

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7217153号

(P7217153)

(45)発行日 令和5年2月2日(2023.2.2)

(24)登録日 令和5年1月25日(2023.1.25)

(51)国際特許分類

F I

G 0 4 G 5/00 (2013.01)

G 0 4 G 5/00 J

H 0 4 Q 9/00 (2006.01)

H 0 4 Q 9/00 3 1 1 H

H 0 4 M 11/00 (2006.01)

H 0 4 M 11/00 3 0 1

請求項の数 6 (全28頁)

(21)出願番号 特願2019-2774(P2019-2774)
(22)出願日 平成31年1月10日(2019.1.10)
(65)公開番号 特開2020-112412(P2020-112412
A)
(43)公開日 令和2年7月27日(2020.7.27)
審査請求日 令和3年10月7日(2021.10.7)

(73)特許権者 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1
番1号
(74)代理人 100121083
弁理士 青木 宏義
(74)代理人 100138391
弁理士 天田 昌行
(74)代理人 100074099
弁理士 大菅 義之
(74)代理人 100133570
弁理士 徳 永 民雄
(72)発明者 清水 貴志
神奈川県川崎市中原区上小田中4丁目1
番1号 富士通株式会社内

最終頁に続く

(54)【発明の名称】 情報処理装置、時刻同期方法、及び時刻同期プログラム

(57)【特許請求の範囲】

【請求項1】

第1システム時刻を記憶する記憶部と、

前記記憶部に前記第1システム時刻が書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第1情報と、前記第1情報の取得時刻を示す第1時刻情報とを受信し、前記第1情報と前記第1時刻情報とを受信した後に、前記情報取得装置から、第2情報と前記第2情報の取得時刻を示す第2時刻情報とを受信する受信部と、
前記第2時刻情報と前記第1時刻情報との差分と、クロック信号と、前記クロック信号の周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算する第1周波数計算部と、

前記受信部が前記第1情報と前記第1時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記受信部が前記第1情報と前記第1時刻情報とを受信した受信時刻として用い、前記第1システム時刻に基づいて、前記受信時刻を第2システム時刻に変換する受信時刻変換部と、

前記第2システム時刻と前記第1時刻情報とに基づいて、前記第2時刻情報を第3システム時刻に変換する時刻情報変換部と、

前記第2システム時刻を前記第1情報に付加し、前記第3システム時刻を前記第2情報に付加する付加部と、

を備え、

前記時刻情報変換部は、

前記第 2 時刻情報と前記第 1 時刻情報との差分を生成する減算部と、
前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用い
て、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算する第 1 差分計算部と、
前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算す
ることで、前記第 3 システム時刻を求める第 1 加算部と、
を含むことを特徴とする情報処理装置。

【請求項 2】

前記受信時刻変換部は、
前記受信時刻と前記クロック信号の周波数とを用いて、前記第 2 システム時刻と前記第 1 システム時刻との差分を計算する第 2 差分計算部と、
前記第 2 システム時刻と前記第 1 システム時刻との差分を前記第 1 システム時刻に加算することで、前記第 2 システム時刻を求める第 2 加算部と、
を含むことを特徴とする請求項 1 記載の情報処理装置。

10

【請求項 3】

前記第 1 システム時刻と前記クロック信号とを用いて、前記クロック信号の周波数を計算する第 2 周波数計算部をさらに備えることを特徴とする請求項 2 記載の情報処理装置。

【請求項 4】

前記受信部は、前記記憶部に前記第 1 システム時刻が書き込まれた後に、複数の情報取得装置のうちいずれかの情報取得装置から、前記いずれかの情報取得装置が取得した第 1 情報と第 1 時刻情報とを受信し、前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記いずれかの情報取得装置から、第 2 情報と第 2 時刻情報とを受信し、
前記付加部は、前記第 2 システム時刻を前記いずれかの情報取得装置から受信した第 1 情報に付加し、前記第 3 システム時刻を前記いずれかの情報取得装置から受信した第 2 情報に付加することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の情報処理装置。

20

【請求項 5】

情報処理装置によって実行される時刻同期方法であって、
前記情報処理装置が、
第 1 システム時刻が記憶部に書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第 1 情報と、前記第 1 情報の取得時刻を示す第 1 時刻情報とを受信し、
前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記情報取得装置から、第 2 情報と前記第 2 情報の取得時刻を示す第 2 時刻情報とを受信し、
前記第 2 時刻情報と前記第 1 時刻情報との差分と、クロック信号と、前記クロック信号の周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算し、
前記第 1 情報と前記第 1 時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記第 1 情報と前記第 1 時刻情報とを受信した受信時刻として用い、前記第 1 システム時刻に基づいて、前記受信時刻を第 2 システム時刻に変換し、
前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用いて、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算し、
前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算することで、第 3 システム時刻を求め、
前記第 2 システム時刻を前記第 1 情報に付加し、
前記第 3 システム時刻を前記第 2 情報に付加する、
ことを特徴とする時刻同期方法。

30

40

【請求項 6】

第 1 システム時刻が記憶部に書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第 1 情報と、前記第 1 情報の取得時刻を示す第 1 時刻情報とを受信し、
前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記情報取得装置から、第 2 情報と前記第 2 情報の取得時刻を示す第 2 時刻情報とを受信し、
前記第 2 時刻情報と前記第 1 時刻情報との差分と、クロック信号と、前記クロック信号の

50

周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算し、

前記第 1 情報と前記第 1 時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記第 1 情報と前記第 1 時刻情報とを受信した受信時刻として用い、前記第 1 システム時刻に基づいて、前記受信時刻を第 2 システム時刻に変換し、

前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用いて、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算し、

前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算することで、第 3 システム時刻を求め、

前記第 2 システム時刻を前記第 1 情報に付加し、

前記第 3 システム時刻を前記第 2 情報に付加する、

処理をコンピュータに実行させるための時刻同期プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置、時刻同期方法、及び時刻同期プログラムに関する。

【背景技術】

【0002】

図 1 は、複数のセンサから情報を収集する、従来の情報処理システムの構成例を示している。図 1 の情報処理システムは、情報処理装置 101、センサ 102 - 1、及びセンサ 102 - 2 を含む。情報処理装置 101 は、FPGA (Field-Programmable Gate Array) 111 を含む。

【0003】

センサ 102 - 1 及びセンサ 102 - 2 は、個別のタイミング信号に同期して動作する。センサ 102 - 1 は、データ 121 - 1 ~ データ 121 - 4 を取得し、取得したデータをバッファリングしながら情報処理装置 101 へ送信する。同様に、センサ 102 - 2 は、データ 122 - 1 ~ データ 122 - 4 を取得し、取得したデータをバッファリングしながら情報処理装置 101 へ送信する。

【0004】

FPGA 111 は、センサ 102 - 1 からデータ 121 - 1 ~ データ 121 - 4 を受信し、センサ 102 - 2 からデータ 122 - 1 ~ データ 122 - 4 を受信する。そして、FPGA 111 は、受信したデータに対する情報処理を行う。

【0005】

センサに関連して、複数のセンサによる計測結果同士を時間的に対応付けるセンサ情報処理装置、及びセンサデータに関する誤差の少ないタイムスタンプが得られるセンシングシステムが知られている (例えば、特許文献 1 及び特許文献 2 を参照)。

【先行技術文献】

【特許文献】

【0006】

【文献】特開 2015 - 19223 号公報

国際公開第 2018 / 151202 号パンフレット

【発明の概要】

【発明が解決しようとする課題】

【0007】

情報処理装置において複数のセンサから収集した情報を処理する際に、各情報に時刻を示すタイムスタンプを付加する場合がある。しかし、各センサが個別のタイミング信号に同期して動作している場合、複数のセンサのタイミング信号に基づく時刻は互いに同期しておらず、情報処理装置のシステム時刻とも同期していない。

【0008】

なお、時刻同期の問題は、複数のセンサから情報を収集する場合に限らず、単一のセン

10

20

30

40

50

サから情報を収集する場合においても生ずるものである。

【 0 0 0 9 】

1つの側面において、本発明は、情報取得装置が情報を取得した時刻を、取得された情報を処理する情報処理装置のシステム時刻と同期させることを目的とする。

【課題を解決するための手段】

【 0 0 1 0 】

1つの案では、情報処理装置は、記憶部、受信部、受信時刻変換部、時刻情報変換部、及び付加部を含む。

【 0 0 1 1 】

記憶部は、第1システム時刻を記憶する。受信部は、記憶部に第1システム時刻が書き込まれた後に、情報取得装置から、情報取得装置が取得した第1情報と、第1情報の取得時刻を示す第1時刻情報とを受信する。そして、受信部は、第1情報と第1時刻情報とを受信した後に、情報取得装置から、第2情報と第2情報の取得時刻を示す第2時刻情報とを受信する。

10

【 0 0 1 2 】

受信時刻変換部は、第1システム時刻に基づいて、受信部が第1情報と第1時刻情報とを受信した受信時刻を、第2システム時刻に変換し、時刻情報変換部は、第2システム時刻と第1時刻情報とに基づいて、第2時刻情報を第3システム時刻に変換する。付加部は、第2システム時刻を第1情報に付加し、第3システム時刻を第2情報に付加する。

【発明の効果】

20

【 0 0 1 3 】

実施形態によれば、情報取得装置が情報を取得した時刻を、取得された情報を処理する情報処理装置のシステム時刻と同期させることができる。

【図面の簡単な説明】

【 0 0 1 4 】

【図1】従来の情報処理システムの構成図である。

【図2】FPGAの第1の比較例を示す図である。

【図3】FPGAの第2の比較例を示す図である。

【図4】情報処理装置の機能的構成図である。

【図5】時刻同期処理のフローチャートである。

30

【図6】実施形態の情報処理システムの構成図である。

【図7】センサモジュールの構成図である。

【図8】ビデオモジュールの構成図である。

【図9】変換部の構成図である。

【図10A】時刻同期処理の具体例を示すフローチャート(その1)である。

【図10B】時刻同期処理の具体例を示すフローチャート(その2)である。

【図11】時刻同期処理を示す図である。

【図12】第2周波数計算部の構成図である。

【図13】第1周波数計算部の構成図である。

【図14】クロック信号の周波数を示す図である。

40

【図15】タイミング信号の周波数を示す図である。

【図16】OS時刻を示す図である。

【図17】パケット消失時の時刻同期処理を示す図である。

【図18】ソフトウェア制御による時刻同期部の構成図である。

【発明を実施するための形態】

【 0 0 1 5 】

以下、図面を参照しながら、実施形態を詳細に説明する。

図2は、データにタイムスタンプを付加するFPGAの第1の比較例を示している。図2のFPGA201は、割り込みコントローラ211、MMIO(Memory-mapped Input Output)レジスタ212、及び検出部213を含む。

