

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4586526号  
(P4586526)

(45) 発行日 平成22年11月24日(2010.11.24)

(24) 登録日 平成22年9月17日(2010.9.17)

(51) Int.Cl. F I  
G O 6 F 9/50 (2006.01) G O 6 F 9/46 4 6 2 Z

請求項の数 11 (全 105 頁)

(21) 出願番号	特願2004-369157 (P2004-369157)	(73) 特許権者	000002185
(22) 出願日	平成16年12月21日(2004.12.21)		ソニー株式会社
(65) 公開番号	特開2005-276164 (P2005-276164A)		東京都港区港南1丁目7番1号
(43) 公開日	平成17年10月6日(2005.10.6)	(74) 代理人	100093241
審査請求日	平成19年9月12日(2007.9.12)		弁理士 官田 正昭
(31) 優先権主張番号	特願2004-54044 (P2004-54044)	(74) 代理人	100101801
(32) 優先日	平成16年2月27日(2004.2.27)		弁理士 山田 英治
(33) 優先権主張国	日本国(JP)	(74) 代理人	100095496
			弁理士 佐々木 榮二
		(74) 代理人	100086531
			弁理士 澤田 俊夫
		(74) 代理人	110000763
			特許業務法人大同特許事務所

最終頁に続く

(54) 【発明の名称】 情報処理装置、情報処理方法、情報処理システムおよび情報処理用プログラム

(57) 【特許請求の範囲】

【請求項1】

ネットワークに接続される情報処理装置であって、  
プログラムを実行する1つまたは複数のプロセッサ手段と、このプロセッサ手段を管理する管理手段とを備え、

前記管理手段は、前記プロセッサ手段を特定するためのプロセッサ識別子、および、このプロセッサ識別子によって特定されるプロセッサ手段がプログラムの実行中または実行予約中であるか否かを示す使用状況情報、プログラムを特定するためのプログラム識別子、ネットワークを介して当該情報処理装置に接続された他の情報処理装置からの確保要求に応じて確保したプロセッサ手段に対して発番したロックシーケンス番号を含む管理テーブルを備え、

前記管理手段は、前記確保要求に応じて前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保したとき、前記プロセッサ手段の確保に応じて前記管理テーブルの書き替えを行い、該書き替え後の前記管理テーブル上の情報を前記他の情報処理装置に返信し、前記管理テーブル上のプログラム識別子とロックシーケンス番号が一致する前記他の情報処理装置からのプログラムを、該一致するプログラム識別子とロックシーケンス番号に対応して確保されているプロセッサ手段に送信することを特徴とする情報処理装置。

【請求項2】

請求項1に記載の情報処理装置において、

前記管理手段は、前記他の情報処理装置からの確保要求に応じて、前記プロセッサ手段中の、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段を、前記他の情報処理装置のために確保することを特徴とする情報処理装置。

【請求項 3】

請求項 1 に記載の情報処理装置において、

前記管理手段は、前記他の情報処理装置からの問い合わせに対して、前記管理テーブル上の情報を前記他の情報処理装置に返信し、その後の前記他の情報処理装置からの確保要求に応じて、前記プロセッサ手段中の、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段を、前記他の情報処理装置のために確保することを特徴とする情報処理装置。

10

【請求項 4】

請求項 1 に記載の情報処理装置において、

前記管理手段は、前記他の情報処理装置からの確保要求に対して、前記プロセッサ手段中に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段が、前記他の情報処理装置から要求された数以上存在するときには、そのプロセッサ手段を前記他の情報処理装置のために確保し、前記プロセッサ手段中に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段が、前記他の情報処理装置から要求された数未満しか存在しないときには、前記他の情報処理装置からの確保要求で示されたプログラム優先度より低いプログラム優先度のプログラムの実行中または実行予約中のプロセッサ手段を、前記他の情報処理装置のために確保することを特徴とする情報処理装置。

20

【請求項 5】

請求項 1 に記載の情報処理装置において、

前記管理手段は、前記他の情報処理装置からの問い合わせに対して、前記管理テーブル上の情報を前記他の情報処理装置に返信し、その後の前記他の情報処理装置からの確保要求に対して、前記プロセッサ手段中に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段が、前記他の情報処理装置から要求された数以上存在するときには、そのプロセッサ手段を前記他の情報処理装置のために確保し、前記プロセッサ手段中に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段が、前記他の情報処理装置から要求された数未満しか存在しないときには、前記他の情報処理装置からの確保要求で示されたプログラム優先度より低いプログラム優先度のプログラムの実行中または実行予約中のプロセッサ手段を、前記他の情報処理装置のために確保することを特徴とする情報処理装置。

30

【請求項 6】

同一のネットワーク上に接続される複数の情報処理装置からなるシステムであって、

前記複数の情報処理装置は、それぞれ、プログラムを実行する 1 つまたは複数のプロセッサ手段と、このプロセッサ手段を管理する管理手段とを備え、

前記管理手段は、それぞれ、自情報処理装置内の前記プロセッサ手段を特定するためのプロセッサ識別子、および、このプロセッサ識別子によって特定されるプロセッサ手段がプログラムの実行中または実行予約中であるか否かを示す使用状況情報、プログラムを特定するためのプログラム識別子、ネットワークを介して当該情報処理装置に接続された他の情報処理装置からの確保要求に応じて確保したプロセッサ手段に対して発番したロックシーケンス番号を含む管理テーブルを備え、

40

前記管理手段は、前記確保要求に応じて自情報処理装置内の前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保したとき、前記プロセッサ手段の確保に応じて前記管理テーブルの書き替えを行い、該書き替え後の前記管理テーブル上の情報を前記他の情報処理装置に返信し、前記管理テーブル上のプログラム識別子とロックシーケンス番号が一致する前記他の情報処理装置からのプログラムを、該一致するプログラム識別子とロックシーケンス番号に対応して確保されているプロセッサ手段に送信することを特徴とする情報処理システム。

【請求項 7】

50

それぞれ、プログラムを実行する1つまたは複数のプロセッサ手段と、このプロセッサ手段を管理する管理手段とを備える複数の情報処理装置が、同一のネットワーク上に接続されたネットワークシステムにおける情報処理方法であって、

前記管理手段は、それぞれ、自情報処理装置内の前記プロセッサ手段を特定するためのプロセッサ識別子、および、このプロセッサ識別子によって特定されるプロセッサ手段がプログラムの実行中または実行予約中であるか否かを示す使用状況情報、プログラムを特定するためのプログラム識別子、ネットワークを介して当該情報処理装置に接続された他の情報処理装置からの確保要求に応じて確保したプロセッサ手段に対して発番したロックシーケンス番号を含む管理テーブルを備え、

一の情報処理装置内の前記管理手段が、自情報処理装置内の前記管理テーブルを参照して、あるプログラムの実行用に自情報処理装置内でプロセッサ手段を確保できるか否かを判断する第1工程と、

前記一の情報処理装置内の前記管理手段が、前記第1工程で自情報処理装置内でプロセッサ手段を確保できると判断したとき、自情報処理装置内にプロセッサ手段を確保して、前記プロセッサ手段の確保に応じて自情報処理装置内の前記管理テーブルを書き替える第2工程と、

前記一の情報処理装置内の前記管理手段が、前記第1工程で自情報処理装置内でプロセッサ手段を確保できないと判断したとき、情報処理装置内の前記管理手段に対してプロセッサ手段の確保を要求する第3工程と、

この確保要求に応じて、前記他の情報処理装置内の前記管理手段が、自情報処理装置内にプロセッサ手段を確保して、前記プロセッサ手段の確保に応じて自情報処理装置内の前記管理テーブルの書き替えを行い、該書き替え後の前記管理テーブル上の情報を前記一の情報処理装置内の前記管理手段に通知する第4工程と、

前記管理テーブル上のプログラム識別子とロックシーケンス番号が一致する前記一の情報処理装置からのプログラムを、該一致するプログラム識別子とロックシーケンス番号に対応して確保されている前記他の情報処理装置内のプロセッサ手段に送信する第5工程と

を備えることを特徴とする情報処理方法。

#### 【請求項8】

請求項7に記載の情報処理方法において、

前記第2工程では、前記一の情報処理装置内の前記管理手段は、第1次的に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段を、前記あるプログラムの実行用に確保し、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段だけでは必要数のプロセッサ手段を確保できないときには、前記あるプログラムより優先度の低いプログラムの実行中または実行予約中のプロセッサ手段を、前記あるプログラムの実行用に確保することを特徴とする情報処理方法。

#### 【請求項9】

請求項7に記載の情報処理方法において、

前記第4工程では、前記他の情報処理装置内の前記管理手段は、第1次的に、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段を、前記あるプログラムの実行用に確保し、プログラムの実行中ではなく、実行予約中でもないプロセッサ手段が、前記一の情報処理装置から要求された数未満しか存在しないときには、前記あるプログラムより優先度の低いプログラムの実行中または実行予約中のプロセッサ手段を、前記あるプログラムの実行用に確保することを特徴とする情報処理方法。

#### 【請求項10】

請求項7に記載の情報処理方法において、

前記第3工程では、前記一の情報処理装置内の前記管理手段は、プロセッサ手段の確保要求先の他の情報処理装置として特定の種別の情報処理装置を優先的に選択することを特徴とする情報処理方法。

#### 【請求項11】

10

20

30

40

50

プログラムを実行する1つまたは複数のプロセッサ手段を備え、ネットワークに接続される情報処理装置において、前記プロセッサ手段の一部または全部を、ネットワークを介して当該情報処理装置に接続された他の情報処理装置のために確保するために、コンピュータを、

前記プロセッサ手段を特定するためのプロセッサ識別子、および、このプロセッサ識別子によって特定されるプロセッサ手段がプログラムの実行中または実行予約中であるか否かを示す使用状況情報、プログラムを特定するためのプログラム識別子、ネットワークを介して当該情報処理装置に接続された他の情報処理装置からの確保要求に応じて確保したプロセッサ手段に対して発番したロックシーケンス番号を含む管理テーブルを生成する手段、

10

前記他の情報処理装置からの確保要求に対して、前記管理テーブルを参照して、前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保できるか否かを判断する手段、および、

前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保できるとき、前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保して、前記管理テーブルの書き替えを行い、該書き替え後の前記管理テーブル上の情報を前記他の情報処理装置に返信し、前記管理テーブル上のプログラム識別子とロックシーケンス番号が一致する前記他の情報処理装置からのプログラムを、該一致するプログラム識別子とロックシーケンス番号に対応して確保されているプロセッサ手段に送信する手段、

として機能させるための情報処理用プログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、情報処理装置、情報処理方法、情報処理システムおよび情報処理用プログラムに関する。

【背景技術】

【0002】

最近、グリッド・コンピューティングが注目されている。グリッド・コンピューティングとは、ネットワークに接続された複数の情報処理装置を協調動作させて、高い演算性能を実現する技術である。

30

【0003】

例えば、特許文献1（特開2002-342165号公報）、特許文献2（特開2002-351850号公報）、特許文献3（特開2002-358289号公報）、特許文献4（特開2002-366533号公報）および特許文献5（特開2002-366534号公報）には、均一なモジュラー構造、共通のコンピューティング・モジュール、および均一なソフトウェアセルを用いることによって、高速処理用コンピュータ・アーキテクチャを実現することが示されている。

【0004】

また、特許文献6（米国特許6,587,906）、特許文献7（米国特許6,667,920）、特許文献8（米国特許6,728,845）、特許文献9（米国特開2004-0039895）、特許文献10（米国特開2004-0054880）および特許文献11（米国特開2004-0098496）には、情報処理装置内に複数個存在するプロセッサを独立・並列に動作させることによって、処理を高速化させることが示されている。

40

【0005】

上に挙げた先行技術文献は、以下の通りである。

【特許文献1】特開2002-342165号公報

【特許文献2】特開2002-351850号公報

【特許文献3】特開2002-358289号公報

【特許文献4】特開2002-366533号公報

50

【特許文献 5】特開 2 0 0 2 - 3 6 6 5 3 4 号公報  
 【特許文献 6】米国特許 6 , 5 8 7 , 9 0 6  
 【特許文献 7】米国特許 6 , 6 6 7 , 9 2 0  
 【特許文献 8】米国特許 6 , 7 2 8 , 8 4 5  
 【特許文献 9】米国特開 2 0 0 4 - 0 0 3 9 8 9 5  
 【特許文献 1 0】米国特開 2 0 0 4 - 0 0 5 4 8 8 0  
 【特許文献 1 1】米国特開 2 0 0 4 - 0 0 9 8 4 9 6

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

しかしながら、ネットワークを介して複数の情報処理装置が接続されたネットワークシステムにおいて、ユーザの当該ネットワークシステムに対する操作に応じた処理を、ネットワークシステム全体で確実に、しかも現に実行中または実行予約中のプログラムの実行をできるだけ妨げないように、分散的に実行するには、それぞれの情報処理装置、またはネットワークシステムを管理する情報処理装置が、自装置内および他装置内のプロセッサ手段の使用状況（プログラム実行状況）を把握し、処理に必要なプロセッサ手段を自装置内または他装置内に確保する必要がある。

10

【 0 0 0 7 】

そこで、この発明は、各情報処理装置内のプロセッサ手段に対して的確にプログラムの実行を割り当てることができ、ネットワークシステム全体で確実に、しかも現に実行中または実行予約中のプログラムの実行をできるだけ妨げないように、分散処理を実行することができるようにしたものである。

20

【課題を解決するための手段】

【 0 0 0 8 】

この発明の情報処理装置は、  
 ネットワークに接続される情報処理装置であって、  
 プログラムを実行する 1 つまたは複数のプロセッサ手段と、このプロセッサ手段を管理する管理手段とを備え、

前記管理手段は、前記プロセッサ手段を特定するためのプロセッサ識別子、および、このプロセッサ識別子によって特定されるプロセッサ手段がプログラムの実行中または実行予約中であるか否かを示す使用状況情報、プログラムを特定するためのプログラム識別子、ネットワークを介して当該情報処理装置に接続された他の情報処理装置からの確保要求に応じて確保したプロセッサ手段に対して発番したロックシーケンス番号を含む管理テーブルを備え、

30

前記管理手段は、前記確保要求に応じて前記プロセッサ手段の一部または全部を前記他の情報処理装置のために確保したとき、前記プロセッサ手段の確保に応じて前記管理テーブルの書き替えを行い、該書き替え後の前記管理テーブル上の情報を前記他の情報処理装置に返信し、前記管理テーブル上のプログラム識別子とロックシーケンス番号が一致する前記他の情報処理装置からのプログラムを、該一致するプログラム識別子とロックシーケンス番号に対応して確保されているプロセッサ手段に送信するものである。

40

【 0 0 0 9 】

このような構成の、この発明の情報処理装置を複数、同一のネットワーク上に接続して、ネットワークシステムを構成すれば、各情報処理装置内のプロセッサ手段に対して的確にプログラムの実行を割り当てることができ、ネットワークシステム全体で確実に、しかも現に実行中または実行予約中のプログラムの実行をできるだけ妨げないように、分散処理を実行することができる。

【 0 0 1 0 】

また、このネットワークシステムにおいて、ある情報処理装置が当該ネットワークから切断され、かつ、その切断された情報処理装置内のプロセッサ手段が、あるプログラムの実行中または実行予約中であつた場合に、切断された情報処理装置内のプロセッサ手段に

50

代わって、切断されていない情報処理装置内のプロセッサ手段が、そのプログラムを実行する場合にも、切断されていない各情報処理装置内のプロセッサ手段に対して的確にプログラムの実行を割り当てることができ、ネットワークシステム全体で確実に、しかも現に実行中または実行予約中のプログラムの実行をできるだけ妨げないように、分散処理を実行することができる。

【発明の効果】

【0011】

以上のように、この発明によれば、各情報処理装置内のプロセッサ手段に対して的確にプログラムの実行を割り当てることができ、ネットワークシステム全体で確実に、しかも現に実行中または実行予約中のプログラムの実行をできるだけ妨げないように、分散処理を実行することができる。

10

【発明を実施するための最良の形態】

【0012】

[1. ネットワークシステムおよび情報処理装置の基本的構成その1]

図1は、この発明のネットワークシステムの一例を示し、ネットワーク9を介して複数の情報処理装置1, 2, 3および4が接続されたものである。

【0013】

(1-1. 情報処理装置および情報処理コントローラ)

情報処理装置1, 2, 3および4は、それぞれ後述のような各種のAV (Audio and Visual) 機器やポータブル機器である。

20

【0014】

情報処理装置1について示すと、情報処理装置1は、コンピュータ機能部として情報処理コントローラ11を備える。情報処理コントローラ11は、メインプロセッサ21-1、サブプロセッサ23-1, 23-2および23-3、DMAC (ダイレクトメモリアクセスコントローラ) 25-1、およびDC (ディスクコントローラ) 27-1を有する。

【0015】

メインプロセッサ21-1は、サブプロセッサ23-1, 23-2および23-3によるサブプロセッサプログラムの実行(データ処理)のスケジュール管理と、情報処理コントローラ11(情報処理装置1)の全般的な管理とを行う。ただし、メインプロセッサ21-1内で管理のためのプログラム以外のプログラムが動作するように構成することもできる。その場合には、メインプロセッサ21-1はサブプロセッサとしても機能することになる。メインプロセッサ21-1は、LS (ローカルストレージ) 22-1を有する。

30

【0016】

サブプロセッサは、1つでもよいが、望ましくは複数とする。この例は、複数の場合である。

【0017】

各サブプロセッサ23-1, 23-2および23-3は、メインプロセッサ21-1の制御によって並列的かつ独立に、サブプロセッサプログラムを実行し、データを処理する。さらに、場合によってメインプロセッサ21-1内のプログラムがサブプロセッサ23-1, 23-2または23-3内のサブプロセッサプログラムと連携して動作するように構成することもできる。後述する機能プログラムもメインプロセッサ21-1内で動作するプログラムである。各サブプロセッサ23-1, 23-2および23-3も、LS (ローカルストレージ) 24-1, 24-2および24-3を有する。

40

【0018】

DMAC 25-1は、情報処理コントローラ11に接続されたDRAM (ダイナミックRAM) などからなるメインメモリ26-1に格納されているプログラムおよびデータにアクセスするものであり、DC 27-1は、情報処理コントローラ11に接続された外部記録部28-1および28-2にアクセスするものである。

【0019】

外部記録部28-1および28-2は、固定ディスク(ハードディスク)でも、リムー

50

バブルディスクでもよく、また、MO、CD±RW、DVD±RWなどの光ディスク、メモリディスク、SRAM（スタティックRAM）、ROMなど、各種のものを用いることができる。したがって、DC27-1は、ディスクコントローラと称するが、外部記録部コントローラである。

【0020】

図1の例のように、情報処理コントローラ11に対して外部記録部28を複数接続できるように、情報処理コントローラ11を構成することができる。

【0021】

メインプロセッサ21-1、各サブプロセッサ23-1、23-2および23-3、DMAC25-1、およびDC27-1は、バス29-1によって接続される。

10

【0022】

情報処理コントローラ11には、当該の情報処理コントローラ11を備える情報処理装置1をネットワーク全体を通して一意的に識別できる識別子が、情報処理装置IDとして割り当てられる。

【0023】

メインプロセッサ21-1、および各サブプロセッサ23-1、23-2および23-3に対しても同様に、それぞれを特定できる識別子が、メインプロセッサIDおよびサブプロセッサIDとして割り当てられる。

【0024】

情報処理コントローラ11は、ワンチップIC（集積回路）として構成することが望ましい。

20

【0025】

他の情報処理装置2、3および4も、同様に構成される。ここで、親番号が同一であるユニットは、枝番号が異なっても、特に断りがない限り、同じ働きをするものとする。また、以下の説明において枝番号が省略されている場合には、枝番号の違いに在る差異を生じないものとする。

【0026】

（1-2、各サブプロセッサからメインメモリへのアクセス）

上述したように、1つの情報処理コントローラ内の各サブプロセッサ23は、独立にサブプロセッサプログラムを実行し、データを処理するが、異なるサブプロセッサがメインメモリ26内の同一領域に対して同時に読み出しまたは書き込みを行った場合には、データの不整合を生じ得る。そこで、サブプロセッサ23からメインメモリ26へのアクセスは、以下のような手順によって行う。

30

【0027】

図2（A）に示すように、メインメモリ26は、アドレス指定可能な複数のメモリロケーション0～Mによって構成される。各メモリロケーションに対しては、データの状態を示す情報を格納するための追加セグメント0～Mが割り振られる。追加セグメントは、F/Eビット、サブプロセッサIDおよびLSアドレス（ローカルストレージアドレス）を含むものとされる。また、各メモリロケーションには、後述のアクセスキー0～Mも割り振られる。F/Eビットは、以下のように定義される。

40

【0028】

F/Eビット=0は、サブプロセッサ23によって読み出されている処理中のデータ、または空き状態であるため最新データではない無効データであり、読み出し不可であることを示す。また、F/Eビット=0は、当該メモリロケーションにデータ書き込み可能であることを示し、書き込み後に1に設定される。

【0029】

F/Eビット=1は、当該メモリロケーションのデータがサブプロセッサ23によって読み出されておらず、未処理の最新データであることを示す。当該メモリロケーションのデータは読み出し可能であり、サブプロセッサ23によって読み出された後に0に設定される。また、F/Eビット=1は、当該メモリロケーションがデータ書き込み不可である

50

ことを示す。

【 0 0 3 0 】

さらに、上記 F / E ビット = 0 (読み出し不可 / 書き込み可) の状態において、当該メモリロケーションについて読み出し予約を設定することは可能である。F / E ビット = 0 のメモリロケーションに対して読み出し予約を行う場合には、サブプロセッサ 2 3 は、読み出し予約を行うメモリロケーションの追加セグメントに、読み出し予約情報として当該サブプロセッサ 2 3 のサブプロセッサ ID および L S アドレスを書き込む。

【 0 0 3 1 】

その後、データ書き込み側のサブプロセッサ 2 3 によって、読み出し予約されたメモリロケーションにデータが書き込まれ、F / E ビット = 1 (読み出し可 / 書き込み不可) に設定されたとき、あらかじめ読み出し予約情報として追加セグメントに書き込まれたサブプロセッサ ID および L S アドレスに読み出される。

10

【 0 0 3 2 】

複数のサブプロセッサによってデータを多段階に処理する必要がある場合、このように各メモリロケーションのデータの読み出し / 書き込みを制御することによって、前段階の処理を行うサブプロセッサ 2 3 が、処理済みのデータをメインメモリ 2 6 内の所定アドレスに書き込んだ後に即座に、後段階の処理を行う別のサブプロセッサ 2 3 が前処理後のデータを読み出すことが可能となる。

【 0 0 3 3 】

図 2 ( B ) に示すように、各サブプロセッサ 2 3 内の L S 2 4 も、アドレス指定可能な複数のメモリロケーション 0 ~ L によって構成される。各メモリロケーションに対しては、同様に追加セグメント 0 ~ L が割り振られる。追加セグメントは、ビジービットを含むものとされる。

20

【 0 0 3 4 】

サブプロセッサ 2 3 がメインメモリ 2 6 内のデータを自身の L S 2 4 のメモリロケーションに読み出すときには、読み出し先のメモリロケーションに対応するビジービットを 1 に設定して予約する。ビジービットが 1 であるメモリロケーションには、他のデータは格納することができない。L S 2 4 のメモリロケーションに読み出し後、ビジービットは 0 になり、任意の目的に使用できるようになる。

【 0 0 3 5 】

図 2 ( A ) に示すように、さらに、各情報処理コントローラと接続されたメインメモリ 2 6 には、複数のサンドボックスが含まれる。サンドボックスは、メインメモリ 2 6 内の領域を画定するものであり、各サンドボックスは、各サブプロセッサ 2 3 に割り当てられ、そのサブプロセッサが排他的に使用することができる。すなわち、各々のサブプロセッサ 2 3 は、自身に割り当てられたサンドボックスを使用できるが、この領域を超えてデータのアクセスを行うことはできない。

30

【 0 0 3 6 】

メインメモリ 2 6 は、複数のメモリロケーション 0 ~ M から構成されるが、サンドボックスは、これらのメモリロケーションの集合である。すなわち、1つのサンドボックスは、1つまたは複数のメモリロケーションから構成される。

40

【 0 0 3 7 】

さらに、メインメモリ 2 6 の排他的な制御を実現するために、図 2 ( C ) に示すようなキー管理テーブルが用いられる。キー管理テーブルは、情報処理コントローラ内の S R A M のような比較的高速のメモリに格納され、D M A C 2 5 と関連付けられる。

【 0 0 3 8 】

キー管理テーブル内には、情報処理コントローラ内のサブプロセッサの数と同数のエントリが存在し、各エントリには、サブプロセッサ ID および、それに対応するサブプロセッサキーおよびキーマスクが関連付けられて格納される。

【 0 0 3 9 】

サブプロセッサ 2 3 がメインメモリ 2 6 を使用する際のプロセスは、以下のとおりであ

50

る。まず、サブプロセッサ 23 は D M A C 25 に、読み出しまたは書き込みのコマンドを送信する。このコマンドには、自身のサブプロセッサ I D と、アクセス要求先であるメインメモリ 26 のアドレスが含まれる。

【 0 0 4 0 】

D M A C 25 は、このコマンドを実行する前に、キー管理テーブルを参照して、アクセス要求元のサブプロセッサ 23 のサブプロセッサキーを調べる。次に、D M A C 25 は、調べたアクセス要求元のサブプロセッサキーと、アクセス要求先であるメインメモリ 26 内の図 2 ( A ) に示したメモリロケーションに割り振られたアクセスキーとを比較して、2 つのキーが一致した場合にのみ、上記のコマンドを実行する。

【 0 0 4 1 】

図 2 ( C ) に示したキー管理テーブル上のキーマスクは、その任意のビットが 1 になることによって、そのキーマスクに関連付けられたサブプロセッサキーの対応するビットが 0 または 1 になることができる。

【 0 0 4 2 】

例えば、サブプロセッサキーが 1 0 1 0 であるとする。通常、このサブプロセッサキーによって 1 0 1 0 のアクセスキーを持つサンドボックスへのアクセスだけが可能になる。しかし、このサブプロセッサキーと関連付けられたキーマスクが 0 0 0 1 に設定されている場合には、キーマスクのビットが 1 に設定された桁のみにつき、サブプロセッサキーとアクセスキーとの一致判定がマスクされ、このサブプロセッサキー 1 0 1 0 によって、アクセスキーが 1 0 1 0 または 1 0 1 1 のいずれかであるアクセスキーを持つサンドボックスへのアクセスが可能となる。

【 0 0 4 3 】

以上のようにして、メインメモリ 26 のサンドボックスの排他性が実現される。すなわち、1 つの情報処理コントローラ内の複数のサブプロセッサによってデータを多段階に処理する必要がある場合、以上のように構成することによって、前段階の処理を行うサブプロセッサと、後段階の処理を行うサブプロセッサのみが、メインメモリ 26 の所定アドレスにアクセスできるようになり、データを保護することができる。

【 0 0 4 4 】

例えば、以下のように使用することが考えられる。まず、情報処理装置の起動直後においては、キーマスクの値は全てゼロである。メインプロセッサ内のプログラムが実行され、サブプロセッサ内のサブプロセッサプログラムと連携動作するものとする。第 1 のサブプロセッサにより実行された処理結果データを一旦、メインメモリ 26 に格納し、第 2 のサブプロセッサに送信したいときには、該当するメインメモリ領域は、当然どちらのサブプロセッサからもアクセス可能である必要がある。そのような場合に、メインプロセッサ内のプログラムは、キーマスクの値を適切に変更し、複数のサブプロセッサからアクセスできるメインメモリ領域を設けることにより、サブプロセッサによる多段階的の処理を可能にする。

【 0 0 4 5 】

より具体的には、他の情報処理装置からのデータ 第 1 のサブプロセッサによる処理 第 1 のメインメモリ領域 第 2 のサブプロセッサによる処理 第 2 のメインメモリ領域、という手順で多段階処理が行われるときには、

第 1 のサブプロセッサのサブプロセッサキー : 0 1 0 0、

第 1 のメインメモリ領域のアクセスキー : 0 1 0 0、

第 2 のサブプロセッサのサブプロセッサキー : 0 1 0 1、

第 2 のメインメモリ領域のアクセスキー : 0 1 0 1、

のような設定のままだと、第 2 のサブプロセッサは第 1 のメインメモリ領域にアクセスすることができない。そこで、第 2 のサブプロセッサのキーマスクを 0 0 0 1 にすることにより、第 2 のサブプロセッサによる第 1 のメインメモリ領域へのアクセスを可能にすることができる。

【 0 0 4 6 】

10

20

30

40

50

( 1 - 3 . ソフトウェアセルの生成および構成 )

図 1 のネットワークシステムでは、情報処理装置 1 , 2 , 3 および 4 の間での分散処理のために、情報処理装置 1 , 2 , 3 および 4 の間でソフトウェアセルが伝送される。すなわち、ある情報処理装置内の情報処理コントローラに含まれるメインプロセッサ 2 1 は、コマンド、プログラムおよびデータを含むソフトウェアセルを生成し、ネットワーク 9 を介して他の情報処理装置に送信することによって、処理を分散することができる。

【 0 0 4 7 】

図 3 に、ソフトウェアセルの構成の一例を示す。この例のソフトウェアセルは、全体として、送信元 I D、送信先 I D、応答先 I D、セルインターフェース、DMA コマンド、プログラム、およびデータによって構成される。

【 0 0 4 8 】

送信元 I D には、ソフトウェアセルの送信元である情報処理装置のネットワークアドレスおよび情報処理装置 I D、さらに、その情報処理装置内の情報処理コントローラが備えるメインプロセッサ 2 1 および各サブプロセッサ 2 3 の識別子 ( メインプロセッサ I D およびサブプロセッサ I D ) が含まれる。

【 0 0 4 9 】

送信先 I D および応答先 I D には、それぞれ、ソフトウェアセルの送信先である情報処理装置、およびソフトウェアセルの実行結果の応答先である情報処理装置についての、同じ情報が含まれる。

【 0 0 5 0 】

セルインターフェースは、ソフトウェアセルの利用に必要な情報であり、グローバル I D、必要なサブプロセッサの情報、サンドボックスサイズ、および前回のソフトウェアセル I D から構成される。

【 0 0 5 1 】

グローバル I D は、ネットワーク全体を通して当該のソフトウェアセルを一意的に識別できるものであり、送信元 I D、およびソフトウェアセルの作成または送信の日時 ( 日付および時刻 ) に基づいて作成される。

【 0 0 5 2 】

必要なサブプロセッサの情報は、当該ソフトウェアセルの実行に必要なサブプロセッサの数が設定される。サンドボックスサイズは、当該ソフトウェアセルの実行に必要なメインメモリ 2 6 内およびサブプロセッサ 2 3 の L S 2 4 内のメモリ量が設定される。

【 0 0 5 3 】

前回のソフトウェアセル I D は、ストリーミングデータなどのシーケンシャルな実行を要求する 1 グループのソフトウェアセル内の、前回のソフトウェアセルの識別子である。

