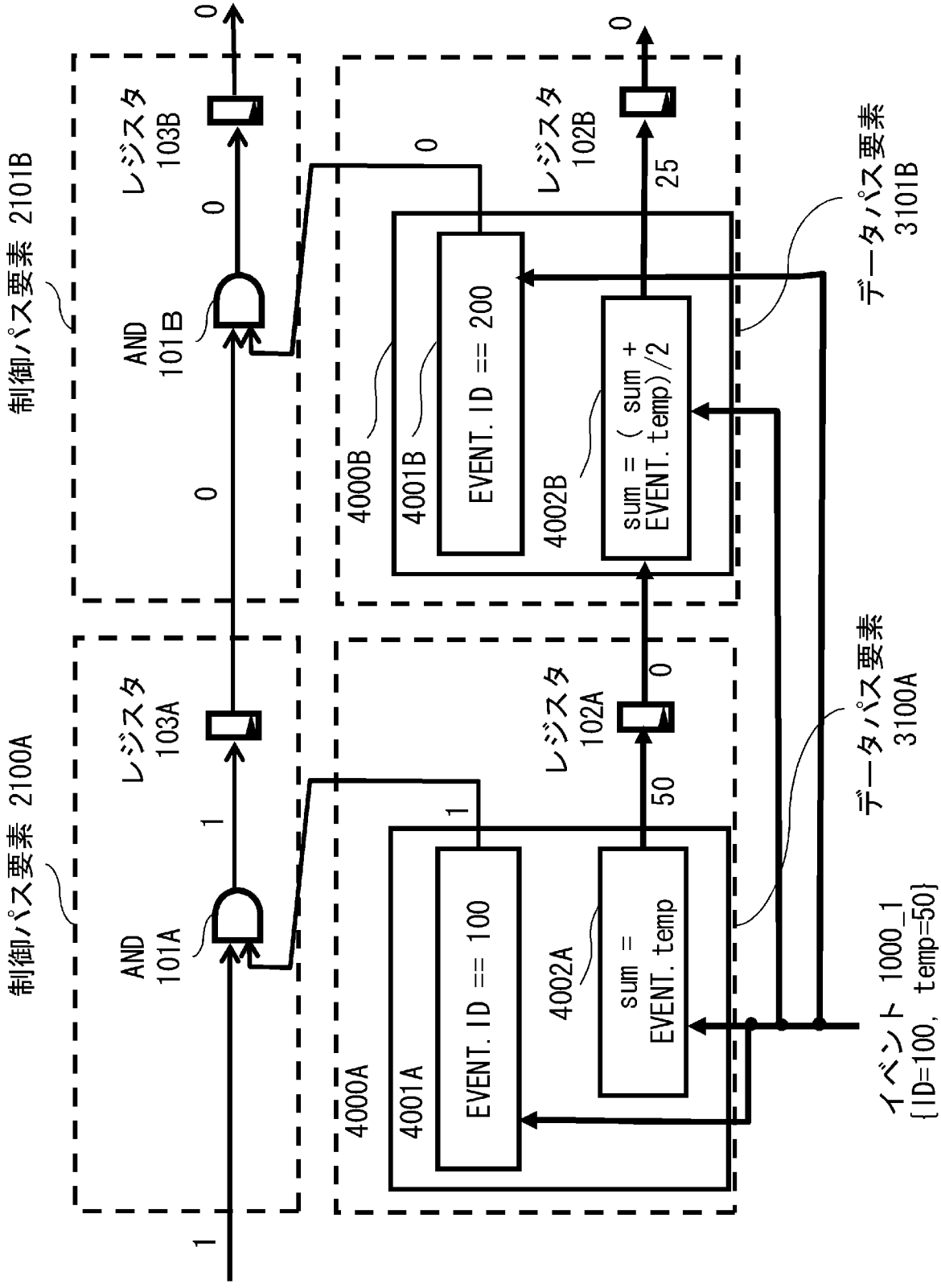
[図8]

```
RULE  = A(sum) B(sum)

EVENT = { ID, temp /* 温度 */ }

A (sum) = {
    /* sumに温度を足す */
    sum = EVENT.temp;

    /* IDが100なら1を返す。でなければ0を返す */
    if ( EVENT. ID == 100 ) return 1 else return 0;
}

B (sum) = {
    /* A()の温度とB()の温度の平均を取る */
    sum = (sum + EVENT.temp ) / 2;

    /* IDが200なら1を返す。でなければ0を返す */
    if ( EVENT.ID == 200 ) return 1 else return 0;
}
```

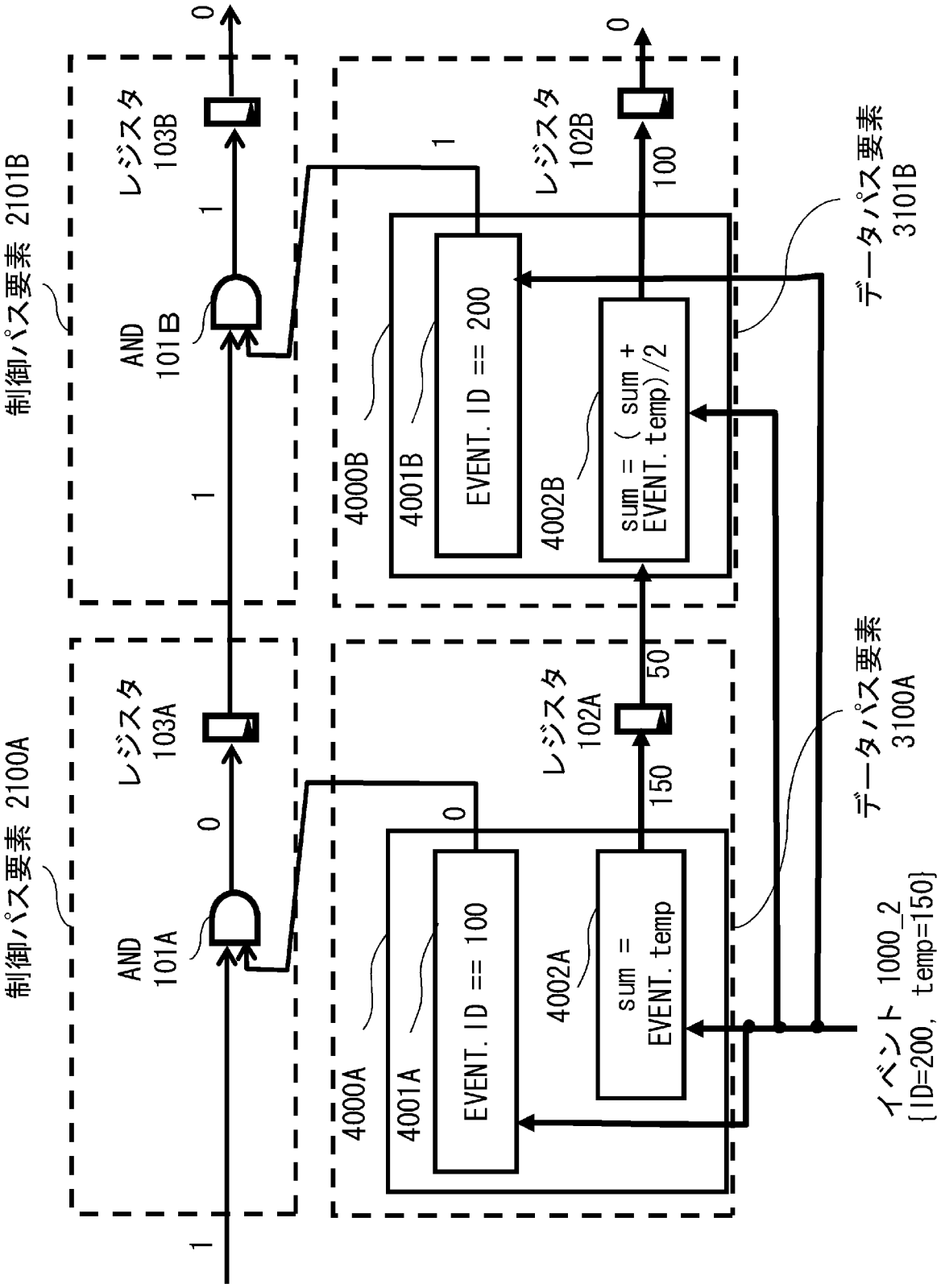
[図9]


制御パス要素 2100A
制御パス要素 2101B
AND 101A
AND 101B
レジスタ 103A
レジスタ 103B
レジスタ 102A
レジスタ 102B
4000A
4000B
4001A
4001B
4002A
4002B
EVENT. ID == 100
EVENT. ID == 200
sum = EVENT. temp
sum = (sum + EVENT. temp)/2
データパス要素 3100A
データパス要素 3101B
イベント 1000
1

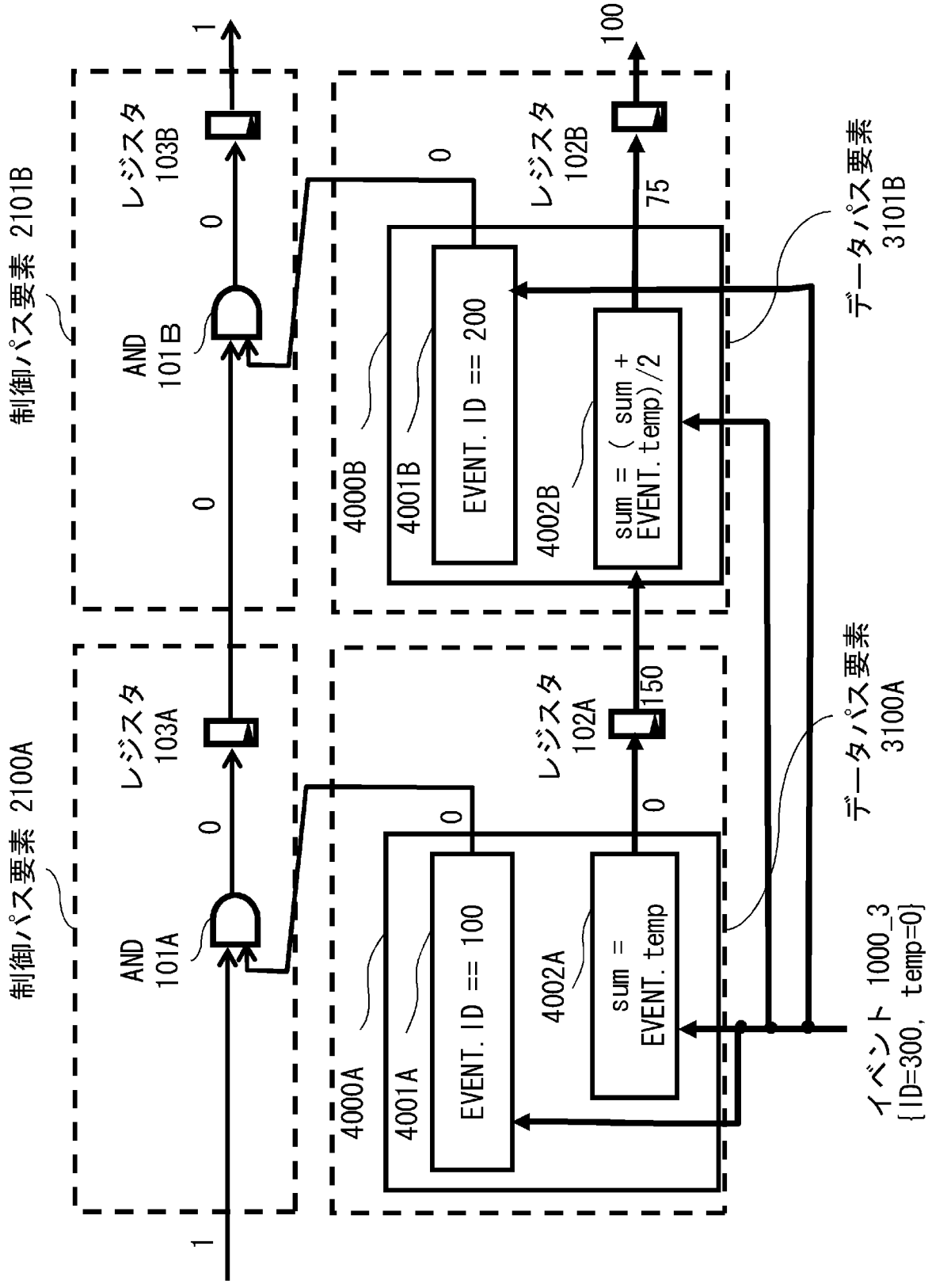
[図10]


制御パス要素 2100A
制御パス要素 2101B
AND 101A
AND 101B
レジスタ 103A
レジスタ 103B
レジスタ 102A
レジスタ 102B
4000A
4000B
4001A
4001B
4002A
4002B
EVENT. ID == 100
EVENT. ID == 200
sum = EVENT. temp
sum = (sum + EVENT. temp)/2
データパス要素 3100A
データパス要素 3101B
イベント 1000_1 {ID=100, temp=50}
0
1
50
25