50

【 0 0 1 6 】

検出部 2 1 3 は、センサから入力データを受信すると、割り込みコントローラ 2 1 1 へ検出信号を出力する。割り込みコントローラ 2 1 1 は、検出信号を受信すると、オペレーティングシステム (Operating System , OS) 2 0 2 に対して、割り込み又はポーリングにより時刻同期を要求する。OS 2 0 2 は、割り込みコントローラ 2 1 1 からの要求に基づいて OS 時刻 2 2 1 を取得し、FPGA 2 0 1 の MMIO レジスタ 2 1 2 に書き込む。OS 時刻 2 2 1 は、システム時刻の一例である。

【 0 0 1 7 】

FPGA 2 0 1 は、MMIO レジスタ 2 1 2 に書き込まれた OS 時刻 2 2 1 を、タイムスタンプとして、検出部 2 1 3 から出力される入力データに付加し、タイムスタンプが付加された出力データを出力する。

10

【 0 0 1 8 】

しかしながら、図 2 の構成では、出力データのタイムスタンプは、FPGA 2 0 1 が入力データを受信したときの OS 時刻を示しており、センサが入力データを取得したときの OS 時刻を示していない。

【 0 0 1 9 】

図 3 は、データにタイムスタンプを付加する FPGA の第 2 の比較例を示している。図 3 の FPGA 3 0 1 は、MMIO レジスタ 3 1 1 及び加算部 3 1 2 を含む。

【 0 0 2 0 】

センサは、タイミング信号に同期してカウント値をインクリメントするカウンタを含み、入力データを取得したときのカウンタ値をその入力データに付加して、FPGA 3 0 1 へ送信する。

20

【 0 0 2 1 】

OS 2 0 2 は、定期的に OS 時刻 2 2 1 を取得し、FPGA 3 0 1 の MMIO レジスタ 3 1 1 に書き込む。FPGA 3 0 1 は、MMIO レジスタ 3 1 1 に OS 時刻 2 2 1 が書き込まれると、リセット信号をセンサへ送信し、センサは、リセット信号を受信すると、カウンタをリセットする。

【 0 0 2 2 】

FPGA 3 0 1 は、センサから入力データを受信すると、入力データからカウンタ値を取得する。加算部 3 1 2 は、MMIO レジスタ 3 1 1 に書き込まれた OS 時刻 2 2 1 に、カウンタ値が示す時間を加算することで、タイムスタンプを生成する。そして、FPGA 3 0 1 は、生成されたタイムスタンプを入力データに付加し、タイムスタンプが付加された出力データを出力する。

30

【 0 0 2 3 】

しかしながら、図 3 の構成では、入力データに付加されたカウンタ値に、FPGA 3 0 1 からセンサへリセット信号を送信する通信のオーバーヘッドが含まれている。このため、そのカウンタ値から生成されるタイムスタンプは、必ずしも、センサが入力データを取得したときの OS 時刻を示していない。

【 0 0 2 4 】

図 4 は、実施形態の情報処理装置の機能的構成例を示している。図 4 の情報処理装置 4 0 1 は、記憶部 4 1 1、受信部 4 1 2、受信時刻変換部 4 1 3、時刻情報変換部 4 1 4、及び付加部 4 1 5 を含む。記憶部 4 1 1 は、第 1 システム時刻を記憶する。

40

【 0 0 2 5 】

図 5 は、図 4 の情報処理装置 4 0 1 が行う時刻同期処理の例を示すフローチャートである。まず、受信部 4 1 2 は、記憶部 4 1 1 に第 1 システム時刻が書き込まれた後に、情報取得装置 4 0 2 から、情報取得装置 4 0 2 が取得した第 1 情報と、第 1 情報の取得時刻を示す第 1 時刻情報とを受信する (ステップ 5 0 1)。そして、受信部 4 1 2 は、第 1 情報と第 1 時刻情報とを受信した後に、情報取得装置 4 0 2 から、第 2 情報と第 2 情報の取得時刻を示す第 2 時刻情報とを受信する (ステップ 5 0 2)。

【 0 0 2 6 】

50

受信時刻変換部 4 1 3 は、第 1 システム時刻に基づいて、受信部 4 1 2 が第 1 情報と第 1 時刻情報とを受信した受信時刻を、第 2 システム時刻に変換する（ステップ 5 0 3）。時刻情報変換部 4 1 4 は、第 2 システム時刻と第 1 時刻情報とに基づいて、第 2 時刻情報を第 3 システム時刻に変換する（ステップ 5 0 4）。そして、付加部 4 1 5 は、第 2 システム時刻を第 1 情報に付加し（ステップ 5 0 5）、第 3 システム時刻を第 2 情報に付加する（ステップ 5 0 6）。

【 0 0 2 7 】

図 4 の情報処理装置 4 0 1 によれば、情報取得装置が情報を取得した時刻を、取得された情報を処理する情報処理装置のシステム時刻と同期させることができる。

【 0 0 2 8 】

図 6 は、実施形態の情報処理システムの構成例を示している。図 6 の情報処理システムは、情報処理装置 6 0 1 と情報取得装置 6 0 2 - 1 ~ 情報取得装置 6 0 2 - N（N は 1 以上の整数）を含む。情報処理装置 6 0 1 及び各情報取得装置 6 0 2 - i（i = 1 ~ N）は、図 4 の情報処理装置 4 0 1 及び情報取得装置 4 0 2 にそれぞれ対応する。情報処理装置 6 0 1 と各情報取得装置 6 0 2 - i は、通信ネットワーク 6 0 3 を介して通信することができる。

【 0 0 2 9 】

情報取得装置 6 0 2 - i は、各種センサ、撮像装置、マイクロフォン等を含み、所定の情報を取得する。例えば、情報取得装置 6 0 2 - i が圧力センサを含む場合、気圧、水圧等を示す情報が取得され、情報取得装置 6 0 2 - i が温度センサを含む場合、温度を示す情報が取得される。情報取得装置 6 0 2 - i が撮像装置を含む場合、画像情報が取得され、情報取得装置 6 0 2 - i がマイクロフォンを含む場合、音声情報が取得される。情報取得装置 6 0 2 - i に含まれるセンサは、湿度センサ、風力センサ等であってもよい。

【 0 0 3 0 】

情報取得装置 6 0 2 - i は、取得した情報に、その情報の取得時刻を示す時刻情報を付加することで、センサデータの packets を生成し、生成した packets を情報処理装置 6 0 1 へ送信する。情報取得装置 6 0 2 - i は、タイミング信号を用いて時刻情報を生成する。

【 0 0 3 1 】

情報処理装置 6 0 1 は、CPU（Central Processing Unit）6 1 1、メモリ 6 1 2、及び時刻同期部 6 1 3 を含む。CPU 6 1 1（プロセッサ）は、NTP（Network Time Protocol）によって定期的に更新される OS 時刻を、時刻同期部 6 1 3 へ出力する。

【 0 0 3 2 】

時刻同期部 6 1 3 は、CPU 6 1 1 が出力する OS 時刻に基づいて、情報取得装置 6 0 2 - i から受信した packets に含まれる所定の情報に、その情報の取得時刻を示す OS 時刻を付加することで、転送データを生成する。そして、時刻同期部 6 1 3 は、生成した転送データをメモリ 6 1 2 へ転送し、メモリ 6 1 2 は、転送された転送データを記憶する。CPU 6 1 1 は、メモリ 6 1 2 が記憶している転送データに対する情報処理を行う。

【 0 0 3 3 】

時刻同期部 6 1 3 は、受信部 6 2 1、第 1 周波数計算部 6 2 2 - 1 ~ 第 1 周波数計算部 6 2 2 - N、第 1 周波数計算部 6 2 3、変換部 6 2 4 - 1 ~ 変換部 6 2 4 - N、及び付加部 6 2 5 を含む。受信部 6 2 1 及び付加部 6 2 5 は、図 4 の受信部 4 1 2 及び付加部 4 1 5 にそれぞれ対応する。

【 0 0 3 4 】

受信部 6 2 1 は、情報取得装置 6 0 2 - i から packets を受信し、packets に含まれる所定の情報を付加部 6 2 5 へ出力し、packets に含まれる時刻情報を、第 1 周波数計算部 6 2 2 - i 及び変換部 6 2 4 - i へ出力する。

【 0 0 3 5 】

第 2 周波数計算部 6 2 3 は、CPU 6 1 1 が出力する OS 時刻と、時刻同期部 6 1 3 内のクロック信号とを用いて、そのクロック信号の周波数を計算する。そして、第 2 周波数計算部 6 2 3 は、計算したクロック信号の周波数を、第 1 周波数計算部 6 2 2 - 1 ~ 第 1

10

20

30

40

50

周波数計算部 6 2 2 - N 及び変換部 6 2 4 - 1 ~ 変換部 6 2 4 - N へ出力する。

【 0 0 3 6 】

第 1 周波数計算部 6 2 2 - i は、2 つのパケットに含まれる時刻情報の差分と、時刻同期部 6 1 3 内のクロック信号と、第 2 周波数計算部 6 2 3 が出力する、クロック信号の周波数とを用いて、情報取得装置 6 0 2 - i 内のタイミング信号の周波数を計算する。そして、第 1 周波数計算部 6 2 2 - i は、計算したタイミング信号の周波数を変換部 6 2 4 - i へ出力する。

【 0 0 3 7 】

変換部 6 2 4 - i は、CPU 6 1 1 が出力する OS 時刻に基づいて、受信部 4 1 2 が出力する時刻情報を OS 時刻に変換し、その OS 時刻を付加部 6 2 5 へ出力する。付加部 6 2 5 は、変換部 6 2 4 - i が出力する OS 時刻を、受信部 4 1 2 が出力する所定の情報にタイムスタンプとして付加することで、転送データを生成し、メモリ 6 1 2 へ転送する。

10

【 0 0 3 8 】

図 7 は、情報取得装置 6 0 2 - i の一例であるセンサモジュールの構成例を示している。図 7 のセンサモジュール 7 0 1 は、圧力センサ 7 1 1、水晶振動子 7 1 2、A D (Analog-to-Digital) コンバータ 7 1 3、カウンタ 7 1 4、書き込み回路 7 1 5、データバッファ 7 1 6、及び I 2 C (Inter-Integrated Circuit) バスマスタ 7 1 7 を含む。この場合、図 6 の通信ネットワーク 6 0 3 として I 2 C バスが用いられる。

【 0 0 3 9 】

圧力センサ 7 1 1 は、圧力を示すアナログ信号を A D コンバータ 7 1 3 へ出力し、水晶振動子 7 1 2 は、正弦波のタイミング信号を A D コンバータ 7 1 3、カウンタ 7 1 4、及び I 2 C バスマスタ 7 1 7 へ出力する。

20

【 0 0 4 0 】

カウンタ 7 1 4 は、水晶振動子 7 1 2 が出力するタイミング信号に同期してカウント動作を行い、カウント値を書き込み回路 7 1 5 へ出力する。カウント値は、情報の取得時刻を示す時刻情報の一例である。A D コンバータ 7 1 3 は、水晶振動子 7 1 2 が出力するタイミング信号に同期して、圧力センサ 7 1 1 が出力するアナログ信号をデジタル信号の情報 7 2 1 に変換し、書き込み回路 7 1 5 へ出力する。