【 0 0 5 4 】

ソフトウェアセルの実行セクションは、DMA コマンド、プログラムおよびデータから構成される。DMA コマンドには、プログラムの起動に必要な一連の DMA コマンドが含まれ、プログラムには、サブプロセッサ 2 3 によって実行されるサブプロセッサプログラムが含まれる。ここでのデータは、このサブプロセッサプログラムを含むプログラムによって処理されるデータである。

【 0 0 5 5 】

さらに、DMA コマンドには、ロードコマンド、キックコマンド、機能プログラム実行コマンド、ステータス要求コマンド、およびステータス返信コマンドが含まれる。

【 0 0 5 6 】

ロードコマンドは、メインメモリ 2 6 内の情報をサブプロセッサ 2 3 内の L S 2 4 にロードするコマンドであり、ロードコマンド自体のほかに、メインメモリアドレス、サブプロセッサ I D および L S アドレスを含む。メインメモリアドレスは、情報のロード元であるメインメモリ 2 6 内の所定領域のアドレスを示す。サブプロセッサ I D および L S アドレスは、情報のロード先であるサブプロセッサ 2 3 の識別子および L S 2 4 のアドレスを示す。

10

20

30

40

50

## 【 0 0 5 7 】

キックコマンドは、サブプロセッサプログラムの実行を開始するコマンドであり、キックコマンド自体のほかに、サブプロセッサIDおよびプログラムカウンタを含む。サブプロセッサIDは、キック対象のサブプロセッサ23を識別し、プログラムカウンタは、サブプロセッサプログラム実行用プログラムカウンタのためのアドレスを与える。

## 【 0 0 5 8 】

機能プログラム実行コマンドは、後述のように、ある情報処理装置が他の情報処理装置に対して、機能プログラムの実行を要求するコマンドである。機能プログラム実行コマンドを受信した情報処理装置内の情報処理コントローラは、後述の機能プログラムIDによって、起動すべき機能プログラムを識別する。

10

## 【 0 0 5 9 】

ステータス要求コマンドは、送信先IDで示される情報処理装置の現在の動作状態（状況）に関する装置情報を、応答先IDで示される情報処理装置宛に送信することを要求するコマンドである。機能プログラムについては後述するが、図6に示す情報処理装置のメインメモリ26が記憶するソフトウェアの構成図において機能プログラムにカテゴライズされるプログラムである。機能プログラムは、メインメモリ26にロードされ、メインプロセッサ21により実行される。

## 【 0 0 6 0 】

ステータス返信コマンドは、上記のステータス要求コマンドを受信した情報処理装置が、自身の装置情報を当該ステータス要求コマンドに含まれる応答先IDで示される情報処理装置に返信するコマンドである。ステータス返信コマンドは、実行セクションのデータ領域に装置情報を格納する。

20

## 【 0 0 6 1 】

図4に、DMAコマンドがステータス返信コマンドである場合におけるソフトウェアセルのデータ領域の構造を示す。

## 【 0 0 6 2 】

情報処理装置IDは、情報処理コントローラを備える情報処理装置を識別するための識別子であり、ステータス返信コマンドを送信する情報処理装置のIDを示す。情報処理装置IDは、電源投入時、その情報処理装置内の情報処理コントローラに含まれるメインプロセッサ21によって、電源投入時の日時、情報処理装置のネットワークアドレスおよび情報処理装置内の情報処理コントローラに含まれるサブプロセッサ23の数などに基づいて生成される。

30

## 【 0 0 6 3 】

情報処理装置種別IDには、当該の情報処理装置の特徴を表す値が含まれる。情報処理装置の特徴とは、例えば、後述のハードディスクレコーダ、PDA（Personal Digital Assistants）、ポータブルCD（Compact Disc）プレーヤなどである。また、情報処理装置種別IDは、映像音声記録、映像音声再生など、情報処理装置の機能を表すものであってもよい。情報処理装置の特徴や機能を表す値は予め決定されているものとし、情報処理装置種別IDを読み出すことにより、当該情報処理装置の特徴や機能を把握することが可能である。

40

## 【 0 0 6 4 】

MS（マスター/スレーブ）ステータスは、後述のように情報処理装置がマスター装置またはスレーブ装置のいずれで動作しているかを表すもので、これが0に設定されている場合にはマスター装置として動作していることを示し、1に設定されている場合にはスレーブ装置として動作していることを示す。

## 【 0 0 6 5 】

メインプロセッサ動作周波数は、情報処理コントローラ内のメインプロセッサ21の動作周波数を表す。メインプロセッサ使用率は、メインプロセッサ21で現在動作している全てのプログラムについての、メインプロセッサ21での使用率を表す。メインプロセッサ使用率は、対象メインプロセッサの全処理能力に対する使用中の処理能力の比率を表し

50

た値で、例えばプロセッサ処理能力評価のための単位であるMIPSを単位として算出され、または単位時間あたりのプロセッサ使用時間に基づいて算出される。後述のサブプロセッサ使用率についても同様である。

【0066】

サブプロセッサ数は、当該の情報処理コントローラが備えるサブプロセッサ23の数を表す。サブプロセッサIDは、当該の情報処理コントローラ内の各サブプロセッサ23を識別するための識別子である。

【0067】

サブプロセッサステータスは、各サブプロセッサ23の状態を表すものであり、`unused`、`reserved`、`busy`などの状態がある。`unused`は、当該のサブプロセッサが現在使用されてなく、使用の予約もされていないことを示す。`reserved`は、現在は使用されていないが、使用が予約されている状態を示す。`busy`は、現在使用中であることを示す。

10

【0068】

サブプロセッサ使用率は、当該のサブプロセッサで現在実行している、または当該のサブプロセッサに実行が予約されているサブプロセッサプログラムについての、当該サブプロセッサでの使用率を表す。すなわち、サブプロセッサ使用率は、サブプロセッサステータスが`busy`である場合には、現在の使用率を示し、サブプロセッサステータスが`reserved`である場合には、後に使用される予定の推定使用率を示す。

【0069】

20

サブプロセッサID、サブプロセッサステータスおよびサブプロセッサ使用率は、1つのサブプロセッサ23に対して一組設定され、1つの情報処理コントローラ内のサブプロセッサ23の数の組数だけ設定される。

【0070】

メインメモリ総容量およびメインメモリ使用量は、それぞれ、当該の情報処理コントローラに接続されているメインメモリ26の総容量および現在使用中の容量を表す。

【0071】

外部記録部数は、当該の情報処理コントローラに接続されている外部記録部28の数を表す。外部記録部IDは、当該の情報処理コントローラに接続されている外部記録部28を一意的に識別する情報である。外部記録部種別IDは、当該の外部記録部の種類（例えば、ハードディスク、`CD±RW`、`DVD±RW`、メモリディスク、`SRAM`、`ROM`など）を表す。

30

【0072】

外部記録部総容量および外部記録部使用量は、それぞれ、外部記録部IDによって識別される外部記録部28の総容量および現在使用中の容量を表す。

【0073】

外部記録部ID、外部記録部種別ID、外部記録部総容量および外部記録部使用量は、1つの外部記録部28に対して一組設定され、当該の情報処理コントローラに接続されている外部記録部28の数の組数だけ設定される。すなわち、1つの情報処理コントローラに複数の外部記録部が接続されている場合、各々の外部記録部には異なる外部記録部IDが割り当てられ、外部記録部種別ID、外部記録部総容量および外部記録部使用量も別々に管理される。

40

【0074】

(1-4. ソフトウェアセルの実行)

ある情報処理装置内の情報処理コントローラに含まれるメインプロセッサ21は、以上のような構成のソフトウェアセルを生成し、ネットワーク9を介して他の情報処理装置および当該装置内の情報処理コントローラに送信する。送信元の情報処理装置、送信先の情報処理装置、応答先の情報処理装置、および各装置内の情報処理コントローラは、それぞれ、上記の送信元ID、送信先IDおよび応答先IDによって識別される。

【0075】

50

ソフトウェアセルを受信した情報処理装置内の情報処理コントローラに含まれるメインプロセッサ21は、そのソフトウェアセルをメインメモリ26に格納する。さらに、送信先のメインプロセッサ21は、ソフトウェアセルを読み出し、それに含まれるDMAコマンドを処理する。

【0076】

具体的に、送信先のメインプロセッサ21は、まず、ロードコマンドを実行する。これによって、ロードコマンドで指示されたメインメモリアドレスから、ロードコマンドに含まれるサブプロセッサIDおよびLSアドレスで特定されるサブプロセッサ内のLS24の所定領域に、情報がロードされる。ここでロードされる情報は、受信したソフトウェアセルに含まれるサブプロセッサプログラムまたはデータ、あるいはその他の指示されたデータである。

10

【0077】

次に、メインプロセッサ21は、キックコマンドを、これに含まれるサブプロセッサIDで指示されたサブプロセッサに、同様にキックコマンドに含まれるプログラムカウンタと共に送信する。

【0078】

指示されたサブプロセッサは、そのキックコマンドおよびプログラムカウンタに従って、サブプロセッサプログラムを実行する。そして、実行結果をメインメモリ26に格納した後、実行を完了したことをメインプロセッサ21に通知する。

【0079】

20

なお、送信先の情報処理装置内の情報処理コントローラにおいてソフトウェアセルを実行するプロセッサはサブプロセッサ23に限定されるものではなく、メインプロセッサ21がソフトウェアセルに含まれる機能プログラムなどのメインメモリ用プログラムを実行するように指定することも可能である。

【0080】

この場合には、送信元の情報処理装置は、送信先の情報処理装置宛に、サブプロセッサプログラムの代わりに、メインメモリ用プログラムおよびそのメインメモリ用プログラムによって処理されるデータを含み、DMAコマンドがロードコマンドであるソフトウェアセルを送信し、メインメモリ26にメインメモリ用プログラムおよびそれによって処理されるデータを記憶させる。

30

【0081】

次に、送信元の情報処理装置は、送信先の情報処理装置宛に、送信先の情報処理装置内の情報処理コントローラについてのメインプロセッサID、メインメモリアドレス、メインメモリ用プログラムを識別するための後述の機能プログラムIDなどの識別子、およびプログラムカウンタを含み、DMAコマンドがキックコマンドまたは機能プログラム実行コマンドであるソフトウェアセルを送信して、メインプロセッサ21に当該メインメモリ用プログラムを実行させる。

【0082】

以上のように、この発明のネットワークシステムでは、送信元の情報処理装置は、サブプロセッサプログラムまたはメインメモリ用プログラムをソフトウェアセルによって送信先の情報処理装置に送信するとともに、当該サブプロセッサプログラムを送信先の情報処理装置内の情報処理コントローラに含まれるサブプロセッサ23にロードさせ、当該サブプロセッサプログラムまたは当該メインメモリ用プログラムを送信先の情報処理装置に実行させることができる。

40

【0083】

送信先の情報処理装置内の情報処理コントローラでは、受信したソフトウェアセルに含まれるプログラムがサブプロセッサプログラムである場合には、当該サブプロセッサプログラムを指定されたサブプロセッサにロードさせる。そして、ソフトウェアセルに含まれるサブプロセッサプログラムまたはメインメモリ用プログラムを実行させる。

【0084】

50

したがって、ユーザが送信先の情報処理装置を操作しなくても自動的に、当該サブプロセッサプログラムまたは当該メインメモリ用プログラムを送信先の情報処理装置内の情報処理コントローラに実行させることができる。

【 0 0 8 5 】

このようにして情報処理装置は、自装置内の情報処理コントローラがサブプロセッサプログラムまたは機能プログラムなどのメインメモリ用プログラムを有していない場合には、ネットワークに接続された他の情報処理装置からそれらを取得することができる。さらに、各サブプロセッサとメインメモリとの間ではDMA方式によりデータが転送され、また上述したサンドボックスを使用することによって、1つの情報処理コントローラ内でデータを多段階に処理する必要がある場合でも、高速かつ高セキュリティに処理を実行することができる。

10

【 0 0 8 6 】

[ 2 . 情報処理装置間の分散処理の例その 1 ]

ソフトウェアセルの使用による分散処理の結果、図 5 の上段に示すようにネットワーク 9 に接続されている複数の情報処理装置 1 , 2 , 3 および 4 は、図 5 の下段に示すように、仮想的な 1 台の情報処理装置 7 として動作する。ただし、そのためには、以下のような構成によって、以下のような処理が実行される必要がある。

【 0 0 8 7 】

( 2 - 1 . システムのソフトウェア構成とプログラムのロード )

図 6 に、個々の情報処理コントローラのメインメモリ 2 6 が記憶するソフトウェアの構成を示す。これらのソフトウェア(プログラム)は、情報処理装置に電源が投入される前においては、当該の情報処理コントローラに接続される外部記録部 2 8 に記録されているものである。

20

【 0 0 8 8 】

各プログラムは、機能または特徴によって、制御プログラム、機能プログラムおよびデバイスドライバにカテゴライズされる。

【 0 0 8 9 】

制御プログラムは、各情報処理コントローラが同じものを備え、各情報処理コントローラのメインプロセッサ 2 1 が実行するもので、後述のMS(マスター/スレーブ)マネージャおよび能力交換プログラムを含む。

30

【 0 0 9 0 】

機能プログラムは、メインプロセッサ 2 1 が実行するもので、記録用、再生用、素材検索用など、情報処理コントローラごとに情報処理装置に応じたものが備えられる。

【 0 0 9 1 】

デバイスドライバは、情報処理コントローラ(情報処理装置)の入出力(送受信)用で、放送受信、モニタ出力、ビットストリーム入出力、ネットワーク入出力など、情報処理コントローラごとに情報処理装置に応じたものが備えられる。

【 0 0 9 2 】

ケーブルの差し込みなどによって情報処理装置が物理的にネットワーク 9 に接続された状態で、情報処理装置に主電源が投入され、情報処理装置が電氣的・機能的にもネットワーク 9 に接続されると、その情報処理装置の情報処理コントローラのメインプロセッサ 2 1 は、制御プログラムに属する各プログラム、およびデバイスドライバに属する各プログラムを、メインメモリ 2 6 にロードする。

40

【 0 0 9 3 】

ロード手順としては、メインプロセッサ 2 1 は、まず、DC 2 7 に読み出し命令を実行させることによって、外部記録部 2 8 からプログラムを読み出し、次に、DMAC 2 5 に書き込み命令を実行させることによって、そのプログラムをメインメモリ 2 6 に書き込む。

【 0 0 9 4 】

機能プログラムに属する各プログラムについては、必要なときに必要なプログラムだけ

50

をロードするように構成してもよく、または、他のカテゴリに属するプログラムと同様に、主電源投入直後に各プログラムをロードするように構成してもよい。

【0095】

機能プログラムに属する各プログラムは、ネットワークに接続された全ての情報処理装置の外部記録部28に記録されている必要はなく、いずれか1つの情報処理装置の外部記録部28に記録されていれば、前述の方法によって他の情報処理装置からロードすることができるので、結果的に図5の下段に示すように、仮想的な1台の情報処理装置7として機能プログラムを実行することができる。

【0096】

また、前述したように、メインプロセッサ21によって処理される機能プログラムは、サブプロセッサ23によって処理されるサブプロセッサプログラムと連携動作する場合があります。そこで、メインプロセッサ21が外部記録部28から機能プログラムを読み出し、メインメモリ26に書き込む際、対象となる機能プログラムと連携動作するサブプロセッサプログラムが存在する場合には、当該サブプロセッサプログラムも併せて同じメインメモリ26に書き込む。この場合、連携動作するサブプロセッサプログラムは、1個である場合もあるが、複数個であることもあり得る。複数個である場合には、全ての連携動作するサブプロセッサプログラムをメインメモリ26に書き込む。

10

【0097】

メインメモリ26に書き込まれたサブプロセッサプログラムは、その後、サブプロセッサ23内のLS24に書き込まれ、メインプロセッサ21によって処理される機能プログラムと連携動作する。

20

【0098】

図3のソフトウェアセルに示したように、機能プログラムには、プログラムごとにプログラムを一意的に識別できる識別子が、機能プログラムIDとして割り当てられる。機能プログラムIDは、機能プログラムの作成の段階で、作成日時や情報処理装置IDなどから決定される。

【0099】

サブプロセッサプログラムにもサブプロセッサプログラムIDが割り当てられ、これによりサブプロセッサプログラムを一意的に識別可能である。割り当てられるサブプロセッサプログラムIDは、連携動作の相手となる機能プログラムの機能プログラムIDと関連性のある識別子、例えば機能プログラムIDを親番号とした上で最後尾に枝番号を付加させた識別子でもよいが、連携動作の相手となる機能プログラムの機能プログラムIDとは関連性のない識別子でもよい。

30

【0100】

いずれにしても、機能プログラムとサブプロセッサプログラムが連携動作する場合には、両者とも相手の識別子であるプログラムIDを自プログラム内に記憶しておく必要がある。機能プログラムが複数個のサブプロセッサプログラムと連携動作する場合にも、当該機能プログラムは、その複数個のサブプロセッサプログラムのサブプロセッサプログラムIDを全て記憶しておく。

【0101】

メインプロセッサ21は、自身が動作する情報処理装置の装置情報(動作状態に関する情報)を格納するための領域をメインメモリ26に確保し、当該情報を自装置の装置情報テーブルとして記録する。ここでの装置情報は、図4に示した情報処理装置ID以下の各情報である。

40

【0102】

(2-2. システムにおけるマスター/スレーブの決定)

上述したネットワークシステムでは、ある情報処理装置への主電源投入時、その情報処理装置の情報処理コントローラのメインプロセッサ21は、マスター/スレーブマネージャ(以下、MSマネージャ)をメインメモリ26にロードし、実行する。

【0103】

50

MSマネージャは、自身が動作する情報処理装置がネットワーク9に接続されていることを検知すると、同じネットワーク9に接続されている他の情報処理装置の存在を確認する。ここでの「接続」または「存在」は、上述したように、情報処理装置が物理的にネットワーク9に接続されているだけでなく、電氣的・機能的にもネットワーク9に接続されていることを示す。

**【0104】**

以下では、自身が動作する情報処理装置を自装置、他の情報処理装置を他装置と称する。当該装置も、当該情報処理装置を示すものとする。

**【0105】**

MSマネージャが同じネットワーク9に接続されている他の情報処理装置の存在を確認する方法を、以下に示す。

**【0106】**

MSマネージャは、DMAコマンドがステータス要求コマンドであり、送信元IDおよび応答先IDが当該情報処理装置で、送信先IDを特定しないソフトウェアセルを生成して、当該情報処理装置が接続されたネットワーク上に送信し、ネットワーク接続確認用のタイマーを設定する。タイマーのタイムアウト時間は、例えば10分とされる。

**【0107】**

当該ネットワークシステム上に他の情報処理装置が接続されている場合、その他装置は、上記ステータス要求コマンドのソフトウェアセルを受信し、上記応答先IDで特定されるステータス要求コマンドを発行した情報処理装置に対して、DMAコマンドがステータス返信コマンドで、かつデータとして自身(その他装置)の装置情報を含むソフトウェアセルを送信する。このステータス返信コマンドのソフトウェアセルには、少なくとも当該他装置を特定する情報(情報処理装置ID、メインプロセッサに関する情報、サブプロセッサに関する情報など)および当該他装置のMSステータスが含まれる。

**【0108】**

ステータス要求コマンドを発行した情報処理装置のMSマネージャは、上記ネットワーク接続確認用のタイマーがタイムアウトするまで、当該ネットワーク上の他装置から送信されるステータス返信コマンドのソフトウェアセルの受信を監視する。その結果、MSステータス=0(マスター装置)を示すステータス返信コマンドが受信された場合には、自装置の装置情報テーブルにおけるMSステータスを1に設定する。これによって、当該装置はスレーブ装置となる。

**【0109】**

一方、上記ネットワーク接続確認用のタイマーがタイムアウトするまでの間にステータス返信コマンドが全く受信されなかった場合、またはMSステータス=0(マスター装置)を示すステータス返信コマンドが受信されなかった場合には、自装置の装置情報テーブルにおけるMSステータスを0に設定する。これによって、当該装置はマスター装置となる。

**【0110】**

すなわち、いずれの装置もネットワーク9に接続されていない状態、またはネットワーク9上にマスター装置が存在しない状態において、新たな情報処理装置がネットワーク9に接続されると、当該装置は自動的にマスター装置として設定される。一方、ネットワーク9上に既にマスター装置が存在する状態において、新たな情報処理装置がネットワーク9に接続されると、当該装置は自動的にスレーブ装置として設定される。

**【0111】**

マスター装置およびスレーブ装置のいずれについても、MSマネージャは、定期的にステータス要求コマンドをネットワーク9上の他装置に送信してステータス情報を照会することにより、他装置の状況を監視する。その結果、ネットワーク9に接続されている情報処理装置の主電源が遮断され、またはネットワーク9から情報処理装置が切り離されることにより、あらかじめ判定用に設定された所定期間内に特定の他装置からステータス返信コマンドが返信されなかった場合や、ネットワーク9に新たな情報処理装置が接続された

10

20

30

40

50

場合など、ネットワーク 9 の接続状態に変化があった場合には、その情報を後述の能力交換プログラムに通知する。

【 0 1 1 2 】

( 2 - 3 . マスター装置およびスレーブ装置における装置情報の取得 )

メインプロセッサ 2 1 は、MS マネージャから、ネットワーク 9 上の他装置の照会および自装置の MS ステータスの設定完了の通知を受けると、能力交換プログラムを実行する。

【 0 1 1 3 】

能力交換プログラムは、自装置がマスター装置である場合には、ネットワーク 9 に接続されている全ての他装置の装置情報、すなわち各スレーブ装置の装置情報を取得する。

10

【 0 1 1 4 】

他装置の装置情報の取得は、上述したように、DMA コマンドがステータス要求コマンドであるソフトウェアセルを生成して他装置に送信し、その後、DMA コマンドがステータス返信コマンドで、かつデータとして他装置の装置情報を含むソフトウェアセルを他装置から受信することによって可能である。

【 0 1 1 5 】

能力交換プログラムは、マスター装置である自装置の装置情報テーブルと同様に、ネットワーク 9 に接続されている全ての他装置 ( 各スレーブ装置 ) の装置情報を格納するための領域を自装置のメインメモリ 2 6 に確保し、これら情報を他装置 ( スレーブ装置 ) の装置情報テーブルとして記録する。

20

【 0 1 1 6 】

すなわち、マスター装置のメインメモリ 2 6 には、自装置を含むネットワーク 9 に接続されている全ての情報処理装置の装置情報が、装置情報テーブルとして記録される。

【 0 1 1 7 】

一方、能力交換プログラムは、自装置がスレーブ装置である場合には、ネットワーク 9 に接続されている全ての他装置の装置情報、すなわちマスター装置および自装置以外の各スレーブ装置の装置情報を取得し、これら装置情報に含まれる情報処理装置 ID および MS ステータスを、自装置のメインメモリ 2 6 に記録する。

【 0 1 1 8 】

すなわち、スレーブ装置のメインメモリ 2 6 には、自装置の装置情報が、装置情報テーブルとして記録されるとともに、自装置以外のネットワーク 9 に接続されているマスター装置および各スレーブ装置についての情報処理装置 ID および MS ステータスが、別の装置情報テーブルとして記録される。

30

【 0 1 1 9 】

また、マスター装置およびスレーブ装置のいずれについても、能力交換プログラムは、上記のように MS マネージャから、新たにネットワーク 9 に情報処理装置が接続されたことが通知されたときには、その情報処理装置の装置情報を取得し、上述したようにメインメモリ 2 6 に記録する。

【 0 1 2 0 】

なお、MS マネージャおよび能力交換プログラムは、メインプロセッサ 2 1 で実行されることに限らず、いずれかのサブプロセッサ 2 3 で実行されてもよい。また、MS マネージャおよび能力交換プログラムは、情報処理装置の主電源が投入されている間は常時動作する常駐プログラムであることが望ましい。

40

【 0 1 2 1 】

( 2 - 4 . 情報処理装置がネットワークから切断された場合 )

マスター装置およびスレーブ装置のいずれについても、能力交換プログラムは、上記のように MS マネージャから、ネットワーク 9 に接続されている情報処理装置の主電源が遮断され、またはネットワーク 9 から情報処理装置が切り離されたことが通知されたときには、その情報処理装置の装置情報テーブルを自装置のメインメモリ 2 6 から削除する。

【 0 1 2 2 】

50

さらに、このようにネットワーク 9 から切断された情報処理装置がマスター装置である場合には、以下のような方法によって、新たにマスター装置が決定される。

【 0 1 2 3 】

具体的に、例えば、ネットワーク 9 から切断されていない情報処理装置は、それぞれ、自装置および他装置の情報処理装置 ID を数値に置き換えて、自装置の情報処理装置 ID を他装置の情報処理装置 ID と比較し、自装置の情報処理装置 ID がネットワーク 9 から切断されていない情報処理装置中で最小である場合、そのスレーブ装置は、マスター装置に移行して、MS ステータスを 0 に設定し、マスター装置として、上述したように、ネットワーク 9 に接続されている全ての他装置（各スレーブ装置）の装置情報を取得して、メインメモリ 26 に記録する。

10

【 0 1 2 4 】

（ 2 - 5 . 装置情報に基づく情報処理装置間の分散処理 ）

図 5 の下段に示したようにネットワーク 9 に接続されている複数の情報処理装置 1 , 2 , 3 および 4 を仮想的な 1 台の情報処理装置 7 として動作させるためには、マスター装置がユーザの操作およびスレーブ装置の動作状態を把握する必要がある。

【 0 1 2 5 】

図 7 に、4 台の情報処理装置が仮想的な 1 台の情報処理装置 7 として動作する様子を示す。情報処理装置 1 がマスター装置、情報処理装置 2 , 3 および 4 がスレーブ装置 A , B および C として、動作しているものとする。

【 0 1 2 6 】

ユーザがネットワーク 9 に接続されている情報処理装置を操作した場合、操作対象がマスター装置 1 であれば、その操作情報は、マスター装置 1 において直接把握され、操作対象がスレーブ装置であれば、その操作情報は、操作されたスレーブ装置からマスター装置 1 に送信される。すなわち、ユーザの操作対象がマスター装置 1 とスレーブ装置のいずれであるかにかかわらず、その操作情報は常にマスター装置 1 において把握される。操作情報の送信は、例えば、DMA コマンドが操作情報送信コマンドであるソフトウェアセルによって行われる。

20

【 0 1 2 7 】

そして、マスター装置 1 内の情報処理コントローラ 11 に含まれるメインプロセッサ 21 - 1 は、その操作情報に従って、実行する機能プログラムを選択する。その際、必要であれば、マスター装置 1 内の情報処理コントローラ 11 に含まれるメインプロセッサ 21 - 1 は、上記の方法によって自装置の外部記録部 28 - 1 または 28 - 2 からメインメモリ 26 - 1 に機能プログラムをロードするが、他の情報処理装置（スレーブ装置）がマスター装置 1 に機能プログラムを送信してもよい。

30

【 0 1 2 8 】

機能プログラムには、その実行単位ごとに必要となる、図 4 に示した各情報として表される情報処理装置種別 ID、メインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件などの、装置に関する要求スペックが規定されている。

【 0 1 2 9 】

マスター装置 1 内の情報処理コントローラ 11 に含まれるメインプロセッサ 21 - 1 は、各機能プログラムについて必要となる上記要求スペックを読み出す。また、あらかじめ能力交換プログラムによってメインメモリ 26 - 1 に記録された装置情報テーブルを参照し、各情報処理装置の装置情報を読み出す。ここでの装置情報は、図 4 に示した情報処理装置 ID 以下の各情報を示し、メインプロセッサ、サブプロセッサ、メインメモリおよび外部記録部に関する情報である。

40

【 0 1 3 0 】

マスター装置 1 内の情報処理コントローラ 11 に含まれるメインプロセッサ 21 - 1 は、ネットワーク 9 上に接続された各情報処理装置の上記装置情報と、機能プログラム実行に必要な上記要求スペックとを順次比較する。

50

## 【 0 1 3 1 】

そして、例えば、機能プログラムが録画機能を必要とする場合には、情報処理装置種別 ID に基づいて、録画機能を有する情報処理装置のみを特定して抽出する。さらに、機能プログラムを実行するために必要なメインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件を確保できるスレーブ装置を、実行要求候補装置として特定する。ここで、複数の実行要求候補装置が特定された場合には、当該候補装置から 1 つの実行要求候補装置を特定して選択する。

## 【 0 1 3 2 】

実行要求するスレーブ装置が特定されたら、マスター装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、その特定されたスレーブ装置について、自装置内の情報処理コントローラ 1 1 に含まれるメインメモリ 2 6 - 1 に記録されている当該スレーブ装置の装置情報テーブルを更新する。

10

## 【 0 1 3 3 】

さらに、マスター装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、DMA コマンドが機能プログラム実行コマンドであるソフトウェアセルを生成し、当該ソフトウェアセルのセルインターフェースに、当該機能プログラムに関する必要なサブプロセッサの情報およびサンドボックスサイズ ( 図 3 参照 ) を設定して、上記実行要求されるスレーブ装置に対して送信する。

## 【 0 1 3 4 】

機能プログラムの実行を要求されたスレーブ装置は、その機能プログラムを実行するとともに、自装置の装置情報テーブルを更新する。その際、必要であれば、スレーブ装置内の情報処理コントローラに含まれるメインプロセッサ 2 1 は、上記の方法によって自装置の外部記録部 2 8 からメインメモリ 2 6 に機能プログラムおよび当該機能プログラムと連携動作するサブプロセッサプログラムをロードする。

20

## 【 0 1 3 5 】

機能プログラムの実行を要求されたスレーブ装置の外部記録部 2 8 に、必要な機能プログラムまたは当該機能プログラムと連携動作するサブプロセッサプログラムが記録されていない場合には、他の情報処理装置が当該機能プログラムまたはサブプロセッサプログラムを、その機能プログラム実行要求先スレーブ装置に送信するように、システムを構成すればよい。

30

## 【 0 1 3 6 】

サブプロセッサプログラムについては、上記のロードコマンドおよびキックコマンドを利用して他の情報処理装置に実行させることもできる。

## 【 0 1 3 7 】

機能プログラムの実行終了後、機能プログラムを実行したスレーブ装置内の情報処理コントローラに含まれるメインプロセッサ 2 1 は、終了通知をマスター装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 に送信するとともに、自装置の装置情報テーブルを更新する。マスター装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、その終了通知を受信して、機能プログラムを実行したスレーブ装置の装置情報テーブルを更新する。

40

## 【 0 1 3 8 】

マスター装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、自装置および他装置の装置情報テーブルの参照結果から、当該の機能プログラムを実行することができる情報処理装置として、自身を選択する場合もあり得る。その場合には、マスター装置 1 が当該の機能プログラムを実行する。

## 【 0 1 3 9 】

図 7 の例で、ユーザがスレーブ装置 A ( 情報処理装置 2 ) を操作し、当該操作に応じた機能プログラムを別のスレーブ装置 B ( 情報処理装置 3 ) が実行する場合につき、図 8 に以上の分散処理の例を示す。