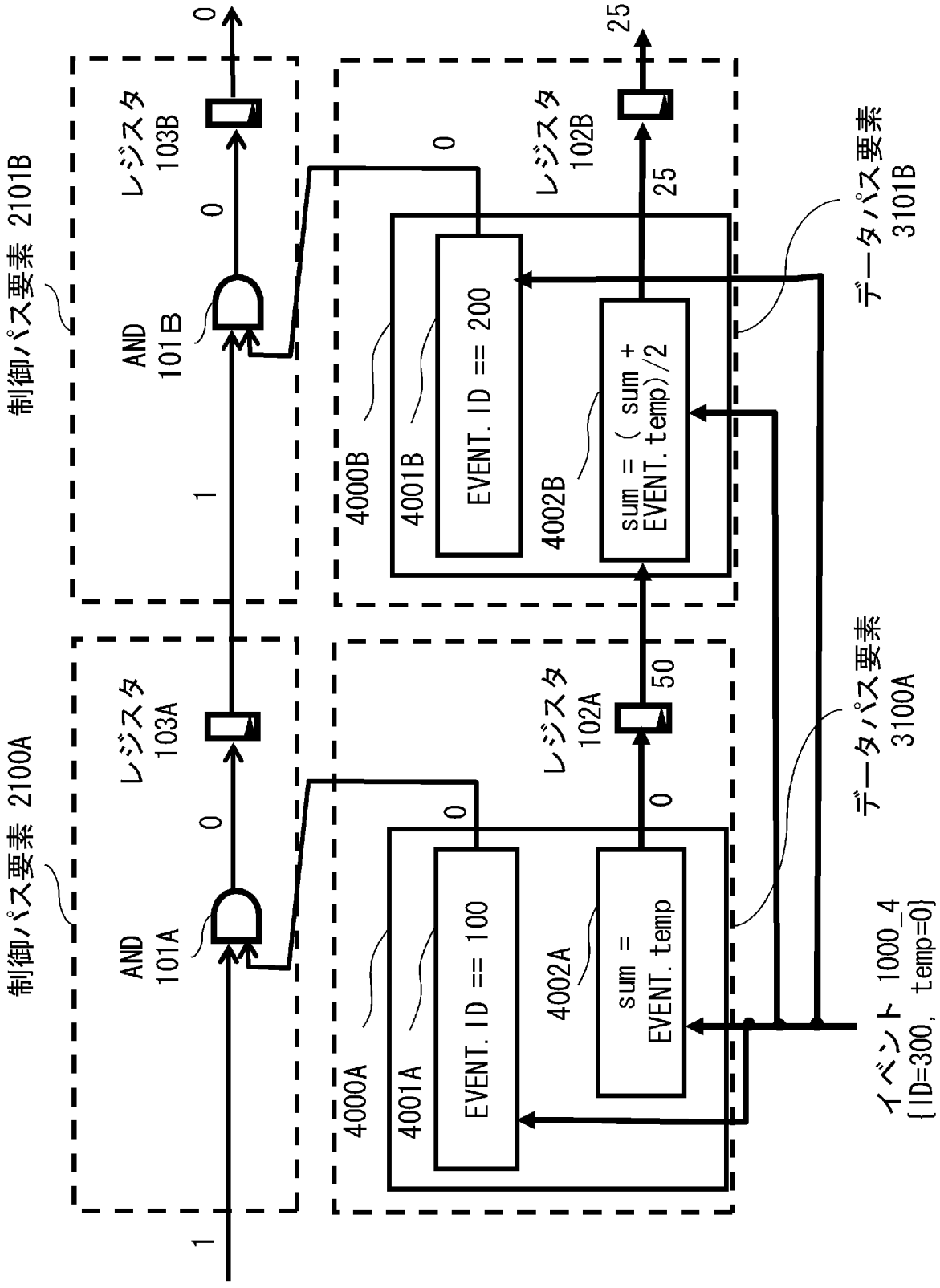
[図11]


制御パス要素 2101B
制御パス要素 2100A
AND 101B
AND 101A
レジスタ 103B
レジスタ 103A
レジスタ 102B
レジスタ 102A
データパス要素 3101B
データパス要素 3100A
4000B
4001B
EVENT. ID == 200
4002B
sum = (sum + EVENT. temp)/2
4000A
4001A
EVENT. ID == 100
4002A
sum = EVENT. temp
イベント 1000_2 {ID=200, temp=150}
0
1
100
50
150

[図12]


制御パス要素 2101B
制御パス要素 2100A
AND 101B
AND 101A
レジスタ 103B
レジスタ 103A
レジスタ 102B
レジスタ 102A
データパス要素 3101B
データパス要素 3100A
4000B
4001B
EVENT. ID == 200
4002B
sum = (sum + EVENT. temp)/2
4000A
4001A
EVENT. ID == 100
4002A
sum = EVENT. temp
イベント 1000_3 {ID=300, temp=0}
0
1
75
100
150

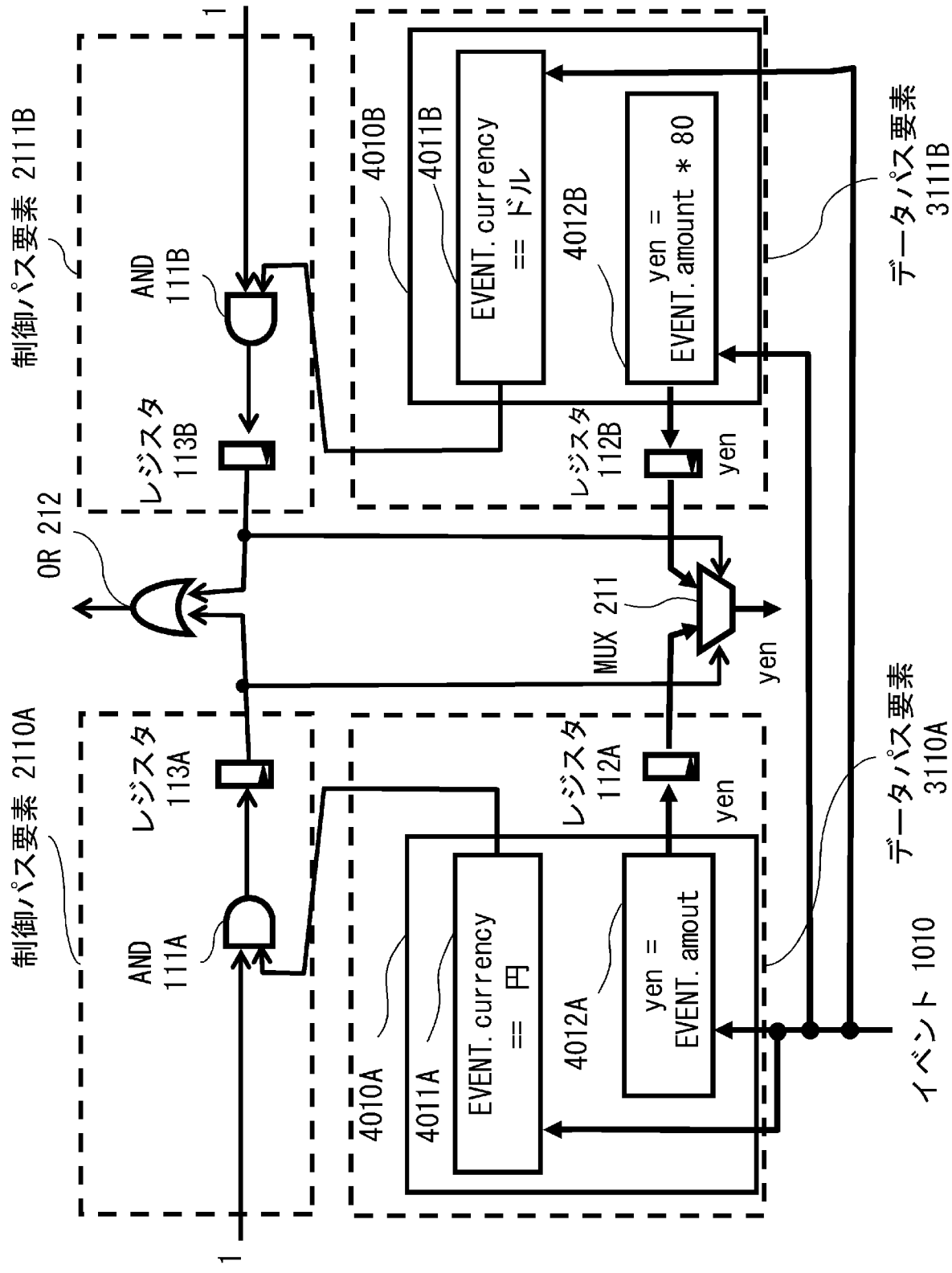
[図13]


制御パス要素 2101B
制御パス要素 2100A
AND 101B
AND 101A
レジスタ 103B
レジスタ 103A
レジスタ 102B
レジスタ 102A
データパス要素 3101B
データパス要素 3100A
4000B
4001B
EVENT. ID == 200
4002B
sum = (sum + EVENT. temp)/2
4000A
4001A
EVENT. ID == 100
4002A
sum = EVENT. temp
イベント 1000_4 {ID=300, temp=0}
0
1
25
50

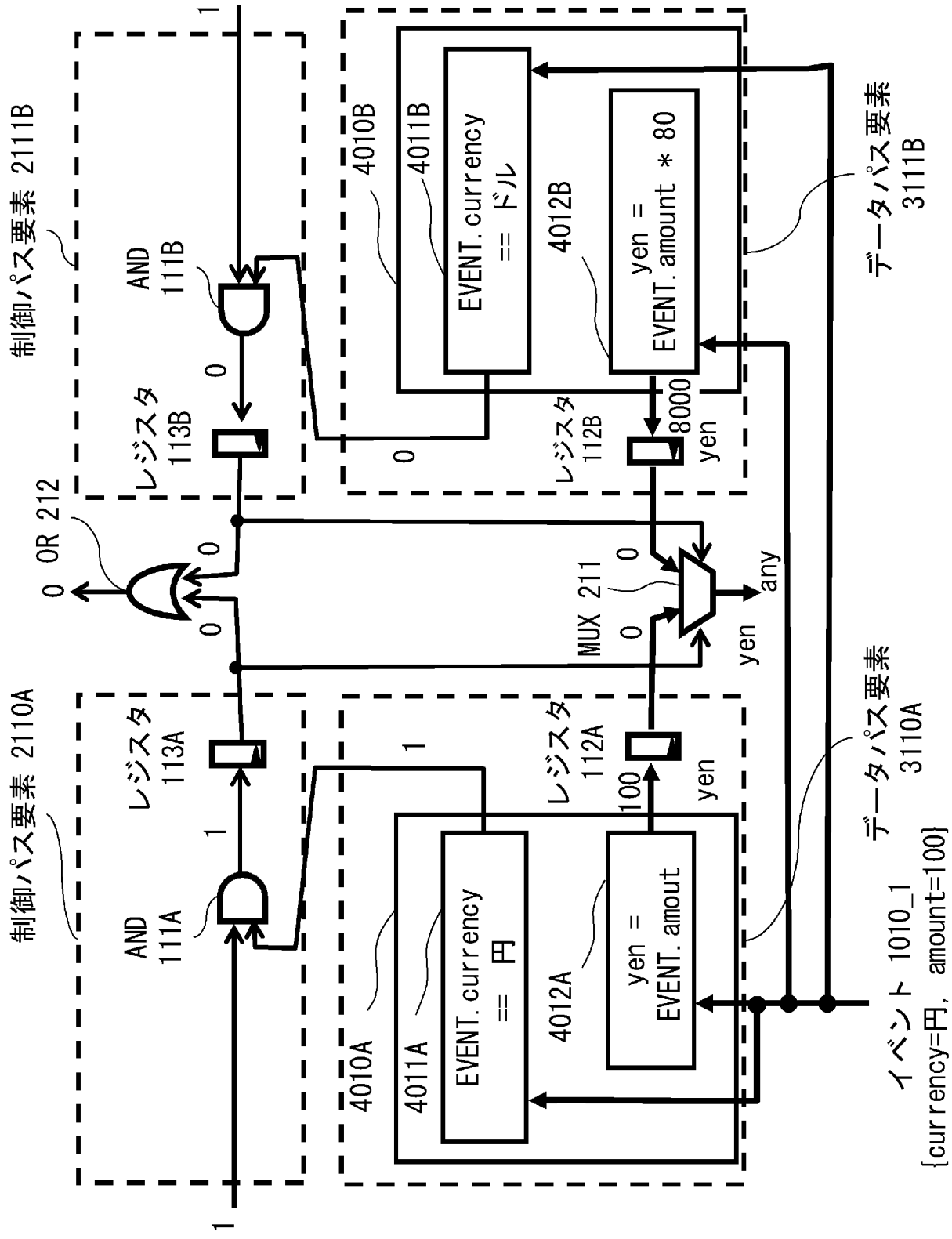
[図14]

```
RULE  = A (yen) | B(yen)
EVENT = { currency, amount}
A (yen) = {
    /* amountがそのまま円建て */
    yen = EVENT.amount;

    /* 通貨が円なら1を返す。でなければ0を返す */
    if ( EVENT.currency == 円 )
        return 1 else return 0;
}
B (yen) = {
    /* ドル建てを円に直す */
    yen = EVENT.amount * 80;

    /* 通貨がドルなら1を返す。でなければ0を返す */
    if ( EVENT.currency == ドル )
        return 1 else return 0;
}
```