【 0 0 4 1 】

書き込み回路 7 1 5 は、A D コンバータ 7 1 3 が出力する情報 7 2 1 に、カウンタ 7 1 4 が出力するカウント値を付加することで、センサデータの packets を生成して、データバッファ 7 1 6 へ出力する。データバッファ 7 1 6 は、複数の packets をバッファリングして、I 2 C バスマスタ 7 1 7 へ出力する。

30

【 0 0 4 2 】

I 2 C バスマスタ 7 1 7 は、水晶振動子 7 1 2 が出力するタイミング信号に同期して動作し、データバッファ 7 1 6 が出力する packets を、シリアルデータ S D A として情報処理装置 6 0 1 へ送信する。I 2 C バスマスタ 7 1 7 は、シリアルデータ S D A とともに、シリアルクロック S C L を情報処理装置 6 0 1 へ送信する。

【 0 0 4 3 】

図 8 は、情報取得装置 6 0 2 - i の一例であるビデオモジュールの構成例を示している。図 8 のビデオモジュール 8 0 1 は、撮像装置 8 1 1、水晶振動子 8 1 2、A D コンバータ 8 1 3、カウンタ 8 1 4、書き込み回路 8 1 5、及びデータバッファ 8 1 6 を含む。

40

【 0 0 4 4 】

撮像装置 8 1 1 は、画像センサを含み、デジタル映像を撮影して、V G A (Video Graphics Array) 信号を A D コンバータ 8 1 3 へ出力する。V G A 信号は、H s y n c (Horizontal synchronizing signal)、V s y n c (Vertical synchronizing signal)、b l u e ビデオ信号、g r e e n ビデオ信号、及び r e d ビデオ信号を含む。H s y n c は、書き込み回路 8 1 5 にも出力される。水晶振動子 8 1 2 は、正弦波のタイミング信号をカウンタ 8 1 4 へ出力する。

【 0 0 4 5 】

50

カウンタ 8 1 4 は、水晶振動子 8 1 2 が出力するタイミング信号に同期してカウント動作を行い、カウント値を書き込み回路 8 1 5 へ出力する。A D コンバータ 8 1 3 は、撮像装置 8 1 1 が出力する V G A 信号をデジタル信号の画像情報 8 2 1 に変換し、書き込み回路 8 1 5 へ出力する。A D コンバータ 8 1 3 の代わりに、デジタルエンコーダを用いてもよい。

【 0 0 4 6 】

書き込み回路 8 1 5 は、H s y n c に同期して動作し、A D コンバータ 8 1 3 が出力する画像情報 8 2 1 に、カウンタ 8 1 4 が出力するカウント値を付加することで、センサデータの packets を生成して、データバッファ 8 1 6 へ出力する。データバッファ 8 1 6 は、複数の packets をバッファリングして、情報処理装置 6 0 1 へ送信する。

10

【 0 0 4 7 】

図 6 の時刻同期部 6 1 3 は、ハードウェア制御又はソフトウェア制御によって実現される。ハードウェア制御による時刻同期部 6 1 3 としては、F P G A のようなハードウェア論理回路が用いられ、ソフトウェア制御による時刻同期部 6 1 3 としては、コンピュータが用いられる。

【 0 0 4 8 】

以下では、図 9 ~ 図 1 7 を参照しながら、時刻同期部 6 1 3 として F P G A を用いた場合の構成及び動作を説明する。

【 0 0 4 9 】

図 9 は、図 6 の変換部 6 2 4 - i の構成例を示している。図 9 の変換部 6 2 4 - i は、書き込み回路 9 0 1、判定回路 9 0 2、セクタ 9 0 3、カウンタ 9 0 4、レジスタ 9 0 5 ~ レジスタ 9 1 1、及び減算器 (s u b) 9 1 2 を含む。変換部 6 2 4 - i は、さらに、D S P (Digital Signal Processor) 9 1 3、D S P 9 1 4、加算器 (a d d) 9 1 5、及び加算器 9 1 6 を含む。

20

【 0 0 5 0 】

レジスタ 9 0 5 は、図 4 の記憶部 4 1 1 に対応し、D S P 9 1 3 及び加算器 9 1 5 は、受信時刻変換部 4 1 3 に対応し、減算器 9 1 2、D S P 9 1 4、及び加算器 9 1 6 は、時刻情報変換部 4 1 4 に対応する。減算器 9 1 2 は減算部の一例であり、D S P 9 1 4 は第 1 差分計算部の一例であり、加算器 9 1 6 は第 1 加算部の一例である。D S P 9 1 3 は第 2 差分計算部の一例であり、加算器 9 1 5 は第 2 加算部の一例である。

30

【 0 0 5 1 】

書き込み回路 9 0 1 は、C P U 6 1 1 が出力する O S 時刻である O S _ s y s t e m _ t i m e を受信すると、O S _ s y s t e m _ t i m e を O S _ t i m e としてレジスタ 9 0 5 に書き込む。レジスタ 9 0 5 は、O S _ t i m e を加算器 9 1 5 へ出力する。

【 0 0 5 2 】

そして、書き込み回路 9 0 1 は、カウンタ 9 0 4 をクリアして、カウント値 F P G A _ c n t に 0 を設定し、レジスタ 9 0 9 が記憶するフラグ f i r s t _ p k t f l a g をクリアして、f i r s t _ p k t f l a g に論理 “ 0 ” を設定する。その後、カウンタ 9 0 4 は、時刻同期部 6 1 3 内のクロック信号 F P G A _ c l k に同期してカウント動作を行い、カウント値 F P G A _ c n t をレジスタ 9 0 6 へ出力する。

40

【 0 0 5 3 】

受信部 6 2 1 は、情報取得装置 6 0 2 - i から packets を受信したとき、packets の受信を示す信号 s e n s o r _ v a l i d と、packets に含まれるカウント値 s e n s o r _ c n t とを、第 1 周波数計算部 6 2 2 - i 及び変換部 6 2 4 - i へ出力する。s e n s o r _ v a l i d 及び s e n s o r _ c n t は、判定回路 9 0 2 及びセクタ 9 0 3 にそれぞれ入力される。

【 0 0 5 4 】

判定回路 9 0 2 は、s e n s o r _ v a l i d を受信した場合、レジスタ 9 0 9 の f i r s t _ p k t f l a g をチェックする。f i r s t _ p k t f l a g が論理 “ 0 ” である場合、判定回路 9 0 2 は、f i r s t _ p k t f l a g に論理 “ 1 ” を設定し、カウンタ 9

50

04が出力するFPGA_cntを、1st_pkt_clkとしてレジスタ906に書き込む。

【0055】

したがって、OS_system_timeがレジスタ905に書き込まれた後、受信部621が最初の packetsを受信したときに、first_pkt_flagに論理“1”が設定され、そのときのFPGA_cntが1st_pkt_clkに設定される。1st_pkt_clkは、最初の packetsを受信した受信時刻を示している。レジスタ906は、1st_pkt_clkをDSP913へ出力する。

【0056】

次に、判定回路902は、セクタ903にレジスタ907を選択させ、セクタ903は、sensor_cntをレジスタ907へ出力し、レジスタ907は、sensor_cntを1st_pkt_cntとして記憶する。そして、判定回路902は、セクタ903にレジスタ908を選択させ、セクタ903は、sensor_cntをレジスタ908へ出力し、レジスタ908は、sensor_cntをcur_pkt_cntとして記憶する。

10

【0057】

したがって、受信部621が最初の packetsを受信したとき、その packetsのsensor_cntが、1st_pkt_cnt及びcur_pkt_cntに設定される。

【0058】

一方、first_pkt_flagが論理“1”である場合、判定回路902は、セクタ903にレジスタ908を選択させ、レジスタ908は、sensor_cntをcur_pkt_cntとして記憶する。この場合、first_pkt_flag、1st_pkt_clk、及び1st_pkt_cntは更新されない。

20

【0059】

したがって、OS_system_timeがレジスタ905に書き込まれた後、受信部621が2番目以降の packetsを受信したとき、その packetsのsensor_cntがcur_pkt_cntに設定され、1st_pkt_clk及び1st_pkt_cntは変化しない。

【0060】

このように、1st_pkt_cntは、最初の packetsのsensor_cntを表し、cur_pkt_cntは、最初の packets又は2番目以降の packetsのsensor_cntを表す。レジスタ907及びレジスタ908は、1st_pkt_cnt及びcur_pkt_cntをそれぞれ減算器912へ出力する。減算器912は、cur_pkt_cntから1st_pkt_cntを減算することで、カウント値の差分を求め、DSP914へ出力する。

30

【0061】

したがって、受信部621が最初の packetsを受信したとき、減算器912は、カウント値の差分として0を出力する。一方、受信部621が2番目以降の packetsを受信したとき、減算器912は、受信した packetsのsensor_cntと最初の packetsのsensor_cntとの差分を出力する。

40

【0062】

第2周波数計算部623は、クロック信号FPGA_clkの周波数FPGA_freqを、第1周波数計算部622-1～第1周波数計算部622-N及び変換部624-1～変換部624-Nへ出力する。レジスタ910は、FPGA_freqを記憶し、DSP913へ出力する。

【0063】

第1周波数計算部622-iは、情報取得装置602-i内のタイミング信号の周波数sensor_freqを変換部624-iへ出力する。レジスタ911は、sensor_freqを記憶し、DSP914へ出力する。

【0064】

50

DSP913は、レジスタ906が出力する1st_pkt_clkと、レジスタ910が出力するFPGA_freqとを用いて、受信部621が最初の packetsを受信したときのOS時刻と、レジスタ905が記憶するOS_timeとの差分を計算する。DSP913は、1st_pkt_clkをFPGA_freqで除算することで、差分を求めることができる。そして、DSP913は、計算した差分を加算器915へ出力する。

【0065】

加算器915は、DSP913が出力する差分を、レジスタ905が出力するOS_timeに加算することで、受信部621が最初の packetsを受信したときのOS時刻を求め、加算器916へ出力する。このように、DSP913及び加算器915によって、1st_pkt_clkを、最初の packetsの受信時刻を示すOS時刻に変換することができる。

10

【0066】

DSP914は、レジスタ911が出力するsensor_freqを用いて、減算器912が出力するカウント値の差分から、受信した packetsのsensor_cntと最初の packetsのsensor_cntとの差分に対応する、OS時刻の差分を計算する。DSP914は、カウント値の差分をsensor_freqで除算することで、OS時刻の差分を求めることができる。そして、DSP914は、計算した差分を加算器916へ出力する。

【0067】

DSP914が出力するOS時刻の差分は、sensor_cntの差分をsensor_freqに基づいて変換した値であるため、情報取得装置602-iと時刻同期部613との間における packetsの転送時間を含んでいない。したがって、通信ネットワーク603の転送遅延とは無関係である。

20

【0068】

加算器916は、DSP914が出力する差分を、加算器915が出力するOS時刻に加算することで、最初の packetsの受信時刻を示すOS時刻から、sensor_cntの差分に対応する相対時間が経過した後のOS時刻を求める。そして、加算器916は、求めたOS時刻をReal_timeとして付加部625へ出力する。