## 【 0 1 4 0 】

50

図 8 の例では、ユーザがスレーブ装置 A を操作することによって、スレーブ装置 A を含むネットワークシステム全体の分散処理が開始して、まず、スレーブ装置 A は、ステップ 8 1 で、その操作情報をマスター装置 1 に送信する。

【 0 1 4 1 】

マスター装置 1 は、ステップ 7 2 で、その操作情報を受信し、さらにステップ 7 3 に進んで、自装置のメインメモリ 2 6 - 1 に記録されている自装置および他装置の装置情報テーブルから、各情報処理装置の動作状態を調べて、受信した操作情報に応じた機能プログラムを実行することができる情報処理装置を選択する。この例は、スレーブ装置 B が選択される場合である。

【 0 1 4 2 】

次に、マスター装置 1 は、ステップ 7 4 で、その選択したスレーブ装置 B に対して機能プログラムの実行を要求する。

【 0 1 4 3 】

スレーブ装置 B は、ステップ 9 5 で、その実行要求を受信し、さらにステップ 9 6 に進んで、実行要求された機能プログラムを実行する。

【 0 1 4 4 】

以上のように、ユーザは、1 台の情報処理装置のみを操作することによって、他の情報処理装置を操作することなく、複数の情報処理装置 1, 2, 3 および 4 を仮想的な 1 台の情報処理装置 7 として動作させることができる。

【 0 1 4 5 】

( 2 - 6 . 各情報処理装置およびシステムの具体例 )

ネットワーク 9 を介して互いに接続される情報処理装置 1, 2, 3 および 4 は、基本的に上記のような情報処理コントローラ 1 1, 1 2, 1 3 および 1 4 によって情報処理を行うものであれば、どのようなものでもよいが、図 9 に、その一例を示す。

【 0 1 4 6 】

この例では、情報処理コントローラ 1 1 を備える情報処理装置 1 は、ハードディスクレコーダで、図 1 0 に示すように、ハードウェア構成としては、図 1 に示した外部記録部 2 8 - 1 として、ハードディスクを内蔵し、図 1 に示した外部記録部 2 8 - 2 として、DVD ± R / RW, CD ± R / RW, Blu-ray - Disc (登録商標) などの光ディスクを装着できるように構成されるとともに、情報処理コントローラ 1 1 のバス 2 9 - 1 に接続されたバス 3 1 - 1 に、放送受信部 3 2 - 1、映像入力部 3 3 - 1、音声入力部 3 4 - 1、映像出力部 3 5 - 1、音声出力部 3 6 - 1、操作パネル部 3 7 - 1、リモコン受光部 3 8 - 1 およびネットワーク接続部 3 9 - 1 が接続されたものである。

【 0 1 4 7 】

放送受信部 3 2 - 1、映像入力部 3 3 - 1 および音声入力部 3 4 - 1 は、放送信号を受信し、または情報処理装置 1 の外部から映像信号および音声信号を入力し、それぞれ所定フォーマットのデジタルデータに変換して、情報処理コントローラ 1 1 での処理のためにバス 3 1 - 1 に送出するものであり、映像出力部 3 5 - 1 および音声出力部 3 6 - 1 は、情報処理コントローラ 1 1 からバス 3 1 - 1 に送出された映像データおよび音声データを処理して、デジタルデータのまま、またはアナログ信号に変換して、情報処理装置 1 の外部に送出するものであり、リモコン受光部 3 8 - 1 は、リモコン送信器 4 3 - 1 からのリモコン (遠隔操作) 赤外線信号を受信するものである。

【 0 1 4 8 】

図 9 および図 1 0 に示すように、情報処理装置 (ハードディスクレコーダ) 1 の映像出力部 3 5 - 1 および音声出力部 3 6 - 1 には、モニタ表示装置 4 1 およびスピーカ装置 4 2 が接続される。

【 0 1 4 9 】

図 9 の例の、情報処理コントローラ 1 2 を備える情報処理装置 2 も、ハードディスクレコーダで、図 1 0 において括弧内に参照番号を付して示すように、情報処理装置 1 と同様に構成されたものである。ただし、例えば、図 9 に示すように、情報処理装置 (ハードデ

10

20

30

40

50

ィスクレコーダ) 2には、モニタ表示装置およびスピーカ装置は接続されない。

【0150】

情報処理装置(ハードディスクレコーダ) 1および2、すなわち情報処理コントローラ 11および12のソフトウェア構成としては、図11に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、映像音声記録、映像音声再生、素材検索および番組録画予約のためのプログラムを備え、デバイスドライバとして、放送受信、映像出力、音声出力、外部記録部入出力およびネットワーク入出力のためのプログラムを備える。

【0151】

図9の例の、情報処理コントローラ13を備える情報処理装置3は、PDA(Personal Digital Assistants)で、図12に示すように、ハードウェア構成としては、図1に示した外部記録部28-5として、メモ리카ードディスクを装着できるように構成されるとともに、情報処理コントローラ13のバス29-3に接続されたバス51に、液晶表示部52、音声出力部53、カメラ部54、音声入力部55、キーボード部56およびネットワーク接続部57が接続されたものである。

10

【0152】

なお、図1では内部を省略した情報処理コントローラ13は、メインプロセッサ21-3、サブプロセッサ23-7, 23-8および23-9、DMAC(ダイレクトメモリアクセスコントローラ)25-3、DC(ディスクコントローラ)27-3、およびバス29-3を備え、そのメインプロセッサ21-3は、LS(ローカルストレージ)22-3を有し、各サブプロセッサ23-7, 23-8および23-9は、LS(ローカルストレージ)24-7, 24-8および24-9を有する。

20

【0153】

情報処理装置(PDA)3、すなわち情報処理コントローラ13のソフトウェア構成としては、図13に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、映像音声記録、映像音声再生、電話帳、ワープロおよび表計算のためのプログラム、およびWebブラウザを備え、デバイスドライバとして、映像出力、音声出力、カメラ映像入力、マイク音声入力およびネットワーク入出力のためのプログラムを備える。

【0154】

図9の例の、情報処理コントローラ14を備える情報処理装置4は、ポータブルCDプレーヤで、図14に示すように、ハードウェア構成としては、図1に示した外部記録部28-6として、CD(Compact Disc)を装着できるように構成されるとともに、情報処理コントローラ14のバス29-4に接続されたバス61に、液晶表示部62、音声出力部63、操作ボタン部64およびネットワーク接続部65が接続されたものである。

30

【0155】

なお、図1では内部を省略した情報処理コントローラ14は、メインプロセッサ21-4、サブプロセッサ23-10, 23-11および23-12、DMAC(ダイレクトメモリアクセスコントローラ)25-4、DC(ディスクコントローラ)27-4、およびバス29-4を備え、そのメインプロセッサ21-4は、LS(ローカルストレージ)22-4を有し、各サブプロセッサ23-10, 23-11および23-12は、LS(ローカルストレージ)24-10, 24-11および24-12を有する。

40

【0156】

情報処理装置(ポータブルCDプレーヤ)4、すなわち情報処理コントローラ14のソフトウェア構成としては、図15に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、音楽再生のためのプログラムを備え、デバイスドライバとして、音声出力、CD制御およびネットワーク入出力のためのプログラムを備える。

【0157】

50

以上のような図9の例のネットワークシステムにおいて、情報処理装置1, 3および4がネットワーク9上に接続されており、情報処理装置1がマスター装置(MSステータス=0)として、情報処理装置3および4がスレーブ装置(MSステータス=1)として、設定されているものとする。

【0158】

この状態で、新たに情報処理装置2がネットワーク9に接続されると、上述した方法によって、情報処理装置2内の情報処理コントローラ12に含まれるメインプロセッサ21-2で実行されているMSマネージャは、他の情報処理装置1, 3および4にMSステータスを照会して、情報処理装置1が既にマスター装置として存在することを認識し、自装置(情報処理装置2)をスレーブ装置(MSステータス=1)に設定する。また、マスター装置に設定されている情報処理装置1は、新たに追加された情報処理装置2を含む各装置の装置情報を収集して、メインメモリ26-1内の装置情報テーブルを更新する。

10

【0159】

このような状態で、ユーザによってスレーブ装置である情報処理装置(PDA)3で2時間の放送番組の録画予約の操作が行われた場合を示す。

【0160】

この場合、スレーブ装置である情報処理装置(PDA)3は、ユーザから録画開始時刻、録画終了時刻、録画対象放送チャネル、録画品質などの情報を含む録画予約情報の入力を受け付け、当該録画予約情報およびDMAコマンドとしての録画予約コマンドを含むソフトウェアセルを生成して、マスター装置である情報処理装置1に送信する。

20

【0161】

DMAコマンドが録画予約コマンドであるソフトウェアセルを受信した情報処理装置1内の情報処理コントローラ11に含まれるメインプロセッサ21-1は、録画予約コマンドを読み出すとともに、メインメモリ26-1内の装置情報テーブルを参照して、当該録画予約コマンドを実行可能な情報処理装置を特定する。

【0162】

まず、メインプロセッサ21-1は、装置情報テーブルに含まれる各情報処理装置1, 2, 3および4の情報処理装置種別IDを読み出して、録画予約コマンドに対応する機能プログラムを実行可能な情報処理装置を抽出する。ここでは、録画機能を示す情報処理装置種別IDを有する情報処理装置1および2が候補装置として特定され、情報処理装置3および4は候補装置から除外される。

30

【0163】

次に、マスター装置である情報処理装置1内の情報処理コントローラ11に含まれるメインプロセッサ21-1は、装置情報テーブルを参照して、情報処理装置1および2のメインプロセッサまたはサブプロセッサの処理能力、メインメモリに関する情報などの、装置に関する情報を読み出し、情報処理装置1および2が録画予約コマンドに対応する機能プログラムの実行に必要な要求スペックを満足するか否かを判断する。ここでは、情報処理装置1, 2とも、録画予約コマンドに対応する機能プログラムの実行に必要な要求スペックを満足するものとする。

【0164】

さらに、メインプロセッサ21-1は、装置情報テーブルを参照して、情報処理装置1および2の外部記録部に関する情報を読み出し、外部記録部の空き容量が当該録画予約コマンドの実行に必要な容量を満足するか否かを判断する。情報処理装置1および2はハードディスクレコーダであるので、それぞれハードディスク28-1および28-3の、総容量と使用量との差分が、それぞれの空き容量に相当する。

40

【0165】

この場合、情報処理装置1のハードディスク28-1の空き容量が、録画時間に換算して10分であり、情報処理装置2のハードディスク28-3の空き容量が、録画時間に換算して20時間であるとする。

【0166】

50

このとき、マスター装置である情報処理装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、当該録画予約コマンドの実行に必要な 2 時間分の空き容量を確保できる情報処理装置を、実行要求先スレーブ装置として特定する。

【 0 1 6 7 】

その結果、情報処理装置 2 のみが実行要求先スレーブ装置として選択され、マスター装置である情報処理装置 1 内の情報処理コントローラ 1 1 に含まれるメインプロセッサ 2 1 - 1 は、ユーザにより操作された情報処理装置 3 から送信された録画予約情報を含む当該録画予約コマンドを情報処理装置 2 に送信して、上記 2 時間の放送番組の録画予約の実行を要求する。

【 0 1 6 8 】

そして、情報処理装置 2 内の情報処理コントローラ 1 2 に含まれるメインプロセッサ 2 1 - 2 は、当該録画予約コマンドを解析して、録画に必要な機能プログラムを外部記録部であるハードディスク 2 8 - 3 からメインメモリ 2 6 - 2 にロードし、録画予約情報に従って録画を実行する。その結果、録画予約された 2 時間の放送番組の映像音声データが情報処理装置 2 のハードディスク 2 8 - 3 に記録される。

【 0 1 6 9 】

このように、図 9 の例のネットワークシステムにおいても、ユーザは、1 台の情報処理装置のみを操作することによって、他の情報処理装置を操作することなく、複数の情報処理装置 1 , 2 , 3 および 4 を仮想的な 1 台の情報処理装置 7 として動作させることができる。

【 0 1 7 0 】

[ 3 . サブプロセッサの割り当てによる分散処理その 1 ]

図 7 または図 9 のネットワークシステムにおける前述の例は、マスター装置である情報処理装置 1 が機能プログラムを実行することができる情報処理装置を特定する際、当該機能プログラムの実行に必要な全ての要求スペックを満足する情報処理装置がネットワークシステム内に存在する場合であるが、このような情報処理装置が存在しない場合もあり得る。

【 0 1 7 1 】

図 9 の例では、マスター装置である情報処理装置 1 は、録画予約コマンドに対応する機能プログラムを実行させる情報処理装置として、最終的に情報処理装置 2 を特定したが、情報処理装置 2 が当該機能プログラムの実行に必要な要求スペックを一部、満たしていないこともある。例えば、他の要求スペックは全て満たしているが、サブプロセッサ数が不足している場合である。そのような場合でも、情報処理装置 2 は、上記のロードコマンドまたはキックコマンドを利用して他の情報処理装置内のサブプロセッサを利用することによって、当該機能プログラムを実行できる可能性がある。

【 0 1 7 2 】

そこで、マスター装置が機能プログラムを実行させる情報処理装置を特定する際、候補となる情報処理装置のサブプロセッサ処理能力を考慮しないで、当該機能プログラムを実行させる情報処理装置を特定する場合の例を示す。

【 0 1 7 3 】

この場合、機能プログラムには、図 4 に示した各情報として表される情報処理装置種別 ID、メインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件などの、装置に関する要求スペックが規定されており、マスター装置が当該機能プログラムを実行させる情報処理装置を特定する際には、サブプロセッサ処理能力以外の要求スペックを満足する情報処理装置を特定するものとする。したがって当然、マスター装置によって特定されて機能プログラムの実行を要求された情報処理装置は、自装置内に当該機能プログラムの実行に必要なサブプロセッサ処理能力を確保できない場合がある。

【 0 1 7 4 】

以下に、分散処理の他の例として、このように機能プログラムの実行を要求された情報

10

20

30

40

50

処理装置が、ネットワークシステムに含まれる他の情報処理装置内のサブプロセッサを利用する場合について示す。

【0175】

(3-1. システム構成)

図16に、4台の情報処理装置1, 2, 3および4がネットワーク9に接続されている状態を示す。

【0176】

情報処理装置1がマスター装置として、情報処理装置2, 3および4がスレーブ装置として、それぞれ動作し、マスター装置である情報処理装置1が機能プログラムを実行させる情報処理装置として自装置を特定するものとする。

10

【0177】

ネットワーク9に接続されている各情報処理装置のソフトウェア構成としては、図17に示すように、制御プログラムとして、図6などに示したMSマネージャおよび能力交換プログラムのほかに、リソースマネージャを含むものとする。

【0178】

リソースマネージャは、自装置内の各サブプロセッサを制御して、プログラムやデータの転送、プログラムの実行の開始や停止、プログラムの実行結果の受信などを行うものである。

【0179】

さらに、リソースマネージャは、自装置内の各サブプロセッサの使用状況を把握して、他の情報処理装置からの問い合わせに対して応答し、または他の情報処理装置に対して、その他装置内の各サブプロセッサの使用状況を問い合わせる。

20

【0180】

そのために、リソースマネージャには、図18に示すようなサブプロセッサ管理テーブルを用意する。

【0181】

この例は、各情報処理装置内の情報処理コントローラが数個以上のサブプロセッサを備える場合で、サブプロセッサ管理テーブル中の情報処理装置IDは、自装置の情報処理装置IDであり、サブプロセッサIDは、自装置内の各サブプロセッサのサブプロセッサIDであり、サブプロセッサステータスは、各サブプロセッサの使用状況を示す。

30

【0182】

サブプロセッサステータスとしては、上述したように、`unused` (現在使用されてなく、使用の予約もされていない状態)、`reserved` (現在は使用されていないが、使用が予約されている状態)、および`busy` (現在使用されている状態)がある。

【0183】

後述のように、`unused`を空いている状態とし、空いているか否か、すなわち、`unused`であるか、`reserved`または`busy`であるかを、空き状況とする。

【0184】

サブプロセッサプログラムIDは、当該のサブプロセッサで実行中の、または当該のサブプロセッサに実行予約中のサブプロセッサプログラムを識別する識別子であり、機能プログラムIDは、上段のサブプロセッサプログラムIDで示されるサブプロセッサプログラムと連携動作する機能プログラムを識別する識別子である。

40

【0185】

サブプロセッサプログラム優先度は、その実行中または実行予約中のサブプロセッサプログラムの優先度である。サブプロセッサプログラム優先度は、当該サブプロセッサプログラムと連携動作する機能プログラムの優先度に応じて、あらかじめ決定され、例えば、`high` (優先度が高い)、`middle` (中位の優先度)および`low` (優先度が低い)の3段階に区分される。

【0186】

ロックシーケンス番号は、後述のようにサブプロセッサを確保する際などに、リソース

50

マネージャが発番し、サブプロセッサ管理テーブルに記述する。リソースマネージャは、機能プログラムから受信したサブプロセッサプログラムIDおよびロックシーケンス番号が、このサブプロセッサ管理テーブルに記述されたそれらと一致するときのみ、そのサブプロセッサの使用を許可する。

【0187】

(3-2. サブプロセッサ割り当て処理)

上述したように、図16において、情報処理装置1がマスター装置として、情報処理装置2, 3および4がスレーブ装置として、それぞれ動作し、マスター装置としての情報処理装置1が機能プログラムを実行させる情報処理装置として自装置を特定したものとす

10

【0188】

そして、情報処理装置1が、自装置のメインメモリ26-1内の機能プログラムと連携動作するサブプロセッサプログラムの実行用に、自装置内または他の情報処理装置2, 3および4内にサブプロセッサを確保する場合の、情報処理装置1のメインメモリ26-1内の機能プログラム、情報処理装置1内のリソースマネージャ、および他の情報処理装置2, 3および4内のリソースマネージャによって実行されるサブプロセッサ割り当て処理の一例を、図19~図27を用いて示す。

【0189】

まず、図19に示すように、情報処理装置1のメインメモリ26-1内の機能プログラムは、ステップ111で、サブプロセッサプログラムの実行に必要なサブプロセッサ数を判断し、次にステップ112で、自装置(情報処理装置1)内のリソースマネージャに、当該の機能プログラムの機能プログラムIDと、当該機能プログラムと連携動作する全てのサブプロセッサプログラムのサブプロセッサプログラムIDおよびサブプロセッサプログラム優先度とを添えて、必要数(n個とする)のサブプロセッサの使用要求を出力する

20

【0190】

これに対して、情報処理装置1内のリソースマネージャは、ステップ113で、自装置(情報処理装置1)の図18に示したサブプロセッサ管理テーブルを参照して、自装置内にn個のサブプロセッサの空きがあるか否かを判断し、n個以上のサブプロセッサが空いている場合には、ステップ113からステップ114に進んで、自装置内にn個のサブ

30

【0191】

さらに、情報処理装置1内のリソースマネージャは、ステップ115に進んで、当該の機能プログラムに、自装置内にn個のサブプロセッサを確保したことと、確保したn個のサブプロセッサのサブプロセッサID、サブプロセッサプログラムID、およびロック

40

【0192】

図20に示すように、当該の機能プログラムは、ステップ117で、この通知を受信して、自装置(情報処理装置1)内のリソースマネージャに、自身の機能プログラムIDと共に、確保した各サブプロセッサに対するサブプロセッサID、サブプロセッサプログラムID、サブプロセッサプログラム本体およびロックシーケンス番号を、確保したサブプロセッサ数の分だけ送信する。

【0193】

情報処理装置1内のリソースマネージャは、ステップ121で、これらを受信して、各サブプロセッサに対するサブプロセッサプログラムを保存し、さらにステップ122に進

50

んで、受信したサブプロセッサプログラムIDおよびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断し、同じであるときには、ステップ123に進んで、確保した各サブプロセッサにサブプロセッサプログラムを送信し、さらにステップ124に進んで、当該の機能プログラムに、各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0194】

以上によって、情報処理装置1内の空いているサブプロセッサだけで必要数のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

【0195】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

10

【0196】

情報処理装置1内のリソースマネージャは、ステップ122で、当該の機能プログラムから受信したサブプロセッサプログラムIDまたはロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと異なると判断した場合には、ステップ122からステップ126に進んで、当該の機能プログラムにエラーを通知する。

【0197】

この場合には、当該の機能プログラムは、ステップ127で、エラー処理を実行する。このエラー処理は、具体的には、再度、ステップ112からサブプロセッサ割り当て処理を実行するなどである。

20

【0198】

一方、情報処理装置1内のリソースマネージャは、図19のステップ113で、自装置内にn個のサブプロセッサの空きがないと判断した場合、すなわち情報処理装置1内に空いているサブプロセッサが(n-1)個以下しかない(1個もない場合を含む)と判断した場合には、図21に示すように、ステップ113からステップ131に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。図16は、他の情報処理装置2,3および4が接続されている場合である。

【0199】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1内のリソースマネージャは、ステップ131からステップ132に進んで、その中の1個である他の情報処理装置(この場合は情報処理装置2)内のリソースマネージャに、サブプロセッサの空き状況を問い合わせる。

30

【0200】

この問い合わせは、図28(A)に示すようなサブプロセッサ空き状況問い合わせコマンドによって行う。具体的に、このサブプロセッサ空き状況問い合わせコマンドは、送信元ID(この場合は情報処理装置1の情報処理装置ID)、送信先ID(この場合は情報処理装置2の情報処理装置ID)および応答先ID(この場合は情報処理装置1の情報処理装置ID)を有し、DMAコマンドとして、サブプロセッサ空き状況問い合わせコマンド本体および機能プログラムIDを含むものとする。

40

【0201】

図21に示すように、この問い合わせに対して、当該他の情報処理装置(この場合は情報処理装置2)内のリソースマネージャは、ステップ133で、自装置内のサブプロセッサ管理テーブルから、自装置内のサブプロセッサの空き状況を判断し、さらにステップ135に進んで、自装置内に空いているサブプロセッサがある場合には、その数およびサブプロセッサIDを、空いているサブプロセッサがない場合には、その旨を、情報処理装置1内のリソースマネージャに通知する。

【0202】

この通知は、図28(B)に示すようなサブプロセッサ空き状況返信コマンドによって

50

行う。具体的に、このサブプロセッサ空き状況返信コマンドは、送信元ID（この場合は情報処理装置2の情報処理装置ID）、送信先ID（この場合は情報処理装置1の情報処理装置ID）および応答先ID（この場合は情報処理装置1の情報処理装置ID）を有し、DMAコマンドとして、サブプロセッサ空き状況返信コマンド本体、機能プログラムID、空いているサブプロセッサの数、および空いている各サブプロセッサのサブプロセッサIDを含むものとする。

**【0203】**

この通知を受けて、情報処理装置1内のリソースマネージャは、図22に示すように、ステップ136で、自装置（情報処理装置1）分と他装置（情報処理装置2）分とを合わせてトータルで、 $n$ 個のサブプロセッサを確保できるか否かを判断する。

10

**【0204】**

例えば、 $n$ が8、9または10というような数である場合に、情報処理装置1内で1個のサブプロセッサしか確保できず、情報処理装置2内で $(n - 2)$ 個のサブプロセッサしか確保できない場合には、トータルで $n$ 個のサブプロセッサを確保できないことになる。

**【0205】**

このようにトータルで $n$ 個のサブプロセッサを確保できない場合には、情報処理装置1内のリソースマネージャは、図21に示すように、ステップ136からステップ131に進んで（戻って）、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。図16は、他の情報処理装置3および4が接続されている場合である。

**【0206】**

20

以下、情報処理装置2の場合と同様に、ネットワークシステム全体で $n$ 個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ132、133、135、136および131の処理を繰り返す。

**【0207】**

情報処理装置1以外にネットワーク上に複数個の情報処理装置が接続されている場合に情報処理装置1内のリソースマネージャがサブプロセッサの空き状況を問い合わせる順番については、例えば、上述したマスター装置の決定方法のように、ネットワーク上の各情報処理装置の情報処理装置IDを数値に置き換え、当該数値が小さい情報処理装置順にサブプロセッサの空き状況を問い合わせることが考えられる。

30

**【0208】**

そして、ネットワークシステム全体で、空いているサブプロセッサによって $n$ 個のサブプロセッサを確保できる場合には、情報処理装置1内のリソースマネージャは、図22に示すように、ステップ136からステップ137に進んで、当該の機能プログラムに、 $n$ 個のサブプロセッサを確保できることを通知する。

**【0209】**

当該の機能プログラムは、ステップ138で、この通知を受信し、後述のように自装置（情報処理装置1）内のリソースマネージャにサブプロセッサ確保要求を出力する。

**【0210】**

一方、情報処理装置1内のリソースマネージャは、図21のステップ131で、それ以上、ネットワーク上に情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体でも（情報処理装置1のみがネットワーク上に存在する場合には情報処理装置1内だけで）、空いているサブプロセッサによって $n$ 個のサブプロセッサを確保できないと判断した場合には、図22に示すように、ステップ131からステップ141に進んで、自装置（情報処理装置1）内に当該の機能プログラムより優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否か、および、ある場合には、その数を検出する。

40

**【0211】**

この場合の「実行中」というのは、`busy`（現在使用されている状態）だけでなく、`reserved`（現在は使用されていないが、使用が予約されている状態）を含む。以

50

下では、当該の機能プログラムより優先度の低いサブプロセッサプログラムを、単に「優先度の低いサブプロセッサプログラム」と称する。

【0212】

さらに、情報処理装置1内のリソースマネージャは、ステップ143に進んで、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置1）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、n個のサブプロセッサを確保できるか否かを判断し、確保できる場合には、ステップ143からステップ144に進んで、当該の機能プログラムに、n個のサブプロセッサを確保できることを通知する。

【0213】

当該の機能プログラムは、ステップ138で、この通知を受信し、後述のように自装置（情報処理装置1）内のリソースマネージャにサブプロセッサ確保要求を出力する。

【0214】

一方、情報処理装置1内のリソースマネージャは、ステップ143で、ネットワークシステム全体の空いているサブプロセッサと、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、n個のサブプロセッサを確保できないと判断した場合には、図23に示すように、ステップ143からステップ151に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。

【0215】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1内のリソースマネージャは、ステップ151からステップ152に進んで、当該他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャに、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの有無および数を問い合わせる。この場合の「実行中」というのも、busyだけでなく、reservedを含む。

【0216】

この問い合わせは、図28(A)に示したようなサブプロセッサ空き状況問い合わせコマンドに、当該の機能プログラムと連携動作するサブプロセッサプログラムのサブプロセッサプログラム優先度を含めたコマンドによって行う。

【0217】

図23に示すように、この問い合わせに対して、当該他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャは、ステップ153で、自装置内のサブプロセッサ管理テーブルから、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否かを判断し、ある場合には、その数およびサブプロセッサIDを情報処理装置1内のリソースマネージャに通知し、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがない場合には、その旨を情報処理装置1内のリソースマネージャに通知する。

【0218】

この通知は、図28(B)に示したようなサブプロセッサ空き状況返信コマンドにおいて、空いているサブプロセッサの数およびサブプロセッサIDに代えて、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの数およびサブプロセッサIDを記述したコマンドによって行う。

【0219】

この通知を受けて、情報処理装置1内のリソースマネージャは、図23に示すように、ステップ154で、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置1）内および他装置（この場合は情報処理装置2）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、n個のサブプロセッサを確保できるか否かを判断する。

【0220】

例えば、n=7の場合に、ネットワークシステム全体で空いているサブプロセッサが2個しかなく、情報処理装置1内に優先度の低いサブプロセッサプログラムを実行中のサブ

10

20

30

40

50

プロセッサが2個しかなく、情報処理装置2内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサが2個しかない場合には、トータルでn個のサブプロセッサを確保できないことになる。

【0221】

このようにトータルでn個のサブプロセッサを確保できない場合には、情報処理装置1内のリソースマネージャは、ステップ154からステップ151に進んで(戻って)、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。

【0222】

以下、情報処理装置2の場合と同様に、ネットワークシステム全体でn個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ152, 153, 154および151の処理を繰り返す。

10

【0223】

そして、空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、ネットワークシステム全体でn個のサブプロセッサを確保できる場合には、情報処理装置1内のリソースマネージャは、ステップ154からステップ155に進んで、当該の機能プログラムに、n個のサブプロセッサを確保できることを通知する。

【0224】

当該の機能プログラムは、ステップ158で、この通知を受信し、以下のように自装置(情報処理装置1)内のリソースマネージャにサブプロセッサ確保要求を出力する。

20

【0225】

すなわち、当該の機能プログラムは、図22のステップ138、または図23のステップ158で、情報処理装置1内のリソースマネージャから、n個のサブプロセッサを確保できる旨の通知を受信したら、図24に示すように、ステップ161で、自装置(情報処理装置1)内のリソースマネージャに、n個のサブプロセッサの確保要求を出力する。

【0226】

図24に示すように、この確保要求に対して、情報処理装置1内のリソースマネージャは、ステップ162で、自装置(情報処理装置1)内に確保すべき分がある場合には、その分のサブプロセッサを自装置内に確保し、さらにステップ163に進んで、他の情報処理装置内に確保されるべき分があるか否かを判断し、他の情報処理装置内に確保されるべき分がない場合には、ステップ163からステップ164に進んで、当該の機能プログラムに、n個のサブプロセッサを確保したことを通知する。

30

【0227】

情報処理装置1内の空いているサブプロセッサだけではn個のサブプロセッサを確保できず、かつ情報処理装置1以外にネットワーク上に接続されている情報処理装置が存在しないが、情報処理装置1内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、n個のサブプロセッサを確保できる場合には、このように情報処理装置1内だけでn個のサブプロセッサが確保されることになる。

【0228】

この場合、情報処理装置1内のリソースマネージャは、ステップ162では、自装置内に確保した各サブプロセッサにつき、自装置のサブプロセッサ管理テーブル内のサブプロセッサステータスが `unused` となっているものについては、`reserved` に書き替え、サブプロセッサプログラムID、機能プログラムIDおよびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替えるとともに、ロックシーケンス番号を発番してサブプロセッサ管理テーブルに記述し、ステップ164では、これら情報を、当該の機能プログラムに送信する。

40

【0229】

具体的に、ステップ164で送信される情報は、確保したサブプロセッサの数、確保した各サブプロセッサのサブプロセッサIDおよびロックシーケンス番号を含むものとする

50

## 【 0 2 3 0 】

図 2 4 に示すように、当該の機能プログラムは、ステップ 1 6 7 で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置 1）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

## 【 0 2 3 1 】

なお、情報処理装置 1 内のリソースマネージャは、ステップ 1 6 2 で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを、当該の機能プログラムと連携動作するサブプロセッサプログラムの実行用に確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブプロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

10

## 【 0 2 3 2 】

一方、確保されるべき  $n$  個のサブプロセッサに他の情報処理装置内に確保されるべき分がある場合には、情報処理装置 1 内のリソースマネージャは、ステップ 1 6 3 からステップ 1 7 1 に進んで、他の情報処理装置 2, 3 および 4 内のリソースマネージャにサブプロセッサの確保を要求し、他の情報処理装置 2, 3 および 4 内のリソースマネージャは、ステップ 1 7 2 で、それぞれ自装置内に自装置分のサブプロセッサを確保する。