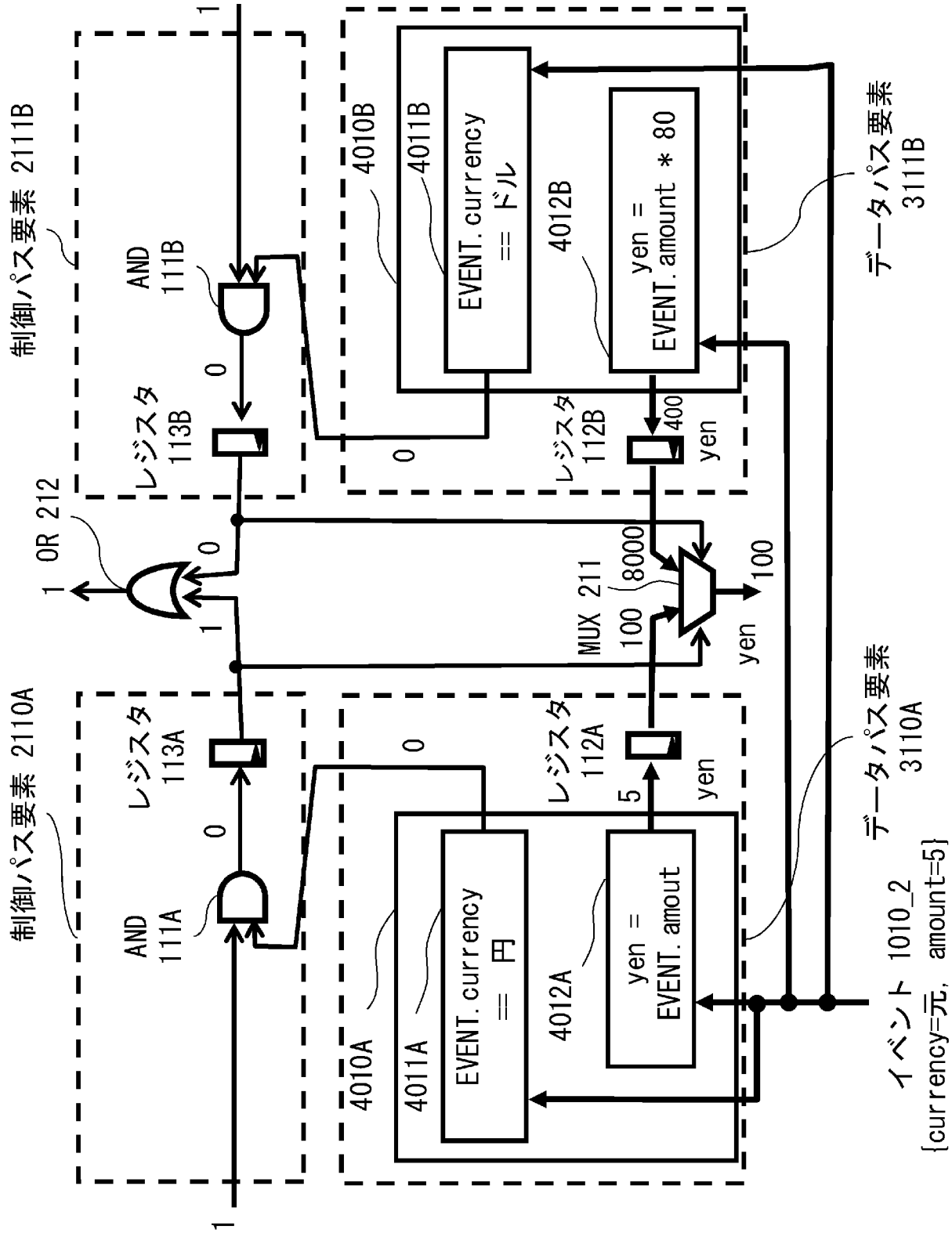
[図15]


制御パス要素 2111B
AND 111B
1
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
レジスタ 113B
レジスタ 112B
yen
OR 212
MUX 211
yen
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
yen
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010
1

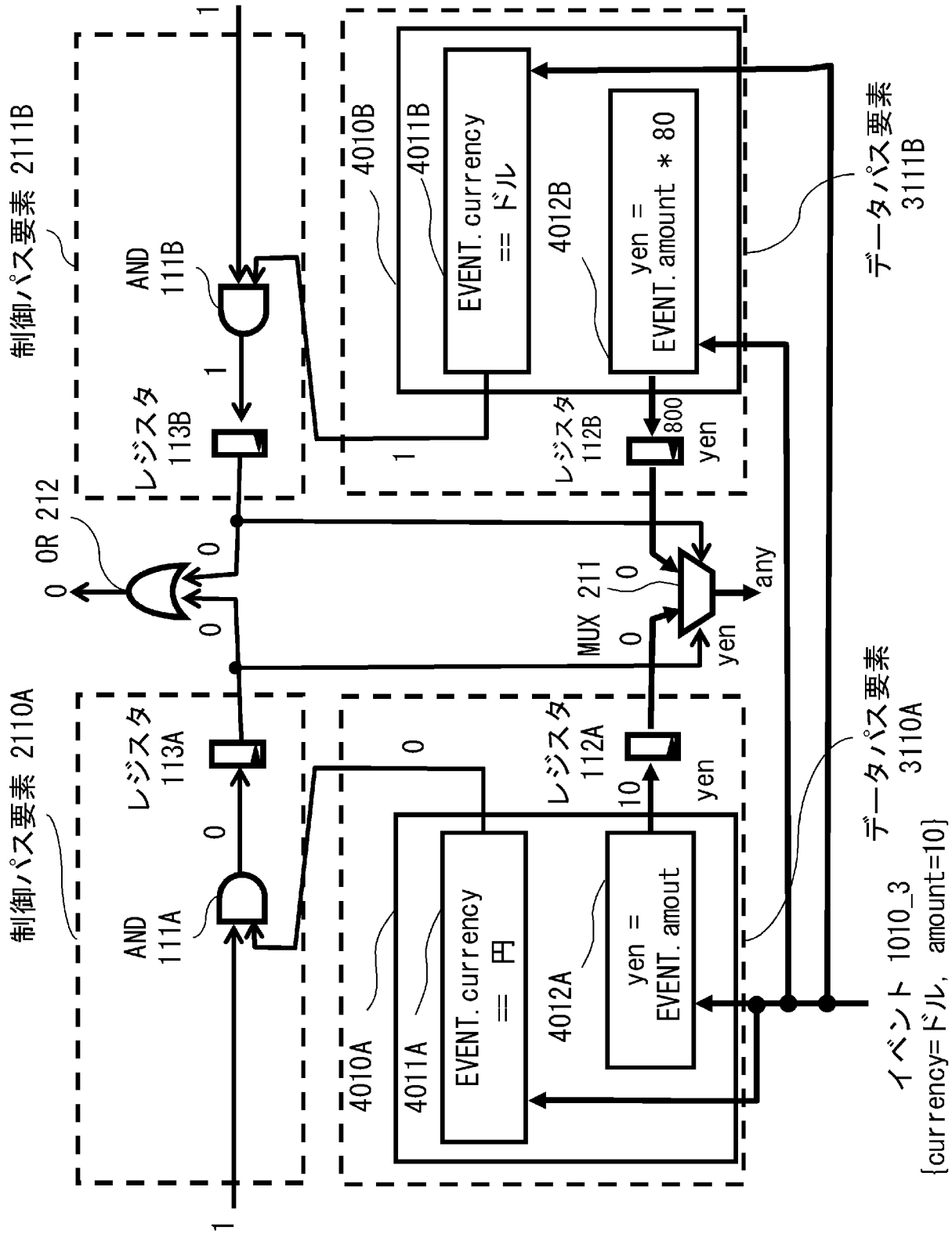
[図16]


制御パス要素 2111B
AND 111B
レジスタ 113B
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
OR 212
MUX 211
レジスタ 112B
8000
yen
any
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
100
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010_1
{currency=円, amount=100}

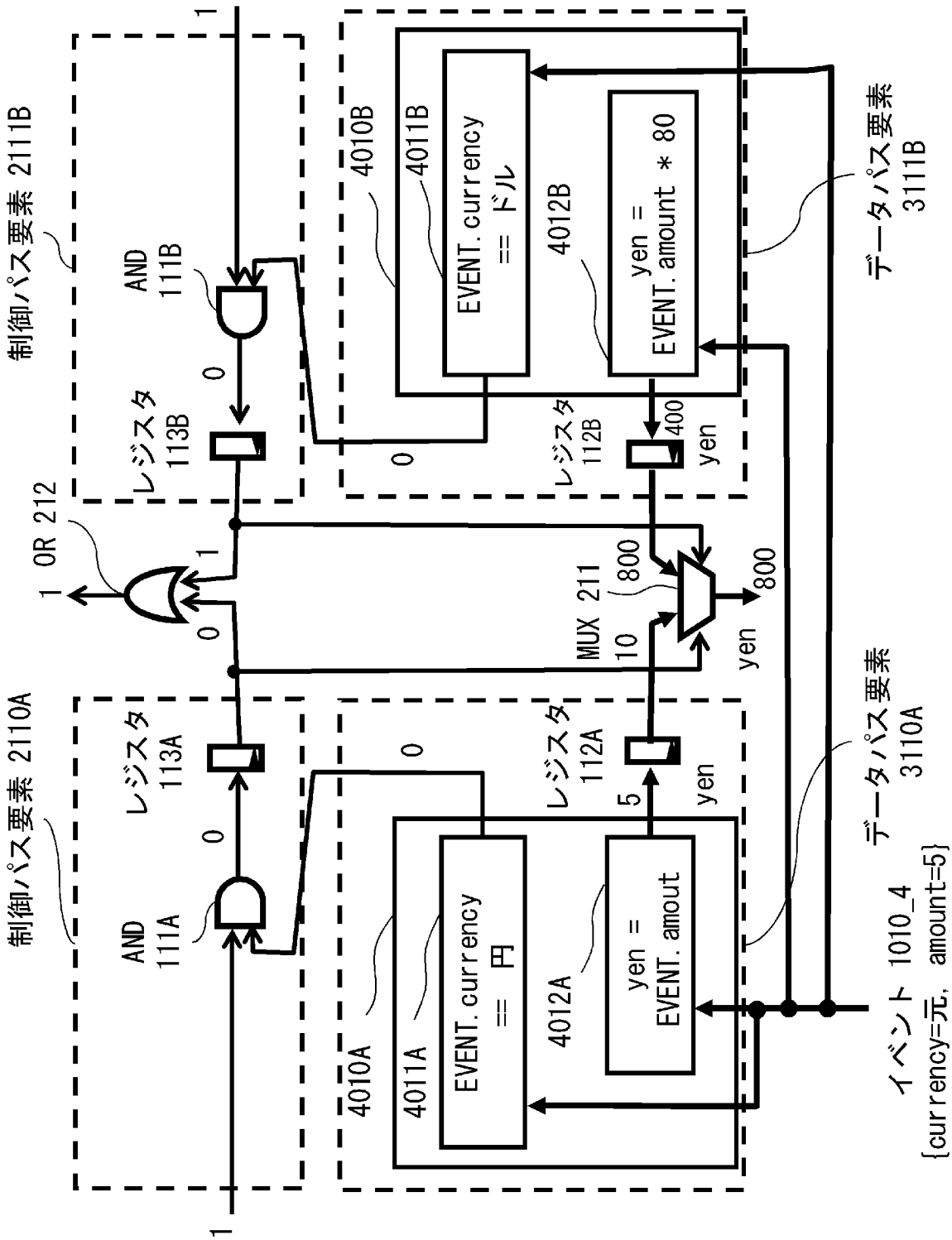
[図17]


制御パス要素 2111B
AND 111B
レジスタ 113B
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
レジスタ 112B
400
yen
OR 212
MUX 211
8000
100
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010_2
{currency=元, amount=5}