【0069】

このように、減算器912、DSP914、及び加算器916によって、受信した packetsのsensor_cntを、その packetsに含まれる情報の取得時刻を示すOS時刻に変換することができる。加算器916が出力するReal_timeは、受信した packetsに含まれる情報にタイムスタンプとして付加される。

30

【0070】

図10A及び図10Bは、図9の変換部624-iが行う時刻同期処理の具体例を示すフローチャートである。まず、書き込み回路901は、CPU611からOS_system_timeを受信したか否かをチェックする(ステップ1001)。

【0071】

OS_system_timeを受信した場合(ステップ1001, YES)、書き込み回路901は、OS_system_timeをOS_timeとしてレジスタ905に書き込む(ステップ1003)。そして、書き込み回路901は、カウンタ904のFPGA_cntをクリアし(ステップ1004)、レジスタ909のfirst_pkt_flagをクリアする(ステップ1005)。

40

【0072】

次に、判定回路902は、受信部621からsensor_validを受信したか否かをチェックする(ステップ1002)。sensor_validを受信した場合(ステップ1002, YES)、判定回路902は、レジスタ909のfirst_pkt_flagをチェックする(ステップ1006)。

【0073】

first_pkt_flagが論理“0”である場合(ステップ1006, YES)、判

50

定回路 902 は、`first_pkt_flag` に論理 “1” を設定する (ステップ 1009)。そして、判定回路 902 は、カウンタ 904 が出力する `FPGA_cnt` を、`1st_pkt_clk` としてレジスタ 906 に書き込む (ステップ 1010)。

【0074】

次に、セクタ 903 は、`sensor_cnt` をレジスタ 907 へ出力し、レジスタ 907 は、`sensor_cnt` を `1st_pkt_cnt` として記憶する (ステップ 1011)。そして、セクタ 903 は、`sensor_cnt` をレジスタ 908 へ出力し、レジスタ 908 は、`sensor_cnt` を `cur_pkt_cnt` として記憶する (ステップ 1007)。

【0075】

次に、減算器 912 は、`cur_pkt_cnt` から `1st_pkt_cnt` を減算することで、カウント値の差分を求める (ステップ 1008)。そして、DSP 914 は、レジスタ 911 の `sensor_freq` を用いて、減算器 912 が出力するカウント値の差分を OS 時刻の差分に変換する (ステップ 1012)。

【0076】

次に、DSP 913 は、レジスタ 910 の `FPGA_freq` を用いて、レジスタ 906 の `1st_pkt_clk` を、受信部 621 が最初の packets を受信したときの OS 時刻と、レジスタ 905 の `OS_time` との差分に変換する (ステップ 1013)。

【0077】

次に、加算器 915 は、DSP 913 が出力する差分を、レジスタ 905 の `OS_time` に加算し、加算器 916 は、DSP 914 が出力する差分を、加算器 915 が出力する加算結果に加算する (ステップ 1014)。これにより、`Real_time` が求められる。

【0078】

次に、カウンタ 904 は、`FPGA_cnt` をカウントアップする (ステップ 1015)。そして、変換部 624-i は、レジスタ 911 の `sensor_freq` を更新し (ステップ 1016)、レジスタ 910 の `FPGA_freq` を更新して (ステップ 1017)、ステップ 1001 以降の処理を繰り返す。

【0079】

`OS_system_time` を受信していない場合 (ステップ 1001, NO)、変換部 624-i は、ステップ 1002 以降の処理を行う。`first_pkt_flag` が論理 “1” である場合 (ステップ 1006, NO)、変換部 624-i は、ステップ 1007 以降の処理を行う。`sensor_valid` を受信していない場合 (ステップ 1002, NO)、変換部 624-i は、ステップ 1015 以降の処理を行う。

【0080】

図 11 は、図 9 の変換部 624-i が行う時刻同期処理の例を示している。CPU 611 は、所定のタイミングで `OS_system_time` を出力し、レジスタ 905 の `OS_time` を更新する。

【0081】

図 11 の例では、15:00:05、15:00:10、及び 15:00:16 が順番に `OS_time` に設定されている。タイムフレーム 1101 は、15:00:05 から 15:00:10 までの区間を表し、タイムフレーム 1102 は、15:00:10 から 15:00:16 までの区間を表す。

【0082】

タイムフレーム 1101 において、受信部 621 は、`FPGA_cnt` = 50 のときに最初の packets を受信し、`FPGA_cnt` = 71 のときに 2 番目の packets を受信し、`FPGA_cnt` = 125 のときに 3 番目の packets を受信する。最初の packets の `sensor_cnt` は 1010 であり、2 番目の packets の `sensor_cnt` は 1050 であり、3 番目の packets の `sensor_cnt` は 1150 である。

【0083】

10

20

30

40

50

この場合、2番目のパケットの `sensor_cnt` と最初のパケットの `sensor_cnt` との差分“40”が、OS時刻の差分1111に変換される。そして、`FPGA_cnt = 50` に対応するOS時刻に差分1111が加算されて、`sensor_cnt = 1050` に対応する `Real_time` が求められる。

【0084】

また、3番目のパケットの `sensor_cnt` と最初のパケットの `sensor_cnt` との差分“140”が、OS時刻の差分1112に変換される。そして、`FPGA_cnt = 50` に対応するOS時刻に差分1112が加算されて、`sensor_cnt = 1150` に対応する `Real_time` が求められる。

【0085】

タイムフレーム1102において、受信部621は、`FPGA_cnt = 36` のときに最初のパケットを受信し、`FPGA_cnt = 73` のときに2番目のパケットを受信し、`FPGA_cnt = 114` のときに3番目のパケットを受信する。最初のパケットの `sensor_cnt` は1213であり、2番目のパケットの `sensor_cnt` は1255であり、3番目のパケットの `sensor_cnt` は1291である。

【0086】

タイムフレーム1102においても、タイムフレーム1101の場合と同様に、各 `sensor_cnt` に対応する `Real_time` が求められる。

【0087】

図6の情報処理システムによれば、各タイムフレーム内において受信した複数のパケットに含まれる情報に、情報取得装置602-iが情報を取得したときのOS時刻を示すタイムスタンプが付加される。複数の情報それぞれに付加されたタイムスタンプの差分は正確な相対時間を表しているため、それらの情報の取得時刻を情報処理装置601のOS時刻と同期させることができる。この場合、図3の構成のように、情報処理装置601から情報取得装置602-iへリセット信号を送信する必要はない。

【0088】

また、各情報取得装置602-iが個別のタイミング信号に同期して動作している場合であっても、情報取得装置602-1～情報取得装置602-Nの間で、情報の取得時刻を同期させることが可能になる。

【0089】

図12は、図6の第2周波数計算部623の構成例を示している。図12の第2周波数計算部623は、書き込み回路1201、ロード回路()1202、カウンタ1203、レジスタ1204～レジスタ1206、減算器1207、減算器1208、DSP1209、及び統計値計算回路1210を含む。

【0090】

書き込み回路1201は、CPU611からOS_system_timeを受信すると、ロード指示をロード回路1202へ出力し、OS_system_timeをcurrent_timeとしてレジスタ1205に書き込む。レジスタ1205は、current_timeを減算器1208へ出力する。

【0091】

そして、書き込み回路1201は、カウンタ1203をクリアして、カウント値に0を設定する。その後、カウンタ1203は、FPGA_clkに同期してカウント動作を行い、カウント値をレジスタ1204及び減算器1207へ出力する。

【0092】

ロード回路1202は、書き込み回路1201が出力するロード指示に従って、OS_system_timeが書き込まれる直前のcurrent_timeを、last_timeとしてレジスタ1206に書き込む。レジスタ1206は、last_timeを減算器1208へ出力する。

【0093】

また、ロード回路1202は、書き込み回路1201が出力するロード指示に従って、

10

20

30

40

50

カウンタ 1203 がクリアされる直前のカウント値を、レジスタ 1204 に書き込む。レジスタ 1204 は、カウント値を減算器 1207 へ出力する。

【0094】

減算器 1207 は、レジスタ 1204 が出力するカウント値から、カウンタ 1203 が出力するカウント値を減算することで、カウント値の差分を求め、DSP 1209 へ出力する。減算器 1208 は、レジスタ 1205 が出力する `current_time` から、レジスタ 1206 が出力する `last_time` を減算することで、OS 時刻の差分を求め、DSP 1209 へ出力する。

【0095】

DSP 1209 は、減算器 1207 が出力する差分を、減算器 1208 が出力する差分で除算することで、FPGA `clk` の周波数を求め、統計値計算回路 1210 へ出力する。統計値計算回路 1210 は、DSP 1209 が出力する複数の周波数の統計値を求め、求めた統計値を FPGA `freq` として出力する。統計値としては、平均値、中央値、最頻値等を用いることができる。統計値を求める必要がない場合は、統計値計算回路 1210 を省略してもよい。

10

【0096】

図 12 の第 2 周波数計算部 623 によれば、環境の変化によって、時刻同期部 613 内のクロック信号 FPGA `clk` の周波数が変動する場合であっても、正確な周波数を求めることができる。そして、求められた周波数を用いることで、図 9 の Real `time` を補正することが可能になる。

20

【0097】

図 13 は、図 6 の第 1 周波数計算部 622 - i の構成例を示している。図 13 の第 1 周波数計算部 622 - i は、カウンタ 1301、レジスタ 1302 ~ レジスタ 1304、減算器 1305、減算器 1306、DSP 1307、DSP 1308、及び統計値計算回路 1309 を含む。

【0098】

カウンタ 1301 は、FPGA `clk` に同期してカウント動作を行い、カウント値 FPGA `cur_clk` をレジスタ 1302 及び減算器 1305 へ出力する。第 1 周波数計算部 622 - i は、受信部 621 から `sensor_valid` を受信したとき、カウンタ 1301 が出力する FPGA `cur_clk` を、FPGA `last_clk` としてレジスタ 1302 に書き込む。レジスタ 1302 は、FPGA `last_clk` を減算器 1305 へ出力する。

30

【0099】

レジスタ 1303 は、受信部 621 が出力する `sensor_cnt` を、`sensor_cur_clk` として記憶し、`sensor_cur_clk` を減算器 1306 へ出力する。第 1 周波数計算部 622 - i は、受信部 621 から `sensor_valid` を受信したとき、レジスタ 1303 の `sensor_cur_clk` を、`sensor_last_clk` としてレジスタ 1304 に書き込む。レジスタ 1304 は、`sensor_last_clk` を減算器 1306 へ出力する。

【0100】

40

減算器 1305 は、カウンタ 1301 が出力する FPGA `cur_clk` から、レジスタ 1302 が出力する FPGA `last_clk` を減算することで、カウント値の差分を求め、DSP 1307 へ出力する。減算器 1306 は、レジスタ 1303 が出力する `sensor_cur_clk` から、レジスタ 1304 が出力する `sensor_last_clk` を減算することで、`sensor_cnt` の差分を求め、DSP 1307 へ出力する。