## 【 0 2 3 3 】

他の情報処理装置に対するサブプロセッサの確保要求は、図 2 9 (A) に示すような DMA コマンドがサブプロセッサ確保要求コマンドであるソフトウェアセルによって行う。具体的に、このサブプロセッサ確保要求コマンドは、DMA コマンドとして、サブプロセッサ確保要求コマンド本体、機能プログラム ID、サブプロセッサプログラム優先度、確保要求サブプロセッサ数および確保要求サブプロセッサ数の分のサブプロセッサプログラム ID を含むものとする。

20

## 【 0 2 3 4 】

情報処理装置 1 内の空いているサブプロセッサだけでは  $n$  個のサブプロセッサを確保できないが、情報処理装置 1 以外にもネットワーク上に情報処理装置が存在していて、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、 $n$  個のサブプロセッサを確保できる場合には、このように他の情報処理装置を含めたネットワークシステム全体で  $n$  個のサブプロセッサが確保されることになる。

30

## 【 0 2 3 5 】

この場合、他の情報処理装置 2, 3 および 4 内のリソースマネージャは、ステップ 1 7 2 では、それぞれ自装置内に確保した各サブプロセッサにつき、自装置のサブプロセッサ管理テーブル内のサブプロセッサステータスが `u n u s e d` となっているものについては、`r e s e r v e d` に書き替え、サブプロセッサプログラム ID、機能プログラム ID およびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替えるとともに、ロックシーケンス番号を発番してサブプロセッサ管理テーブルに記述する。

## 【 0 2 3 6 】

また、他の情報処理装置 2, 3 および 4 内のリソースマネージャは、ステップ 1 7 2 で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを、当該の機能プログラムと連携動作するサブプロセッサプログラムの実行用に確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブプロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

40

## 【 0 2 3 7 】

さらに、他の情報処理装置 2, 3 および 4 内のリソースマネージャは、図 2 5 に示すように、ステップ 1 7 3 で、それぞれ自装置内に自装置分のサブプロセッサを確保したことを、情報処理装置 1 内のリソースマネージャに通知する。

## 【 0 2 3 8 】

50

この通知には、図 29 (B) に示すような DMA コマンドがサブプロセッサ確保返信コマンドであるソフトウェアセルを用いる。具体的に、このサブプロセッサ確保返信コマンドは、DMA コマンドとして、サブプロセッサ確保返信コマンド本体、機能プログラム ID、サブプロセッサプログラム優先度、確保したサブプロセッサの数、確保した各サブプロセッサのサブプロセッサ ID、サブプロセッサプログラム ID およびロックシーケンス番号を含むものとする。

【0239】

図 25 に示すように、情報処理装置 1 内のリソースマネージャは、他の情報処理装置 2, 3 および 4 内のリソースマネージャからのサブプロセッサ確保通知を受信したら、ステップ 174 で、当該の機能プログラムに、n 個のサブプロセッサを確保したことを通知する。

10

【0240】

当該の機能プログラムは、ステップ 177 で、このサブプロセッサ確保通知を受信し、自装置 (情報処理装置 1) 内のリソースマネージャに、各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0241】

すなわち、当該の機能プログラムは、図 24 のステップ 167、または図 25 のステップ 177 で、情報処理装置 1 内のリソースマネージャから、n 個のサブプロセッサを確保した旨の通知を受信したら、図 25 に示すように、ステップ 181 で、情報処理装置 1 内のリソースマネージャに、確保された各サブプロセッサに対するサブプロセッサプログラムを、当該サブプロセッサプログラムに対応するサブプロセッサプログラム ID、サブプロセッサ ID、ロックシーケンス番号およびサブプロセッサが確保された各情報処理装置の情報処理装置 ID と共に送信する。

20

【0242】

図 25 および図 26 に示すように、情報処理装置 1 内のリソースマネージャは、ステップ 182 で、当該の機能プログラムから送信されたサブプロセッサプログラムを受信して、自装置 (情報処理装置 1) 分のサブプロセッサプログラムがある場合には、それを自装置内の確保されたサブプロセッサに送信し、さらにステップ 183 に進んで、他の情報処理装置分のサブプロセッサプログラムがあるか否かを判断し、他の情報処理装置分のサブプロセッサプログラムがない場合には、ステップ 183 からステップ 184 に進んで、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

30

【0243】

一方、当該の機能プログラムから送信されたサブプロセッサプログラムに他の情報処理装置分のサブプロセッサプログラムがある場合には、情報処理装置 1 内のリソースマネージャは、ステップ 183 からステップ 191 に進んで、他の情報処理装置 2, 3 および 4 内のリソースマネージャに各情報処理装置分のサブプロセッサプログラムを送信する。

【0244】

この送信には、図 30 に示すような DMA コマンドがサブプロセッサプログラム送信コマンドであるソフトウェアセルを用いる。具体的に、このサブプロセッサプログラム送信コマンドは、DMA コマンドとして、サブプロセッサプログラム送信コマンド本体、確保された各サブプロセッサのサブプロセッサ ID、ロックシーケンス番号、確保された各サブプロセッサに対するサブプロセッサプログラムのサブプロセッサプログラム ID、確保された各サブプロセッサに対するサブプロセッサプログラム本体を含むものとする。

40

【0245】

他の情報処理装置 2, 3 および 4 内のリソースマネージャは、ステップ 192 で、それぞれ自装置分のサブプロセッサプログラムを、自装置内の確保されたサブプロセッサに送信し、さらにステップ 193 に進んで、情報処理装置 1 内のリソースマネージャに、確保されたサブプロセッサにサブプロセッサプログラムを送信したことを通知する。この通知を受けて、情報処理装置 1 内のリソースマネージャは、ステップ 184 で、当該の機能プ

50

ログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0246】

なお、情報処理装置1内のリソースマネージャ、および他の情報処理装置2, 3および4内のリソースマネージャは、図25のステップ182、および図26のステップ192では、当該の機能プログラムから受信した確保された各サブプロセッサに対するサブプロセッサプログラムのサブプロセッサプログラムID、およびロックシーケンス番号が、自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断して、異なる場合には、情報処理装置1内のリソースマネージャが直接、または他の情報処理装置2, 3および4内のリソースマネージャが情報処理装置1内のリソースマネージャを介して、当該の機能プログラムにエラーを通知するが、図25および図26では、この点を省略して、エラーを生じない場合のみを示した。

10

【0247】

図26に示すように、当該の機能プログラムは、ステップ187で、情報処理装置1内のリソースマネージャからの、確保された各サブプロセッサにサブプロセッサプログラムを送信した旨の通知を受信する。

【0248】

これによって、情報処理装置1内の空いているサブプロセッサだけではn個のサブプロセッサを確保できないが、情報処理装置1内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、情報処理装置1内だけでn個のサブプロセッサを確保できる場合、あるいは、情報処理装置1内の空いているサブプロセッサだけではn個のサブプロセッサを確保できないが、情報処理装置1以外にもネットワーク上に情報処理装置が存在していて、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、n個のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

20

【0249】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

30

【0250】

一方、情報処理装置1内のリソースマネージャは、図23のステップ151で、ネットワーク上に他の情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体の空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、n個のサブプロセッサを確保できないと判断した場合には、図27に示すように、ステップ151からステップ195に進んで、当該の機能プログラムに、n個のサブプロセッサを確保できないことを通知する。

【0251】

当該の機能プログラムは、ステップ197で、この通知を受信し、エラー処理を実行する。この場合のエラー処理は、ユーザに対して直ちに処理を実行できない旨を通知して、時間を置いて再度、図19のステップ112からサブプロセッサ割り当て処理を実行し、またはユーザが操作することをユーザに促すなどである。

40

【0252】

なお、上述したように優先度の低いサブプロセッサプログラムに対してサブプロセッサが奪われたことが通知された場合には、当該のサブプロセッサプログラムにつき、それに対応する別の処理が実行される。

【0253】

上述した例は、情報処理装置IDの置き換えられた数値の昇順に、すなわち情報処理装置1 情報処理装置2 情報処理装置3 情報処理装置4の順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求する場合であるが、例えば、各情報処理

50

装置内のリソースマネージャは、実行されるべきサブプロセッサプログラムの優先度が高い場合には、上述した装置情報の一部であるメインプロセッサ動作周波数により、ネットワーク 9 上に接続されている各情報処理装置の動作周波数を判断して、動作周波数が高い情報処理装置から順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求するように、システムを構成してもよい。

【 0 2 5 4 】

あるいはまた、装置情報の一部である情報処理装置種別 I D により、ネットワーク 9 上に接続されている各情報処理装置の種別を判断することによって、ハードディスクレコーダのようにネットワーク 9 から切断される可能性が低い情報処理装置から、P D A やポータブル C D プレーヤのようにネットワーク 9 から切断される可能性が高い情報処理装置への順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求するように、システムを構成してもよい。

10

【 0 2 5 5 】

また、上述した例は、情報処理装置 1 内の空いているサブプロセッサだけでは必要数のサブプロセッサを確保できない場合、情報処理装置 1 内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、まずサブプロセッサの空き状況を問い合わせ、その結果、ネットワークシステム全体で必要数のサブプロセッサを確保できる場合には、改めてサブプロセッサの確保を要求する場合であるが、サブプロセッサの空き状況を問い合わせることなく、情報処理装置 1 内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、サブプロセッサの確保を要求するように、システム

20

【 0 2 5 6 】

さらに、上述した例は、情報処理装置 1 がマスター装置として動作していて、そのマスター装置を起点としてサブプロセッサ割り当て処理を実行する場合であるが、例えば、スレーブ装置として動作している情報処理装置がユーザによって操作された場合に、その操作されたスレーブ装置を起点としてサブプロセッサ割り当て処理を実行するように、システムを構成することもできる。

【 0 2 5 7 】

[ 4 . 情報処理装置がネットワークから切断された場合のサブプロセッサの割り当てによる分散処理その 1 ]

30

図 1 6 のように複数の情報処理装置 1 , 2 , 3 および 4 がネットワーク 9 に接続されている状態で、ユーザが誤って、ある情報処理装置をネットワーク 9 から切断した場合 ( 情報処理装置を物理的にネットワーク 9 から切り離した場合、または情報処理装置の主電源を遮断した場合 )、または、強い震動などの外的要因によって、ある情報処理装置がネットワーク 9 から切断された場合、その切断された情報処理装置内のサブプロセッサが、切断されていない情報処理装置が有するサブプロセッサプログラムの実行中または実行予約中であったときには、切断された情報処理装置内のサブプロセッサに代わって、切断されていない情報処理装置内のサブプロセッサが、そのサブプロセッサプログラムを実行することが望ましい。

【 0 2 5 8 】

40

そこで、このような場合には、以下のように、切断された情報処理装置内のサブプロセッサが実行中または実行予約中であったサブプロセッサプログラムの実行用に、切断されていない情報処理装置内のサブプロセッサを割り当てる。

【 0 2 5 9 】

( 4 - 1 . システム構成 )

図 3 1 に、図 1 6 のように 4 台の情報処理装置 1 , 2 , 3 および 4 がネットワーク 9 に接続されている状態で、情報処理装置 4 がネットワーク 9 から切断された場合を示す。

【 0 2 6 0 】

各情報処理装置のソフトウェア構成としては、図 1 7 に示したように、制御プログラムとしてリソースマネージャを含むものとし、そのリソースマネージャには、図 1 8 に示し

50

たようなサブプロセッサ管理テーブルを用意する。サブプロセッサ管理テーブルの内容は、上述したとおりである。

【0261】

(4-2. サブプロセッサ割り当て処理)

図32~図40に、図31のように情報処理装置1, 2, 3および4がネットワーク9に接続されている状態で、情報処理装置4がネットワーク9から切断された場合に、情報処理装置1が、自装置のメインメモリ26-1内の機能プログラムと連携動作するサブプロセッサプログラムの実行用に、情報処理装置1内または他の情報処理装置2, 3内にサブプロセッサを確保する場合の、情報処理装置1内のMS(マスター/スレーブ)マネージャ、情報処理装置1のメインメモリ26-1内の機能プログラム、情報処理装置1内のリソースマネージャ、および他の情報処理装置2および3内のリソースマネージャによって実行されるサブプロセッサ割り当て処理の一例を示す。

10

【0262】

情報処理装置4がネットワーク9から切断されたことは、情報処理装置1, 2および3内のMSマネージャによって検出される。

【0263】

そして、情報処理装置1内のMSマネージャは、情報処理装置4がネットワーク9から切断されたことを検出したら、図32に示すように、ステップ201で、自装置(情報処理装置1)内のリソースマネージャに、切断された情報処理装置4の情報処理装置IDを通知し、情報処理装置1内のリソースマネージャは、ステップ202で、自装置のメインメモリ26-1内の機能プログラムに、その情報処理装置IDを転送する。

20

【0264】

これを受けて、情報処理装置1のメインメモリ26-1内の機能プログラムは、ステップ203で、図19~図30に示して上述したようなサブプロセッサ割り当て処理の結果から、切断された情報処理装置4内のサブプロセッサが、自身(当該の機能プログラム)と連携動作するサブプロセッサプログラムを実行中であったか否かを判断する。

【0265】

この場合の「実行中」というのも、busy(現在使用されている状態)だけでなく、reserved(現在は使用されていないが、使用が予約されている状態)を含む。

【0266】

そして、切断された情報処理装置4内の全てのサブプロセッサが、当該の機能プログラムと連携動作するサブプロセッサプログラムを実行中でなかった場合には、以後の処理は不要であるので、そのままサブプロセッサ割り当て処理を終了する。

30

【0267】

一方、切断された情報処理装置4内の一部または全部のサブプロセッサが、当該の機能プログラムと連携動作するサブプロセッサプログラムを実行中であった場合には、当該の機能プログラムは、ステップ203からステップ212に進んで、自装置(情報処理装置1)内のリソースマネージャに、当該の機能プログラムの機能プログラムIDおよび当該機能プログラムと連携動作する、切断された情報処理装置4内のサブプロセッサにより実行中(上記のように実行予約中を含む)のサブプロセッサプログラムのサブプロセッサプログラムIDを添えて、切断された情報処理装置4内のサブプロセッサに代わって当該の機能プログラムと連携動作するサブプロセッサプログラムを実行するのに必要な数(m個とする)のサブプロセッサの使用要求を出力する。

40

【0268】

これに対して、情報処理装置1内のリソースマネージャは、図33に示すように、ステップ213で、自装置(情報処理装置1)の図18に示したサブプロセッサ管理テーブルを参照して、自装置内にm個のサブプロセッサの空きがあるか否かを判断し、m個以上のサブプロセッサが空いている場合には、ステップ213からステップ214に進んで、自装置内にm個のサブプロセッサを確保するとともに、その確保した各サブプロセッサにつき、サブプロセッサ管理テーブル内のサブプロセッサステータスを、unusedからr

50

reservedに書き替え、サブプロセッサプログラムID、機能プログラムIDおよびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替え、ロックシーケンス番号を発番して、サブプロセッサ管理テーブルに記述する。

【0269】

さらに、情報処理装置1内のリソースマネージャは、ステップ215に進んで、当該の機能プログラムに、自装置内にm個のサブプロセッサを確保したことと、確保したm個のサブプロセッサのサブプロセッサIDおよびロックシーケンス番号とを通知する。

【0270】

当該の機能プログラムは、ステップ217で、この通知を受信して、自装置（情報処理装置1）内のリソースマネージャに、自身の機能プログラムIDと共に、確保された各サブプロセッサに対するサブプロセッサID、サブプロセッサプログラムID、サブプロセッサプログラム本体およびロックシーケンス番号を、確保したサブプロセッサ数の分だけ送信する。

【0271】

図34に示すように、情報処理装置1内のリソースマネージャは、ステップ221で、これらを受信して、各サブプロセッサに対するサブプロセッサプログラムを保存し、さらにステップ222に進んで、受信したサブプロセッサプログラムIDおよびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断し、同じであるときには、ステップ223に進んで、確保した各サブプロセッサにサブプロセッサプログラムを送信し、さらにステップ224に進んで、当該の機能プログラムに、各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0272】

以上によって、情報処理装置1内の空いているサブプロセッサだけで必要数のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

【0273】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

【0274】

情報処理装置1内のリソースマネージャは、ステップ222で、当該の機能プログラムから受信したサブプロセッサプログラムIDまたはロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと異なると判断した場合には、ステップ222からステップ226に進んで、当該の機能プログラムにエラーを通知する。この場合には、当該の機能プログラムは、ステップ227で、エラー処理を実行する。

【0275】

一方、情報処理装置1内のリソースマネージャは、図33のステップ213で、自装置内にm個のサブプロセッサの空きがないと判断した場合、すなわち情報処理装置1内に空いているサブプロセッサが(m-1)個以下しかない(1個もない場合を含む)と判断した場合には、図35に示すように、ステップ213からステップ229に進んで、自装置内に空いているサブプロセッサがある場合には、そのサブプロセッサを自装置分として確保する。

【0276】

情報処理装置1内のリソースマネージャは、さらにステップ231に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。図31は、他の情報処理装置2および3が接続されている場合である。

【0277】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1内のリソースマネージャは、ステップ231からステップ232に進んで、その中の1

10

20

30

40

50

個である他の情報処理装置（この場合は情報処理装置 2）内のリソースマネージャに、空いているサブプロセッサの確保を要求する。

【0278】

この確保要求は、図 29（A）に示したようなサブプロセッサ確保要求コマンドによって行い、その確保要求サブプロセッサ数は、 $(m - a)$  とする。a は、ステップ 229 で情報処理装置 1 内に確保されたサブプロセッサ数で、 $0 < a < m$  である。

【0279】

図 35 に示すように、この確保要求に対して、他の情報処理装置（この場合は情報処理装置 2）内のリソースマネージャは、ステップ 233 で、自装置内のサブプロセッサ管理テーブルから、自装置内のサブプロセッサの空き状況を判断し、次にステップ 234 に進んで、自装置内に空いているサブプロセッサがある場合には、最大  $(m - a)$  個の範囲で自装置内にサブプロセッサを確保し、さらにステップ 235 に進んで、自装置内でサブプロセッサを確保した場合には、その数および確保した各サブプロセッサのサブプロセッサ ID、サブプロセッサプログラム ID およびロックシーケンス番号を情報処理装置 1 内のリソースマネージャに通知し、自装置内でサブプロセッサを確保できない場合には、その旨を情報処理装置 1 内のリソースマネージャに通知する。この通知は、図 29（B）に示したようなサブプロセッサ確保返信コマンドによって行う。

【0280】

この通知を受けて、情報処理装置 1 内のリソースマネージャは、図 36 に示すように、ステップ 236 で、自装置（情報処理装置 1）分と他装置（情報処理装置 2）分とを合わせてトータルで、m 個のサブプロセッサを確保できたか否かを判断する。

【0281】

そして、トータルで m 個のサブプロセッサを確保できなかった場合には、情報処理装置 1 内のリソースマネージャは、図 35 に示すように、ステップ 236 からステップ 231 に進んで（戻って）、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。図 31 は、情報処理装置 2 以外に他の情報処理装置 3 が接続されている場合である。

【0282】

以下、情報処理装置 2 の場合と同様に、ネットワークシステム全体で m 個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ 232、233、234、235、236 および 231 の処理を繰り返す。

【0283】

そして、ネットワークシステム全体で、空いているサブプロセッサによって m 個のサブプロセッサを確保できた場合には、情報処理装置 1 内のリソースマネージャは、図 36 に示すように、ステップ 236 からステップ 237 に進んで、当該の機能プログラムに、m 個のサブプロセッサを確保したことを通知する。

【0284】

当該の機能プログラムは、ステップ 238 で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置 1）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0285】

一方、情報処理装置 1 内のリソースマネージャは、図 35 のステップ 231 で、それ以上、ネットワーク上に情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体でも（情報処理装置 1 のみがネットワーク上に存在する場合には情報処理装置 1 内だけで）、空いているサブプロセッサによって m 個のサブプロセッサを確保できないと判断した場合には、図 36 に示すように、ステップ 231 からステップ 241 に進んで、自装置（情報処理装置 1）内に当該の機能プログラムより優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否か、および、ある場合には、その数を検出する。

10

20

30

40

50

## 【0286】

この場合の「実行中」というのも、`busy`（現在使用されている状態）だけでなく、`reserved`（現在は使用されていないが、使用が予約されている状態）を含む。以下でも、当該の機能プログラムより優先度の低いサブプロセッサプログラムを、単に「優先度の低いサブプロセッサプログラム」と称する。

## 【0287】

次に、情報処理装置1内のリソースマネージャは、ステップ242に進んで、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがある場合には、ネットワークシステム全体の空いているサブプロセッサと合わせてm個のサブプロセッサを確保できる範囲で、その優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保するとともに、その優先度の低いサブプロセッサプログラムに対してサブプロセッサが奪われたことを通知する。

10

## 【0288】

さらに、情報処理装置1内のリソースマネージャは、図37に示すように、ステップ243に進んで、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置1）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、m個のサブプロセッサを確保できたか否かを判断し、確保できた場合には、ステップ243からステップ244に進んで、当該の機能プログラムに、m個のサブプロセッサを確保したことを通知する。

## 【0289】

当該の機能プログラムは、ステップ248で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置1）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

20

## 【0290】

一方、情報処理装置1内のリソースマネージャは、ステップ243で、ネットワークシステム全体の空いているサブプロセッサと、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、m個のサブプロセッサを確保できないと判断した場合には、ステップ243からステップ251に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。

## 【0291】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1内のリソースマネージャは、ステップ251からステップ252に進んで、当該他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャに、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの確保を要求する。この確保要求は、図29（A）に示したようなサブプロセッサ確保要求コマンドによって行う。

30

## 【0292】

図38に示すように、この確保要求に対して、他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャは、ステップ253で、自装置内のサブプロセッサ管理テーブルから、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否かを判断し、ある場合には、その優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保して、その数および確保した各サブプロセッサのサブプロセッサID、サブプロセッサプログラムIDおよび発番したロックシーケンス番号を情報処理装置1内のリソースマネージャに通知し、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがない場合には、その旨を情報処理装置1内のリソースマネージャに通知する。この通知は、図29（B）に示したようなサブプロセッサ確保返信コマンドによって行う。

40

## 【0293】

なお、他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャは、ステップ253で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブ

50

プロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

【0294】

他の情報処理装置（この場合は情報処理装置2）内のリソースマネージャからの確保通知を受けて、情報処理装置1内のリソースマネージャは、図38に示すように、ステップ254で、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置1）内および他装置（情報処理装置2）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、m個のサブプロセッサを確保できたか否かを判断し、確保できない場合には、図37に示すように、ステップ254からステップ251に進んで（戻って）、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。

10

【0295】

以下、情報処理装置2の場合と同様に、ネットワークシステム全体でm個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ252、253、254および251の処理を繰り返す。

【0296】

そして、空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、ネットワークシステム全体でm個のサブプロセッサを確保できた場合には、情報処理装置1内のリソースマネージャは、ステップ254からステップ255に進んで、当該の機能プログラムに、m個のサブプロセッサを確保したことを通知する。

20

【0297】

当該の機能プログラムは、ステップ258で、このサブプロセッサ確保通知を受信し、以下のように自装置（情報処理装置1）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0298】

すなわち、当該の機能プログラムは、図36のステップ238、図37のステップ248、または図38のステップ258で、情報処理装置1内のリソースマネージャから、m個のサブプロセッサを確保した旨の通知を受信したら、図39に示すように、ステップ281で、情報処理装置1内のリソースマネージャに、確保された各サブプロセッサに対するサブプロセッサプログラムを送信する。

30

【0299】

図39に示すように、情報処理装置1内のリソースマネージャは、ステップ282で、当該の機能プログラムから送信されたサブプロセッサプログラムを受信して、自装置（情報処理装置1）分のサブプロセッサプログラムがある場合には、それを自装置内の確保されたサブプロセッサに送信し、さらにステップ283に進んで、他の情報処理装置分のサブプロセッサプログラムがあるか否かを判断し、他の情報処理装置分のサブプロセッサプログラムがない場合には、ステップ283からステップ284に進んで、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

40

【0300】

一方、当該の機能プログラムから送信されたサブプロセッサプログラムに他の情報処理装置分のサブプロセッサプログラムがある場合には、情報処理装置1内のリソースマネージャは、ステップ283からステップ291に進んで、他の情報処理装置2、3のリソースマネージャに各情報処理装置分のサブプロセッサプログラムを送信する。

【0301】

他の情報処理装置2、3のリソースマネージャは、ステップ292で、それぞれ自装置分のサブプロセッサプログラムを、自装置内の確保されたサブプロセッサに送信し、さらにステップ293に進んで、情報処理装置1内のリソースマネージャに、確保されたサブ

50

プロセッサにサブプロセッサプログラムを送信したことを通知する。この通知を受けて、情報処理装置 1 内のリソースマネージャは、ステップ 2 8 4 で、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【 0 3 0 2 】

なお、情報処理装置 1 内のリソースマネージャ、および他の情報処理装置 2 , 3 内のリソースマネージャは、図 3 5 のステップ 2 2 9 , 2 3 4、図 3 6 のステップ 2 4 2、または図 3 8 のステップ 2 5 3 で、自装置内にサブプロセッサを確保した場合には、その確保したサブプロセッサにつき、自装置内のサブプロセッサ管理テーブルを書き替えるとともに、情報処理装置 1 , 2 または 3 内のリソースマネージャが発番したロックシーケンス番号をサブプロセッサ管理テーブルに記述し、図 3 9 のステップ 2 8 2 および 2 9 2 では、当該の機能プログラムから受信したサブプロセッサプログラム ID およびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断して、異なる場合には、情報処理装置 1 内のリソースマネージャが直接、または他の情報処理装置 2 , 3 内のリソースマネージャが情報処理装置 1 内のリソースマネージャを介して、当該の機能プログラムにエラーを通知するが、図 3 6 ~ 図 3 9 では、この点を省略して、エラーを生じない場合のみを示した。

【 0 3 0 3 】

図 3 9 に示すように、当該の機能プログラムは、ステップ 2 8 7 で、情報処理装置 1 内のリソースマネージャからの、確保された各サブプロセッサにサブプロセッサプログラムを送信した旨の通知を受信する。

【 0 3 0 4 】

これによって、情報処理装置 1 内の空いているサブプロセッサだけでは m 個のサブプロセッサを確保できないが、情報処理装置 1 内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、情報処理装置 1 内だけで m 個のサブプロセッサを確保できる場合、あるいは、情報処理装置 1 内の空いているサブプロセッサだけでは m 個のサブプロセッサを確保できないが、情報処理装置 1 以外にもネットワーク上に情報処理装置が存在していて、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、m 個のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

【 0 3 0 5 】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

【 0 3 0 6 】

一方、情報処理装置 1 内のリソースマネージャは、図 3 7 のステップ 2 5 1 で、ネットワーク上に他の情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体の空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、m 個のサブプロセッサを確保できないと判断した場合には、図 4 0 に示すように、ステップ 2 5 1 からステップ 2 9 5 に進んで、当該の機能プログラムに、m 個のサブプロセッサを確保できないことを通知する。当該の機能プログラムは、ステップ 2 9 7 で、この通知を受信し、エラー処理を実行する。

【 0 3 0 7 】

上述した例は、情報処理装置 ID の置き換えられた数値の昇順に、すなわち情報処理装置 1 情報処理装置 2 情報処理装置 3 の順に、サブプロセッサの確保を要求する場合であるが、例えば、各情報処理装置内のリソースマネージャは、実行されるべきサブプロセッサプログラムの優先度が高い場合には、上述した装置情報の一部であるメインプロセッサ動作周波数により、ネットワーク 9 上に接続されている各情報処理装置の動作周波数を判断して、動作周波数が高い情報処理装置から順に、サブプロセッサの確保を要求するように、システムを構成してもよい。

10

20

30

40

50

## 【 0 3 0 8 】

あるいはまた、装置情報の一部である情報処理装置種別IDにより、ネットワーク9上に接続されている各情報処理装置の種別を判断することによって、ハードディスクレコーダのようにネットワーク9から切断される可能性が低い情報処理装置から、PDAやポータブルCDプレーヤのようにネットワーク9から切断される可能性が高い情報処理装置への順に、サブプロセッサの確保を要求するように、システムを構成してもよい。

## 【 0 3 0 9 】

また、上述した例は、情報処理装置1内の空いているサブプロセッサだけでは必要数のサブプロセッサを確保できない場合、情報処理装置1内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、直ちにサブプロセッサの確保を要求する  
10  
場合であるが、図19～図27に示して上述した例のように、情報処理装置1内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、まずサブプロセッサの空き状況を問い合わせ、その結果、ネットワークシステム全体で必要数のサブプロセッサを確保できる場合には、改めてサブプロセッサの確保を要求するように、システムを構成してもよい。

## 【 0 3 1 0 】

[ 5 . ネットワークシステムおよび情報処理装置の基本的構成その2 ]

図41は、この発明のネットワークシステムの他の例を示し、ネットワーク1009を介して複数の情報処理装置1001, 1002, 1003および1004が接続されたもの  
20  
のである。

## 【 0 3 1 1 】

( 5 - 1 . 情報処理装置および情報処理コントローラ )

情報処理装置1001, 1002, 1003および1004は、それぞれ後述のような各種のAV ( Audio and Visual ) 機器やポータブル機器である。

## 【 0 3 1 2 】

情報処理装置1001について示すと、情報処理装置1001は、コンピュータ機能部として情報処理コントローラ1011を備える。情報処理コントローラ1011は、メインプロセッサ1021-1、サブプロセッサ1023-1, 1023-2および1023-3、DMAC ( ダイレクトメモリアクセスコントローラ ) 1025-1、コントロールレジスタ1028-1、ワークメモリ1029-1、およびDC ( ディスクコントローラ ) 1030-1を有する。  
30

## 【 0 3 1 3 】

メインプロセッサ1021-1は、サブプロセッサ1023-1, 1023-2および1023-3によるサブプロセッサプログラムの実行 ( データ処理 ) のスケジュール管理と、情報処理コントローラ1011 ( 情報処理装置1001 ) の全般的な管理とを行う。ただし、メインプロセッサ1021-1内で管理のためのプログラム以外のプログラムが動作するように構成することもできる。その場合には、メインプロセッサ1021-1はサブプロセッサとしても機能することになる。メインプロセッサ1021-1は、LS ( ローカルストレージ ) 1022-1を有する。  
40

## 【 0 3 1 4 】

サブプロセッサは、1つでもよいが、望ましくは複数とする。この例は、複数の場合である。

## 【 0 3 1 5 】

各サブプロセッサ1023-1, 1023-2および1023-3は、メインプロセッサ1021-1の制御によって並列的かつ独立に、サブプロセッサプログラムを実行し、データを処理する。さらに、場合によってメインプロセッサ1021-1内のプログラムがサブプロセッサ1023-1, 1023-2または1023-3内のサブプロセッサプログラムと連携して動作するように構成することもできる。後述する機能プログラムも、メインプロセッサ1021-1内で動作するプログラムである。各サブプロセッサ1023-1, 1023-2および1023-3も、LS ( ローカルストレージ ) 1024-1  
50