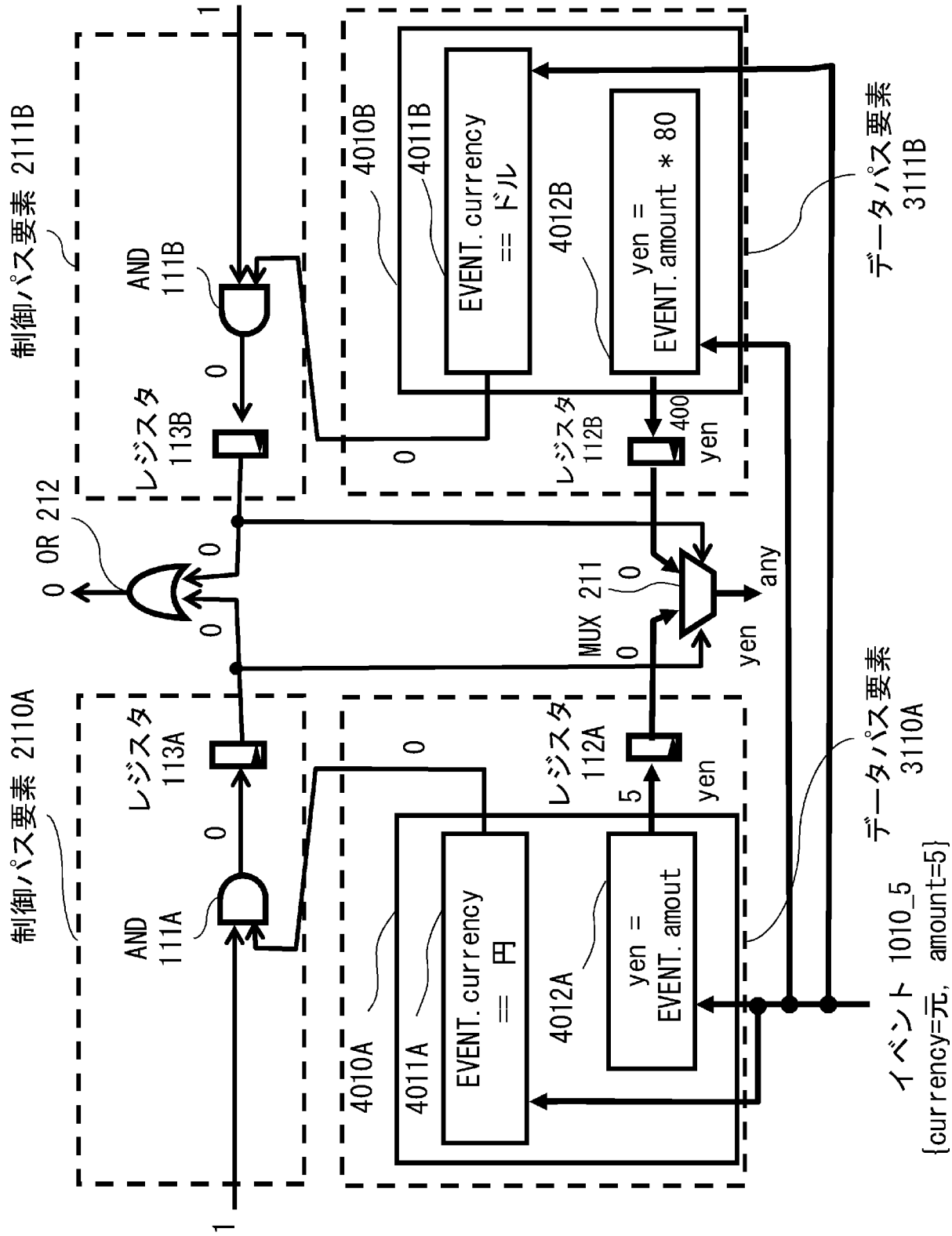
[図18]


制御パス要素 2111B
AND 111B
レジスタ 113B
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
OR 212
MUX 211
any
yen
800
レジスタ 112B
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010_3
{currency=ドル, amount=10}

[図19]


制御パス要素 2111B
AND 111B
レジスタ 113B
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
レジスタ 112B
400
yen
1 OR 212
MUX 211
800
10
800
yen
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
5
yen
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010_4
{currency=元, amount=5}

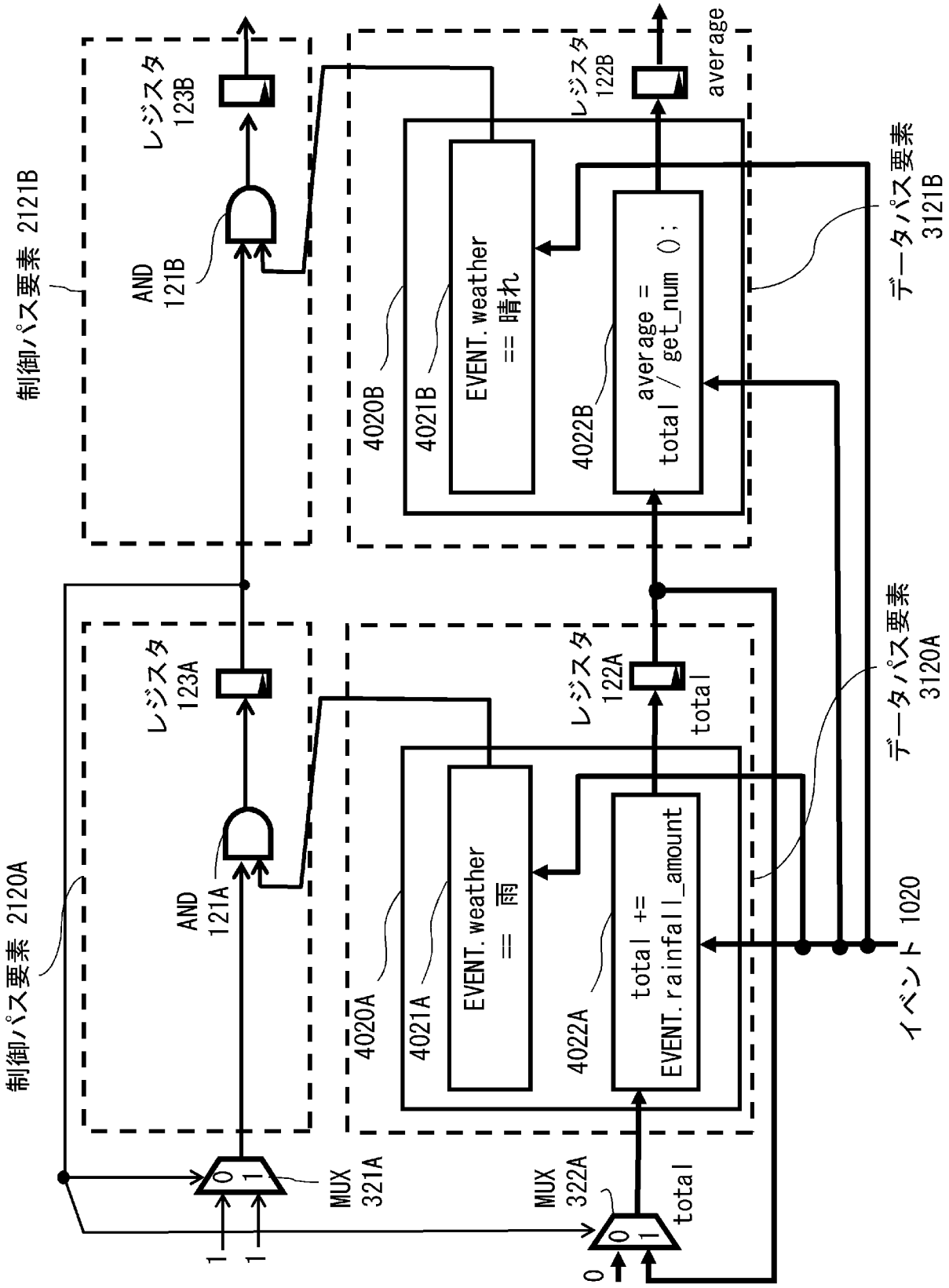
[図20]


制御パス要素 2111B
AND 111B
レジスタ 113B
4010B
4011B
EVENT.currency == ドル
4012B
yen = EVENT.amount * 80
データパス要素 3111B
レジスタ 112B
400
yen
OR 212
MUX 211
any
制御パス要素 2110A
AND 111A
レジスタ 113A
レジスタ 112A
5
4010A
4011A
EVENT.currency == 円
4012A
yen = EVENT.amout
データパス要素 3110A
イベント 1010_5
{currency=元, amount=5}

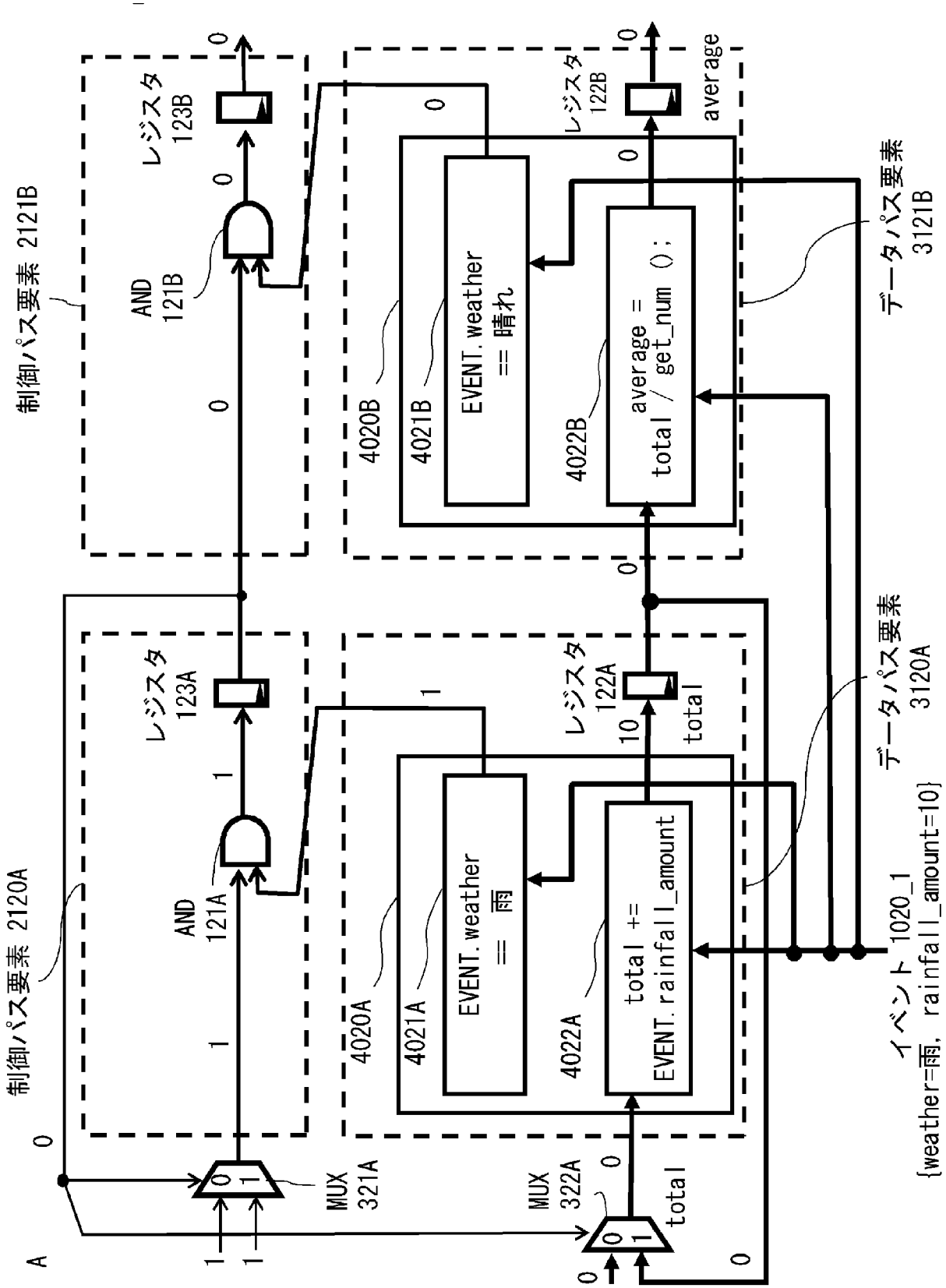
[図21]

```
RULE  = A (total)+ B (total, average)

EVENT = { weather, rainfall_amount}

A (total) = {
    /* 毎秒降水量を積算 */
    total += EVENT.rainfall_amount;

    /* 天気が雨なら1を返す。でなければ0を返す */
    if ( EVENT.weather == 雨 )
        return 1 else return 0;
}


B (total, average) = {
    /* 雨天中の積算降水量の毎秒平均を出す*/
    average = total / get_num_event ();

    /* 天気が晴れなら1を返す。でなければ0を返す */
    if ( EVENT.weather == 晴れ )
        return 1 else return 0;
}
```