【0101】

DSP 1307 は、減算器 1306 が出力する差分を、減算器 1305 が出力する差分で除算し、除算結果を DSP 1308 へ出力する。DSP 1308 は、DSP 1307 が出力する除算結果に、第 2 周波数計算部 623 が出力する FPGA `freq` を乗算する

50

ことで、情報取得装置 602 - i 内のタイミング信号の周波数を求め、統計値計算回路 1309 へ出力する。統計値計算回路 1309 は、DSP 1308 が出力する複数の周波数の統計値を求め、求めた統計値を `sensor_freq` として出力する。統計値としては、平均値、中央値、最頻値等を用いることができる。統計値を求める必要がない場合は、統計値計算回路 1309 を省略してもよい。

【0102】

図 13 の第 1 周波数計算部 622 - i によれば、環境の変化によって、情報取得装置 602 - i 内のタイミング信号の周波数が変動する場合であっても、正確な周波数を求めることができる。そして、求められた周波数を用いることで、図 9 の `Real_time` を補正することが可能になる。

10

【0103】

図 14 は、図 12 の第 2 周波数計算部 623 によって計算される、クロック信号 `FPGA_clk` の周波数 `FPGA_freq` の例を示している。この例では、統計値計算回路 1210 による統計値計算は省略されている。図 14 の `Sub` 出力は、減算器 1207 が出力するカウント値の差分を表す。`FPGA_freq` は、次式により計算される。

$$\begin{aligned} & \text{FPGA_freq} \\ &= \text{Sub 出力} / (\text{current_time} - \text{last_time}) \quad (1) \end{aligned}$$

【0104】

図 11 の時刻同期処理において、`OS_time = 15:00:05` のとき、`Sub` 出力 = 110000 である。次に、`OS_time = 15:00:10` のとき、カウンタ 1203、レジスタ 1204、`current_time`、及び `last_time` の値は、以下のようになる。

20

```
カウンタ 1203  00000
レジスタ 1204  10000
current_time  15:00:10
last_time     15:00:05
```

【0105】

この場合、`Sub` 出力 = 10000 となり、`FPGA_freq` は、次式により計算される。

$$\text{FPGA_freq} = 10000 / (10 - 5) = 2000 [\text{Hz}] \quad (2)$$

30

【0106】

次に、`OS_time = 15:00:16` のとき、カウンタ 1203、レジスタ 1204、`current_time`、及び `last_time` の値は、以下のようになる。

```
カウンタ 1203  00000
レジスタ 1204  11000
current_time  15:00:16
last_time     15:00:10
```

【0107】

この場合、`Sub` 出力 = 11000 となり、`FPGA_freq` は、次式により計算される。

40

$$\text{FPGA_freq} = 11000 / (16 - 10) = 1833 [\text{Hz}] \quad (3)$$

【0108】

図 15 は、図 13 の第 1 周波数計算部 622 - i によって計算される、タイミング信号の周波数 `sensor_freq` の例を示している。この例では、統計値計算回路 1309 による統計値計算は省略されている。`sensor_freq` は、次式により計算される。

$$\begin{aligned} & \text{sensor_freq} \\ &= \{ (\text{sensor_cur_clk} - \text{sensor_last_clk}) \\ & \quad / (\text{FPGA_cur_clk} - \text{FPGA_last_clk}) \} \\ & \quad * \text{FPGA_freq} \quad (4) \end{aligned}$$

50

【 0 1 0 9 】

図 1 1 のタイムフレーム 1 1 0 1 において、最初の packets を受信したとき、`sensor_cnt`、`FPGA_cur_clk`、及び `FPGA_freq` の値は、以下のようになる。

```
sensor_cnt = 1 0 1 0
FPGA_cur_clk = 2 0 5 0
FPGA_freq = 2 0 0 0
```

【 0 1 1 0 】

次に、2 番目の packets を受信したとき、`sensor_cnt`、`FPGA_cur_clk`、`FPGA_last_clk`、`sensor_cur_clk`、`sensor_last_clk`、及び `FPGA_freq` の値は、以下のようになる。

```
sensor_cnt = 1 0 5 0
FPGA_cur_clk = 2 0 7 1
FPGA_last_clk = 2 0 5 0
sensor_cur_clk = 1 0 5 0
sensor_last_clk = 1 0 1 0
FPGA_freq = 2 0 0 0
```

【 0 1 1 1 】

この場合、`sensor_freq` は、次式により計算される。

```
sensor_freq
= { ( 1 0 5 0 - 1 0 1 0 ) / ( 2 0 7 1 - 2 0 5 0 ) } * 2 0 0 0
= 3 8 1 0 [ Hz ] ( 5 )
```

【 0 1 1 2 】

次に、3 番目の packets を受信したとき、`sensor_cnt`、`FPGA_cur_clk`、`FPGA_last_clk`、`sensor_cur_clk`、`sensor_last_clk`、及び `FPGA_freq` の値は、以下のようになる。

```
sensor_cnt = 1 1 5 0
FPGA_cur_clk = 2 1 2 5
FPGA_last_clk = 2 0 7 1
sensor_cur_clk = 1 1 5 0
sensor_last_clk = 1 0 5 0
FPGA_freq = 2 0 0 0
```

【 0 1 1 3 】

この場合、`sensor_freq` は、次式により計算される。

```
sensor_freq
= { ( 1 1 5 0 - 1 0 5 0 ) / ( 2 1 2 5 - 2 0 7 1 ) } * 2 0 0 0
= 3 7 0 4 [ Hz ] ( 6 )
```

【 0 1 1 4 】

図 1 6 は、図 9 の変換部 6 2 4 - i によって計算される OS 時刻 `Real_time` の例を示している。`Real_time` は、次式により計算される。

```
Real_time
= ( cur_pkt_cnt - 1 st_pkt_cnt ) / sensor_freq
+ 1 st_pkt_clk / FPGA_freq + OS_time ( 7 )
```

【 0 1 1 5 】

図 1 1 の時刻同期処理において、`OS_time = 15 : 00 : 05` のとき、`1st_pkt_clk`、`first_pkt_flag`、`FPGA_freq`、及び `sensor_freq` の値は、以下のようになる。

```
1st_pkt_clk = 0
first_pkt_flag = 0
```

10

20

30

40

50

```
FPGA_freq = 2000
sensor_freq = 4000
```

【0116】

次に、最初の packets を受信したとき、`sensor_cnt`、`1st_pkt_clk`、`first_pktflag`、`1st_pkt_cnt`、`cur_pkt_cnt`、`FPGA_freq`、及び `sensor_freq` の値は、以下のようになる。

```
sensor_cnt = 1010
1st_pkt_clk = 50
first_pktflag = 1
1st_pkt_cnt = 1010
cur_pkt_cnt = 1010
FPGA_freq = 2000
sensor_freq = 4000
```

10

【0117】

この場合、`Real_time` は、次式により計算される。

```
Real_time
= (1010 - 1010) / 4000 + 50 / 2000 + 15 : 00 : 05
= 15 : 00 : 05 : 025 (8)
```

【0118】

次に、2 番目の packets を受信したとき、`sensor_cnt`、`1st_pkt_clk`、`first_pktflag`、`1st_pkt_cnt`、`cur_pkt_cnt`、`FPGA_freq`、及び `sensor_freq` の値は、以下のようになる。

20

```
sensor_cnt = 1050
1st_pkt_clk = 50
first_pktflag = 1
1st_pkt_cnt = 1010
cur_pkt_cnt = 1050
FPGA_freq = 2000
sensor_freq = 3810
```

【0119】

30

この場合、`Real_time` は、次式により計算される。

```
Real_time
= (1050 - 1010) / 3810 + 50 / 2000 + 15 : 00 : 05
= 15 : 00 : 05 : 035 (9)
```

【0120】

次に、3 番目の packets を受信したとき、`sensor_cnt`、`1st_pkt_clk`、`first_pktflag`、`1st_pkt_cnt`、`cur_pkt_cnt`、`FPGA_freq`、及び `sensor_freq` の値は、以下のようになる。

```
sensor_cnt = 1150
1st_pkt_clk = 50
first_pktflag = 1
1st_pkt_cnt = 1010
cur_pkt_cnt = 1150
FPGA_freq = 2000
sensor_freq = 3704
```

40

【0121】

この場合、`Real_time` は、次式により計算される。

```
Real_time
= (1150 - 1010) / 3704 + 50 / 2000 + 15 : 00 : 05
= 15 : 00 : 05 : 063 (10)
```

50

【 0 1 2 2 】

ところで、通信ネットワーク 6 0 3 において障害等が発生した場合、情報取得装置 6 0 2 - i から送信されたセンサデータの packets が消失して、情報処理装置 6 0 1 に到達しないことがある。このような場合であっても、時刻同期部 6 1 3 は、情報処理装置 6 0 1 に到達した packets に含まれる情報に対して、タイムスタンプを付加することが可能である。

【 0 1 2 3 】

図 1 7 は、通信ネットワーク 6 0 3 における packets 消失時の時刻同期処理の例を示している。図 1 7 (a) は、図 1 1 のタイムフレーム 1 1 0 1 において、最初の packets が消失した場合の時刻同期処理の例を示している。

10

【 0 1 2 4 】

図 1 7 (a) のタイムフレーム 1 1 0 1 において、最初の packets が消失した場合、受信部 6 2 1 は、 $FPGA_cnt = 71$ のときに 2 番目の packets を受信し、 $FPGA_cnt = 125$ のときに 3 番目の packets を受信する。

【 0 1 2 5 】

この場合、3 番目の packets の $sensor_cnt$ と 2 番目の packets の $sensor_cnt$ との差分 “ 1 0 0 ” が、OS 時刻の差分 1 7 0 1 に変換される。そして、 $FPGA_cnt = 71$ に対応する OS 時刻に差分 1 7 0 1 が加算されて、 $sensor_cnt = 1150$ に対応する $Real_time$ が求められる。

【 0 1 2 6 】

20

図 1 7 (b) は、図 1 1 のタイムフレーム 1 1 0 1 において、2 番目の packets が消失した場合の時刻同期処理の例を示している。

【 0 1 2 7 】

図 1 7 (b) のタイムフレーム 1 1 0 1 において、2 番目の packets が消失した場合、受信部 6 2 1 は、 $FPGA_cnt = 50$ のときに最初の packets を受信し、 $FPGA_cnt = 125$ のときに 3 番目の packets を受信する。

【 0 1 2 8 】

この場合、3 番目の packets の $sensor_cnt$ と最初の packets の $sensor_cnt$ との差分 “ 1 4 0 ” が、OS 時刻の差分 1 1 1 2 に変換される。そして、 $FPGA_cnt = 50$ に対応する OS 時刻に差分 1 1 1 2 が加算されて、 $sensor_cnt = 1150$ に対応する $Real_time$ が求められる。

30

【 0 1 2 9 】

このように、図 6 の情報処理システムによれば、タイムフレーム内のいずれの packets が消失した場合であっても、いずれかの packets を受信した OS 時刻を基準として、後続する packets に含まれる情報にタイムスタンプを付加することができる。