、1024-2および1024-3を有する。

【0316】

DMAC1025-1は、それぞれ情報処理コントローラ1011に接続された、DRAM(ダイナミックRAM)などからなるメインメモリ1026-1およびSRAM(スタティックRAM)などからなるサブメモリ1027-1に格納されているプログラムおよびデータにアクセスするためのものである。DMAC1025-1を介すことにより、各サブプロセッサとメインメモリ1026-1との間ではDMA方式によりデータ転送を行うことができ、高速化を実現することができる。

【0317】

コントロールレジスタ1028-1は、情報処理コントローラ1011内で処理されるべきサブプロセッサプログラムを、どのサブプロセッサが、さらにはサブプロセッサ内に複数個存在する後述のプレセッシングスレッドのいずれが、処理するかを決定するとともに、サブプロセッサによるサブプロセッサプログラム実行の進捗を管理するために使用される。

10

【0318】

ワークメモリ1029-1は、情報処理コントローラ1011内に含まれるSRAMなどからなる作業用メモリであり、メインプロセッサ1021-1、およびサブプロセッサ1023-1、1023-2および1023-3によってアクセスされる。

【0319】

DC1030-1は、情報処理コントローラ1011に接続された外部記録部1031-1および1031-2にアクセスするものである。

20

【0320】

外部記録部1031-1および1031-2は、固定ディスク(ハードディスク)でも、リムーバブルディスクでもよく、また、MO、CD±RW、DVD±RWなどの光ディスク、メモリディスク、SRAM(スタティックRAM)、ROMなど、各種のものを利用することができる。したがって、DC1030-1は、ディスクコントローラと称するが、外部記録部コントローラである。

【0321】

図41の例のように、情報処理コントローラ1011に対して外部記録部1031を複数接続できるように、情報処理コントローラ1011を構成することができる。

30

【0322】

メインプロセッサ1021-1、各サブプロセッサ1023-1、1023-2および1023-3、DMAC1025-1、コントロールレジスタ1028-1、ワークメモリ1029-1、およびDC1030-1は、バス1032-1によって接続される。

【0323】

情報処理コントローラ1011には、当該の情報処理コントローラ1011を備える情報処理装置1001をネットワーク全体を通して一意的に識別できる識別子が、情報処理装置IDとして割り当てられる。

【0324】

メインプロセッサ1021-1、および各サブプロセッサ1023-1、1023-2および1023-3に対しても同様に、それぞれを特定できる識別子が、メインプロセッサIDおよびサブプロセッサIDとして割り当てられる。

40

【0325】

情報処理コントローラ1011は、ワンチップIC(集積回路)として構成することが望ましい。

【0326】

他の情報処理装置1002、1003および1004も、同様に構成される。ここで、親番号が同一であるユニットは、枝番号が異なっても、特に断りがない限り、同じ働きをするものとする。また、以下の説明において枝番号が省略されている場合には、枝番号の違いに在る差異を生じないものとする。

50

## 【 0 3 2 7 】

( 5 - 2 . 各サブプロセッサからメインメモリへのアクセス )

上述したように、1つの情報処理コントローラ内の各サブプロセッサ1023は、独立にサブプロセッサプログラムを実行し、データを処理するが、異なるサブプロセッサがメインメモリ1026内の同一領域に対して同時に読み出したりは書き込みを行った場合には、データの不整合を生じ得る。そこで、サブプロセッサ1023からメインメモリ1026へのアクセスは、以下のような手順によって行う。

## 【 0 3 2 8 】

サブメモリ1027およびワークメモリ1029についても同様に、異なるサブプロセッサが同一領域に対して同時にアクセスすることが考えられるが、ここでは、メインメモリ1026について示す。

## 【 0 3 2 9 】

図42(A)に示すように、メインメモリ1026は、アドレス指定可能な複数のメモリロケーション0~Nによって構成される。各メモリロケーションに対しては、データの状態を示す情報を格納するための追加セグメント0~Nが割り振られる。追加セグメントは、F/Eビット、サブプロセッサIDおよびLSアドレス(ローカルストレージアドレス)を含むものとされる。また、各メモリロケーションには、後述のアクセスキー0~Nも割り振られる。F/Eビットは、以下のように定義される。

## 【 0 3 3 0 】

F/Eビット=0は、サブプロセッサ1023によって読み出されている処理中のデータ、または空き状態であるため最新データではない無効データであり、読み出し不可であることを示す。また、F/Eビット=0は、当該メモリロケーションにデータ書き込み可能であることを示し、書き込み後に1に設定される。

## 【 0 3 3 1 】

F/Eビット=1は、当該メモリロケーションのデータがサブプロセッサ1023によって読み出されておらず、未処理の最新データであることを示す。当該メモリロケーションのデータは読み出し可能であり、サブプロセッサ1023によって読み出された後に0に設定される。また、F/Eビット=1は、当該メモリロケーションがデータ書き込み不可であることを示す。

## 【 0 3 3 2 】

さらに、上記F/Eビット=0(読み出し不可/書き込み可)の状態において、当該メモリロケーションについて読み出し予約を設定することは可能である。F/Eビット=0のメモリロケーションに対して読み出し予約を行う場合には、サブプロセッサ1023は、読み出し予約を行うメモリロケーションの追加セグメントに、読み出し予約情報として当該サブプロセッサ1023のサブプロセッサIDおよびLSアドレスを書き込む。

## 【 0 3 3 3 】

その後、データ書き込み側のサブプロセッサ1023によって、読み出し予約されたメモリロケーションにデータが書き込まれ、F/Eビット=1(読み出し可/書き込み不可)に設定されたとき、あらかじめ読み出し予約情報として追加セグメントに書き込まれたサブプロセッサIDおよびLSアドレスに読み出される。

## 【 0 3 3 4 】

複数のサブプロセッサによってデータを多段階に処理する必要がある場合、このように各メモリロケーションのデータの読み出し/書き込みを制御することによって、前段階の処理を行うサブプロセッサ1023が、処理済みのデータをメインメモリ1026内の所定アドレスに書き込んだ後に即座に、後段階の処理を行う別のサブプロセッサ1023が前処理後のデータを読み出すことが可能となる。

## 【 0 3 3 5 】

図42(B)に示すように、各サブプロセッサ1023内のLS1024も、アドレス指定可能な複数のメモリロケーション0~Lによって構成される。各メモリロケーションに対しては、同様に追加セグメント0~Lが割り振られる。追加セグメントは、ビジービ

10

20

30

40

50

ットを含むものとされる。

【0336】

サブプロセッサ1023がメインメモリ1026内のデータを自身のLS1024のメモリロケーションに読み出すときには、読み出し先のメモリロケーションに対応するビジービットを1に設定して予約する。ビジービットが1であるメモリロケーションには、他のデータは格納することができない。LS1024のメモリロケーションに読み出し後、ビジービットは0になり、任意の目的に使用できるようになる。

【0337】

図42(A)に示すように、さらに、各情報処理コントローラと接続されたメインメモリ1026には、複数のサンドボックスが含まれる。サンドボックスは、メインメモリ1026内の領域を画定するものであり、各サンドボックスは、各サブプロセッサ1023に割り当てられ、そのサブプロセッサが排他的に使用することができる。すなわち、各々のサブプロセッサ1023は、自身に割り当てられたサンドボックスを使用できるが、この領域を超えてデータのアクセスを行うことはできない。

10

【0338】

メインメモリ1026は、複数のメモリロケーション0~Nから構成されるが、サンドボックスは、これらのメモリロケーションの集合である。すなわち、1つのサンドボックスは、1つまたは複数のメモリロケーションから構成される。

【0339】

さらに、メインメモリ1026の排他的な制御を実現するために、図42(C)に示すようなキー管理テーブルが用いられる。キー管理テーブルは、情報処理コントローラ内のSRAMのような比較的高速のメモリに格納され、DMAC1025と関連付けられる。ただし、キー管理テーブルは、ワークメモリ1029に格納されてもよい。

20

【0340】

キー管理テーブル内には、情報処理コントローラ内のサブプロセッサの数と同数のエントリが存在し、各エントリには、サブプロセッサIDおよび、それに対応するサブプロセッサキーおよびキーマスクが関連付けられて格納される。

【0341】

サブプロセッサ1023がメインメモリ1026を使用する際のプロセスは、以下のとおりである。まず、サブプロセッサ1023はDMAC1025に、読み出しまたは書き込みのコマンドを送信する。このコマンドには、自身のサブプロセッサIDと、アクセス要求先であるメインメモリ1026内のアドレスが含まれる。

30

【0342】

DMAC1025は、このコマンドを実行する前に、キー管理テーブルを参照して、アクセス要求元のサブプロセッサ1023のサブプロセッサキーを調べる。次に、DMAC1025は、調べたアクセス要求元のサブプロセッサキーと、アクセス要求先であるメインメモリ1026内の図42(A)に示したメモリロケーションに割り振られたアクセスキーとを比較して、2つのキーが一致した場合にのみ、上記のコマンドを実行する。

【0343】

図42(C)に示したキー管理テーブル上のキーマスクは、その任意のビットが1になることによって、そのキーマスクに関連付けられたサブプロセッサキーの対応するビットが0または1になることができる。

40

【0344】

例えば、サブプロセッサキーが1010であるとする。通常、このサブプロセッサキーによって1010のアクセスキーを持つサンドボックスへのアクセスだけが可能である。しかし、このサブプロセッサキーと関連付けられたキーマスクが0001に設定されている場合には、キーマスクのビットが1に設定された桁のみにつき、サブプロセッサキーとアクセスキーとの一致判定がマスクされ、このサブプロセッサキー1010によって、アクセスキーが1010または1011のいずれかであるアクセスキーを持つサンドボックスへのアクセスが可能となる。

50

## 【 0 3 4 5 】

以上のようにして、メインメモリ 1 0 2 6 のサンドボックスの排他性が実現される。すなわち、1つの情報処理コントローラ内の複数のサブプロセッサによってデータを多段階に処理する必要がある場合、以上のように構成することによって、前段階の処理を行うサブプロセッサと、後段階の処理を行うサブプロセッサのみが、メインメモリ 1 0 2 6 内の所定アドレスにアクセスできるようになり、データを保護することができる。

## 【 0 3 4 6 】

例えば、以下のように使用することが考えられる。まず、情報処理装置の起動直後においては、キーマスクの値は全てゼロである。メインプロセッサ内のプログラムが実行され、サブプロセッサ内のサブプロセッサプログラムと連携動作するものとする。第1のサブ  
10  
プロセッサにより実行された処理結果データを一旦、メインメモリ 1 0 2 6 に格納し、第2のサブプロセッサに送信したいときには、該当するメインメモリ領域は、当然どちらのサブプロセッサからもアクセス可能である必要がある。そのような場合に、メインプロセッサ内のプログラムは、キーマスクの値を適切に変更し、複数のサブプロセッサからアクセスできるメインメモリ領域を設けることにより、サブプロセッサによる多段階的の処理を可能にする。

## 【 0 3 4 7 】

より具体的には、他の情報処理装置からのデータ 第1のサブプロセッサによる処理  
第1のメインメモリ領域 第2のサブプロセッサによる処理 第2のメインメモリ領域、  
という手順で多段階処理が行われるときには、  
20

第1のサブプロセッサのサブプロセッサキー：0 1 0 0、

第1のメインメモリ領域のアクセスキー：0 1 0 0、

第2のサブプロセッサのサブプロセッサキー：0 1 0 1、

第2のメインメモリ領域のアクセスキー：0 1 0 1、

のような設定のままだと、第2のサブプロセッサは第1のメインメモリ領域にアクセスすることができない。そこで、第2のサブプロセッサのキーマスクを0 0 0 1にすることにより、第2のサブプロセッサによる第1のメインメモリ領域へのアクセスを可能にすることができる。

## 【 0 3 4 8 】

( 5 - 3 . 各サブプロセッサからメインメモリおよびサブメモリへのアクセス )  
30

1つの情報処理コントローラ内の各サブプロセッサ 1 0 2 3 は、メインメモリ 1 0 2 6 に対してと同様に、サブメモリ 1 0 2 7 に対しても、読み出しまたは書き込みが可能であるが、そのコマンドには、いくつかの種類が考えられ、優先度の高低があることもあり得る。そこで以下に、各サブプロセッサ 1 0 2 3 が正確にメインメモリ 1 0 2 6 およびサブメモリ 1 0 2 7 を使用するための構成および手順を示す。

## 【 0 3 4 9 】

図 4 3 に示すように、D M A C 1 0 2 5 は内部に、メインメモリコントロールレジスタ 1 0 3 3 およびサブメモリコントロールレジスタ 1 0 3 4 を有する。メインメモリコントロールレジスタ 1 0 3 3 は、メインメモリ 1 0 2 6 にアクセスするためのものであり、サブメモリコントロールレジスタ 1 0 3 4 は、サブメモリ 1 0 2 7 にアクセスするためのもの  
40

## 【 0 3 5 0 】

メインメモリコントロールレジスタ 1 0 3 3 は、以下の4つのブロックから構成され、以下のように各々がメインメモリ 1 0 2 6 にアクセスするためのコマンドを格納する。

## 【 0 3 5 1 】

優先コマンドブロックには、優先度の高いコマンドを複数個格納する。そのコマンドは優先的に処理される。

## 【 0 3 5 2 】

通常コマンドブロック 1 には、通常のコマンドを複数個格納し、通常コマンドブロック 2 にも、通常のコマンドを複数個格納する。通常コマンドブロック 1 と通常コマンドプロ  
50

ック2との間には、機能的な差はないが、例えば、メインメモリ1026へのアクセス実行前のコマンドを一方のブロックに格納し、実行後のコマンドを他方のブロックに格納することにより、実行前と実行後のコマンドを、それぞれ集中的および連続的に処理することができる効果がある。

**【0353】**

オーダーコマンドブロックには、正しい順番で処理される必要のあるコマンドを複数個格納する。すなわち、各サブプロセッサ1023からのコマンドを、受信した順番に、コマンド送信元のサブプロセッサのサブプロセッサIDと共に格納する。これにより、受信した順番にコマンドを処理し、かつ同じ順番でコマンド送信元サブプロセッサに実行結果を返信することができる。

10

**【0354】**

サブメモリコントロールレジスタ1034も、以下の4つのブロックから構成され、以下のように各々がサブメモリ1027にアクセスするためのコマンドを格納する。

**【0355】**

優先コマンドブロックには、優先度の高いコマンドを複数個格納する。そのコマンドは優先的に処理される。

**【0356】**

リードコマンドブロックには、リードコマンドを複数個格納し、ライトコマンドブロックには、ライトコマンドを複数個格納する。

**【0357】**

待ちコマンドブロックには、サブメモリ1027内の任意の領域に対してアクセスを試みたが、対象領域がロックされていてアクセス不可であったコマンドを複数個格納する。ロック解除時には、そのコマンドは優先コマンドブロックに移動する。

20

**【0358】**

各サブプロセッサ1023は、例えば、図44に示すような構造のコマンドによって、メインメモリ1026またはサブメモリ1027にアクセスする。アクセス実行後のメインメモリ1026またはサブメモリ1027からの実行結果のレスポンスも、同一構造とされる。

**【0359】**

このコマンド/レスポンス構造において、コマンドタイプとしては、リードとライトがある。優先コマンド識別子は、当該コマンドが優先度の高いコマンドであることを示す。通常コマンド識別子は、メインメモリ1026へのアクセス時のものであり、当該コマンドが通常コマンドブロック1または通常コマンドブロック2に格納されることを示す。

30

**【0360】**

チェーンコマンド識別子も、メインメモリ1026へのアクセス時のものであり、当該コマンドが直前または直後のコマンドと共に連続的なアクセスが必要であることを示す。または、連続するコマンド列の通し番号でもよい。チェーンコマンド識別子がセットされたコマンドは、オーダーコマンドブロックに格納されるが、処理の優先度は、優先コマンドブロック内のコマンドより高いものとする。

**【0361】**

ここでのアドレスは、コマンドを実行するメインメモリ1026またはサブメモリ1027内のアドレスを示す。または、ワークメモリ1029内のアドレスでもよい。

40

**【0362】**

サブプロセッサ識別子は、コマンド送信元のサブプロセッサのサブプロセッサIDである。プレッシングスレッド識別子は、コマンド送信元の後述するプレッシングスレッドの識別子である。

**【0363】**

OK/NGは、コマンドの成功/失敗を示す。ここでのデータは、リードコマンド実行時にレスポンスに含まれる読み出されたデータ、またはライトコマンド実行時にコマンドに含まれる書き込まれるデータである。

50

## 【0364】

図43に示すように、DMAC1025内には、上述した各サブプロセッサ1023がメインメモリ1026に正確にアクセスするためのメインメモリコントロールレジスタ1033、および各サブプロセッサ1023がサブメモリ1027に正確にアクセスするためのサブメモリコントロールレジスタ1034のほかに、メインプロセッサ1021がメインメモリ1026またはサブメモリ1027にアクセスするためのリードコマンドおよびライトコマンドをそれぞれ複数個格納できるメインプロセッサ用リードコマンドブロック1035およびメインプロセッサ用ライトコマンドブロック1036を設けることもできる。アドレス変換レジスタ1037については、後述する。

## 【0365】

さらに、複数のサブプロセッサがDMAC1025に対して、優先度の同じメモリアクセスコマンドを同じタイミングで送信した場合には、以下の手順に従うものとする。すなわち、DMAC1025は、内部にメモリアクセスに成功した最後のサブプロセッサを示すポインタを有するものとし、複数のサブプロセッサから優先度の同じメモリアクセスコマンドを同じタイミングで受信したときには、当該ポインタより値が大きく、かつ当該ポインタとの差分が最も小さいコマンドを優先させるものとする。この場合、一番大きいポインタ値の次は、一番小さいポインタ値とする。

## 【0366】

(5-4. 各サブプロセッサからメインメモリに正確にアクセスするためのサブメモリへのアクセス)

DRAMなどにより構成されるメインメモリ1026、およびSRAMなどにより構成されるサブメモリ1027は、当然のことながら、各々の素子の構造が異なるため、使用目的も異なってくる。

## 【0367】

そこで、メインメモリ1026とサブメモリ1027を組み合わせる方法の一例として、各サブプロセッサ1023がメインメモリ1026にアクセスするときのアドレス変換をサブメモリ1027が担うことが考えられる。そのための構成および手順を、図45を用いて示す。

## 【0368】

前述したように、メインメモリ1026は、複数のメモリロケーションによって構成され、さらに1つのサンドボックスは、1つまたは複数のメモリロケーションから構成される。そして、各サブプロセッサ1023は、自身に割り当てられたサンドボックスを排他的に使用することができる。

## 【0369】

例えば、図45(A)に示すように、あるサブプロセッサ1023に、メインメモリ1026内のサンドボックス1, 2, 3が割り当てられているものとする。サンドボックス1は0x800~0x8FF、サンドボックス2は0x200~0x2FF、サンドボックス3は0xFF0~0xFFFとし、各サンドボックスは先頭アドレスによって識別されるものとする。すなわち、割り当てられる各サンドボックスは、アドレスが連続していてもよい。

## 【0370】

当該のサブプロセッサ1023は、これらサンドボックス1, 2, 3からデータを読み出すときには、まず、DMAC1025内のアドレス変換レジスタ1037にアクセスする。

## 【0371】

アドレス変換レジスタ1037は、サブプロセッサ1023と、当該サブプロセッサ1023に割り当てられたサンドボックスとを関連付けるレジスタであり、サブプロセッサ1023の数と同数のエントリから構成される。ここでは、Q1で示されるエントリが当該サブプロセッサ1023に対応するものとする。

## 【0372】

10

20

30

40

50

このとき、サブプロセッサ1023は、自身に対応するQ1の値を読み出す。さらに、読み出されたQ1の値(0x80)で示される、サブメモリ1027内の第1のアドレスから、データ(0x20)を読み出す。

【0373】

サンドボックス1の領域は、Q1の値に基づいて特定される。例えば、Q1の値を16倍した値がサンドボックス1の先頭アドレスを示し、その先頭アドレスに255を加算した値がサンドボックス1の終了アドレスを示す。したがって、上記のようにQ1の値が0x80であるときには、サンドボックス1の領域は0x800~0x8FFである。

【0374】

また、サブプロセッサ1023は、上記の第1のアドレスから読み出されたデータ(0x20)で示される、同じサブメモリ1027内の第2のアドレスから、データ(0xFF)を読み出す。

10

【0375】

サンドボックス2の領域は、この第2のアドレスから読み出されたデータに基づいて特定される。具体的に、サンドボックス1と同様に、第2のアドレスから読み出されたデータの値を16倍した値がサンドボックス2の先頭アドレスを示し、その先頭アドレスに255を加算した値がサンドボックス2の終了アドレスを示す。したがって、上記のように第2のアドレスから読み出されたデータが0xF0であるときには、サンドボックス2の領域は0x200~0x2FFである。

【0376】

20

さらに、サブプロセッサ1023は、上記の第2のアドレスから読み出されたデータ(0xF0)で示される、同じサブメモリ1027内の第3のアドレスから、データ(0x00)を読み出す。0x00は、割り当てられたサンドボックスの終了を示す。

【0377】

サンドボックス3の領域は、この第3のアドレスから読み出されたデータに基づいて特定される。具体的に、サンドボックス1,2と同様に、第3のアドレスから読み出されたデータの値を16倍した値がサンドボックス3の先頭アドレスを示し、その先頭アドレスに255を加算した値がサンドボックス3の終了アドレスを示す。したがって、上記のように第3のアドレスから読み出されたデータが0x00であるときには、サンドボックス3の領域は0xF00~0xFFFFである。

30

【0378】

以上のように、各サブプロセッサ1023は、自身に割り当てられたサンドボックスのアドレスが非連続であっても、割り当てられたサンドボックスに正確にアクセスすることができ、割り当てられたサンドボックスからデータを確実に読み出すことができる。

【0379】

次に、同じサブプロセッサ1023が、メインメモリ1026内の新たなサンドボックスにデータを書き込み、当該サンドボックスをアドレス変換レジスタ1037内の自身に対応したエントリで管理されるサンドボックスグループに追加するための手順を、図45(B)を用いて示す。

【0380】

40

当該サブプロセッサ1023が、メインメモリ1026内の0x000~0x0FFの領域であるサンドボックス4にデータを書き込んだとする。このとき、当該サブプロセッサ1023は、まず、Q1の値(Q1の初期値、例えば上記のように0x80)を読み出す。

【0381】

次に、当該サブプロセッサ1023は、新たにデータを書き込んだサンドボックス4を特定できるサブメモリ1027内のアドレスに、先に読み出したQ1の値(0x80)を書き込む。例えば、今回は0x000~0x0FFの領域にデータを書き込んだので、先頭アドレスである0x000を16で割った0x00で示されるサブメモリ1027内のアドレスにQ1の値(0x80)を書き込む。さらに、Q1の値(0x80)を書き込ん

50

だサブメモリ1027内のアドレス(0x00)を新たなQ1の値として、アドレス変換レジスタ1037内のQ1に書き込む。

【0382】

以上のようにして、各サブプロセッサ1023は、新たなサンドボックスを既存のサンドボックスグループに追加することができる。この場合も、各サンドボックスのアドレスが非連続であってもよい。また、既存のサンドボックスグループが存在せず、最初のサンドボックスをサブメモリ1027およびアドレス変換レジスタ1037と関連付ける際にも、上記の方法を用いることができる。

【0383】

(5-5. メインプロセッサおよび各サブプロセッサからワークメモリへのアクセス)

10

前述したように、メインメモリ1026は、DRAMなどにより構成され、さらにDMA方式によりデータが転送されるため、各サブプロセッサ1023は、大容量のメインメモリ1026を高速に使用することができる。また、サブメモリ1027は、SRAMなどにより構成され、同様に高速に使用することができる。

【0384】

さらに、メインプロセッサ1021および各サブプロセッサ1023が、情報処理コントローラに接続されたメインメモリ1026およびサブメモリ1027と共に、情報処理コントローラ内に含まれるワークメモリ1029を作業用メモリとして共有して使用できれば、さらなる高速化を実現することができる。

20

【0385】

また、当該ワークメモリ1029により簡単な数値演算ができれば、効率的である。ワークメモリ1029は、前述したようにSRAMなどにより構成されるため、DRAMのような大容量化は期待できないものの、非常に高速である。

【0386】

以下に、メインプロセッサ1021および各サブプロセッサ1023がワークメモリ1029にアクセスする場合の構成および手順を示す。

【0387】

図46に示すように、ワークメモリ1029は、コントローラ1038とRAM1039とから構成される。RAM1039としては、SRAMを用いることができるが、それに限定されるものではない。RAM1039は、複数のブロックから構成される。各ブロックには、アドレスが割り当てられ、データが記憶される。

30

【0388】

メインプロセッサ1021および各サブプロセッサ1023は、コントローラ1038を介してRAM1039にアクセスする。具体的に、メインプロセッサ1021および各サブプロセッサ1023は、コントローラ1038にコマンド、アドレスまたはデータなどを送信し、コントローラ1038は、それに従ってRAM1039にアクセスする。

【0389】

処理実行後、コントローラ1038は、コマンド送信元のメインプロセッサ1021またはサブプロセッサ1023にコマンド実行結果を返信する。

40

【0390】

各サブプロセッサ1023がワークメモリ1029にアクセスするときのコマンドは、例えば図44に示したような、メインメモリ1026またはサブメモリ1027へのアクセス時に用いるものと同じである。処理実行後のワークメモリ1029からの実行結果のレスポンスも、同一構造である。

【0391】

ただし、ワークメモリ1029にアクセスする場合には、図44に示したコマンド中の優先コマンド識別子、通常コマンド識別子、チェーンコマンド識別子、プレセッシングスレッド識別子は、基本的に使用しない。しかし、ワークメモリ1029が、これらの識別子に対応可能であれば、使用してもよい。コマンドタイプとしては、以下に示すような、い

50

くつかが考えられる。

【0392】

第1は、リードコマンドである。これは、ワークメモリ1029内のデータを読み出すために使用される。メインプロセッサ1021および各サブプロセッサ1023は、リードコマンドと共に、所望のデータを記憶するRAM1039内のブロックのアドレスも送信する。コントローラ1038からは、実行結果として、リードコマンドの成功/失敗を示すOK/NGと、読み出されたデータが返信される。

【0393】

第2は、ライトコマンドである。これは、ワークメモリ1029内にデータを書き込むために使用される。メインプロセッサ1021および各サブプロセッサ1023は、ライトコマンド、データ、当該データを記憶させるRAM1039内のブロックのアドレスを送信する。コントローラ1038からは、実行結果として、ライトコマンドの成功/失敗を示すOK/NGが返信される。

【0394】

第3は、加算コマンドである。これは、ワークメモリ1029内のデータを加算するために使用される。メインプロセッサ1021および各サブプロセッサ1023は、加算コマンド、加算させるデータを記憶するRAM1039内のブロックのアドレスを送信する。コントローラ1038は、受信したアドレスのブロックにおけるデータに1を加算し、上書きする。実行結果として、加算コマンドの成功/失敗を示すOK/NGが返信される。

【0395】

第4は、セットコマンドである。これは、ワークメモリ1029内のデータをビット単位で操作するために使用される。メインプロセッサ1021および各サブプロセッサ1023は、セットコマンド、操作するデータを記憶するRAM1039内のブロックのアドレス、およびマスクデータを送信する。

【0396】

これに対して、コントローラ1038は、受信したマスクデータと、受信したアドレスのブロックにおけるデータとを比較し、マスクデータにおいて値が1であるビットと同じ位置の、データにおけるビットの値を1にする。実行結果として、セットコマンドの完了を示すCompletedが返信される。このとき、セットコマンド実行前のデータも返信することにより、コマンドの成功/失敗を確認できるようにしてもよい。

【0397】

第5は、クリアコマンドである。これも、ワークメモリ1029内のデータをビット単位で操作するために使用される。メインプロセッサ1021および各サブプロセッサ1023は、クリアコマンド、操作するデータを記憶するRAM1039内のブロックのアドレス、およびマスクデータを送信する。

【0398】

これに対して、コントローラ1038は、受信したマスクデータと、受信したアドレスのブロックにおけるデータとを比較し、マスクデータにおいて値が1であるビットと同じ位置の、データにおけるビットの値を0にする。実行結果として、クリアコマンドの完了を示すCompletedが返信される。このとき、クリアコマンド実行前のデータも返信することにより、コマンドの成功/失敗を確認できるようにしてもよい。

【0399】

以上のように、メインプロセッサ1021および各サブプロセッサ1023は、大容量のメインメモリ1026および高速のサブメモリ1027に加えて、ワークメモリ1029を併用することができる。さらに、ワークメモリ1029をメインメモリ1026またはサブメモリ1027のキャッシュとして利用すれば、さらなる高速化を期待することができる。

【0400】

(5-6. 各サブプロセッサ内のプレセッシングスレッド)

10

20

30

40

50

前述したように、1つの情報処理コントローラ内の各サブプロセッサ1023は構造上独立している。したがって、各々のサブプロセッサ1023が独立にサブプロセッサプログラムを実行し、データを処理することができる。さらに、各サブプロセッサ1023が内部に、仮想的に独立した複数のプレセッシングスレッドを有することも考えられる。その構造を図47に示す。

**【0401】**

サブプロセッサ1023は、内部に有するアービタ1040を介して、バス1032と接続される。サブプロセッサ1023内には、また、LS(ローカルストレージ)1024とプレセッシングスレッド1041, 1042, 1043および1044が含まれ、アービタ1040は、外部からの信号を適切なプレセッシングスレッドに通知する役割を担う。

10

**【0402】**

図47では各プレセッシングスレッド1041, 1042, 1043および1044を独立に示しているが、あくまで仮想的な独立である。プレセッシングスレッド1041, 1042, 1043および1044は、それぞれプレセッシングスレッド識別子が割り当てられ、独立かつ並列に動作可能である。

**【0403】**

各プレセッシングスレッド1041, 1042, 1043および1044は、メインメモリ1026、サブメモリ1027またはワークメモリ1029に対して独立にアクセスを行うため、コマンド送信元のプレセッシングスレッドに確実にレスポンスが返信されるようにする必要がある。そのための手順を以下に示す。

20

**【0404】**

各プレセッシングスレッド1041, 1042, 1043および1044が各々のメモリにアクセスするときのコマンドは、例えば図44に示したような、各サブプロセッサ1023がメモリにアクセスする時に用いるものと同じである。処理実行後の各メモリからの実行結果のレスポンスも、同一構造である。

**【0405】**

図44のコマンド/レスポンス構造については前述したが、サブプロセッサ識別子はコマンド送信元のサブプロセッサのサブプロセッサIDであり、さらにプレセッシングスレッド識別子は、当該サブプロセッサ内のいずれのプレセッシングスレッドがコマンド送信元であることを識別するためのものである。

30

**【0406】**

メインメモリ1026、サブメモリ1027またはワークメモリ1029からのレスポンスは、まず、サブプロセッサ識別子に基づいて、コマンド送信元のサブプロセッサ1023に返信される。さらに、プレセッシングスレッド識別子に基づいて、サブプロセッサ1023内のアービタ1040により、コマンド送信元のプレセッシングスレッドに返信される。