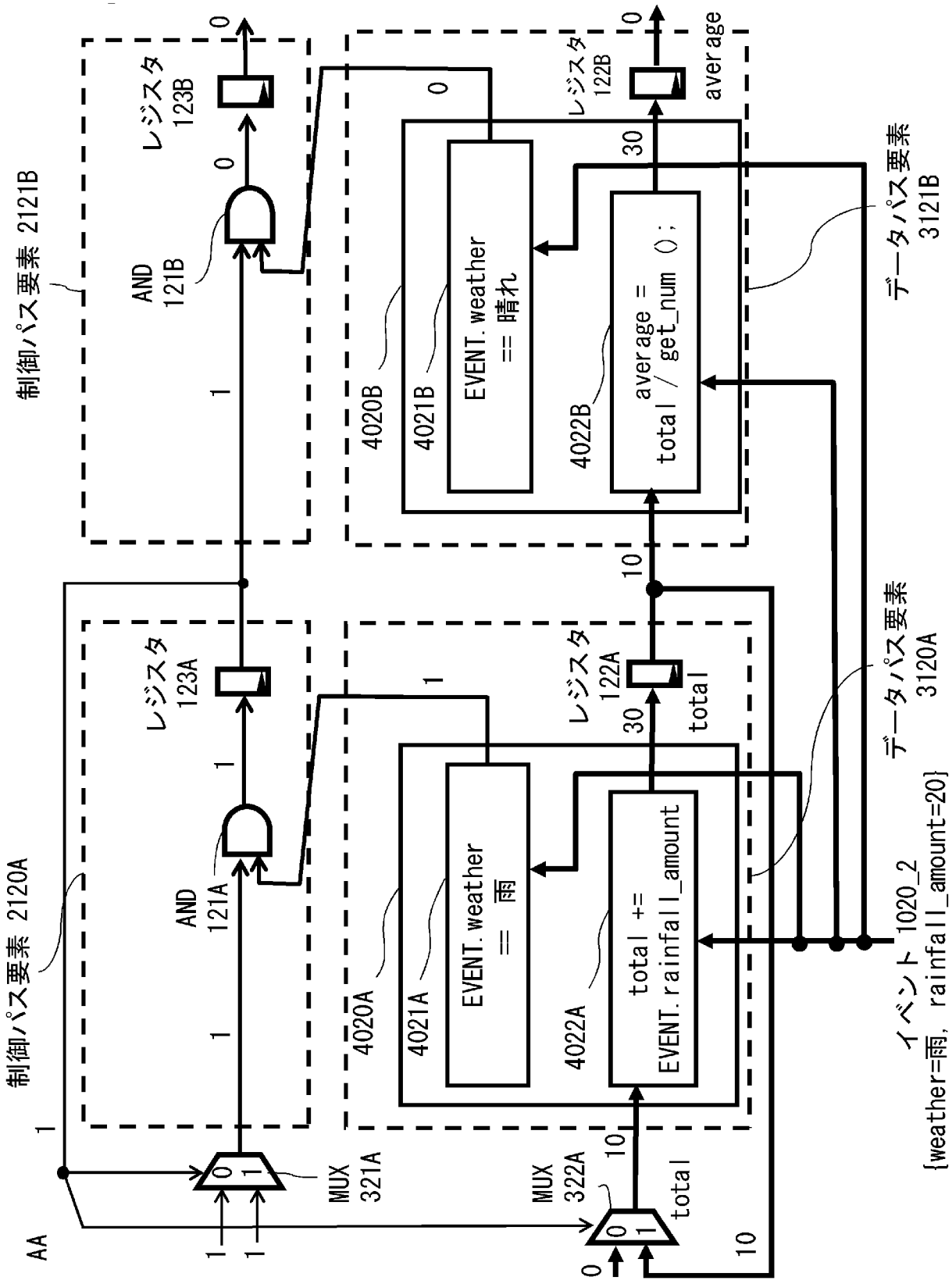
[図22]


制御パス要素 2120A
制御パス要素 2121B
AND 121A
AND 121B
レジスタ 123A
レジスタ 123B
レジスタ 122A
レジスタ 122B
total
average
4020A
4021A
EVENT.weather == 雨
4022A
total += EVENT.rainfall_amount
4020B
4021B
EVENT.weather == 晴れ
4022B
average = total / get_num ();
MUX 321A
MUX 322A
total
データパス要素 3120A
データパス要素 3121B
イベント 1020
0
1

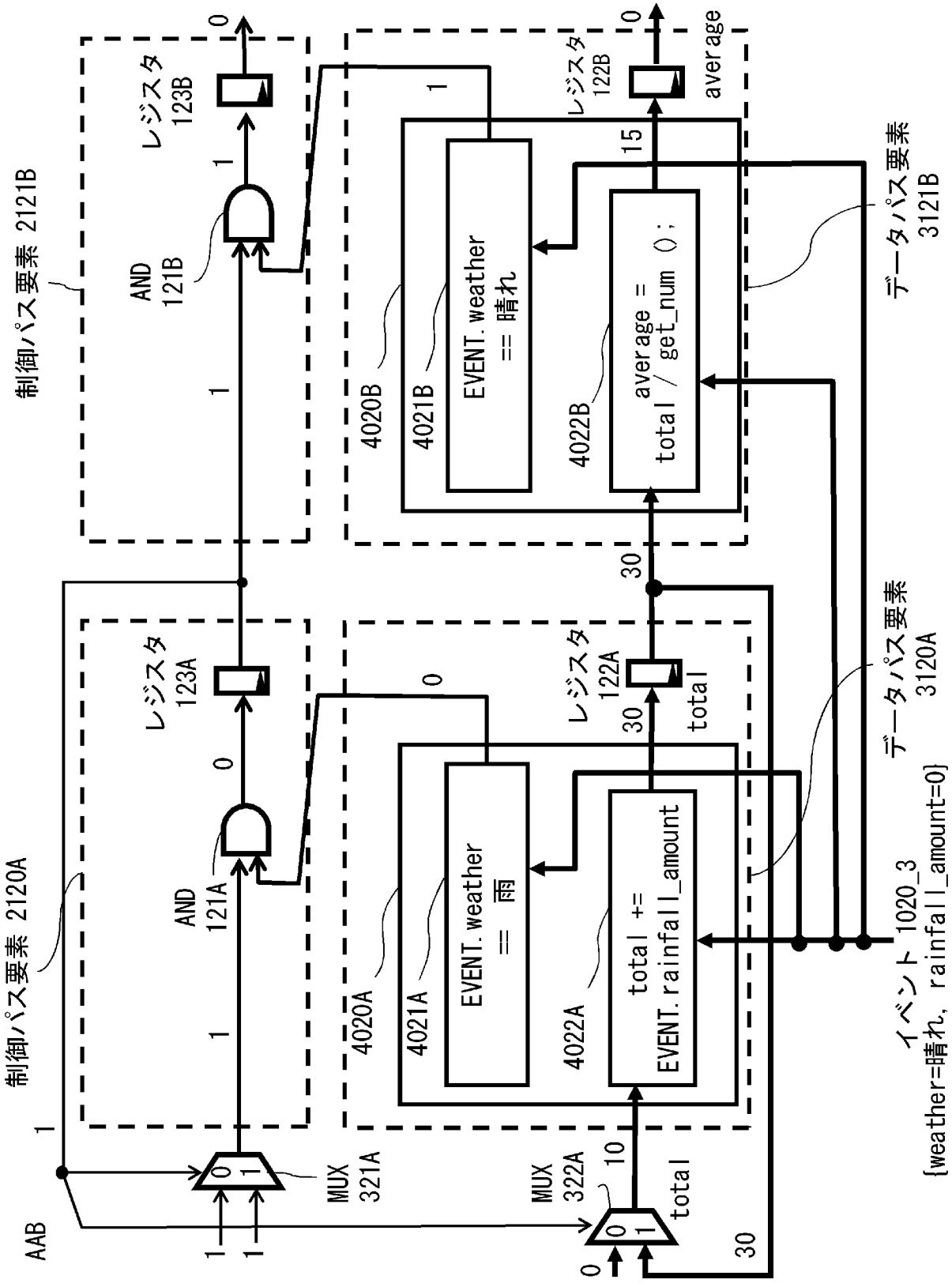
[図23]


制御パス要素 2120A
制御パス要素 2121B
AND 121A
AND 121B
レジスタ 123A
レジスタ 123B
レジスタ 122A
レジスタ 122B
total
average
MUX 321A
MUX 322A
A
4020A
4021A
EVENT.weather == 雨
4022A
total += EVENT.rainfall_amount
4020B
4021B
EVENT.weather == 晴れ
4022B
average = total / get_num ();
データパス要素 3120A
データパス要素 3121B
イベント 1020_1
{weather=雨, rainfall_amount=10}

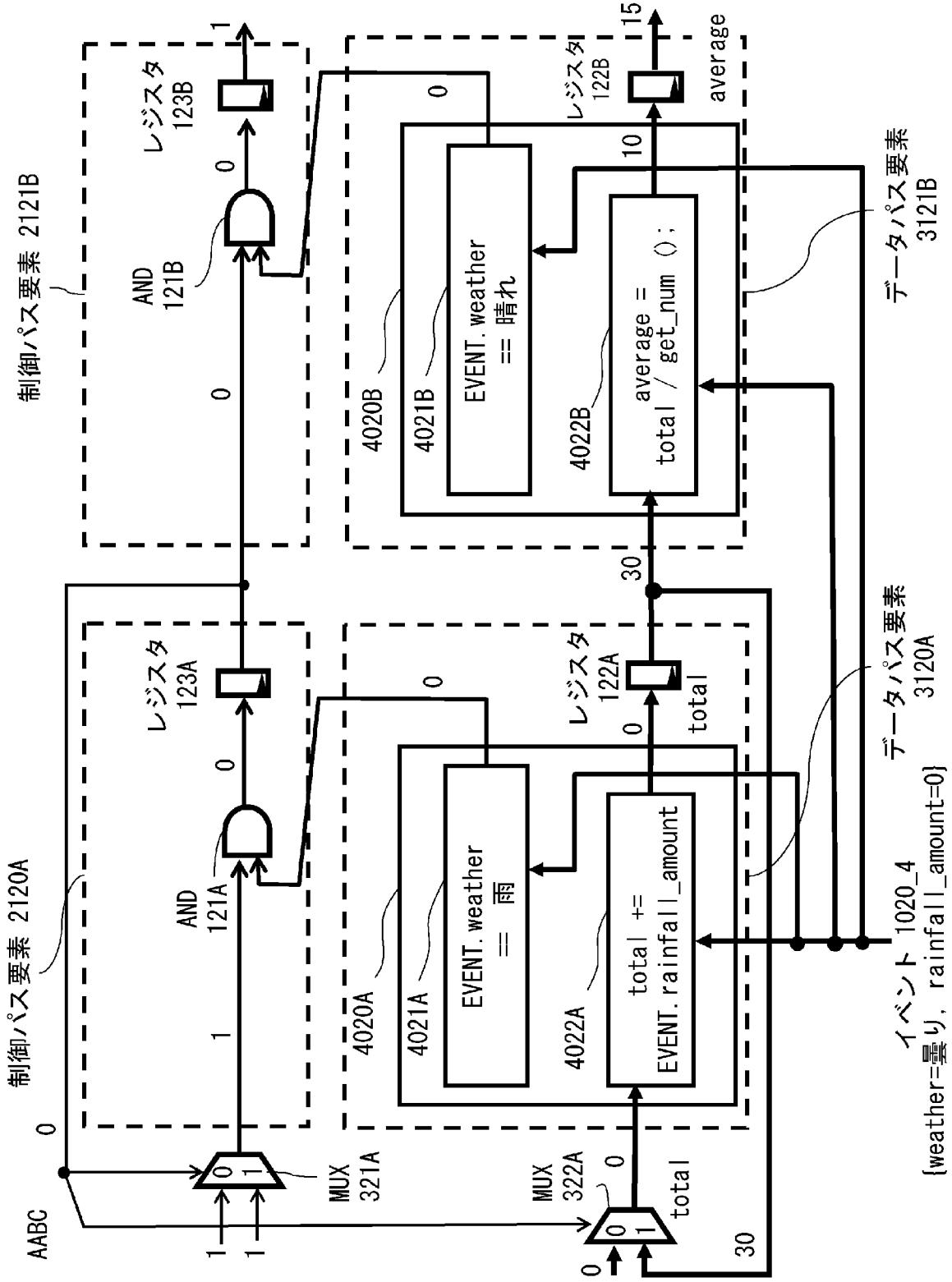
[図24]


制御パス要素 2121B
制御パス要素 2120A
レジスタ 123B
レジスタ 123A
AND 121B
AND 121A
レジスタ 122B
レジスタ 122A
average
total
4020B
4021B
4022B
4020A
4021A
4022A
EVENT.weather == 晴れ
average = total / get_num ();
EVENT.weather == 雨
total += EVENT.rainfall_amount
データパス要素 3121B
データパス要素 3120A
イベント 1020_2
{weather=雨, rainfall_amount=20}
MUX 321A
MUX 322A
AA
total