【 0 1 3 0 】

図 1 8 は、ソフトウェア制御による時刻同期部 6 1 3 の構成例を示している。図 1 8 の時刻同期部 6 1 3 は、CPU 1 8 0 1、メモリ 1 8 0 2、及びインタフェース 1 8 0 3 を含むコンピュータである。

【 0 1 3 1 】

40

メモリ 1 8 0 2 は、例えば、ROM (Read Only Memory)、RAM (Random Access Memory)、フラッシュメモリ等の半導体メモリであり、処理に用いられるプログラム及びデータを格納する。メモリ 1 8 0 2 は、図 4 の記憶部 4 1 1 として用いることができる。

【 0 1 3 2 】

CPU 1 8 0 1 (プロセッサ) は、例えば、メモリ 1 8 0 2 を利用してプログラムを実行することにより、図 4 の受信時刻変換部 4 1 3、時刻情報変換部 4 1 4、及び付加部 4 1 5 として動作する。CPU 1 8 0 1 は、メモリ 1 8 0 2 を利用してプログラムを実行することにより、第 1 周波数計算部、第 2 周波数計算部、減算部、第 1 差分計算部、第 1 加算部、第 2 差分計算部、及び第 2 加算部としても動作する。

50

【 0 1 3 3 】

インタフェース 1 8 0 3 は、通信ネットワーク 6 0 3 に接続され、通信に伴うデータ変換を行う通信インタフェース回路である。時刻同期部 6 1 3 は、プログラム及びデータを外部の装置からインタフェース 1 8 0 3 を介して受信し、それらをメモリ 1 8 0 2 にロードして使用することができる。インタフェース 1 8 0 3 は、図 4 の受信部 4 1 2 又は図 6 の受信部 6 2 1 として用いることができる。

【 0 1 3 4 】

図 1 8 の時刻同期部 6 1 3 は、さらに、可搬型記録媒体を駆動する媒体駆動装置を含んでいてもよい。可搬型記録媒体は、メモリデバイス、フレキシブルディスク、光ディスク、光磁気ディスク等である。可搬型記録媒体は、C D - R O M (Compact Disk Read Only Memory)、D V D (Digital Versatile Disk)、U S B (Universal Serial Bus) メモリ等であってもよい。オペレータ又はユーザは、この可搬型記録媒体にプログラム及びデータを格納しておき、それらをメモリ 1 8 0 2 にロードして使用することができる。

【 0 1 3 5 】

このように、処理に用いられるプログラム及びデータを格納するコンピュータ読み取り可能な記録媒体は、メモリ 1 8 0 2 又は可搬型記録媒体のような、物理的な（非一時的な）記録媒体である。

【 0 1 3 6 】

図 4 の情報処理装置 4 0 1 の構成は一例に過ぎず、情報処理装置 4 0 1 の用途又は条件に応じて一部の構成要素を省略又は変更してもよい。

【 0 1 3 7 】

図 6 の情報処理システムの構成は一例に過ぎず、情報処理システムの用途又は条件に応じて一部の構成要素を省略又は変更してもよい。例えば、時刻同期部 6 1 3 内のクロック信号の周波数があまり変動せず、第 1 周波数計算部 6 2 2 - i 及び変換部 6 2 4 - i がその周波数を示す固定値を保持している場合は、第 2 周波数計算部 6 2 3 を省略することができる。情報取得装置 6 0 2 - i 内のタイミング信号の周波数があまり変動せず、変換部 6 2 4 - i がその周波数を示す固定値を保持している場合は、第 1 周波数計算部 6 2 2 - i を省略することができる。

【 0 1 3 8 】

情報取得装置 6 0 2 - 1 ~ 情報取得装置 6 0 2 - N は、情報処理装置 6 0 1 内に設けられていてもよい。

【 0 1 3 9 】

図 7 のセンサモジュール 7 0 1 及び図 8 のビデオモジュール 8 0 1 の構成は一例に過ぎず、情報処理システムの用途又は条件に応じて一部の構成要素を省略又は変更してもよい。図 7 の圧力センサ 7 1 1 の代わりに、温度センサ、湿度センサ、風力センサ等を用いることもできる。

【 0 1 4 0 】

図 9 の変換部 6 2 4 - i、図 1 2 の第 2 周波数計算部 6 2 3、図 1 3 の第 1 周波数計算部 6 2 2 - i、及び図 1 8 の時刻同期部 6 1 3 の構成は一例に過ぎず、情報処理システムの用途又は条件に応じて一部の構成要素を省略又は変更してもよい。

【 0 1 4 1 】

図 5、図 1 0 A、及び図 1 0 B のフローチャートは一例に過ぎず、変換部 6 2 4 - i の構成又は条件に応じて一部の処理を省略又は変更してもよい。

【 0 1 4 2 】

図 1 1 及び図 1 7 に示した時刻同期処理は一例に過ぎず、`sensor_cnt`、`FGA_cnt`、及び`OS_time`は、情報取得装置 6 0 2 - i が取得する情報に応じて変化する。図 1 4 ~ 図 1 6 に示した計算結果は一例に過ぎず、計算結果は、情報取得装置 6 0 2 - i が取得する情報に応じて変化する。

【 0 1 4 3 】

式 (1) ~ 式 (1 0) の計算式は一例に過ぎず、情報処理装置 6 0 1 の構成又は条件に

10

20

30

40

50

応じて別の計算式を用いてもよい。

【 0 1 4 4 】

開示の実施形態とその利点について詳しく説明したが、当業者は、特許請求の範囲に明確に記載した本発明の範囲から逸脱することなく、様々な変更、追加、省略をすることができるであろう。

【 0 1 4 5 】

図 2 乃至図 1 8 を参照しながら説明した実施形態に関し、さらに以下の付記を開示する。

(付記 1)

第 1 システム時刻を記憶する記憶部と、

前記記憶部に前記第 1 システム時刻が書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第 1 情報と、前記第 1 情報の取得時刻を示す第 1 時刻情報とを受信し、前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記情報取得装置から、第 2 情報と前記第 2 情報の取得時刻を示す第 2 時刻情報とを受信する受信部と、

前記第 1 システム時刻に基づいて、前記受信部が前記第 1 情報と前記第 1 時刻情報とを受信した受信時刻を、第 2 システム時刻に変換する受信時刻変換部と、

前記第 2 システム時刻と前記第 1 時刻情報とに基づいて、前記第 2 時刻情報を第 3 システム時刻に変換する時刻情報変換部と、

前記第 2 システム時刻を前記第 1 情報に付加し、前記第 3 システム時刻を前記第 2 情報に付加する付加部と、

を備えることを特徴とする情報処理装置。

(付記 2)

前記時刻情報変換部は、

前記第 2 時刻情報と前記第 1 時刻情報との差分を生成する減算部と、

前記第 2 時刻情報と前記第 1 時刻情報との差分から、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算する第 1 差分計算部と、

前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算することで、前記第 3 システム時刻を求める第 1 加算部と、

を含むことを特徴とする付記 1 記載の情報処理装置。

(付記 3)

前記第 2 時刻情報と前記第 1 時刻情報との差分と、クロック信号と、前記クロック信号の周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算する第 1 周波数計算部をさらに備え、

前記受信時刻変換部は、前記受信部が前記第 1 情報と前記第 1 時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記受信時刻として用い、

前記第 1 差分計算部は、前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用いて、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算することを特徴とする付記 2 記載の情報処理装置。

(付記 4)

前記受信時刻変換部は、

前記受信時刻と前記クロック信号の周波数とを用いて、前記第 2 システム時刻と前記第 1 システム時刻との差分を計算する第 2 差分計算部と、

前記第 2 システム時刻と前記第 1 システム時刻との差分を前記第 1 システム時刻に加算することで、前記第 2 システム時刻を求める第 2 加算部と、

を含むことを特徴とする付記 3 記載の情報処理装置。

(付記 5)

前記第 1 システム時刻と前記クロック信号とを用いて、前記クロック信号の周波数を計算する第 2 周波数計算部をさらに備えることを特徴とする付記 4 記載の情報処理装置。

(付記 6)

前記受信部は、前記記憶部に前記第 1 システム時刻が書き込まれた後に、複数の情報取得装置のうちいずれかの情報取得装置から、前記いずれかの情報取得装置が取得した第 1

10

20

30

40

50

情報と第 1 時刻情報とを受信し、前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記いずれかの情報取得装置から、第 2 情報と第 2 時刻情報とを受信し、

前記付加部は、前記第 2 システム時刻を前記いずれかの情報取得装置から受信した第 1 情報に付加し、前記第 3 システム時刻を前記いずれかの情報取得装置から受信した第 2 情報に付加することを特徴とする付記 1 乃至 5 のいずれか 1 項に記載の情報処理装置。

(付記 7)

情報処理装置によって実行される時刻同期方法であって、

前記情報処理装置が、

第 1 システム時刻が記憶部に書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第 1 情報と、前記第 1 情報の取得時刻を示す第 1 時刻情報とを受信し、

10

前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記情報取得装置から、第 2 情報と前記第 2 情報の取得時刻を示す第 2 時刻情報とを受信し、

前記第 1 システム時刻に基づいて、前記第 1 情報と前記第 1 時刻情報とを受信した受信時刻を、第 2 システム時刻に変換し、

前記第 2 システム時刻と前記第 1 時刻情報とに基づいて、前記第 2 時刻情報を第 3 システム時刻に変換し、

前記第 2 システム時刻を前記第 1 情報に付加し、

前記第 3 システム時刻を前記第 2 情報に付加する、
ことを特徴とする時刻同期方法。

(付記 8)

20

前記情報処理装置は、

前記第 2 時刻情報と前記第 1 時刻情報との差分を生成し、

前記第 2 時刻情報と前記第 1 時刻情報との差分から、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算し、

前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算することで、前記第 3 システム時刻を求めることを特徴とする付記 7 記載の時刻同期方法。

(付記 9)

前記情報処理装置は、

前記第 2 時刻情報と前記第 1 時刻情報との差分と、クロック信号と、前記クロック信号の周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算し、

30

前記第 1 情報と前記第 1 時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記受信時刻として用い、

前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用いて、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算することを特徴とする付記 8 記載の時刻同期方法。

(付記 10)

前記情報処理装置は、

前記受信時刻と前記クロック信号の周波数とを用いて、前記第 2 システム時刻と前記第 1 システム時刻との差分を計算し、

40

前記第 2 システム時刻と前記第 1 システム時刻との差分を前記第 1 システム時刻に加算することで、前記第 2 システム時刻を求めることを特徴とする付記 9 記載の時刻同期方法。

(付記 11)

第 1 システム時刻が記憶部に書き込まれた後に、情報取得装置から、前記情報取得装置が取得した第 1 情報と、前記第 1 情報の取得時刻を示す第 1 時刻情報とを受信し、

前記第 1 情報と前記第 1 時刻情報とを受信した後に、前記情報取得装置から、第 2 情報と前記第 2 情報の取得時刻を示す第 2 時刻情報とを受信し、

前記第 1 システム時刻に基づいて、前記第 1 情報と前記第 1 時刻情報とを受信した受信時刻を、第 2 システム時刻に変換し、

前記第 2 システム時刻と前記第 1 時刻情報とに基づいて、前記第 2 時刻情報を第 3 シス

50

テム時刻に変換し、

前記第 2 システム時刻を前記第 1 情報に付加し、

前記第 3 システム時刻を前記第 2 情報に付加する、

処理をコンピュータに実行させるための時刻同期プログラム。

(付記 1 2)