**【0407】**

ただし、同じ処理を行うようなときなど、サブプロセッサ内の複数のプレセッシングスレッド中のいずれでもよい場合には、プレセッシングスレッド識別子に基づかずに、処理負荷の軽いプレセッシングスレッドに返信するようにしてもよい。また、メインメモリ1026、サブメモリ1027またはワークメモリ1029からレスポンスを受信するたびに、複数のプレセッシングスレッド中の1つを順繰りに選択し、その選択されたプレセッシングスレッドに当該レスポンスを返信するようにしてもよい。

40

**【0408】**

以上のように、各サブプロセッサ1023内の複数のプレセッシングスレッドが、メインメモリ1026、サブメモリ1027またはワークメモリ1029に対して独立にアクセスを行う場合でも、コマンド送信元のプレセッシングスレッドに確実にレスポンスが返信されるようにすることができる。

**【0409】**

50

## ( 5 - 7 . コントロールレジスタによるサブプロセッサプログラムの管理 )

1つの情報処理コントローラ内に複数のサブプロセッサ1023が存在し、さらに各サブプロセッサ1023内に複数のプレセッシングスレッドを有する場合に、情報処理コントローラ内で処理されるべきサブプロセッサプログラムを、どのプレセッシングスレッドに処理させるかを決定することは、情報処理コントローラの高速度を実現する上で重要である。

## 【0410】

そこで以下に、図41に示したコントロールレジスタ1028を使用することにより、各プレセッシングスレッドに対して適切にサブプロセッサプログラムの処理を割り当て、情報処理コントローラを効率的に動作させる構成および手順を示す。

10

## 【0411】

図48に示すように、コントロールレジスタ1028は、処理待ちサブプロセッサプログラムレジスタ1045とサブプロセッサプログラム処理進捗レジスタ1046とから構成される。

## 【0412】

処理待ちサブプロセッサプログラムレジスタ1045について示すと、メインプロセッサ1021は、情報処理コントローラ内で処理されるべきサブプロセッサプログラムが発生した場合には、処理待ちサブプロセッサプログラムレジスタ1045内に、当該サブプロセッサプログラムまたはこれに関連するデータが格納されているメインメモリ1026、サブメモリ1027、ワークメモリ1029、またはLS（ローカルストレージ）1022または1024におけるアドレスを書き込む。

20

## 【0413】

処理されるべきサブプロセッサプログラムが存在しない状態では、処理待ちサブプロセッサプログラムレジスタ1045の値はゼロである。全てのプレセッシングスレッドは、自身が何もサブプロセッサプログラムを実行していない間は、定期的または非定期的処理待ちサブプロセッサプログラムレジスタ1045の値を読み出し、その結果、ゼロでない値を読み出したプレセッシングスレッドが処理を行うものとする。

## 【0414】

これとともに、当該プレセッシングスレッドは、処理待ちサブプロセッサプログラムレジスタ1045にゼロの値を書き込む。処理を行うプレセッシングスレッドは、読み出した処理待ちサブプロセッサプログラムレジスタ1045の値に基づいて、処理対象となるサブプロセッサプログラムまたは関連データを読み出し、処理する。このとき、サブプロセッサプログラムは既に当該プレセッシングスレッドを有するサブプロセッサ1023に読み出されており、サブプロセッサプログラムの読み出しは不要であることも考えられる。

30

## 【0415】

このようにして、サブプロセッサプログラムを実行していない、処理能力に余裕のあるプレセッシングスレッドに迅速にサブプロセッサプログラムの処理を割り当てることができ、情報処理コントローラを効率的に動作させることができる。

## 【0416】

サブプロセッサプログラム処理進捗レジスタ1046は、サブプロセッサプログラムの処理を割り当てられたプレセッシングスレッドが、処理進捗状況を書き込むための2ビット(x, y)のレジスタである。例えば、(0, 0)は未処理を示し、(0, 1)は処理中の段階1を示し、(1, 0)は処理中の段階2を示し、(1, 1)は処理完了を示す。さらに、この2ビットとともに、プロセッシングスレッド識別子を書き込むことにより、処理を割り当てられたプレセッシングスレッドを示すようにしてもよい。

40

## 【0417】

または、図48にプレセッシングスレッド0, 1, 2, 3として示すように、情報処理コントローラ内の各プロセッシングスレッドごとに、サブプロセッサプログラムの処理進捗状況を書き込むためのレジスタを設けてもよい。

50

## 【 0 4 1 8 】

サブプロセッサプログラム処理進捗レジスタ 1 0 4 6 は、情報処理コントローラ内のメインプロセッサ 1 0 2 1、全てのサブプロセッサ 1 0 2 3 および全てのプレセッシングスレッドがアクセス可能であり、これにより、サブプロセッサプログラムの処理進捗状況を的確に把握することができる。さらに、各プレセッシングスレッドごとに処理進捗レジスタを設ける場合には、複数のサブプロセッサプログラムが同時に実行される場合にも処理進捗状況の把握が可能となる。

## 【 0 4 1 9 】

以上が、各サブプロセッサ 1 0 2 3 が内部に、仮想的に独立した複数のプレセッシングスレッドを有する場合のプレセッシングスレッド管理方法の例である。

10

## 【 0 4 2 0 】

なお、以下においては、サブプロセッサが何らかの処理を行う場合には、当該サブプロセッサ内部のいずれのプレセッシングスレッドが処理を行っても構わないものとし、プレセッシングスレッドの違いによる実行結果の差異は生じないものとする。したがって、サブプロセッサの処理内容につき、サブプロセッサ内部の複数のプレセッシングスレッドがどの様に処理を分担するかについては説明を省略する。

## 【 0 4 2 1 】

( 5 - 8 . ソフトウェアセルの生成および構成 )

図 4 1 のネットワークシステムでは、情報処理装置 1 0 0 1 , 1 0 0 2 , 1 0 0 3 および 1 0 0 4 の間での分散処理のために、情報処理装置 1 0 0 1 , 1 0 0 2 , 1 0 0 3 および 1 0 0 4 の間でソフトウェアセルが伝送される。すなわち、ある情報処理装置内の情報処理コントローラに含まれるメインプロセッサ 1 0 2 1 は、コマンド、プログラムおよびデータを含むソフトウェアセルを生成し、ネットワーク 1 0 0 9 を介して他の情報処理装置に送信することによって、処理を分散することができる。

20

## 【 0 4 2 2 】

図 3 に、ソフトウェアセルの構成の一例を示す。この例のソフトウェアセルは、全体として、送信元 ID、送信先 ID、応答先 ID、セルインターフェース、DMA コマンド、プログラム、およびデータによって構成される。

## 【 0 4 2 3 】

送信元 ID には、ソフトウェアセルの送信元である情報処理装置のネットワークアドレスおよび情報処理装置 ID、さらに、その情報処理装置内の情報処理コントローラが備えるメインプロセッサ 1 0 2 1 および各サブプロセッサ 1 0 2 3 の識別子 ( メインプロセッサ ID およびサブプロセッサ ID ) が含まれる。

30

## 【 0 4 2 4 】

送信先 ID および応答先 ID には、それぞれ、ソフトウェアセルの送信先である情報処理装置、およびソフトウェアセルの実行結果の応答先である情報処理装置についての、同じ情報が含まれる。

## 【 0 4 2 5 】

セルインターフェースは、ソフトウェアセルの利用に必要な情報であり、グローバル ID、必要なサブプロセッサの情報、サンドボックスサイズ、および前回のソフトウェアセル ID から構成される。

40

## 【 0 4 2 6 】

グローバル ID は、ネットワーク全体を通して当該のソフトウェアセルを一意的に識別できるものであり、送信元 ID、およびソフトウェアセルの作成または送信の日時 ( 日付および時刻 ) に基づいて作成される。

## 【 0 4 2 7 】

必要なサブプロセッサの情報は、当該ソフトウェアセルの実行に必要なサブプロセッサの数が設定される。サンドボックスサイズは、当該ソフトウェアセルの実行に必要なメインメモリ 1 0 2 6 内およびサブプロセッサ 1 0 2 3 の L S 1 0 2 4 内のメモリ量が設定される。

50

## 【 0 4 2 8 】

前回のソフトウェアセル I D は、ストリーミングデータなどのシーケンシャルな実行を要求する 1 グループのソフトウェアセル内の、前回のソフトウェアセルの識別子である。

## 【 0 4 2 9 】

ソフトウェアセルの実行セクションは、DMA コマンド、プログラムおよびデータから構成される。DMA コマンドには、プログラムの起動に必要な一連の DMA コマンドが含まれ、プログラムには、サブプロセッサ 1 0 2 3 によって実行されるサブプロセッサプログラムが含まれる。ここでのデータは、このサブプロセッサプログラムを含むプログラムによって処理されるデータである。

## 【 0 4 3 0 】

さらに、DMA コマンドには、ロードコマンド、キックコマンド、機能プログラム実行コマンド、ステータス要求コマンド、およびステータス返信コマンドが含まれる。

## 【 0 4 3 1 】

ロードコマンドは、メインメモリ 1 0 2 6 内の情報をサブプロセッサ 1 0 2 3 内の L S 1 0 2 4 にロードするコマンドであり、ロードコマンド自体のほかに、メインメモリアドレス、サブプロセッサ I D および L S アドレスを含む。メインメモリアドレスは、情報のロード元であるメインメモリ 1 0 2 6 内の所定領域のアドレスを示す。サブプロセッサ I D および L S アドレスは、情報のロード先であるサブプロセッサ 1 0 2 3 の識別子および L S 1 0 2 4 のアドレスを示す。

## 【 0 4 3 2 】

キックコマンドは、サブプロセッサプログラムの実行を開始するコマンドであり、キックコマンド自体のほかに、サブプロセッサ I D およびプログラムカウンタを含む。サブプロセッサ I D は、キック対象のサブプロセッサ 1 0 2 3 を識別し、プログラムカウンタは、サブプロセッサプログラム実行用プログラムカウンタのためのアドレスを与える。

## 【 0 4 3 3 】

機能プログラム実行コマンドは、後述のように、ある情報処理装置が他の情報処理装置に対して、機能プログラムの実行を要求するコマンドである。機能プログラム実行コマンドを受信した情報処理装置内の情報処理コントローラは、後述の機能プログラム I D によって、起動すべき機能プログラムを識別する。

## 【 0 4 3 4 】

ステータス要求コマンドは、送信先 I D で示される情報処理装置の現在の動作状態（状況）に関する装置情報を、応答先 I D で示される情報処理装置宛に送信することを要求するコマンドである。機能プログラムについては後述するが、図 6 に示す情報処理装置のメインメモリ 1 0 2 6 が記憶するソフトウェアの構成図において機能プログラムにカテゴライズされるプログラムである。機能プログラムは、メインメモリ 1 0 2 6 にロードされ、メインプロセッサ 1 0 2 1 により実行される。

## 【 0 4 3 5 】

ステータス返信コマンドは、上記のステータス要求コマンドを受信した情報処理装置が、自身の装置情報を当該ステータス要求コマンドに含まれる応答先 I D で示される情報処理装置に返信するコマンドである。ステータス返信コマンドは、実行セクションのデータ領域に装置情報を格納する。

## 【 0 4 3 6 】

図 4 に、DMA コマンドがステータス返信コマンドである場合におけるソフトウェアセルのデータ領域の構造を示す。

## 【 0 4 3 7 】

情報処理装置 I D は、情報処理コントローラを備える情報処理装置を識別するための識別子であり、ステータス返信コマンドを送信する情報処理装置の I D を示す。情報処理装置 I D は、電源投入時、その情報処理装置内の情報処理コントローラに含まれるメインプロセッサ 1 0 2 1 によって、電源投入時の日時、情報処理装置のネットワークアドレスおよび情報処理装置内の情報処理コントローラに含まれるサブプロセッサ 1 0 2 3 の数など

10

20

30

40

50

に基づいて生成される。

【0438】

情報処理装置種別IDには、当該の情報処理装置の特徴を表す値が含まれる。情報処理装置の特徴とは、例えば、後述のハードディスクレコーダ、PDA(Personal Digital Assistants)、ポータブルCD(Compact Disc)プレーヤなどである。また、情報処理装置種別IDは、映像音声記録、映像音声再生など、情報処理装置の機能を表すものであってもよい。情報処理装置の特徴や機能を表す値は予め決定されているものとし、情報処理装置種別IDを読み出すことにより、当該情報処理装置の特徴や機能を把握することが可能である。

【0439】

MS(マスター/スレーブ)ステータスは、後述のように情報処理装置がマスター装置またはスレーブ装置のいずれで動作しているかを表すもので、これが0に設定されている場合にはマスター装置として動作していることを示し、1に設定されている場合にはスレーブ装置として動作していることを示す。

【0440】

メインプロセッサ動作周波数は、情報処理コントローラ内のメインプロセッサ1021の動作周波数を表す。メインプロセッサ使用率は、メインプロセッサ1021で現在動作している全てのプログラムについての、メインプロセッサ1021での使用率を表す。メインプロセッサ使用率は、対象メインプロセッサの全処理能力に対する使用中の処理能力の比率を表した値で、例えばプロセッサ処理能力評価のための単位であるMIPSを単位として算出され、または単位時間あたりのプロセッサ使用時間に基づいて算出される。後述のサブプロセッサ使用率についても同様である。

【0441】

サブプロセッサ数は、当該の情報処理コントローラが備えるサブプロセッサ1023の数を表す。サブプロセッサIDは、当該の情報処理コントローラ内の各サブプロセッサ1023を識別するための識別子である。

【0442】

サブプロセッサステータスは、各サブプロセッサ1023の状態を表すものであり、unused、reserved、busyなどの状態がある。unusedは、当該のサブプロセッサが現在使用されてなく、使用の予約もされていないことを示す。reservedは、現在は使用されていないが、予約されている状態を示す。busyは、現在使用中であることを示す。

【0443】

サブプロセッサ使用率は、当該のサブプロセッサで現在実行している、または当該のサブプロセッサに実行が予約されているサブプロセッサプログラムについての、当該サブプロセッサでの使用率を表す。すなわち、サブプロセッサ使用率は、サブプロセッサステータスがbusyである場合には、現在の使用率を示し、サブプロセッサステータスがreservedである場合には、後に使用される予定の推定使用率を示す。

【0444】

サブプロセッサID、サブプロセッサステータスおよびサブプロセッサ使用率は、1つのサブプロセッサ1023に対して一組設定され、1つの情報処理コントローラ内のサブプロセッサ1023の数の組数だけ設定される。

【0445】

メインメモリ総容量およびメインメモリ使用量は、それぞれ、当該の情報処理コントローラに接続されているメインメモリ1026の総容量および現在使用中の容量を表す。

【0446】

外部記録部数は、当該の情報処理コントローラに接続されている外部記録部1031の数を表す。外部記録部IDは、当該の情報処理コントローラに接続されている外部記録部1031を一意的に識別する情報である。外部記録部種別IDは、当該の外部記録部の種類(例えば、ハードディスク、CD±RW、DVD±RW、メモリディスク、SRAM、

10

20

30

40

50

ROMなど)を表す。

【0447】

外部記録部総容量および外部記録部使用量は、それぞれ、外部記録部IDによって識別される外部記録部1031の総容量および現在使用中の容量を表す。

【0448】

外部記録部ID、外部記録部種別ID、外部記録部総容量および外部記録部使用量は、1つの外部記録部1031に対して一組設定され、当該の情報処理コントローラに接続されている外部記録部1031の数の組数だけ設定される。すなわち、1つの情報処理コントローラに複数の外部記録部が接続されている場合、各々の外部記録部には異なる外部記録部IDが割り当てられ、外部記録部種別ID、外部記録部総容量および外部記録部使用量も別々に管理される。

10

【0449】

(5-9.ソフトウェアセルの実行)

ある情報処理装置内の情報処理コントローラに含まれるメインプロセッサ1021は、以上のような構成のソフトウェアセルを生成し、ネットワーク1009を介して他の情報処理装置および当該装置内の情報処理コントローラに送信する。送信元の情報処理装置、送信先の情報処理装置、応答先の情報処理装置、および各装置内の情報処理コントローラは、それぞれ、上記の送信元ID、送信先IDおよび応答先IDによって識別される。

【0450】

ソフトウェアセルを受信した情報処理装置内の情報処理コントローラに含まれるメインプロセッサ1021は、そのソフトウェアセルをメインメモリ1026に格納する。さらに、送信先のメインプロセッサ1021は、ソフトウェアセルを読み出し、それに含まれるDMAコマンドを処理する。

20

【0451】

具体的に、送信先のメインプロセッサ1021は、まず、ロードコマンドを実行する。これによって、ロードコマンドで指示されたメインメモリアドレスから、ロードコマンドに含まれるサブプロセッサIDおよびLSアドレスで特定されるサブプロセッサ内のLS1024の所定領域に、情報がロードされる。ここでロードされる情報は、受信したソフトウェアセルに含まれるサブプロセッサプログラムまたはデータ、あるいはその他の指示されたデータである。

30

【0452】

次に、メインプロセッサ1021は、キックコマンドを、これに含まれるサブプロセッサIDで指示されたサブプロセッサに、同様にキックコマンドに含まれるプログラムカウンタと共に送信する。

【0453】

指示されたサブプロセッサは、そのキックコマンドおよびプログラムカウンタに従って、サブプロセッサプログラムを実行する。そして、実行結果をメインメモリ1026に格納した後、実行を完了したことをメインプロセッサ1021に通知する。

【0454】

なお、送信先の情報処理装置内の情報処理コントローラにおいてソフトウェアセルを実行するプロセッサはサブプロセッサ1023に限定されるものではなく、メインプロセッサ1021がソフトウェアセルに含まれる機能プログラムなどのメインメモリ用プログラムを実行するように指定することも可能である。

40

【0455】

この場合には、送信元の情報処理装置は、送信先の情報処理装置宛に、サブプロセッサプログラムの代わりに、メインメモリ用プログラムおよびそのメインメモリ用プログラムによって処理されるデータを含み、DMAコマンドがロードコマンドであるソフトウェアセルを送信し、メインメモリ1026にメインメモリ用プログラムおよびそれによって処理されるデータを記憶させる。

【0456】

50

次に、送信元の情報処理装置は、送信先の情報処理装置宛に、送信先の情報処理装置内の情報処理コントローラについてのメインプロセッサID、メインメモリアドレス、メインメモリ用プログラムを識別するための後述の機能プログラムIDなどの識別子、およびプログラムカウンタを含み、DMAコマンドがキックコマンドまたは機能プログラム実行コマンドであるソフトウェアセルを送信して、メインプロセッサ1021に当該メインメモリ用プログラムを実行させる。

【0457】

以上のように、この発明のネットワークシステムでは、送信元の情報処理装置は、サブプロセッサプログラムまたはメインメモリ用プログラムをソフトウェアセルによって送信先の情報処理装置に送信するとともに、当該サブプロセッサプログラムを送信先の情報処理装置内の情報処理コントローラに含まれるサブプロセッサ1023にロードさせ、当該サブプロセッサプログラムまたは当該メインメモリ用プログラムを送信先の情報処理装置に実行させることができる。

10

【0458】

送信先の情報処理装置内の情報処理コントローラでは、受信したソフトウェアセルに含まれるプログラムがサブプロセッサプログラムである場合には、当該サブプロセッサプログラムを指定されたサブプロセッサにロードさせる。そして、ソフトウェアセルに含まれるサブプロセッサプログラムまたはメインメモリ用プログラムを実行させる。

【0459】

したがって、ユーザが送信先の情報処理装置を操作しなくても自動的に、当該サブプロセッサプログラムまたは当該メインメモリ用プログラムを送信先の情報処理装置内の情報処理コントローラに実行させることができる。

20

【0460】

このようにして情報処理装置は、自装置内の情報処理コントローラがサブプロセッサプログラムまたは機能プログラムなどのメインメモリ用プログラムを有していない場合には、ネットワークに接続された他の情報処理装置からそれらを取得することができる。さらに、各サブプロセッサとメインメモリとの間ではDMA方式によりデータ転送を行い、また上述したサンドボックスを使用することによって、1つの情報処理コントローラ内でデータを多段階に処理する必要がある場合でも、高速かつ高セキュリティに処理を実行することができる。

30

【0461】

[6. ネットワークシステムとしての分散処理その2]

ソフトウェアセルの使用による分散処理の結果、図49の上段に示すようにネットワーク1009に接続されている複数の情報処理装置1001, 1002, 1003および1004は、図49の下段に示すように、仮想的な1台の情報処理装置1007として動作する。ただし、そのためには、以下のような構成によって、以下のような処理が実行される必要がある。

【0462】

(6-1. システムのソフトウェア構成とプログラムのロード)

図6に、個々の情報処理コントローラのメインメモリ1026が記憶するソフトウェアの構成を示す。これらのソフトウェア(プログラム)は、情報処理装置に電源が投入される前においては、当該の情報処理コントローラに接続される外部記録部1031に記録されているものである。

40

【0463】

各プログラムは、機能または特徴によって、制御プログラム、機能プログラムおよびデバイスドライバにカテゴライズされる。

【0464】

制御プログラムは、各情報処理コントローラが同じものを備え、各情報処理コントローラのメインプロセッサ1021が実行するもので、後述のMS(マスター/スレーブ)マネージャおよび能力交換プログラムを含む。

50

## 【 0 4 6 5 】

機能プログラムは、メインプロセッサ 1 0 2 1 が実行するもので、記録用、再生用、素材検索用など、情報処理コントローラごとに情報処理装置に応じたものが備えられる。

## 【 0 4 6 6 】

デバイスドライバは、情報処理コントローラ（情報処理装置）の入出力（送受信）用で、放送受信、モニタ出力、ビットストリーム入出力、ネットワーク入出力など、情報処理コントローラごとに情報処理装置に応じたものが備えられる。

## 【 0 4 6 7 】

ケーブルの差し込みなどによって情報処理装置が物理的にネットワーク 1 0 0 9 に接続された状態で、情報処理装置に主電源が投入され、情報処理装置が電氣的・機能的にもネットワーク 1 0 0 9 に接続されると、その情報処理装置の情報処理コントローラのメインプロセッサ 1 0 2 1 は、制御プログラムに属する各プログラム、およびデバイスドライバに属する各プログラムを、メインメモリ 1 0 2 6 にロードする。

## 【 0 4 6 8 】

ロード手順としては、メインプロセッサ 1 0 2 1 は、まず、D C 1 0 3 0 に読み出し命令を実行させることによって、外部記録部 1 0 3 1 からプログラムを読み出し、次に、D M A C 1 0 2 5 に書き込み命令を実行させることによって、そのプログラムをメインメモリ 1 0 2 6 に書き込む。

## 【 0 4 6 9 】

機能プログラムに属する各プログラムについては、必要なときに必要なプログラムだけをロードするように構成してもよく、または、他のカテゴリに属するプログラムと同様に、主電源投入直後に各プログラムをロードするように構成してもよい。

## 【 0 4 7 0 】

機能プログラムに属する各プログラムは、ネットワークに接続された全ての情報処理装置の外部記録部 1 0 3 1 に記録されている必要はなく、いずれか 1 つの情報処理装置の外部記録部 1 0 3 1 に記録されていれば、前述の方法によって他の情報処理装置からロードすることができるので、結果的に図 4 9 の下段に示すように、仮想的な 1 台の情報処理装置 1 0 0 7 として機能プログラムを実行することができる。

## 【 0 4 7 1 】

また、前述したように、メインプロセッサ 1 0 2 1 によって処理される機能プログラムは、サブプロセッサ 1 0 2 3 によって処理されるサブプロセッサプログラムと連携動作する場合がある。そこで、メインプロセッサ 1 0 2 1 が外部記録部 1 0 3 1 から機能プログラムを読み出し、メインメモリ 1 0 2 6 に書き込む際、対象となる機能プログラムと連携動作するサブプロセッサプログラムが存在する場合には、当該サブプロセッサプログラムも併せて同じメインメモリ 1 0 2 6 に書き込む。この場合、連携動作するサブプロセッサプログラムは、1 個である場合もあるが、複数個であることもあり得る。複数個である場合には、全ての連携動作するサブプロセッサプログラムをメインメモリ 1 0 2 6 に書き込む。

## 【 0 4 7 2 】

メインメモリ 1 0 2 6 に書き込まれたサブプロセッサプログラムは、その後、サブプロセッサ 1 0 2 3 内の L S 1 0 2 4 に書き込まれ、メインプロセッサ 1 0 2 1 によって処理される機能プログラムと連携動作する。

## 【 0 4 7 3 】

図 3 のソフトウェアセルに示したように、機能プログラムには、プログラムごとにプログラムを一意的に識別できる識別子が、機能プログラム I D として割り当てられる。機能プログラム I D は、機能プログラムの作成の段階で、作成日時や情報処理装置 I D などから決定される。

## 【 0 4 7 4 】

サブプロセッサプログラムにもサブプロセッサプログラム I D が割り当てられ、これによりサブプロセッサプログラムを一意的に識別可能である。割り当てられるサブプロセ

10

20

30

40

50

サブプログラムIDは、連携動作の相手となる機能プログラムの機能プログラムIDと関連性のある識別子、例えば機能プログラムIDを親番号とした上で最後尾に枝番号を付加させた識別子でもよいが、連携動作の相手となる機能プログラムの機能プログラムIDとは関連性のない識別子でもよい。

【0475】

いずれにしても、機能プログラムとサブプロセッサプログラムが連携動作する場合には、両者とも相手の識別子であるプログラムIDを自プログラム内に記憶しておく必要がある。機能プログラムが複数個のサブプロセッサプログラムと連携動作する場合にも、当該機能プログラムは、その複数個のサブプロセッサプログラムのサブプロセッサプログラムIDを全て記憶しておく。

10

【0476】

メインプロセッサ1021は、自身が動作する情報処理装置の装置情報（動作状態に関する情報）を格納するための領域をメインメモリ1026に確保し、当該情報を自装置の装置情報テーブルとして記録する。ここでの装置情報は、図4に示した情報処理装置ID以下の各情報である。

【0477】

（6-2. システムにおけるマスター/スレーブの決定）

上述したネットワークシステムでは、ある情報処理装置への主電源投入時、その情報処理装置の情報処理コントローラのメインプロセッサ1021は、マスター/スレーブマネージャ（以下、MSマネージャ）をメインメモリ1026にロードし、実行する。

20

【0478】

MSマネージャは、自身が動作する情報処理装置がネットワーク1009に接続されていることを検知すると、同じネットワーク1009に接続されている他の情報処理装置の存在を確認する。ここでの「接続」または「存在」は、上述したように、情報処理装置が物理的にネットワーク1009に接続されているだけでなく、電氣的・機能的にもネットワーク1009に接続されていることを示す。

【0479】

以下では、自身が動作する情報処理装置を自装置、他の情報処理装置を他装置と称する。当該装置も、当該情報処理装置を示すものとする。

【0480】

MSマネージャが同じネットワーク1009に接続されている他の情報処理装置の存在を確認する方法を、以下に示す。

30

【0481】

MSマネージャは、DMAコマンドがステータス要求コマンドであり、送信元IDおよび応答先IDが当該情報処理装置で、送信先IDを特定しないソフトウェアセルを生成して、当該情報処理装置が接続されたネットワーク上に送信し、ネットワーク接続確認用のタイマーを設定する。タイマーのタイムアウト時間は、例えば10分とされる。

【0482】

当該ネットワークシステム上に他の情報処理装置が接続されている場合、その他装置は、上記ステータス要求コマンドのソフトウェアセルを受信し、上記応答先IDで特定されるステータス要求コマンドを発行した情報処理装置に対して、DMAコマンドがステータス返信コマンドで、かつデータとして自身（その他装置）の装置情報を含むソフトウェアセルを送信する。このステータス返信コマンドのソフトウェアセルには、少なくとも当該他装置を特定する情報（情報処理装置ID、メインプロセッサに関する情報、サブプロセッサに関する情報など）および当該他装置のMSステータスが含まれる。

40

【0483】

ステータス要求コマンドを発行した情報処理装置のMSマネージャは、上記ネットワーク接続確認用のタイマーがタイムアウトするまで、当該ネットワーク上の他装置から送信されるステータス返信コマンドのソフトウェアセルの受信を監視する。その結果、MSステータス=0（マスター装置）を示すステータス返信コマンドが受信された場合には、自

50

装置の装置情報テーブルにおけるMSステータスを1に設定する。これによって、当該装置はスレーブ装置となる。

【0484】

一方、上記ネットワーク接続確認用のタイマーがタイムアウトするまでの間にステータス返信コマンドが全く受信されなかった場合、またはMSステータス=0(マスター装置)を示すステータス返信コマンドが受信されなかった場合には、自装置の装置情報テーブルにおけるMSステータスを0に設定する。これによって、当該装置はマスター装置となる。

【0485】

すなわち、いずれの装置もネットワーク1009に接続されていない状態、またはネットワーク1009上にマスター装置が存在しない状態において、新たな情報処理装置がネットワーク1009に接続されると、当該装置は自動的にマスター装置として設定される。一方、ネットワーク1009上に既にマスター装置が存在する状態において、新たな情報処理装置がネットワーク1009に接続されると、当該装置は自動的にスレーブ装置として設定される。

【0486】

マスター装置およびスレーブ装置のいずれについても、MSマネージャは、定期的にステータス要求コマンドをネットワーク1009上の他装置に送信してステータス情報を照会することにより、他装置の状況を監視する。その結果、ネットワーク1009に接続されている情報処理装置の主電源が遮断され、またはネットワーク1009から情報処理装置が切り離されることにより、あらかじめ判定用に設定された所定期間内に特定の他装置からステータス返信コマンドが返信されなかった場合や、ネットワーク1009に新たな情報処理装置が接続された場合など、ネットワーク1009の接続状態に変化があった場合には、その情報を後述の能力交換プログラムに通知する。

【0487】

(6-3. マスター装置およびスレーブ装置における装置情報の取得)

メインプロセッサ1021は、MSマネージャから、ネットワーク1009上の他装置の照会および自装置のMSステータスの設定完了の通知を受けると、能力交換プログラムを実行する。

【0488】

能力交換プログラムは、自装置がマスター装置である場合には、ネットワーク1009に接続されている全ての他装置の装置情報、すなわち各スレーブ装置の装置情報を取得する。

【0489】

他装置の装置情報の取得は、上述したように、DMAコマンドがステータス要求コマンドであるソフトウェアセルを生成して他装置に送信し、その後、DMAコマンドがステータス返信コマンドで、かつデータとして他装置の装置情報を含むソフトウェアセルを他装置から受信することによって可能である。

【0490】

能力交換プログラムは、マスター装置である自装置の装置情報テーブルと同様に、ネットワーク1009に接続されている全ての他装置(各スレーブ装置)の装置情報を格納するための領域を自装置のメインメモリ1026に確保し、これら情報を他装置(スレーブ装置)の装置情報テーブルとして記録する。

【0491】

すなわち、マスター装置のメインメモリ1026には、自装置を含むネットワーク1009に接続されている全ての情報処理装置の装置情報が、装置情報テーブルとして記録される。