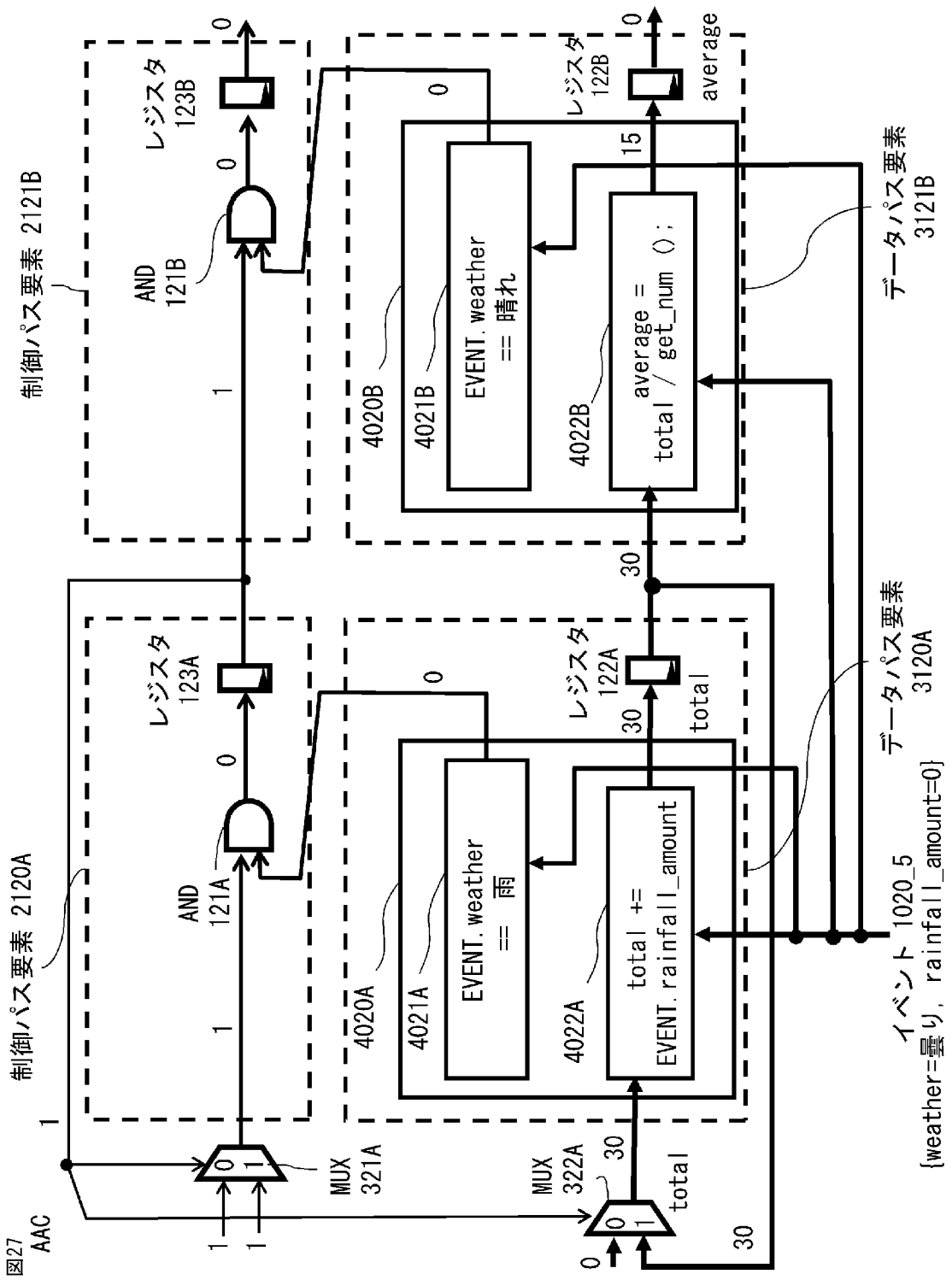
[図25]


制御パス要素 2121B
制御パス要素 2120A
AND 121B
AND 121A
レジスタ 123B
レジスタ 123A
レジスタ 122B
レジスタ 122A
average
total
4020B
4021B
EVENT.weather == 晴れ
4022B
average = total / get_num ();
4020A
4021A
EVENT.weather == 雨
4022A
total += EVENT.rainfall_amount
データパス要素 3121B
データパス要素 3120A
イベント 1020_3
{weather=晴れ, rainfall_amount=0}
MUX 321A
MUX 322A
AAB
total
0
1
10
15
30

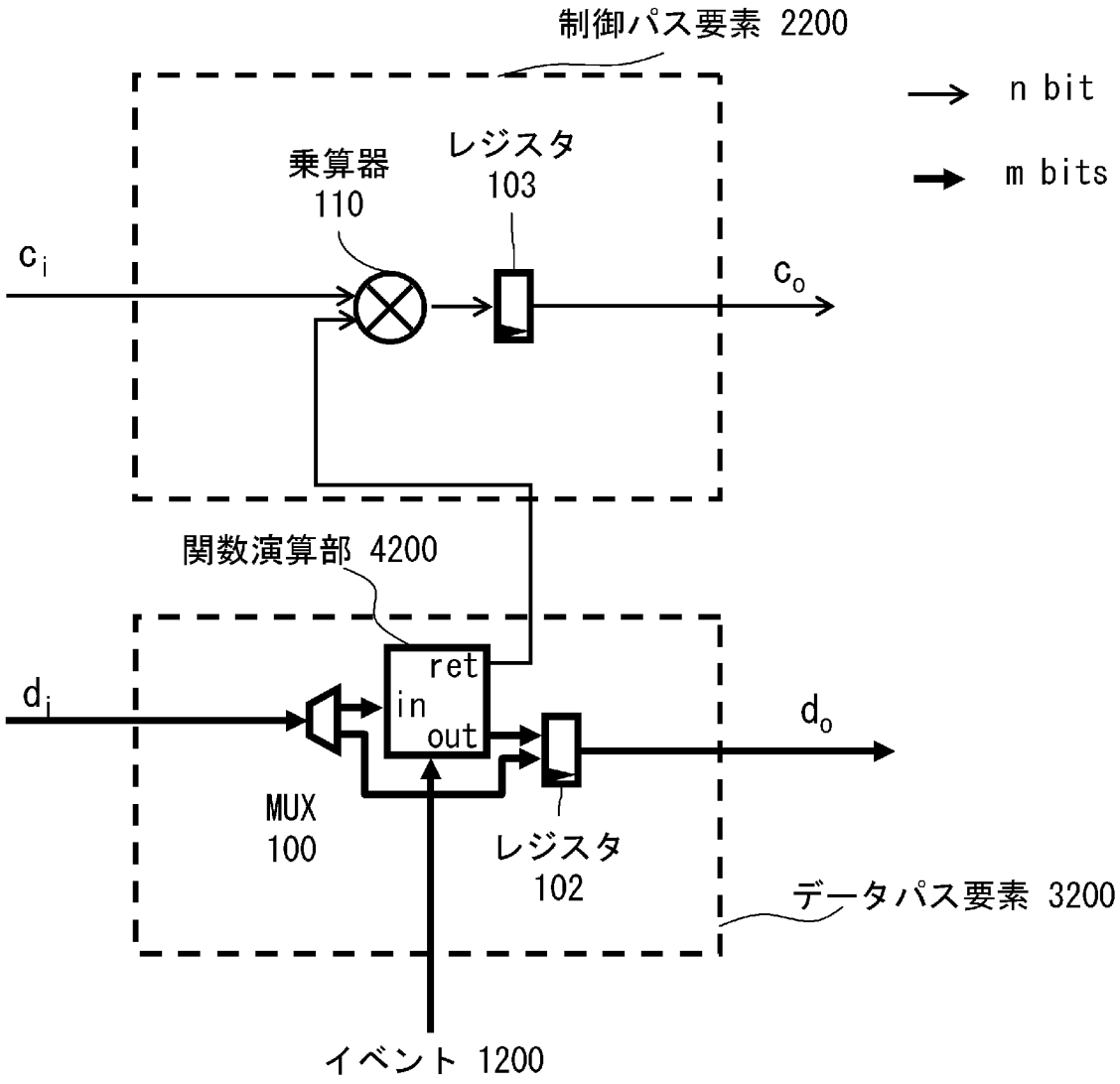
[図26]


制御パス要素 2121B
制御パス要素 2120A
AABC
AND 121B
AND 121A
レジスタ 123B
レジスタ 123A
レジスタ 122B
レジスタ 122A
average
total
MUX 321A
MUX 322A
4020B
4021B
4022B
4020A
4021A
4022A
EVENT.weather == 晴れ
EVENT.weather == 雨
average = total / get_num ();
total += EVENT.rainfall_amount
データパス要素 3121B
データパス要素 3120A
イベント 1020_4
{weather=曇り, rainfall_amount=0}

[図27]

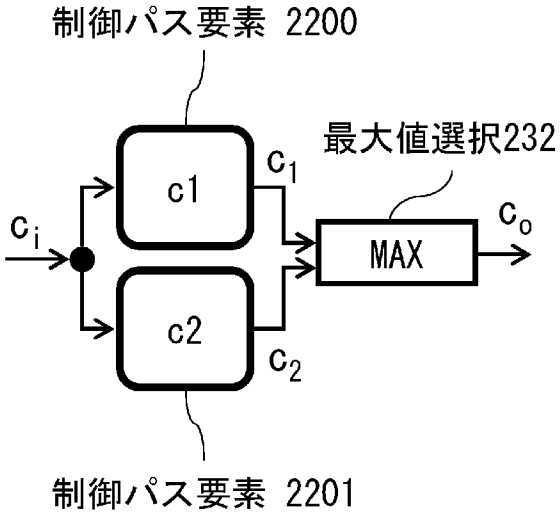

制御パス要素 2120A
制御パス要素 2121B
AND 121A
AND 121B
レジスタ 123A
レジスタ 123B
レジスタ 122A
レジスタ 122B
MUX 321A
MUX 322A
4020A
4021A
EVENT.weather == 雨
4022A
total += EVENT.rainfall_amount
4020B
4021B
EVENT.weather == 晴れ
4022B
average = total / get_num ();
total
average
データパス要素 3120A
データパス要素 3121B
イベント 1020_5
{weather=曇り, rainfall_amount=0}
AAC
図27

[図28]


制御パス要素 2200
n bit
m bits
乗算器
110
レジスタ
103
c_i
c_o
関数演算部 4200
ret
in
out
d_i
d_o
MUX
100
レジスタ
102
データパス要素 3200
イベント 1200

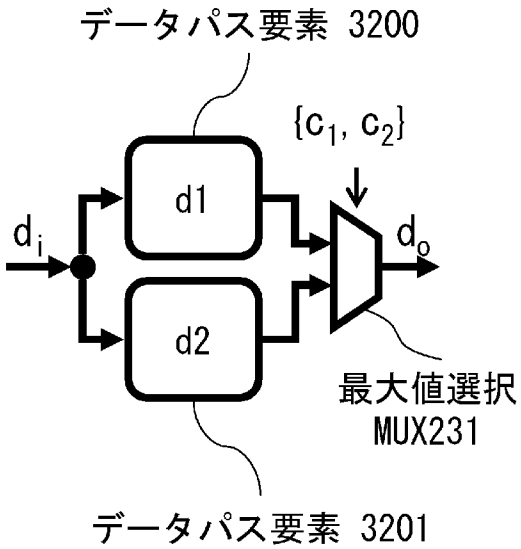
[図29A]

制御パスの和: c1 | c2



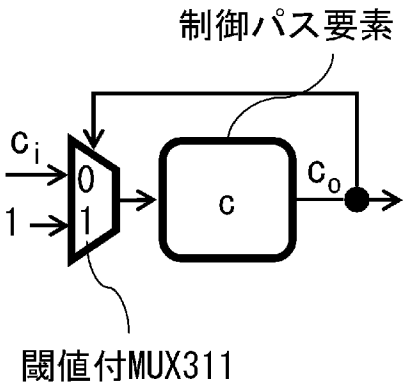
[図29B]

データパスの和: d1 | d2



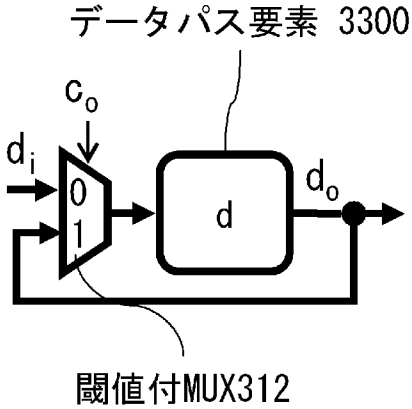
[図30A]