前記コンピュータは、

前記第 2 時刻情報と前記第 1 時刻情報との差分を生成し、

前記第 2 時刻情報と前記第 1 時刻情報との差分から、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算し、

前記第 3 システム時刻と前記第 2 システム時刻との差分を前記第 2 システム時刻に加算することで、前記第 3 システム時刻を求めることを特徴とする付記 1 1 記載の時刻同期プログラム。

10

(付記 1 3)

前記コンピュータは、

前記第 2 時刻情報と前記第 1 時刻情報との差分と、クロック信号と、前記クロック信号の周波数とを用いて、前記情報取得装置において取得時刻を生成するタイミング信号の周波数を計算し、

前記第 1 情報と前記第 1 時刻情報とを受信したときに、前記クロック信号に基づいて特定された時刻を、前記受信時刻として用い、

前記第 2 時刻情報と前記第 1 時刻情報との差分と、前記タイミング信号の周波数とを用いて、前記第 3 システム時刻と前記第 2 システム時刻との差分を計算することを特徴とする付記 1 2 記載の時刻同期プログラム。

20

(付記 1 4)

前記コンピュータは、

前記受信時刻と前記クロック信号の周波数とを用いて、前記第 2 システム時刻と前記第 1 システム時刻との差分を計算し、

前記第 2 システム時刻と前記第 1 システム時刻との差分を前記第 1 システム時刻に加算することで、前記第 2 システム時刻を求めることを特徴とする付記 1 3 記載の時刻同期プログラム。

【符号の説明】

30

【 0 1 4 6 】

1 0 1、4 0 1、6 0 1 情報処理装置

1 0 2 - 1、1 0 2 - 2 センサ

1 2 1 - 1 ~ 1 2 1 - 4、1 2 2 - 1 ~ 1 2 2 - 4 データ

1 1 1、2 0 1、3 0 1 F P G A

2 0 2 O S

2 1 1 割り込みコントローラ

2 1 2、3 1 1 M M I O レジスタ

2 1 3 検出部

2 2 1 O S 時刻

3 1 2 加算部

4 0 2、6 0 2 - 1 ~ 6 0 2 - N 情報取得装置

4 1 1 記憶部

4 1 2、6 2 1 受信部

4 1 3 受信時刻変換部

4 1 4 時刻情報変換部

4 1 5、6 2 5 付加部

6 0 3 通信ネットワーク

6 1 1 C P U

6 1 2 メモリ

40

50

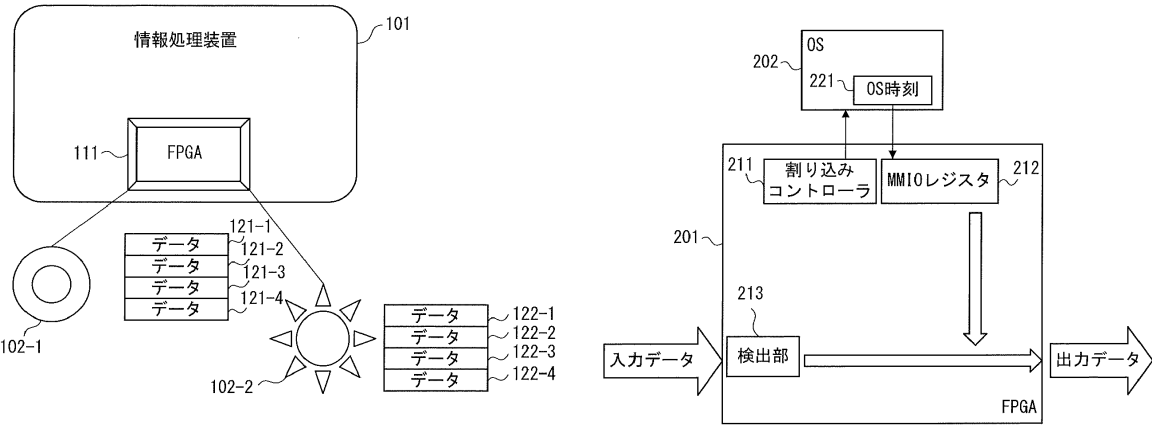
6 1 3	時刻同期部	
6 2 2 - 1 ~ 6 2 2 - N	第 1 周波数計算部	
6 2 3	第 2 周波数計算部	
6 2 4 - 1 ~ 6 2 4 - N	変換部	
7 0 1	センサモジュール	
7 1 1	圧力センサ	
7 1 2、8 1 2	水晶振動子	
7 1 3、8 1 3	A D コンバータ	
7 1 4、8 1 4、9 0 4、1 2 0 3、1 3 0 1	カウンタ	
7 1 5、8 1 5、9 0 1、1 2 0 1	書き込み回路	10
7 1 6、8 1 6	データバッファ	
7 1 7	I 2 C バスマスタ	
7 2 1	情報	
8 0 1	ビデオモジュール	
8 1 1	撮像装置	
8 2 1	画像情報	
9 0 2	判定回路	
9 0 3	セレクタ	
9 0 5 ~ 9 1 1、1 2 0 4 ~ 1 2 0 6、1 3 0 2 ~ 1 3 0 4	レジスタ	
9 1 2、1 2 0 7、1 2 0 8、1 3 0 5、1 3 0 6	減算器	20
9 1 3、9 1 4、1 2 0 9、1 3 0 7、1 3 0 8	D S P	
9 1 5、9 1 6	加算器	
1 1 0 1、1 1 0 2	タイムフレーム	
1 1 1 1、1 1 1 2、1 7 0 1	O S 時刻の差分	
1 2 0 2	ロード回路	
1 2 1 0、1 3 0 9	統計値計算回路	
1 8 0 1	C P U	
1 8 0 2	メモリ	
1 8 0 3	インタフェース	