【0492】

一方、能力交換プログラムは、自装置がスレーブ装置である場合には、ネットワーク1009に接続されている全ての他装置の装置情報、すなわちマスター装置および自装置以

10

20

30

40

50

外の各スレーブ装置の装置情報を取得し、これら装置情報に含まれる情報処理装置IDおよびMSステータスを、自装置のメインメモリ1026に記録する。

【0493】

すなわち、スレーブ装置のメインメモリ1026には、自装置の装置情報が、装置情報テーブルとして記録されるとともに、自装置以外のネットワーク1009に接続されているマスター装置および各スレーブ装置についての情報処理装置IDおよびMSステータスが、別の装置情報テーブルとして記録される。

【0494】

また、マスター装置およびスレーブ装置のいずれについても、能力交換プログラムは、上記のようにMSマネージャから、新たにネットワーク1009に情報処理装置が接続されたことが通知されたときには、その情報処理装置の装置情報を取得し、上述したようにメインメモリ1026に記録する。

【0495】

なお、MSマネージャおよび能力交換プログラムは、メインプロセッサ1021で実行されることに限らず、いずれかのサブプロセッサ1023で実行されてもよい。また、MSマネージャおよび能力交換プログラムは、情報処理装置の主電源が投入されている間は常時動作する常駐プログラムであることが望ましい。

【0496】

(6-4. 情報処理装置がネットワークから切断された場合)

マスター装置およびスレーブ装置のいずれについても、能力交換プログラムは、上記のようにMSマネージャから、ネットワーク1009に接続されている情報処理装置の主電源が遮断され、またはネットワーク1009から情報処理装置が切り離されたことが通知されたときには、その情報処理装置の装置情報テーブルを自装置のメインメモリ1026から削除する。

【0497】

さらに、このようにネットワーク1009から切断された情報処理装置がマスター装置である場合には、以下のような方法によって、新たにマスター装置が決定される。

【0498】

具体的に、例えば、ネットワーク1009から切断されていない情報処理装置は、それぞれ、自装置および他装置の情報処理装置IDを数値に置き換えて、自装置の情報処理装置IDを他装置の情報処理装置IDと比較し、自装置の情報処理装置IDがネットワーク1009から切断されていない情報処理装置中で最小である場合、そのスレーブ装置は、マスター装置に移行して、MSステータスを0に設定し、マスター装置として、上述したように、ネットワーク1009に接続されている全ての他装置(各スレーブ装置)の装置情報を取得して、メインメモリ1026に記録する。

【0499】

(6-5. 装置情報に基づく分散処理)

図49の下段に示したようにネットワーク1009に接続されている複数の情報処理装置1001, 1002, 1003および1004を仮想的な1台の情報処理装置1007として動作させるためには、マスター装置がユーザの操作およびスレーブ装置の動作状態を把握する必要がある。

【0500】

図50に、4台の情報処理装置が仮想的な1台の情報処理装置1007として動作する様子を示す。情報処理装置1001がマスター装置、情報処理装置1002, 1003および1004がスレーブ装置A, BおよびCとして、動作しているものとする。

【0501】

ユーザがネットワーク1009に接続されている情報処理装置を操作した場合、操作対象がマスター装置1001であれば、その操作情報は、マスター装置1001において直接把握され、操作対象がスレーブ装置であれば、その操作情報は、操作されたスレーブ装置からマスター装置1001に送信される。すなわち、ユーザの操作対象がマスター装置

10

20

30

40

50

1001とスレーブ装置のいずれであるかにかかわらず、その操作情報は常にマスター装置1001において把握される。操作情報の送信は、例えば、DMAコマンドが操作情報送信コマンドであるソフトウェアセルによって行われる。

【0502】

そして、マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、その操作情報に従って、実行する機能プログラムを選択する。その際、必要であれば、マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、上記の方法によって自装置の外部記録部1031-1または1031-2からメインメモリ1026-1に機能プログラムをロードするが、他の情報処理装置（スレーブ装置）がマスター装置1001に機能プログラムを送信してもよい。

10

【0503】

機能プログラムには、その実行単位ごとに必要となる、図4に示した各情報として表される情報処理装置種別ID、メインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件などの、装置に関する要求スペックが規定されている。

【0504】

マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、各機能プログラムについて必要となる上記要求スペックを読み出す。また、あらかじめ能力交換プログラムによってメインメモリ1026-1に記録された装置情報テーブルを参照し、各情報処理装置の装置情報を読み出す。ここでの装置情報は、図4に示した情報処理装置ID以下の各情報を示し、メインプロセッサ、サブプロセッサ、メインメモリおよび外部記録部に関する情報である。

20

【0505】

マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、ネットワーク1009上に接続された各情報処理装置の上記装置情報と、機能プログラム実行に必要な上記要求スペックとを順次比較する。

【0506】

そして、例えば、機能プログラムが録画機能を必要とする場合には、情報処理装置種別IDに基づいて、録画機能を有する情報処理装置のみを特定して抽出する。さらに、機能プログラムを実行するために必要なメインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件を確保できるスレーブ装置を、実行要求候補装置として特定する。ここで、複数の実行要求候補装置が特定された場合には、当該候補装置から1つの実行要求候補装置を特定して選択する。

30

【0507】

実行要求するスレーブ装置が特定されたら、マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、その特定されたスレーブ装置について、自装置内の情報処理コントローラ1011に含まれるメインメモリ1026-1に記録されている当該スレーブ装置の装置情報テーブルを更新する。

【0508】

さらに、マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、DMAコマンドが機能プログラム実行コマンドであるソフトウェアセルを生成し、当該ソフトウェアセルのセルインターフェースに、当該機能プログラムに関する必要なサブプロセッサの情報およびサンドボックスサイズ（図17参照）を設定して、上記実行要求されるスレーブ装置に対して送信する。

40

【0509】

機能プログラムの実行を要求されたスレーブ装置は、その機能プログラムを実行するとともに、自装置の装置情報テーブルを更新する。その際、必要であれば、スレーブ装置内の情報処理コントローラに含まれるメインプロセッサ1021は、上記の方法によって自装置の外部記録部1031からメインメモリ1026に機能プログラムおよび当該機能プ

50

プログラムと連携動作するサブプロセッサプログラムをロードする。

【0510】

機能プログラムの実行を要求されたスレーブ装置の外部記録部1031に、必要な機能プログラムまたは当該機能プログラムと連携動作するサブプロセッサプログラムが記録されていない場合には、他の情報処理装置が当該機能プログラムまたはサブプロセッサプログラムを、その機能プログラム実行要求先スレーブ装置に送信するように、システムを構成すればよい。

【0511】

サブプロセッサプログラムについては、前述のロードコマンドおよびキックコマンドを利用して他の情報処理装置に実行させることもできる。

10

【0512】

機能プログラムの実行終了後、機能プログラムを実行したスレーブ装置内の情報処理コントローラに含まれるメインプロセッサ1021は、終了通知をマスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1に送信するとともに、自装置の装置情報テーブルを更新する。マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、その終了通知を受信して、機能プログラムを実行したスレーブ装置の装置情報テーブルを更新する。

【0513】

マスター装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、自装置および他装置の装置情報テーブルの参照結果から、当該の機能プログラムを実行することができる情報処理装置として、自身を選択する場合もあり得る。その場合には、マスター装置1001が当該の機能プログラムを実行する。

20

【0514】

図50の例で、ユーザがスレーブ装置A（情報処理装置1002）を操作し、当該操作に応じた機能プログラムを別のスレーブ装置B（情報処理装置1003）が実行する場合につき、図51に以上の分散処理の例を示す。

【0515】

図51の例では、ユーザがスレーブ装置Aを操作することによって、スレーブ装置Aを含むネットワークシステム全体の分散処理が開始して、まず、スレーブ装置Aは、ステップ1091で、その操作情報をマスター装置1001に送信する。

30

【0516】

マスター装置1001は、ステップ1092で、その操作情報を受信し、さらにステップ1093に進んで、自装置のメインメモリ1026-1に記録されている自装置および他装置の装置情報テーブルから、各情報処理装置の動作状態を調べて、受信した操作情報に応じた機能プログラムを実行することができる情報処理装置を選択する。この例は、スレーブ装置Bが選択される場合である。

【0517】

次に、マスター装置1001は、ステップ1094で、その選択したスレーブ装置Bに対して機能プログラムの実行を要求する。

【0518】

スレーブ装置Bは、ステップ1095で、その実行要求を受信し、さらにステップ1096に進んで、実行要求された機能プログラムを実行する。

40

【0519】

以上のように、ユーザは、1台の情報処理装置のみを操作することによって、他の情報処理装置を操作することなく、複数の情報処理装置1001, 1002, 1003および1004を仮想的な1台の情報処理装置1007として動作させることができる。

【0520】

（6-6. 各情報処理装置およびシステムの具体例）

ネットワーク1009を介して互いに接続される情報処理装置1001, 1002, 1003および1004は、基本的に上記のような情報処理コントローラ1011, 101

50

2, 1013および1014によって情報処理を行うものであれば、どのようなものでもよいが、図52に、その一例を示す。

【0521】

この例では、情報処理コントローラ1011を備える情報処理装置1001は、ハードディスクレコーダで、図53に示すように、ハードウェア構成としては、図41に示した外部記録部1031-1として、ハードディスクを内蔵し、図41に示した外部記録部1031-2として、DVD±R/RW、CD±R/RW、Blu-ray Disc（登録商標）などの光ディスクを装着できるように構成されるとともに、情報処理コントローラ1011のバス1032-1に接続されたバス1051-1に、放送受信部1052-1、映像入力部1053-1、音声入力部1054-1、映像出力部1055-1、音声出力部1056-1、操作パネル部1057-1、リモコン受光部1058-1およびネットワーク接続部1059-1が接続されたものである。

10

【0522】

放送受信部1052-1、映像入力部1053-1および音声入力部1054-1は、放送信号を受信し、または情報処理装置1001の外部から映像信号および音声信号を入力し、それぞれ所定フォーマットのデジタルデータに変換して、情報処理コントローラ1011での処理のためにバス1051-1に送出するものであり、映像出力部1055-1および音声出力部1056-1は、情報処理コントローラ1011からバス1051-1に送出された映像データおよび音声データを処理して、デジタルデータのまま、またはアナログ信号に変換して、情報処理装置1001の外部に送出するものであり、リモコン受光部1058-1は、リモコン送信器1063-1からのリモコン（遠隔操作）赤外線信号を受信するものである。

20

【0523】

図52および図53に示すように、情報処理装置（ハードディスクレコーダ）1001の映像出力部1055-1および音声出力部1056-1には、モニタ表示装置1061およびスピーカ装置1062が接続される。

【0524】

図52の例の、情報処理コントローラ1012を備える情報処理装置1002も、ハードディスクレコーダで、図53において括弧内に参照番号を付して示すように、情報処理装置1001と同様に構成されたものである。ただし、例えば、図52に示すように、情報処理装置（ハードディスクレコーダ）1002には、モニタ表示装置およびスピーカ装置は接続されない。

30

【0525】

情報処理装置（ハードディスクレコーダ）1001および1002、すなわち情報処理コントローラ1011および1012のソフトウェア構成としては、図54に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、映像音声記録、映像音声再生、素材検索および番組録画予約のためのプログラムを備え、デバイスドライバとして、放送受信、映像出力、音声出力、外部記録部入出力およびネットワーク入出力のためのプログラムを備える。

【0526】

図52の例の、情報処理コントローラ1013を備える情報処理装置1003は、PDA（Personal Digital Assistants）で、図55に示すように、ハードウェア構成としては、図41に示した外部記録部1031-5として、メモリカードディスクを装着できるように構成されるとともに、情報処理コントローラ1013のバス1032-3に接続されたバス1071に、液晶表示部1072、音声出力部1073、カメラ部1074、音声入力部1075、キーボード部1076およびネットワーク接続部1077が接続されたものである。

40

【0527】

なお、図41では内部を省略した情報処理コントローラ1013は、メインプロセッサ1021-3、サブプロセッサ1023-7, 1023-8および1023-9、DMA

50

C (ダイレクトメモリアクセスコントローラ) 1025-3、コントロールレジスタ1028-3、ワークメモリ1029-3、DC (ディスクコントローラ) 1030-3、およびバス1032-3を備え、そのメインプロセッサ1021-3は、LS (ローカルストレージ) 1022-3を有し、各サブプロセッサ1023-7, 1023-8および1023-9は、LS (ローカルストレージ) 1024-7, 1024-8および1024-9を有する。

【0528】

情報処理装置 (PDA) 1003、すなわち情報処理コントローラ1013のソフトウェア構成としては、図56に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、映像音声記録、映像音声再生、電話帳、ワープロおよび表計算のためのプログラム、およびWebブラウザを備え、デバイスドライバとして、映像出力、音声出力、カメラ映像入力、マイク音声入力およびネットワーク入出力のためのプログラムを備える。

10

【0529】

図52の例の、情報処理コントローラ1014を備える情報処理装置1004は、ポータブルCDプレーヤで、図57に示すように、ハードウェア構成としては、図41に示した外部記録部1031-6として、CD (Compact Disc) を装着できるように構成されるとともに、情報処理コントローラ1014のバス1032-4に接続されたバス1081に、液晶表示部1082、音声出力部1083、操作ボタン部1084およびネットワーク接続部1085が接続されたものである。

20

【0530】

なお、図41では内部を省略した情報処理コントローラ1014は、メインプロセッサ1021-4、サブプロセッサ1023-10, 1023-11および1023-12、DMAC (ダイレクトメモリアクセスコントローラ) 1025-4、コントロールレジスタ1028-4、ワークメモリ1029-4、DC (ディスクコントローラ) 1030-4、およびバス1032-4を備え、そのメインプロセッサ1021-4は、LS (ローカルストレージ) 1022-4を有し、各サブプロセッサ1023-10, 1023-11および1023-12は、LS (ローカルストレージ) 1024-10, 1024-11および1024-12を有する。

【0531】

情報処理装置 (ポータブルCDプレーヤ) 1004、すなわち情報処理コントローラ1014のソフトウェア構成としては、図58に示すように、制御プログラムとして、MSマネージャおよび能力交換プログラムを備え、機能プログラムとして、音楽再生のためのプログラムを備え、デバイスドライバとして、音声出力、CD制御およびネットワーク入出力のためのプログラムを備える。

30

【0532】

以上のような図52の例のネットワークシステムにおいて、情報処理装置1001, 1003および1004がネットワーク1009上に接続されており、情報処理装置1001がマスター装置 (MSステータス=0) として、情報処理装置1003および1004がスレーブ装置 (MSステータス=1) として、設定されているものとする。

40

【0533】

この状態で、新たに情報処理装置1002がネットワーク1009に接続されると、上述した方法によって、情報処理装置1002内の情報処理コントローラ1012に含まれるメインプロセッサ1021-2で実行されているMSマネージャは、他の情報処理装置1001, 1003および1004にMSステータスを照会して、情報処理装置1001が既にマスター装置として存在することを認識し、自装置 (情報処理装置1002) をスレーブ装置 (MSステータス=1) に設定する。また、マスター装置に設定されている情報処理装置1001は、新たに追加された情報処理装置1002を含む各装置の装置情報を収集して、メインメモリ1026-1内の装置情報テーブルを更新する。

【0534】

50

このような状態で、ユーザによってスレーブ装置である情報処理装置(PDA)1003で2時間の放送番組の録画予約の操作が行われた場合を示す。

【0535】

この場合、スレーブ装置である情報処理装置(PDA)1003は、ユーザから録画開始時刻、録画終了時刻、録画対象放送チャンネル、録画品質などの情報を含む録画予約情報の入力を受け付け、当該録画予約情報およびDMAコマンドとしての録画予約コマンドを含むソフトウェアセルを生成して、マスター装置である情報処理装置1001に送信する。

【0536】

DMAコマンドが録画予約コマンドであるソフトウェアセルを受信した情報処理装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、録画予約コマンドを読み出すとともに、メインメモリ1026-1内の装置情報テーブルを参照して、当該録画予約コマンドを実行可能な情報処理装置を特定する。

10

【0537】

まず、メインプロセッサ1021-1は、装置情報テーブルに含まれる各情報処理装置1001,1002,1003および1004の情報処理装置種別IDを読み出して、録画予約コマンドに対応する機能プログラムを実行可能な情報処理装置を抽出する。ここでは、録画機能を示す情報処理装置種別IDを有する情報処理装置1001および1002が候補装置として特定され、情報処理装置1003および1004は候補装置から除外される。

20

【0538】

次に、マスター装置である情報処理装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、装置情報テーブルを参照して、情報処理装置1001および1002のメインプロセッサまたはサブプロセッサの処理能力、メインメモリに関する情報などの、装置に関する情報を読み出し、情報処理装置1001および1002が録画予約コマンドに対応する機能プログラムの実行に必要な要求スペックを満足するか否かを判断する。ここでは、情報処理装置1001,1002とも、録画予約コマンドに対応する機能プログラムの実行に必要な要求スペックを満足するものとする。

【0539】

さらに、メインプロセッサ1021-1は、装置情報テーブルを参照して、情報処理装置1001および1002の外部記録部に関する情報を読み出し、外部記録部の空き容量が当該録画予約コマンドの実行に必要な容量を満足するか否かを判断する。情報処理装置1001および1002はハードディスクレコーダであるので、それぞれハードディスク1031-1および1031-3の、総容量と使用量との差分が、それぞれの空き容量に相当する。

30

【0540】

この場合、情報処理装置1001のハードディスク1031-1の空き容量が、録画時間に換算して10分であり、情報処理装置1002のハードディスク1031-3の空き容量が、録画時間に換算して20時間であるとする。

【0541】

このとき、マスター装置である情報処理装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、当該録画予約コマンドの実行に必要な2時間分の空き容量を確保できる情報処理装置を、実行要求先スレーブ装置として特定する。

40

【0542】

その結果、情報処理装置1002のみが実行要求先スレーブ装置として選択され、マスター装置である情報処理装置1001内の情報処理コントローラ1011に含まれるメインプロセッサ1021-1は、ユーザにより操作された情報処理装置1003から送信された録画予約情報を含む当該録画予約コマンドを情報処理装置1002に送信して、上記2時間の放送番組の録画予約の実行を要求する。

50

## 【 0 5 4 3 】

そして、情報処理装置 1 0 0 2 内の情報処理コントローラ 1 0 1 2 に含まれるメインプロセッサ 1 0 2 1 - 2 は、当該録画予約コマンドを解析して、録画に必要な機能プログラムを外部記録部であるハードディスク 1 0 3 1 - 3 からメインメモリ 1 0 2 6 - 2 にロードし、録画予約情報に従って録画を実行する。その結果、録画予約された 2 時間の放送番組の映像音声データが情報処理装置 1 0 0 2 のハードディスク 1 0 3 1 - 3 に記録される。

## 【 0 5 4 4 】

このように、図 5 2 の例のネットワークシステムにおいても、ユーザは、1 台の情報処理装置のみを操作することによって、他の情報処理装置を操作することなく、複数の情報処理装置 1 0 0 1 , 1 0 0 2 , 1 0 0 3 および 1 0 0 4 を仮想的な 1 台の情報処理装置 1 0 0 7 として動作させることができる。

10

## 【 0 5 4 5 】

[ 7 . サブプロセッサの割り当てによる分散処理その 2 ]

図 5 0 または図 5 2 のネットワークシステムにおける前述の例は、マスター装置である情報処理装置 1 0 0 1 が機能プログラムを実行することができる情報処理装置を特定する際、当該機能プログラムの実行に必要な全ての要求スペックを満足する情報処理装置がネットワークシステム内に存在する場合であるが、このような情報処理装置が存在しない場合もあり得る。

## 【 0 5 4 6 】

20

図 5 2 の例では、マスター装置である情報処理装置 1 0 0 1 は、録画予約コマンドに対応する機能プログラムを実行させる情報処理装置として、最終的に情報処理装置 1 0 0 2 を特定したが、情報処理装置 1 0 0 2 が当該機能プログラムの実行に必要な要求スペックの一部、満たしていないこともある。例えば、他の要求スペックは全て満たしているが、サブプロセッサ数が不足している場合である。そのような場合でも、情報処理装置 1 0 0 2 は、上記のロードコマンドまたはキックコマンドを利用して他の情報処理装置内のサブプロセッサを利用することによって、当該機能プログラムを実行できる可能性がある。

## 【 0 5 4 7 】

そこで、マスター装置が機能プログラムを実行させる情報処理装置を特定する際、候補となる情報処理装置のサブプロセッサ処理能力を考慮しないで、当該機能プログラムを実行させる情報処理装置を特定する場合の例を示す。

30

## 【 0 5 4 8 】

この場合、機能プログラムには、図 4 に示した各情報として表される情報処理装置種別 ID、メインプロセッサまたはサブプロセッサの処理能力、メインメモリ使用量、外部記録部に関する条件などの、装置に関する要求スペックが規定されており、マスター装置が当該機能プログラムを実行させる情報処理装置を特定する際には、サブプロセッサ処理能力以外の要求スペックを満足する情報処理装置を特定するものとする。したがって当然、マスター装置によって特定されて機能プログラムの実行を要求された情報処理装置は、自装置内に当該機能プログラムの実行に必要なサブプロセッサ処理能力を確保できない場合がある。

40

## 【 0 5 4 9 】

以下に、分散処理の他の例として、このように機能プログラムの実行を要求された情報処理装置が、ネットワークシステムに含まれる他の情報処理装置内のサブプロセッサを利用する場合について示す。

## 【 0 5 5 0 】

( 7 - 1 . システム構成 )

図 5 9 に、4 台の情報処理装置 1 0 0 1 , 1 0 0 2 , 1 0 0 3 および 1 0 0 4 がネットワーク 1 0 0 9 に接続されている状態を示す。

## 【 0 5 5 1 】

情報処理装置 1 0 0 1 がマスター装置として、情報処理装置 1 0 0 2 , 1 0 0 3 および

50

1004がスレーブ装置として、それぞれ動作し、マスター装置である情報処理装置1001が機能プログラムを実行させる情報処理装置として自装置を特定するものとする。

【0552】

ネットワーク1009に接続されている各情報処理装置のソフトウェア構成としては、図17に示すように、制御プログラムとして、図6などに示したMSマネージャおよび能力交換プログラムのほかに、リソースマネージャを含むものとする。

【0553】

リソースマネージャは、自装置内の各サブプロセッサを制御して、プログラムやデータの転送、プログラムの実行の開始や停止、プログラムの実行結果の受信などを行うものである。

【0554】

さらに、リソースマネージャは、自装置内の各サブプロセッサの使用状況を把握して、他の情報処理装置からの問い合わせに対して応答し、または他の情報処理装置に対して、その他装置内の各サブプロセッサの使用状況を問い合わせる。

【0555】

そのために、リソースマネージャには、図18に示すようなサブプロセッサ管理テーブルを用意する。

【0556】

この例は、各情報処理装置内の情報処理コントローラが数個以上のサブプロセッサを備える場合で、サブプロセッサ管理テーブル中の情報処理装置IDは、自装置の情報処理装置IDであり、サブプロセッサIDは、自装置内の各サブプロセッサのサブプロセッサIDであり、サブプロセッサステータスは、各サブプロセッサの使用状況を示す。

【0557】

サブプロセッサステータスとしては、上述したように、`unused`（現在使用されてなく、使用の予約もされていない状態）、`reserved`（現在は使用されていないが、使用が予約されている状態）、および`busy`（現在使用されている状態）がある。

【0558】

後述のように、`unused`を空いている状態とし、空いているか否か、すなわち、`unused`であるか、`reserved`または`busy`であるかを、空き状況とする。

【0559】

サブプロセッサプログラムIDは、当該のサブプロセッサで実行中の、または当該のサブプロセッサに実行予約中のサブプロセッサプログラムを識別する識別子であり、機能プログラムIDは、上段のサブプロセッサプログラムIDで示されるサブプロセッサプログラムと連携動作する機能プログラムを識別する識別子である。

【0560】

サブプロセッサプログラム優先度は、その実行中または実行予約中のサブプロセッサプログラムの優先度である。サブプロセッサプログラム優先度は、当該サブプロセッサプログラムと連携動作する機能プログラムの優先度に応じて、あらかじめ決定され、例えば、`high`（優先度が高い）、`middle`（中位の優先度）および`low`（優先度が低い）の3段階に区分される。

【0561】

ロックシーケンス番号は、後述のようにサブプロセッサを確保する際などに、リソースマネージャが発番し、サブプロセッサ管理テーブルに記述する。リソースマネージャは、機能プログラムから受信したサブプロセッサプログラムIDおよびロックシーケンス番号が、このサブプロセッサ管理テーブルに記述されたそれらと一致するときのみ、そのサブプロセッサの使用を許可する。

【0562】

（7-2. サブプロセッサ割り当て処理）

上述したように、図59において、情報処理装置1001がマスター装置として、情報処理装置1002、1003および1004がスレーブ装置として、それぞれ動作し、マ

10

20

30

40

50

スター装置としての情報処理装置1001が機能プログラムを実行させる情報処理装置として自装置を特定したものとする。

【0563】

そして、情報処理装置1001が、自装置のメインメモリ1026-1内の機能プログラムと連携動作するサブプロセッサプログラムの実行用に、自装置内または他の情報処理装置1002, 1003および1004内にサブプロセッサを確保する場合の、情報処理装置1001のメインメモリ1026-1内の機能プログラム、情報処理装置1001内のリソースマネージャ、および他の情報処理装置1002, 1003および1004内のリソースマネージャによって実行されるサブプロセッサ割り当て処理の一例を、図60～図68を用いて示す。

10

【0564】

まず、図60に示すように、情報処理装置1001のメインメモリ1026-1内の機能プログラムは、ステップ1111で、サブプロセッサプログラムの実行に必要なサブプロセッサ数を判断し、次にステップ1112で、自装置(情報処理装置1001)内のリソースマネージャに、当該の機能プログラムの機能プログラムIDと、当該機能プログラムと連携動作する全てのサブプロセッサプログラムのサブプロセッサプログラムIDおよびサブプロセッサプログラム優先度とを添えて、必要数(n個とする)のサブプロセッサの使用要求を出力する。

【0565】

これに対して、情報処理装置1001内のリソースマネージャは、ステップ1113で、自装置(情報処理装置1001)の図18に示したサブプロセッサ管理テーブルを参照して、自装置内にn個のサブプロセッサの空きがあるか否かを判断し、n個以上のサブプロセッサが空いている場合には、ステップ1113からステップ1114に進んで、自装置内にn個のサブプロセッサを確保するとともに、その確保した各サブプロセッサにつき、サブプロセッサ管理テーブル内のサブプロセッサステータスを、`unused`から`reserved`に書き替え、サブプロセッサプログラムID、機能プログラムIDおよびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替え、ロックシーケンス番号を発番して、サブプロセッサ管理テーブルに記述する。

20

【0566】

さらに、情報処理装置1001内のリソースマネージャは、ステップ1115に進んで、当該の機能プログラムに、自装置内にn個のサブプロセッサを確保したことと、確保したn個のサブプロセッサのサブプロセッサID、サブプロセッサプログラムID、およびロックシーケンス番号とを通知する。

30

【0567】

図61に示すように、当該の機能プログラムは、ステップ1117で、この通知を受信して、自装置(情報処理装置1001)内のリソースマネージャに、自身の機能プログラムIDと共に、確保した各サブプロセッサに対するサブプロセッサID、サブプロセッサプログラムID、サブプロセッサプログラム本体およびロックシーケンス番号を、確保したサブプロセッサ数の分だけ送信する。

40

【0568】

情報処理装置1001内のリソースマネージャは、ステップ1121で、これらを受信して、各サブプロセッサに対するサブプロセッサプログラムを保存し、さらにステップ1122に進んで、受信したサブプロセッサプログラムIDおよびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断し、同じであるときには、ステップ1123に進んで、確保した各サブプロセッサにサブプロセッサプログラムを送信し、さらにステップ1124に進んで、当該の機能プログラムに、各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0569】

以上によって、情報処理装置1001内の空いているサブプロセッサだけで必要数のサ

50

プロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

【0570】

情報処理装置1001内のリソースマネージャは、ステップ1122で、当該の機能プログラムから受信したサブプロセッサプログラムIDまたはロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと異なると判断した場合には、ステップ1122からステップ1126に進んで、当該の機能プログラムにエラーを通知する。

10

【0571】

この場合には、当該の機能プログラムは、ステップ1127で、エラー処理を実行する。このエラー処理は、具体的には、再度、ステップ1112からサブプロセッサ割り当て処理を実行するなどである。

【0572】

一方、情報処理装置1001内のリソースマネージャは、図60のステップ1113で、自装置内にn個のサブプロセッサの空きがないと判断した場合、すなわち情報処理装置1001内に空いているサブプロセッサが(n-1)個以下しかない(1個もない場合を含む)と判断した場合には、図62に示すように、ステップ1113からステップ1131に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。図59は、他の情報処理装置1002, 1003および1004が接続されている場合である。

20

【0573】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1001内のリソースマネージャは、ステップ1131からステップ1132に進んで、その中の1個である他の情報処理装置(この場合は情報処理装置1002)内のリソースマネージャに、サブプロセッサの空き状況を問い合わせる。

【0574】

この問い合わせは、図28(A)に示すようなサブプロセッサ空き状況問い合わせコマンドによって行う。具体的に、このサブプロセッサ空き状況問い合わせコマンドは、送信元ID(この場合は情報処理装置1001の情報処理装置ID)、送信先ID(この場合は情報処理装置1002の情報処理装置ID)および応答先ID(この場合は情報処理装置1001の情報処理装置ID)を有し、DMAコマンドとして、サブプロセッサ空き状況問い合わせコマンド本体および機能プログラムIDを含むものとする。

30

【0575】

図62に示すように、この問い合わせに対して、当該他の情報処理装置(この場合は情報処理装置1002)内のリソースマネージャは、ステップ1133で、自装置内のサブプロセッサ管理テーブルから、自装置内のサブプロセッサの空き状況を判断し、さらにステップ1135に進んで、自装置内に空いているサブプロセッサがある場合には、その数およびサブプロセッサIDを、空いているサブプロセッサがない場合には、その旨を、情報処理装置1001内のリソースマネージャに通知する。

40

【0576】

この通知は、図28(B)に示すようなサブプロセッサ空き状況返信コマンドによって行う。具体的に、このサブプロセッサ空き状況返信コマンドは、送信元ID(この場合は情報処理装置1002の情報処理装置ID)、送信先ID(この場合は情報処理装置1001の情報処理装置ID)および応答先ID(この場合は情報処理装置1001の情報処理装置ID)を有し、DMAコマンドとして、サブプロセッサ空き状況返信コマンド本体、機能プログラムID、空いているサブプロセッサの数、および空いている各サブプロセッサのサブプロセッサIDを含むものとする。

【0577】

50

この通知を受けて、情報処理装置 1 0 0 1 内のリソースマネージャは、図 6 3 に示すように、ステップ 1 1 3 6 で、自装置（情報処理装置 1 0 0 1）分と他装置（情報処理装置 1 0 0 2）分とを合わせてトータルで、 $n$  個のサブプロセッサを確保できるか否かを判断する。