制御パスの繰り返し：c+

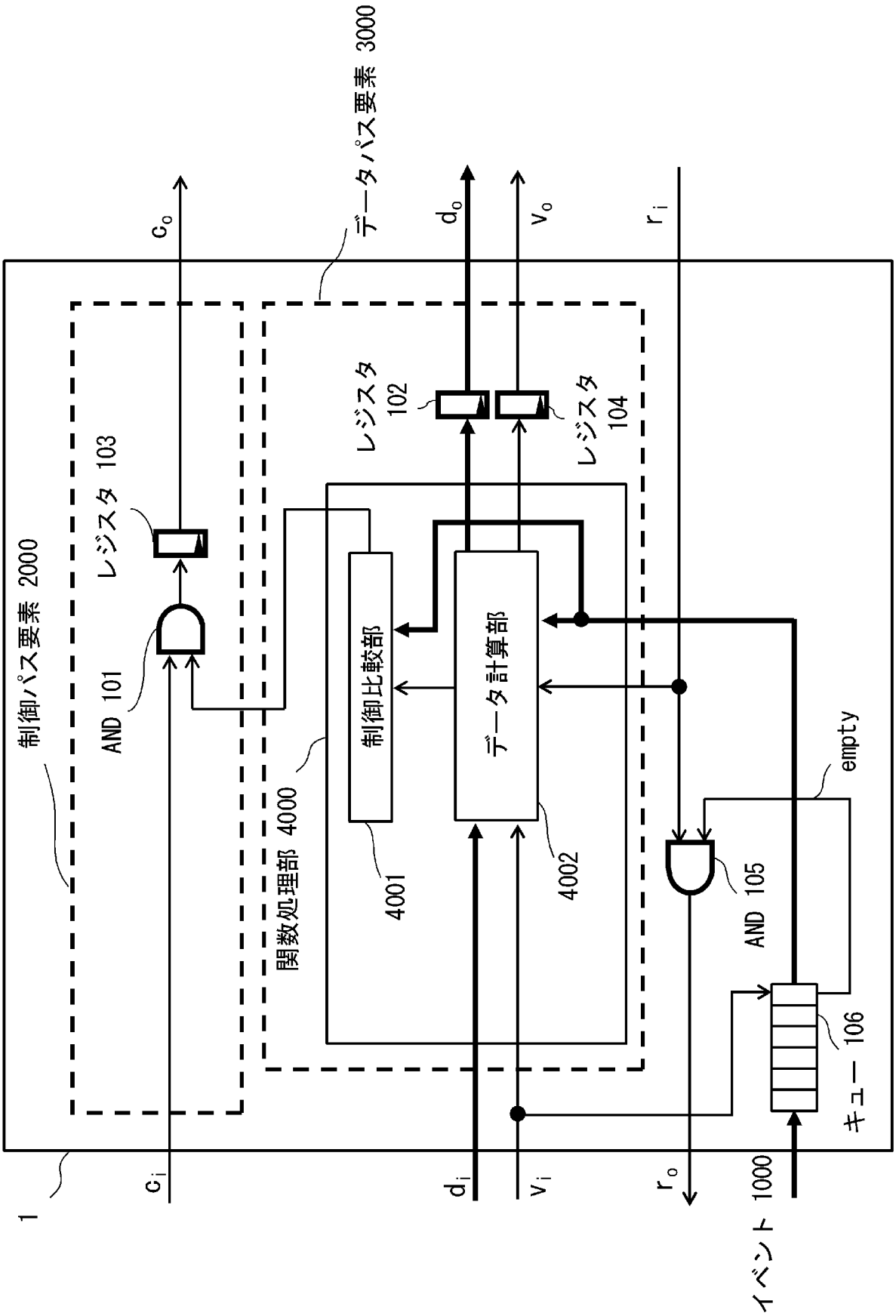


[図30B]

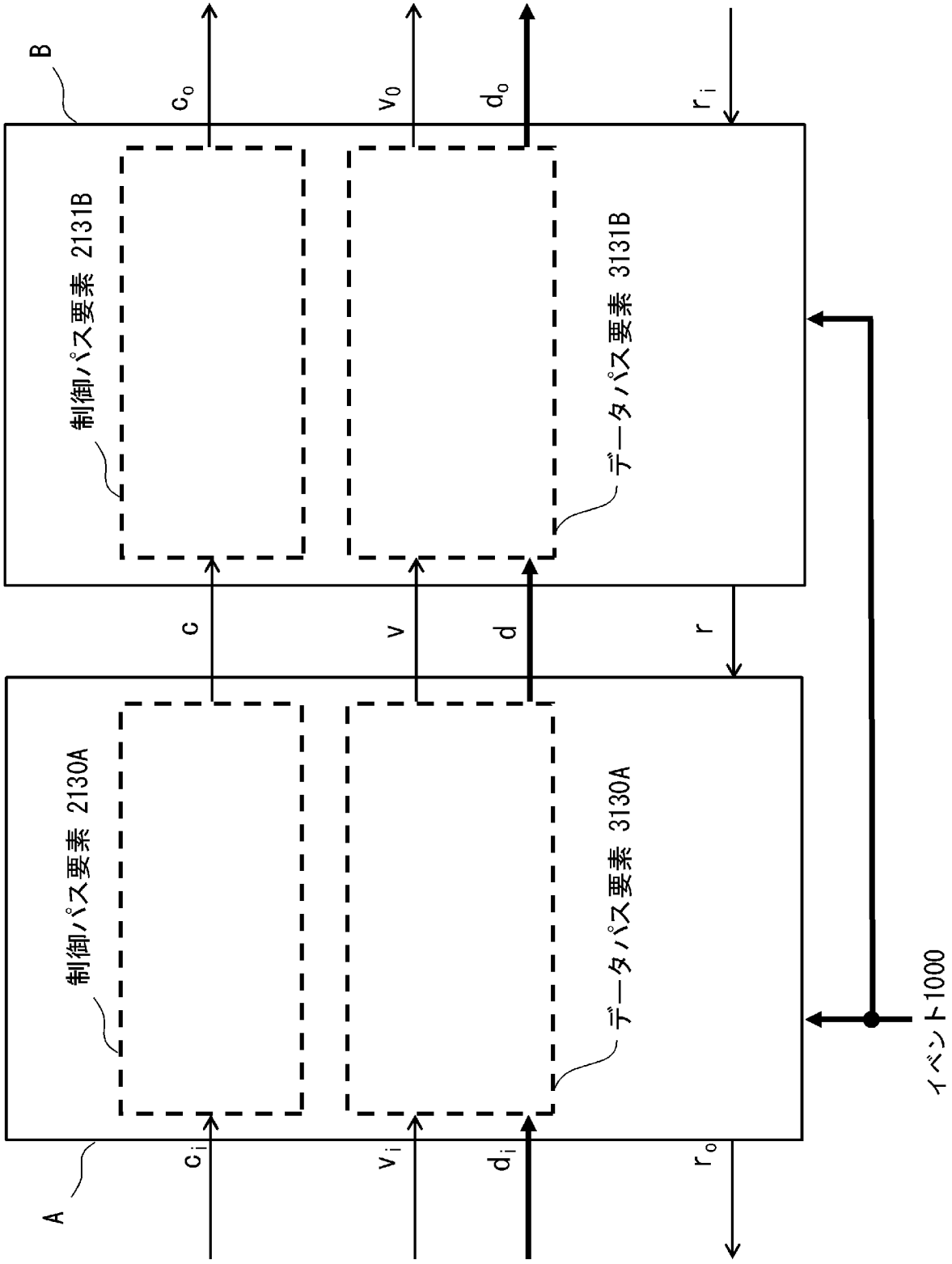
データパスの繰り返し：d+



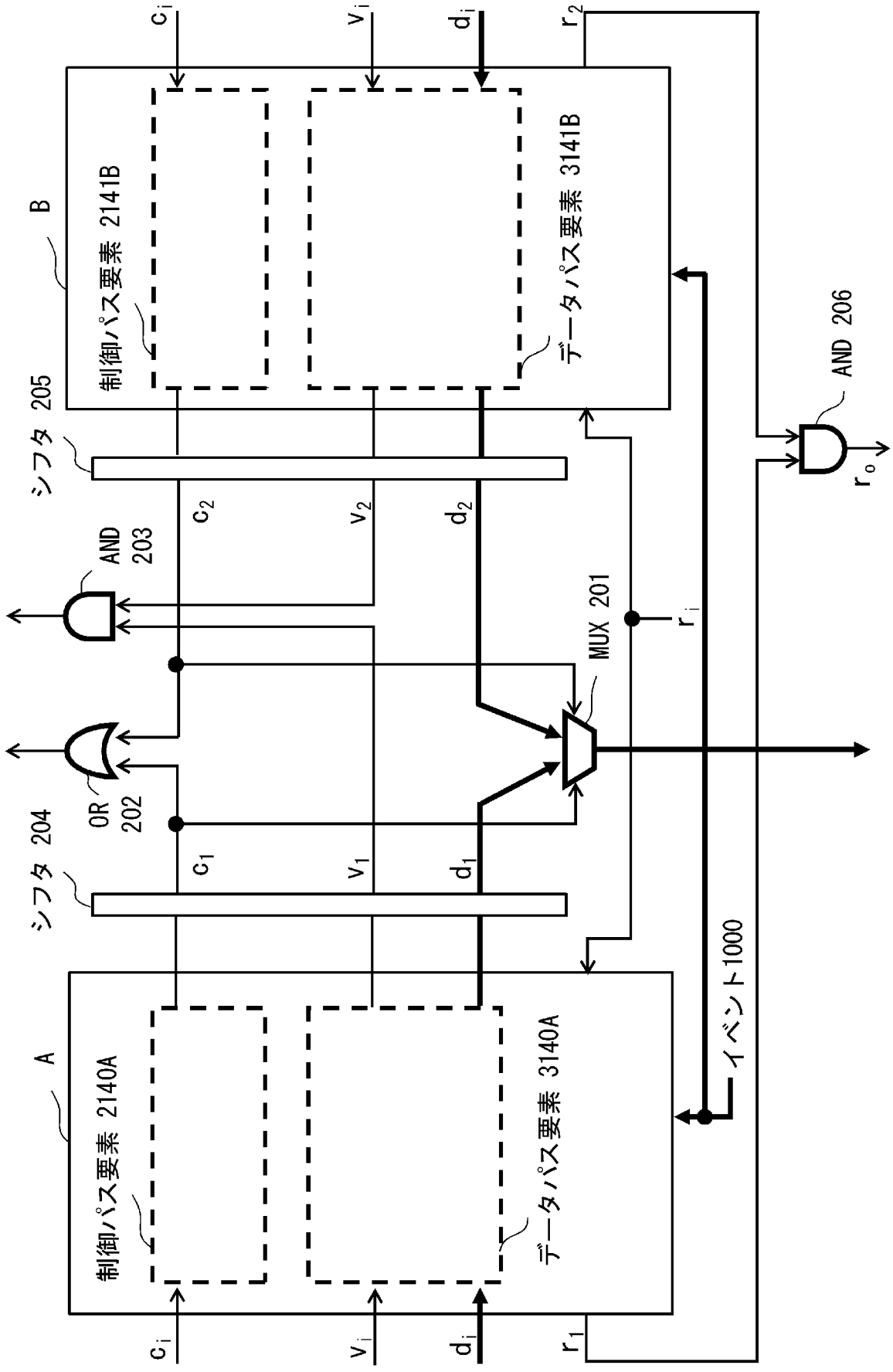
[図31]


データパス要素 3000
制御パス要素 2000
関数処理部 4000
レジスタ 103
AND 101
レジスタ 102
レジスタ 104
制御比較部
データ計算部
4001
4002
AND 105
empty
キュー 106
イベント 1000
1
c_o
c_i
d_o
d_i
v_o
v_i
r_i
r_o