【 図 面 】 30

【 図 1 】 【 図 2 】

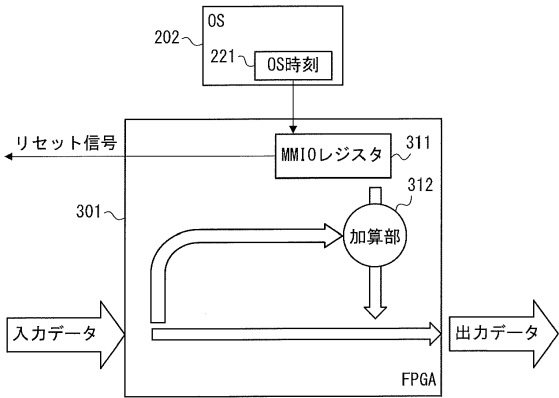
従来の情報処理システムの構成図

FPGAの第1の比較例を示す図



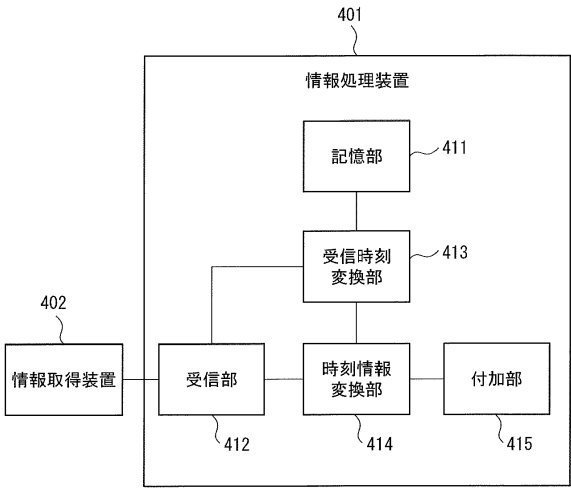
【図 3】

FPGAの第2の比較例を示す図



【図 4】

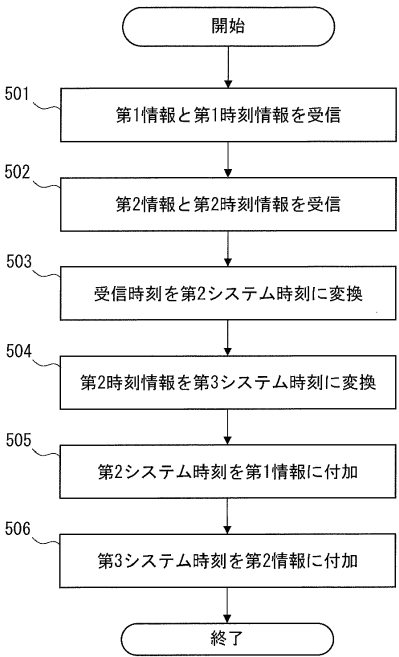
情報処理装置の機能的構成図



10

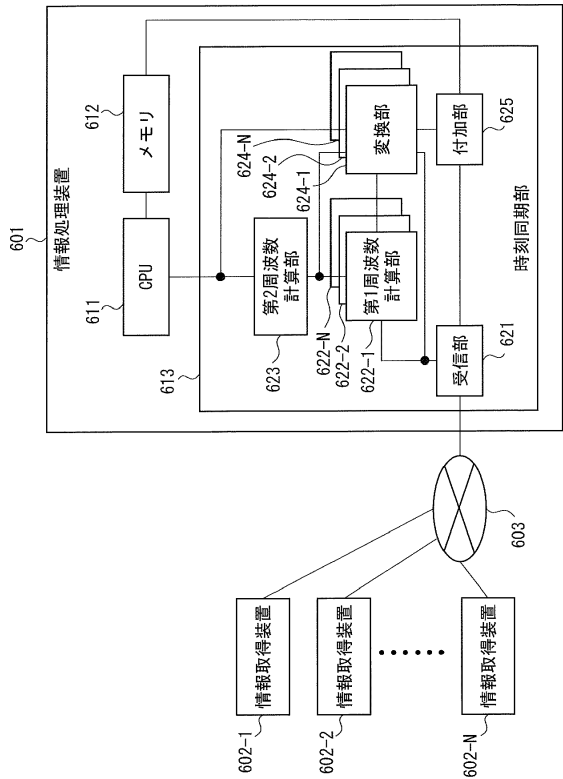
【図 5】

時刻同期処理のフローチャート



【図 6】

実施形態の情報処理システムの構成図



20

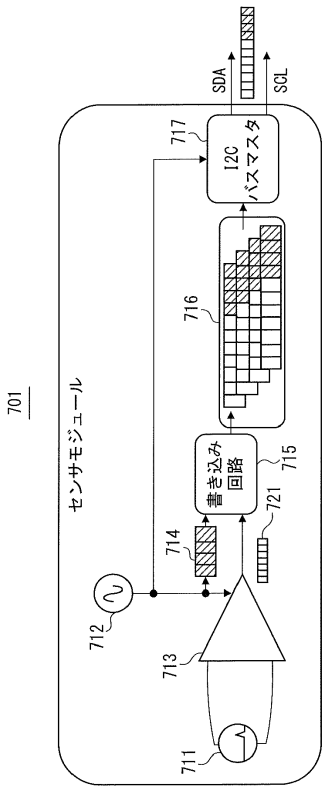
30

40

50

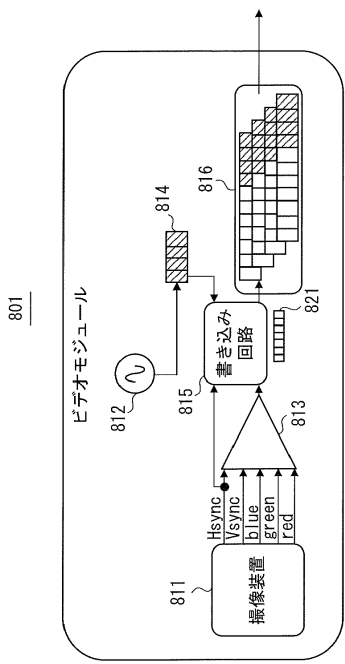
【図 7】

センサモジュールの構成図



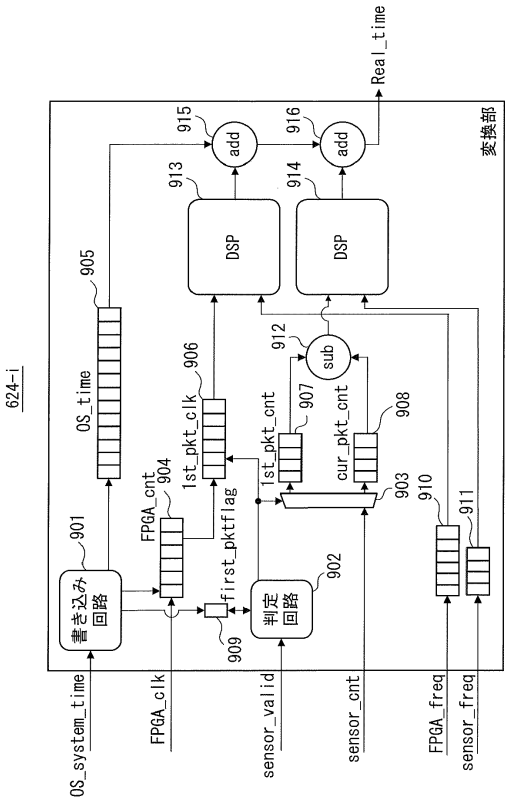
【図 8】

ビデオモジュールの構成図



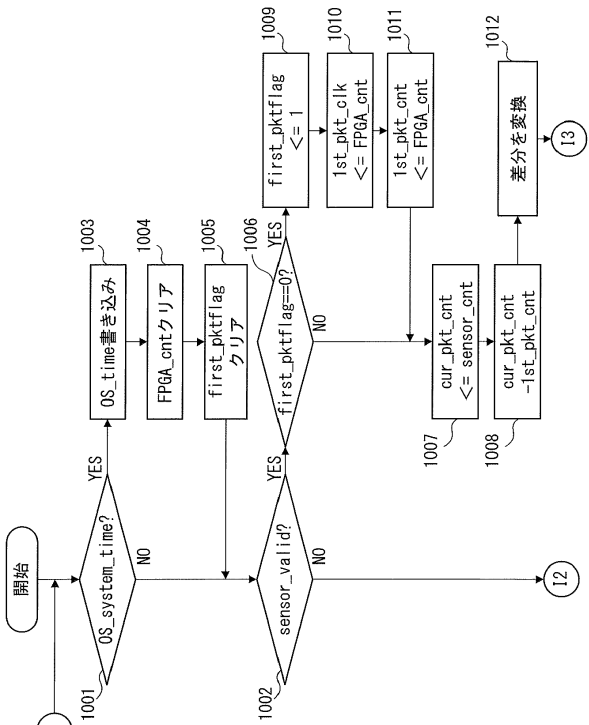
【図 9】

変換部の構成図



【図 10 A】

時刻同期処理の具体例を示すフローチャート(その1)



10

20

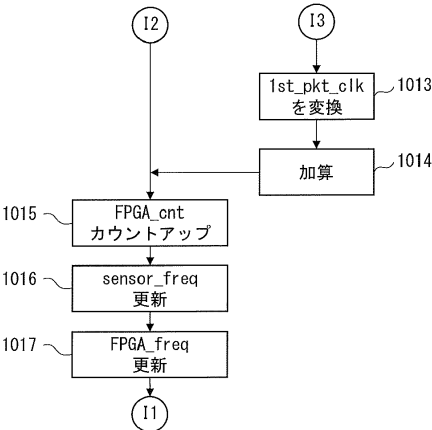
30

40

50

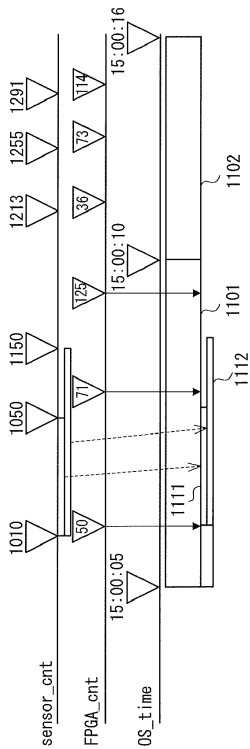
【図 1 0 B】

時刻同期処理の具体例を示すフローチャート(その2)



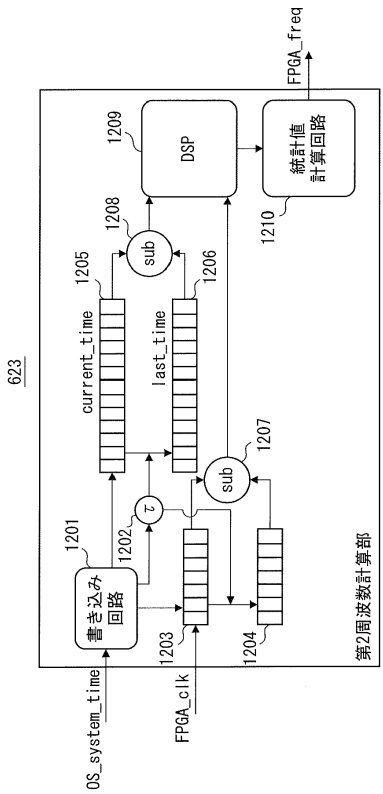
【図 1 1】

時刻同期処理を示す図



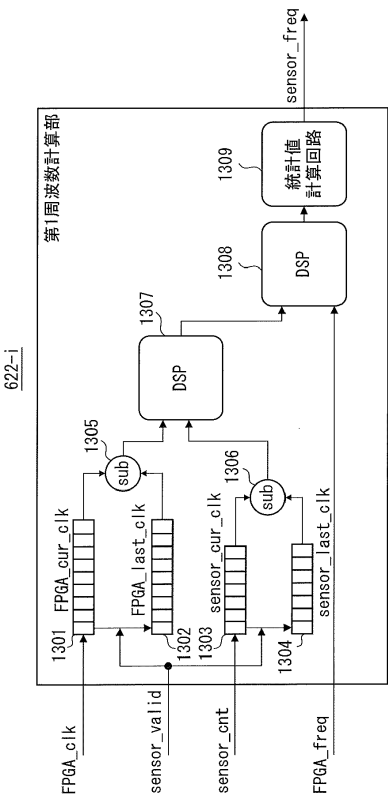
【図 1 2】

第2周波数計算部の構成図



【図 1 3】

第1周波数計算部の構成図



【図 1 4】

クロック信号の周波数を示す図

OS_time	Sub出力	FPGA_freq
15:00:05	11000	
15:00:10	10000	$10000 / (10 - 5) = 2000$
15:00:16	11000	$11000 / (16 - 10) = 1833$

【図 1 5】

タイミング信号の周波数を示す図

sensor_cnt	FPGA_cur_clk	FPGA_freq	sensor_freq
1010	2050	2000	
1050	2071	2000	$\{ (1050 - 1010) / (2071 - 2050) \} * 2000 = 3810$
1150	2125	2000	$\{ (1150 - 1050) / (2125 - 2071) \} * 2000 = 3704$

10

20

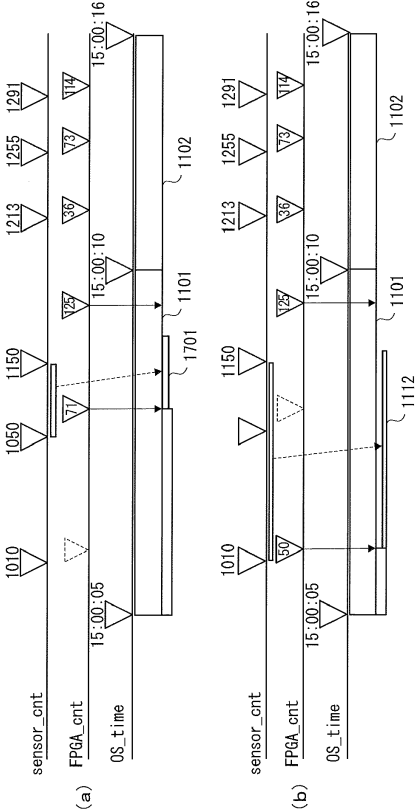
【図 1 6】

OS時刻を示す図

OS_time	sensor_cnt	1st_pkt_clk	first_pktflag	FPGA_freq	sensor_freq	Real_time
15:00:05	—	0	0	2000	4000	
	1010	50	1	2000	4000	$50 / 2000 + 15:00:05 = 15:00:05.025$
	1050	50	1	2000	3810	$(1050 - 1010) / 3810 + 50 / 2000 + 15:00:05 = 15:00:05.035$
	1150	50	1	2000	3704	$(1150 - 1010) / 3740 + 50 / 2000 + 15:00:05 = 15:00:05.063$

【図 1 7】

パケット消失時の時刻同期処理を示す図



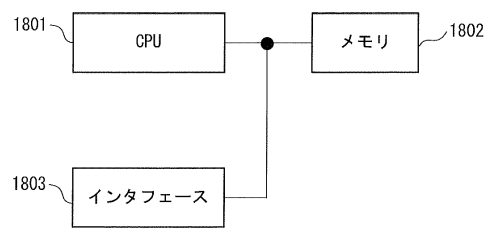
30

40

50

【図 18】

ソフトウェア制御による時刻同期部の構成図



10

20

30

40

50

フロントページの続き

- (72)発明者 横山 乾
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 村川 博
大阪府大阪市中央区城見二丁目2番53号 富士通関西中部ネットテック株式会社内
- 審査官 榮永 雅夫
- (56)参考文献 特開2008-209995(JP,A)
- (58)調査した分野 (Int.Cl., DB名)
- | | |
|---------|-----------------------|
| G 0 4 G | 5 / 0 0 |
| G 0 4 G | 7 / 0 0 |
| H 0 4 Q | 9 / 0 0 - 9 / 1 6 |
| H 0 4 M | 1 1 / 0 0 - 1 1 / 1 0 |