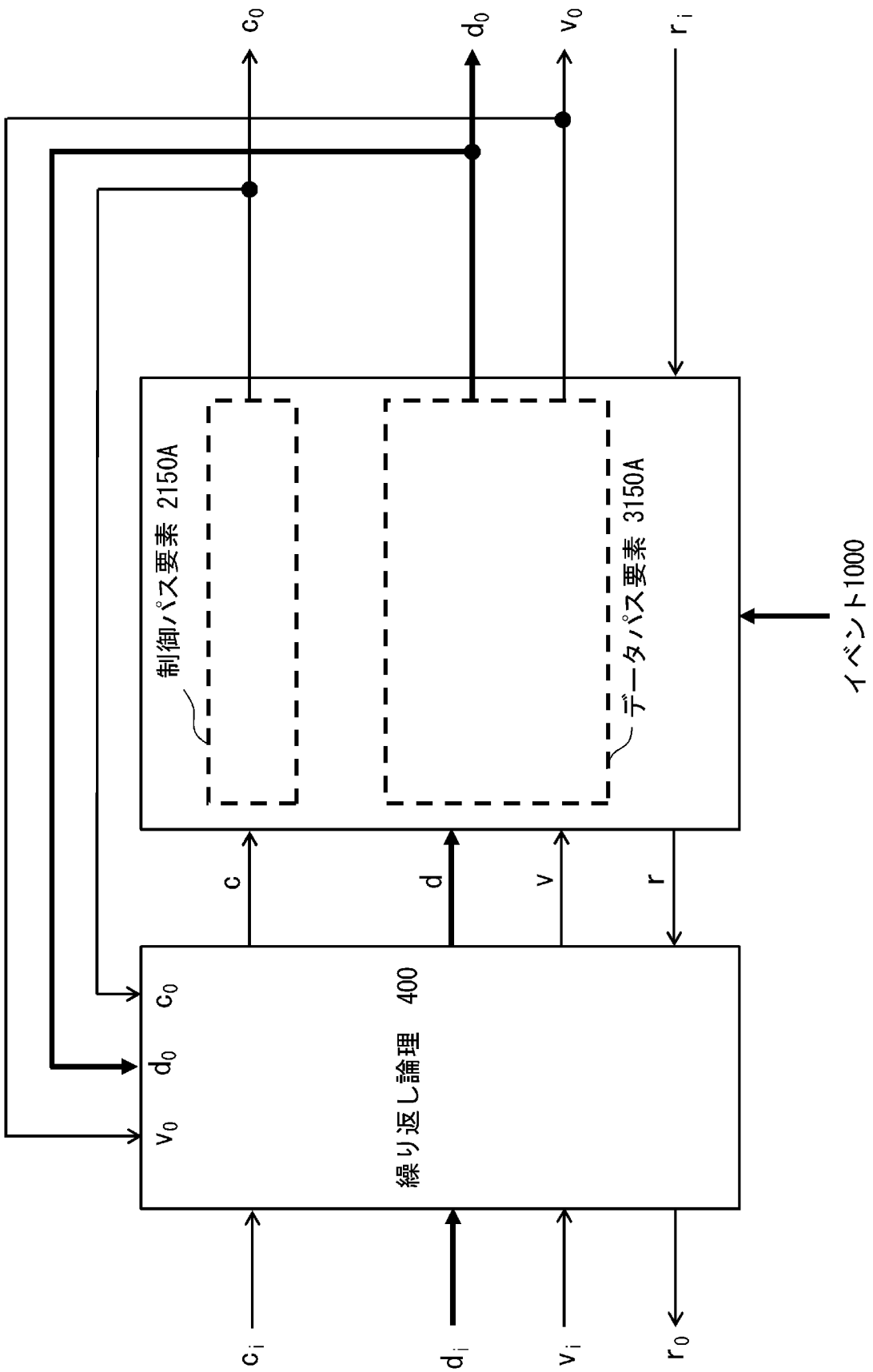
[図32]


B
c_o
v_0
d_o
r_i
制御パス要素 2131B
データパス要素 3131B
c
v
d
r
制御パス要素 2130A
データパス要素 3130A
イベント1000
A
c_i
v_i
d_i
r_o

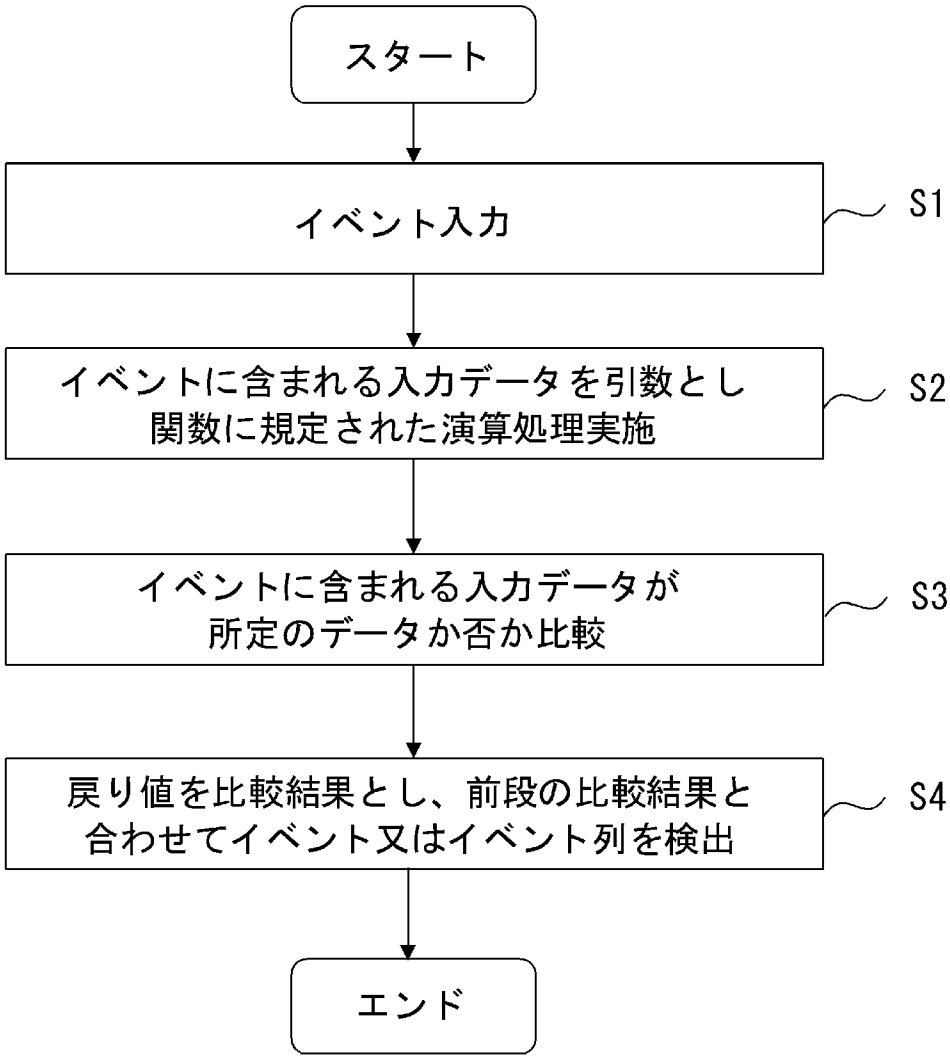
[図33]


c i
v i
d i
r 2
B
制御パス要素 2141B
データパス要素 3141B
AND 206
r o
シフタ 205
c 2
v 2
d 2
AND 203
MUX 201
r i
OR 202
シフタ 204
c 1
v 1
d 1
A
制御パス要素 2140A
データパス要素 3140A
イベント1000
c i
v i
d i
r 1

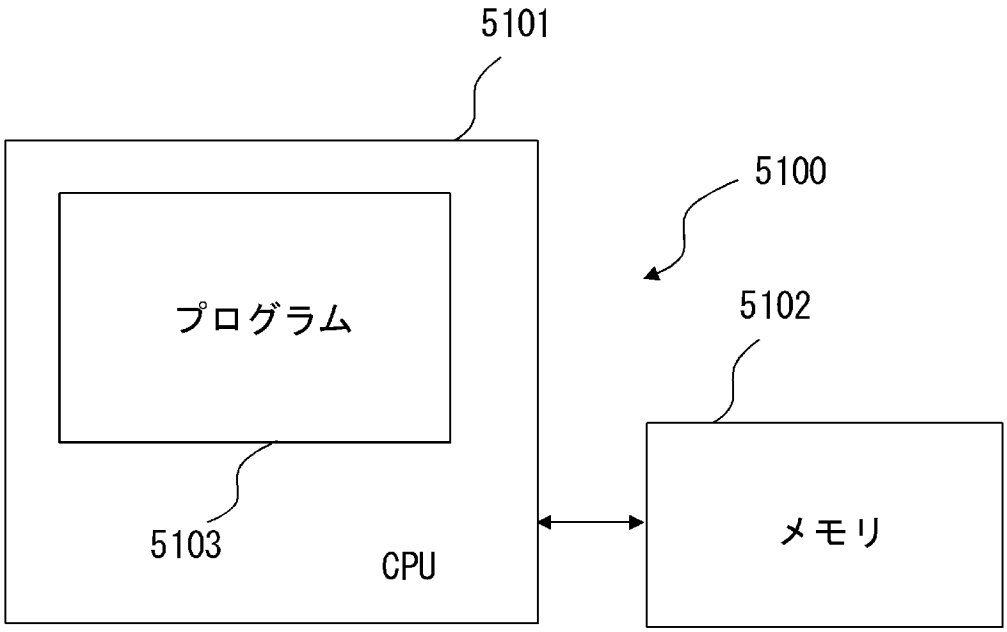
[図34]


c_0
d_0
v_0
r_i
制御パス要素 2150A
データパス要素 3150A
イベント1000
c
d
v
r
c_0
d_0
v_0
繰り返し論理 400
c_i
d_i
v_i
r_0

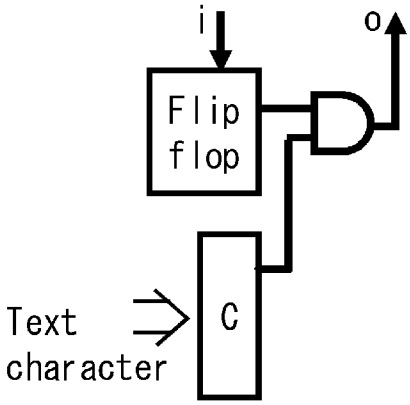
[図35]


スタート
イベント入力
S1
イベントに含まれる入力データを引数とし
関数に規定された演算処理実施
S2
イベントに含まれる入力データが
所定のデータか否か比較
S3
戻り値を比較結果とし、前段の比較結果と
合わせてイベント又はイベント列を検出
S4
エンド

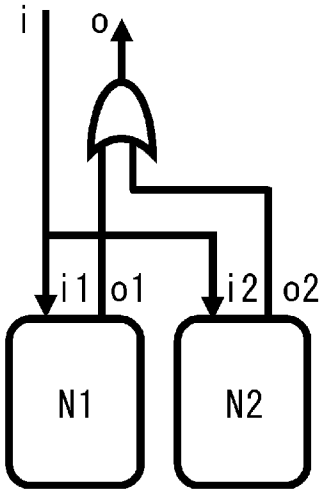
[図36]


5101
5100
プログラム
5102
5103
CPU
メモリ

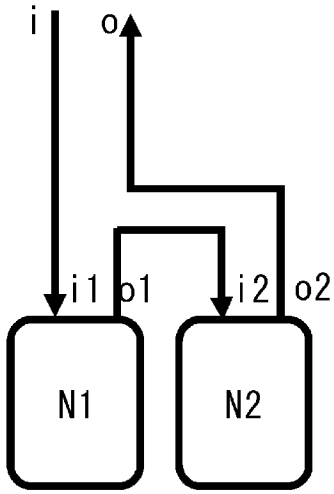
[図37A]


i
o
Flip
flop
C
Text
character

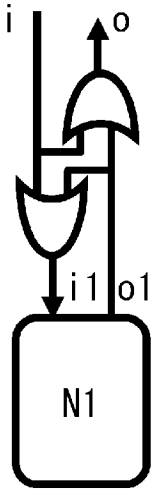
[図37B]



[図37C]



[図37D]



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2011/005905

A. CLASSIFICATION OF SUBJECT MATTER
G06F7/00(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)
G06F7/00

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	1996-2011
Kokai Jitsuyo Shinan Koho	1971-2011	Toroku Jitsuyo Shinan Koho	1994-2011

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	Sidhu, R. et al., "Fast Regular Expression Matching using FPGAs", The 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'01), 2001.04, P.227-238	1-35
A	Yosuke KAWANAKA et al., "A Pattern Independent Regular Expression String Matching Machine and Its FPGA Implementation", The Transactions of the Institute of Electronics, Information and Communication Engineers, 01 December 2009 (01.12.2009), D vol.J92-D, no.12, pages 2159 to 2167	1-35

☒ 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	Date of mailing of the international search report
02 December, 2011 (02.12.11)	13 December, 2011 (13.12.11)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2011/005905

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2010/107114 A1 (NEC Corp.), 23 September 2010 (23.09.2010), entire text; all drawings (Family: none)	1-35

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１１／００５９０５

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G06F7/00(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.　G06F7/00

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	１９７１－２０１１年
日本国実用新案登録公報	１９９６－２０１１年
日本国登録実用新案公報	１９９４－２０１１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	Sidhu, R. et al., “Fast Regular Expression Matching using FPGAs”, The 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'01), 2001.04, P.227-238	1-35
A	川中洋祐、外, “パターン非依存型正規表現ストリングマッチングマシンとそのＦＰＧＡ実装”, 電子情報通信学会論文誌, 2009.12.01, D Vol.J92-D, No.12, P.2159-2167	1-35
A	WO 2010/107114 A1 (日本電気株式会社) 2010.09.23, 全文、全図（ファミリーなし）	1-35

☐　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	国際調査報告の発送日
０２．１２．２０１１	１３．１２．２０１１

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
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）　５Ｅ　９３７６
田中　友章
電話番号　０３－３５８１－１１０１　内線　３５２１