【 0 5 7 8 】

例えば、 $n$  が 8 , 9 または 1 0 というような数である場合に、情報処理装置 1 0 0 1 内で 1 個のサブプロセッサしか確保できず、情報処理装置 1 0 0 2 内で  $(n - 2)$  個のサブプロセッサしか確保できない場合には、トータルで  $n$  個のサブプロセッサを確保できないことになる。

【 0 5 7 9 】

このようにトータルで  $n$  個のサブプロセッサを確保できない場合には、情報処理装置 1 0 0 1 内のリソースマネージャは、図 6 2 に示すように、ステップ 1 1 3 6 からステップ 1 1 3 1 に進んで（戻って）、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。図 5 9 は、他の情報処理装置 1 0 0 3 および 1 0 0 4 が接続されている場合である。

【 0 5 8 0 】

以下、情報処理装置 1 0 0 2 の場合と同様に、ネットワークシステム全体で  $n$  個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ 1 1 3 2 , 1 1 3 3 , 1 1 3 5 , 1 1 3 6 および 1 1 3 1 の処理を繰り返す。

【 0 5 8 1 】

情報処理装置 1 0 0 1 以外にネットワーク上に複数個の情報処理装置が接続されている場合に情報処理装置 1 0 0 1 内のリソースマネージャがサブプロセッサの空き状況を問い合わせる順番については、例えば、上述したマスター装置の決定方法のように、ネットワーク上の各情報処理装置の情報処理装置 ID を数値に置き換え、当該数値が小さい情報処理装置順にサブプロセッサの空き状況を問い合わせることが考えられる。

【 0 5 8 2 】

そして、ネットワークシステム全体で、空いているサブプロセッサによって  $n$  個のサブプロセッサを確保できる場合には、情報処理装置 1 0 0 1 内のリソースマネージャは、図 6 3 に示すように、ステップ 1 1 3 6 からステップ 1 1 3 7 に進んで、当該の機能プログラムに、 $n$  個のサブプロセッサを確保できることを通知する。

【 0 5 8 3 】

当該の機能プログラムは、ステップ 1 1 3 8 で、この通知を受信し、後述のように自装置（情報処理装置 1 0 0 1）内のリソースマネージャにサブプロセッサ確保要求を出力する。

【 0 5 8 4 】

一方、情報処理装置 1 0 0 1 内のリソースマネージャは、図 6 2 のステップ 1 1 3 1 で、それ以上、ネットワーク上に情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体でも（情報処理装置 1 0 0 1 のみがネットワーク上に存在する場合には情報処理装置 1 0 0 1 内だけで）、空いているサブプロセッサによって  $n$  個のサブプロセッサを確保できないと判断した場合には、図 6 3 に示すように、ステップ 1 1 3 1 からステップ 1 1 4 1 に進んで、自装置（情報処理装置 1 0 0 1）内に当該の機能プログラムより優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否か、および、ある場合には、その数を検出する。

【 0 5 8 5 】

この場合の「実行中」というのは、`busy`（現在使用されている状態）だけでなく、`reserved`（現在は使用されていないが、使用が予約されている状態）を含む。以下では、当該の機能プログラムより優先度の低いサブプロセッサプログラムを、単に「優先度の低いサブプロセッサプログラム」と称する。

【 0 5 8 6 】

10

20

30

40

50

さらに、情報処理装置 1001 内のリソースマネージャは、ステップ 1143 に進んで、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置 1001）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、 $n$  個のサブプロセッサを確保できるか否かを判断し、確保できる場合には、ステップ 1143 からステップ 1144 に進んで、当該の機能プログラムに、 $n$  個のサブプロセッサを確保できることを通知する。

【0587】

当該の機能プログラムは、ステップ 1138 で、この通知を受信し、後述のように自装置（情報処理装置 1001）内のリソースマネージャにサブプロセッサ確保要求を出力する。

10

【0588】

一方、情報処理装置 1001 内のリソースマネージャは、ステップ 1143 で、ネットワークシステム全体の空いているサブプロセッサと、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、 $n$  個のサブプロセッサを確保できないと判断した場合には、図 64 に示すように、ステップ 1143 からステップ 1151 に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。

【0589】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置 1001 内のリソースマネージャは、ステップ 1151 からステップ 1152 に進んで、当該他の情報処理装置（この場合は情報処理装置 1002）内のリソースマネージャに、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの有無および数を問い合わせる。この場合の「実行中」というのも、`busy` だけでなく、`reserved` を含む。

20

【0590】

この問い合わせは、図 28 (A) に示したようなサブプロセッサ空き状況問い合わせコマンドに、当該の機能プログラムと連携動作するサブプロセッサプログラムのサブプロセッサプログラム優先度を含めたコマンドによって行う。

【0591】

図 64 に示すように、この問い合わせに対して、当該他の情報処理装置（この場合は情報処理装置 1002）内のリソースマネージャは、ステップ 1153 で、自装置内のサブプロセッサ管理テーブルから、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否かを判断し、ある場合には、その数およびサブプロセッサ ID を情報処理装置 1001 内のリソースマネージャに通知し、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがない場合には、その旨を情報処理装置 1001 内のリソースマネージャに通知する。

30

【0592】

この通知は、図 28 (B) に示したようなサブプロセッサ空き状況返信コマンドにおいて、空いているサブプロセッサの数およびサブプロセッサ ID に代えて、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの数およびサブプロセッサ ID を記述したコマンドによって行う。

40

【0593】

この通知を受けて、情報処理装置 1001 内のリソースマネージャは、図 64 に示すように、ステップ 1154 で、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置 1001）内および他装置（この場合は情報処理装置 1002）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、 $n$  個のサブプロセッサを確保できるか否かを判断する。

【0594】

例えば、 $n = 7$  の場合に、ネットワークシステム全体で空いているサブプロセッサが 2 個しかなく、情報処理装置 1001 内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサが 2 個しかなく、情報処理装置 1002 内に優先度の低いサブプロセッサ

50

サブプログラムを実行中のサブプロセッサが2個しかない場合には、トータルでn個のサブプロセッサを確保できないことになる。

【0595】

このようにトータルでn個のサブプロセッサを確保できない場合には、情報処理装置1001内のリソースマネージャは、ステップ1154からステップ1151に進んで(戻って)、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。

【0596】

以下、情報処理装置1002の場合と同様に、ネットワークシステム全体でn個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ1152, 1153, 1154および1151の処理を繰り返す。

【0597】

そして、空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、ネットワークシステム全体でn個のサブプロセッサを確保できる場合には、情報処理装置1001内のリソースマネージャは、ステップ1154からステップ1155に進んで、当該の機能プログラムに、n個のサブプロセッサを確保できることを通知する。

【0598】

当該の機能プログラムは、ステップ1158で、この通知を受信し、以下のように自装置(情報処理装置1001)内のリソースマネージャにサブプロセッサ確保要求を出力する。

【0599】

すなわち、当該の機能プログラムは、図63のステップ1138、または図64のステップ1158で、情報処理装置1001内のリソースマネージャから、n個のサブプロセッサを確保できる旨の通知を受信したら、図65に示すように、ステップ1161で、自装置(情報処理装置1001)内のリソースマネージャに、n個のサブプロセッサの確保要求を出力する。

【0600】

図65に示すように、この確保要求に対して、情報処理装置1001内のリソースマネージャは、ステップ1162で、自装置(情報処理装置1001)内に確保すべき分がある場合には、その分のサブプロセッサを自装置内に確保し、さらにステップ1163に進んで、他の情報処理装置内に確保されるべき分があるか否かを判断し、他の情報処理装置内に確保されるべき分がない場合には、ステップ1163からステップ1164に進んで、当該の機能プログラムに、n個のサブプロセッサを確保したことを通知する。

【0601】

情報処理装置1001内の空いているサブプロセッサだけではn個のサブプロセッサを確保できず、かつ情報処理装置1001以外にネットワーク上に接続されている情報処理装置が存在しないが、情報処理装置1001内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、n個のサブプロセッサを確保できる場合には、このように情報処理装置1001内だけでn個のサブプロセッサが確保されることになる。

【0602】

この場合、情報処理装置1001内のリソースマネージャは、ステップ1162では、自装置内に確保した各サブプロセッサにつき、自装置のサブプロセッサ管理テーブル内のサブプロセッサステータスが `unused` となっているものについては、`reserved` に書き替え、サブプロセッサプログラムID、機能プログラムIDおよびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替えるとともに、ロックシーケンス番号を発番してサブプロセッサ管理テーブルに記述し、ステップ1164では、これら情報を、当該の機能プログラムに送信する。

【0603】

10

20

30

40

50

具体的に、ステップ 1 1 6 4 で送信される情報は、確保したサブプロセッサの数、確保した各サブプロセッサのサブプロセッサ ID およびロックシーケンス番号を含むものとする。

【 0 6 0 4 】

図 6 5 に示すように、当該の機能プログラムは、ステップ 1 1 6 7 で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置 1 0 0 1）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

【 0 6 0 5 】

なお、情報処理装置 1 0 0 1 内のリソースマネージャは、ステップ 1 1 6 2 で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを、当該の機能プログラムと連携動作するサブプロセッサプログラムの実行用に確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブプロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

【 0 6 0 6 】

一方、確保されるべき  $n$  個のサブプロセッサに他の情報処理装置内に確保されるべき分がある場合には、情報処理装置 1 0 0 1 内のリソースマネージャは、ステップ 1 1 6 3 からステップ 1 1 7 1 に進んで、他の情報処理装置 1 0 0 2 , 1 0 0 3 および 1 0 0 4 内のリソースマネージャにサブプロセッサの確保を要求し、他の情報処理装置 1 0 0 2 , 1 0 0 3 および 1 0 0 4 内のリソースマネージャは、ステップ 1 1 7 2 で、それぞれ自装置内に自装置分のサブプロセッサを確保する。

【 0 6 0 7 】

他の情報処理装置に対するサブプロセッサの確保要求は、図 2 9 ( A ) に示すような DMA コマンドがサブプロセッサ確保要求コマンドであるソフトウェアセルによって行う。具体的に、このサブプロセッサ確保要求コマンドは、DMA コマンドとして、サブプロセッサ確保要求コマンド本体、機能プログラム ID、サブプロセッサプログラム優先度、確保要求サブプロセッサ数および確保要求サブプロセッサ数の分のサブプロセッサプログラム ID を含むものとする。

【 0 6 0 8 】

情報処理装置 1 0 0 1 内の空いているサブプロセッサだけでは  $n$  個のサブプロセッサを確保できないが、情報処理装置 1 0 0 1 以外にもネットワーク上に情報処理装置が存在して、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、 $n$  個のサブプロセッサを確保できる場合には、このように他の情報処理装置を含めたネットワークシステム全体で  $n$  個のサブプロセッサが確保されることになる。

【 0 6 0 9 】

この場合、他の情報処理装置 1 0 0 2 , 1 0 0 3 および 1 0 0 4 内のリソースマネージャは、ステップ 1 1 7 2 では、それぞれ自装置内に確保した各サブプロセッサにつき、自装置のサブプロセッサ管理テーブル内のサブプロセッサステータスが `u n u s e d` となっているものについては、`r e s e r v e d` に書き替え、サブプロセッサプログラム ID、機能プログラム ID およびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替えるとともに、ロックシーケンス番号を発番してサブプロセッサ管理テーブルに記述する。

【 0 6 1 0 】

また、他の情報処理装置 1 0 0 2 , 1 0 0 3 および 1 0 0 4 内のリソースマネージャは、ステップ 1 1 7 2 で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを、当該の機能プログラムと連携動作するサブプロセッサプログラムの実行用に確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブプロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

【 0 6 1 1 】

さらに、他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャは、図 66 に示すように、ステップ 1173 で、それぞれ自装置内に自装置分のサブプロセッサを確保したことを、情報処理装置 1001 内のリソースマネージャに通知する。

【0612】

この通知には、図 29 (B) に示すような DMA コマンドがサブプロセッサ確保返信コマンドであるソフトウェアセルを用いる。具体的に、このサブプロセッサ確保返信コマンドは、DMA コマンドとして、サブプロセッサ確保返信コマンド本体、機能プログラム ID、サブプロセッサプログラム優先度、確保したサブプロセッサの数、確保した各サブプロセッサのサブプロセッサ ID、サブプロセッサプログラム ID およびロックシーケンス番号を含むものとする。

10

【0613】

図 66 に示すように、情報処理装置 1001 内のリソースマネージャは、他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャからのサブプロセッサ確保通知を受信したら、ステップ 1174 で、当該の機能プログラムに、n 個のサブプロセッサを確保したことを通知する。

【0614】

当該の機能プログラムは、ステップ 1177 で、このサブプロセッサ確保通知を受信し、自装置 (情報処理装置 1001) 内のリソースマネージャに、各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0615】

すなわち、当該の機能プログラムは、図 65 のステップ 1167、または図 66 のステップ 1177 で、情報処理装置 1001 内のリソースマネージャから、n 個のサブプロセッサを確保した旨の通知を受信したら、図 66 に示すように、ステップ 1181 で、情報処理装置 1001 内のリソースマネージャに、確保された各サブプロセッサに対するサブプロセッサプログラムを、当該サブプロセッサプログラムに対応するサブプロセッサプログラム ID、サブプロセッサ ID、ロックシーケンス番号およびサブプロセッサが確保された各情報処理装置の情報処理装置 ID と共に送信する。

20

【0616】

図 66 および図 67 に示すように、情報処理装置 1001 内のリソースマネージャは、ステップ 1182 で、当該の機能プログラムから送信されたサブプロセッサプログラムを受信して、自装置 (情報処理装置 1001) 分のサブプロセッサプログラムがある場合には、それを自装置内の確保されたサブプロセッサに送信し、さらにステップ 1183 に進んで、他の情報処理装置分のサブプロセッサプログラムがあるか否かを判断し、他の情報処理装置分のサブプロセッサプログラムがない場合には、ステップ 1183 からステップ 1184 に進んで、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

30

【0617】

一方、当該の機能プログラムから送信されたサブプロセッサプログラムに他の情報処理装置分のサブプロセッサプログラムがある場合には、情報処理装置 1001 内のリソースマネージャは、ステップ 1183 からステップ 1191 に進んで、他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャに各情報処理装置分のサブプロセッサプログラムを送信する。

40

【0618】

この送信には、図 30 に示すような DMA コマンドがサブプロセッサプログラム送信コマンドであるソフトウェアセルを用いる。具体的に、このサブプロセッサプログラム送信コマンドは、DMA コマンドとして、サブプロセッサプログラム送信コマンド本体、確保された各サブプロセッサのサブプロセッサ ID、ロックシーケンス番号、確保された各サブプロセッサに対するサブプロセッサプログラムのサブプロセッサプログラム ID、確保された各サブプロセッサに対するサブプロセッサプログラム本体を含むものとする。

【0619】

50

他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャは、ステップ 1192 で、それぞれ自装置分のサブプロセッサプログラムを、自装置内の確保されたサブプロセッサに送信し、さらにステップ 1193 に進んで、情報処理装置 1001 内のリソースマネージャに、確保されたサブプロセッサにサブプロセッサプログラムを送信したことを通知する。この通知を受けて、情報処理装置 1001 内のリソースマネージャは、ステップ 1184 で、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0620】

なお、情報処理装置 1001 内のリソースマネージャ、および他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャは、図 66 のステップ 1182、および図 67 のステップ 1192 では、当該の機能プログラムから受信した確保された各サブプロセッサに対するサブプロセッサプログラムのサブプロセッサプログラム ID、およびロックシーケンス番号が、自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断して、異なる場合には、情報処理装置 1001 内のリソースマネージャが直接、または他の情報処理装置 1002, 1003 および 1004 内のリソースマネージャが情報処理装置 1001 内のリソースマネージャを介して、当該の機能プログラムにエラーを通知するが、図 66 および図 67 では、この点を省略して、エラーを生じない場合のみを示した。

【0621】

図 67 に示すように、当該の機能プログラムは、ステップ 1187 で、情報処理装置 1001 内のリソースマネージャからの、確保された各サブプロセッサにサブプロセッサプログラムを送信した旨の通知を受信する。

【0622】

これによって、情報処理装置 1001 内の空いているサブプロセッサだけでは  $n$  個のサブプロセッサを確保できないが、情報処理装置 1001 内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、情報処理装置 1001 内だけで  $n$  個のサブプロセッサを確保できる場合、あるいは、情報処理装置 1001 内の空いているサブプロセッサだけでは  $n$  個のサブプロセッサを確保できないが、情報処理装置 1001 以外にもネットワーク上に情報処理装置が存在していて、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、 $n$  個のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

【0623】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

【0624】

一方、情報処理装置 1001 内のリソースマネージャは、図 64 のステップ 1151 で、ネットワーク上に他の情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体の空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、 $n$  個のサブプロセッサを確保できないと判断した場合には、図 68 に示すように、ステップ 1151 からステップ 1195 に進んで、当該の機能プログラムに、 $n$  個のサブプロセッサを確保できないことを通知する。

【0625】

当該の機能プログラムは、ステップ 1197 で、この通知を受信し、エラー処理を実行する。この場合のエラー処理は、ユーザに対して直ちに処理を実行できない旨を通知して、時間を置いて再度、図 60 のステップ 1112 からサブプロセッサ割り当て処理を実行し、またはユーザが操作することをユーザに促すなどである。

【0626】

なお、上述したように優先度の低いサブプロセッサプログラムに対してサブプロセッサが奪われたことが通知された場合には、当該のサブプロセッサプログラムにつき、それに対応する別の処理が実行される。

【0627】

上述した例は、情報処理装置IDの置き換えられた数値の昇順に、すなわち情報処理装置1001 情報処理装置1002 情報処理装置1003 情報処理装置1004の順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求する場合であるが、例えば、各情報処理装置内のリソースマネージャは、実行されるべきサブプロセッサプログラムの優先度が高い場合には、上述した装置情報の一部であるメインプロセッサ動作周波数により、ネットワーク1009上に接続されている各情報処理装置の動作周波数を判断して、動作周波数が高い情報処理装置から順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求するように、システムを構成してもよい。

10

【0628】

あるいはまた、装置情報の一部である情報処理装置種別IDにより、ネットワーク1009上に接続されている各情報処理装置の種別を判断することによって、ハードディスクレコーダのようにネットワーク1009から切断される可能性が低い情報処理装置から、PDAやポータブルCDプレーヤのようにネットワーク1009から切断される可能性が高い情報処理装置への順に、サブプロセッサの空き状況を問い合わせ、サブプロセッサの確保を要求するように、システムを構成してもよい。

【0629】

20

また、上述した例は、情報処理装置1001内の空いているサブプロセッサだけでは必要数のサブプロセッサを確保できない場合、情報処理装置1001内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、まずサブプロセッサの空き状況を問い合わせ、その結果、ネットワークシステム全体で必要数のサブプロセッサを確保できる場合には、改めてサブプロセッサの確保を要求する場合であるが、サブプロセッサの空き状況を問い合わせることなく、情報処理装置1001内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、サブプロセッサの確保を要求するように、システムを構成してもよい。

【0630】

さらに、上述した例は、情報処理装置1001がマスター装置として動作していて、そのマスター装置を起点としてサブプロセッサ割り当て処理を実行する場合であるが、例えば、スレーブ装置として動作している情報処理装置がユーザによって操作された場合に、その操作されたスレーブ装置を起点としてサブプロセッサ割り当て処理を実行するように、システムを構成することもできる。

30

【0631】

[ 8 . 情報処理装置がネットワークから切断された場合のサブプロセッサの割り当てによる分散処理その2 ]

図59のように複数の情報処理装置1001, 1002, 1003および1004がネットワーク1009に接続されている状態で、ユーザが誤って、ある情報処理装置をネットワーク1009から切断した場合(情報処理装置を物理的にネットワーク1009から切り離した場合、または情報処理装置の主電源を遮断した場合)、または、強い震動などの外的要因によって、ある情報処理装置がネットワーク1009から切断された場合、その切断された情報処理装置内のサブプロセッサが、切断されていない情報処理装置が有するサブプロセッサプログラムの実行中または実行予約中であったときには、切断された情報処理装置内のサブプロセッサに代わって、切断されていない情報処理装置内のサブプロセッサが、そのサブプロセッサプログラムを実行することが望ましい。

40

【0632】

そこで、このような場合には、以下のように、切断された情報処理装置内のサブプロセッサが実行中または実行予約中であったサブプロセッサプログラムの実行用に、切断されていない情報処理装置内のサブプロセッサを割り当てる。

50

## 【0633】

## (8-1. システム構成)

図69に、図59のように4台の情報処理装置1001, 1002, 1003および1004がネットワーク1009に接続されている状態で、情報処理装置1004がネットワーク1009から切断された場合を示す。

## 【0634】

各情報処理装置のソフトウェア構成としては、図17に示したように、制御プログラムとしてリソースマネージャを含むものとし、そのリソースマネージャには、図18に示したようなサブプロセッサ管理テーブルを用意する。サブプロセッサ管理テーブルの内容は、上述したとおりである。

10

## 【0635】

## (8-2. サブプロセッサ割り当て処理)

図70~図78に、図69のように情報処理装置1001, 1002, 1003および1004がネットワーク1009に接続されている状態で、情報処理装置1004がネットワーク1009から切断された場合に、情報処理装置1001が、自装置のメインメモリ1026-1内の機能プログラムと連携動作するサブプロセッサプログラムの実行用に、情報処理装置1001内または他の情報処理装置1002, 1003内にサブプロセッサを確保する場合の、情報処理装置1001内のMS(マスター/スレーブ)マネージャ、情報処理装置1001のメインメモリ1026-1内の機能プログラム、情報処理装置1001内のリソースマネージャ、および他の情報処理装置1002および1003内の

20

## 【0636】

情報処理装置1004がネットワーク1009から切断されたことは、情報処理装置1001, 1002および1003内のMSマネージャによって検出される。

## 【0637】

そして、情報処理装置1001内のMSマネージャは、情報処理装置1004がネットワーク1009から切断されたことを検出したら、図70に示すように、ステップ1201で、自装置(情報処理装置1001)内のリソースマネージャに、切断された情報処理装置1004の情報処理装置IDを通知し、情報処理装置1001内のリソースマネージャは、ステップ1202で、自装置のメインメモリ1026-1内の機能プログラムに、その情報処理装置IDを転送する。

30

## 【0638】

これを受けて、情報処理装置1001のメインメモリ1026-1内の機能プログラムは、ステップ1203で、図28~図30および図60~図68に示して上述したようなサブプロセッサ割り当て処理の結果から、切断された情報処理装置1004内のサブプロセッサが、自身(当該の機能プログラム)と連携動作するサブプロセッサプログラムを実行中であったか否かを判断する。

## 【0639】

この場合の「実行中」というのも、busy(現在使用されている状態)だけでなく、reserved(現在は使用されていないが、使用が予約されている状態)を含む。

40

## 【0640】

そして、切断された情報処理装置1004内の全てのサブプロセッサが、当該の機能プログラムと連携動作するサブプロセッサプログラムを実行中でなかった場合には、以後の処理は不要であるので、そのままサブプロセッサ割り当て処理を終了する。

## 【0641】

一方、切断された情報処理装置1004内の一部または全部のサブプロセッサが、当該の機能プログラムと連携動作するサブプロセッサプログラムを実行中であった場合には、当該の機能プログラムは、ステップ1203からステップ1212に進んで、自装置(情報処理装置1001)内のリソースマネージャに、当該の機能プログラムの機能プログラムIDおよび当該機能プログラムと連携動作する、切断された情報処理装置1004内の

50

サブプロセッサにより実行中（上記のように実行予約中を含む）のサブプロセッサプログラムのサブプロセッサプログラムIDを添えて、切断された情報処理装置1004内のサブプロセッサに代わって当該の機能プログラムと連携動作するサブプロセッサプログラムを実行するのに必要な数（m個とする）のサブプロセッサの使用要求を出力する。

【0642】

これに対して、情報処理装置1001内のリソースマネージャは、図71に示すように、ステップ1213で、自装置（情報処理装置1001）の図18に示したサブプロセッサ管理テーブルを参照して、自装置内にm個のサブプロセッサの空きがあるか否かを判断し、m個以上のサブプロセッサが空いている場合には、ステップ1213からステップ1214に進んで、自装置内にm個のサブプロセッサを確保するとともに、その確保した各サブプロセッサにつき、サブプロセッサ管理テーブル内のサブプロセッサステータスを、`unused`から`reserved`に書き替え、サブプロセッサプログラムID、機能プログラムIDおよびサブプロセッサプログラム優先度を、確保したサブプロセッサで実行されるサブプロセッサプログラムと一致するように書き替え、ロックシーケンス番号を発番して、サブプロセッサ管理テーブルに記述する。

10

【0643】

さらに、情報処理装置1001内のリソースマネージャは、ステップ1215に進んで、当該の機能プログラムに、自装置内にm個のサブプロセッサを確保したことと、確保したm個のサブプロセッサのサブプロセッサIDおよびロックシーケンス番号とを通知する。

20

【0644】

当該の機能プログラムは、ステップ1217で、この通知を受信して、自装置（情報処理装置1001）内のリソースマネージャに、自身の機能プログラムIDと共に、確保された各サブプロセッサに対するサブプロセッサID、サブプロセッサプログラムID、サブプロセッサプログラム本体およびロックシーケンス番号を、確保したサブプロセッサ数の分だけ送信する。

【0645】

図72に示すように、情報処理装置1001内のリソースマネージャは、ステップ1221で、これらを受信して、各サブプロセッサに対するサブプロセッサプログラムを保存し、さらにステップ1222に進んで、受信したサブプロセッサプログラムIDおよびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断し、同じであるときには、ステップ1223に進んで、確保した各サブプロセッサにサブプロセッサプログラムを送信し、さらにステップ1224に進んで、当該の機能プログラムに、各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

30

【0646】

以上によって、情報処理装置1001内の空いているサブプロセッサだけで必要数のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

【0647】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

40

【0648】

情報処理装置1001内のリソースマネージャは、ステップ1222で、当該の機能プログラムから受信したサブプロセッサプログラムIDまたはロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと異なると判断した場合には、ステップ1222からステップ1226に進んで、当該の機能プログラムにエラーを通知する。この場合には、当該の機能プログラムは、ステップ1227で、エラー処理を実行する。

【0649】

50

一方、情報処理装置 1001 内のリソースマネージャは、図 71 のステップ 1213 で、自装置内に  $m$  個のサブプロセッサの空きがないと判断した場合、すなわち情報処理装置 1001 内に空いているサブプロセッサが  $(m - 1)$  個以下しかない (1 個もない場合を含む) と判断した場合には、図 73 に示すように、ステップ 1213 からステップ 1229 に進んで、自装置内に空いているサブプロセッサがある場合には、そのサブプロセッサを自装置分として確保する。

【0650】

情報処理装置 1001 内のリソースマネージャは、さらにステップ 1231 に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。図 69 は、他の情報処理装置 1002 および 1003 が接続されている場合である。

10

【0651】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置 1001 内のリソースマネージャは、ステップ 1231 からステップ 1232 に進んで、その中の 1 個である他の情報処理装置 (この場合は情報処理装置 1002) 内のリソースマネージャに、空いているサブプロセッサの確保を要求する。

【0652】

この確保要求は、図 29 (A) に示したようなサブプロセッサ確保要求コマンドによって行い、その確保要求サブプロセッサ数は、 $(m - a)$  とする。  $a$  は、ステップ 1229 で情報処理装置 1001 内に確保されたサブプロセッサ数で、 $0 < a < m$  である。

【0653】

図 73 に示すように、この確保要求に対して、他の情報処理装置 (この場合は情報処理装置 1002) 内のリソースマネージャは、ステップ 1233 で、自装置内のサブプロセッサ管理テーブルから、自装置内のサブプロセッサの空き状況を判断し、次にステップ 1234 に進んで、自装置内に空いているサブプロセッサがある場合には、最大  $(m - a)$  個の範囲で自装置内にサブプロセッサを確保し、さらにステップ 1235 に進んで、自装置内でサブプロセッサを確保した場合には、その数および確保した各サブプロセッサのサブプロセッサ ID、サブプロセッサプログラム ID およびロックシーケンス番号を情報処理装置 1001 内のリソースマネージャに通知し、自装置内でサブプロセッサを確保できない場合には、その旨を情報処理装置 1001 内のリソースマネージャに通知する。この通知は、図 29 (B) に示したようなサブプロセッサ確保返信コマンドによって行う。

20

30

【0654】

この通知を受けて、情報処理装置 1001 内のリソースマネージャは、図 74 に示すように、ステップ 1236 で、自装置 (情報処理装置 1001) 分と他装置 (情報処理装置 1002) 分とを合わせてトータルで、 $m$  個のサブプロセッサを確保できたか否かを判断する。

【0655】

そして、トータルで  $m$  個のサブプロセッサを確保できなかった場合には、情報処理装置 1001 内のリソースマネージャは、図 73 に示すように、ステップ 1236 からステップ 1231 に進んで (戻って)、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。図 69 は、情報処理装置 1002 以外に他の情報処理装置 1003 が接続されている場合である。

40

【0656】

以下、情報処理装置 1002 の場合と同様に、ネットワークシステム全体で  $m$  個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ 1232, 1233, 1234, 1235, 1236 および 1231 の処理を繰り返す。

【0657】

そして、ネットワークシステム全体で、空いているサブプロセッサによって  $m$  個のサブプロセッサを確保できた場合には、情報処理装置 1001 内のリソースマネージャは、図 74 に示すように、ステップ 1236 からステップ 1237 に進んで、当該の機能プログ

50

ラムに、m個のサブプロセッサを確保したことを通知する。

【0658】

当該の機能プログラムは、ステップ1238で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置1001）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0659】

一方、情報処理装置1001内のリソースマネージャは、図73のステップ1231で、それ以上、ネットワーク上に情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体でも（情報処理装置1001のみがネットワーク上に存在する場合には情報処理装置1001内だけで）、空いているサブプロセッサによってm個のサブプロセッサを確保できないと判断した場合には、図74に示すように、ステップ1231からステップ1241に進んで、自装置（情報処理装置1001）内に当該の機能プログラムより優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否か、および、ある場合には、その数を検出する。

10

【0660】

この場合の「実行中」というのも、busy（現在使用されている状態）だけでなく、reserved（現在は使用されていないが、使用が予約されている状態）を含む。以下でも、当該の機能プログラムより優先度の低いサブプロセッサプログラムを、単に「優先度の低いサブプロセッサプログラム」と称する。

【0661】

次に、情報処理装置1001内のリソースマネージャは、ステップ1242に進んで、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがある場合には、ネットワークシステム全体の空いているサブプロセッサと合わせてm個のサブプロセッサを確保できる範囲で、その優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保するとともに、その優先度の低いサブプロセッサプログラムに対してサブプロセッサが奪われたことを通知する。

20

【0662】

さらに、情報処理装置1001内のリソースマネージャは、図75に示すように、ステップ1243に進んで、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置1001）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、m個のサブプロセッサを確保できたか否かを判断し、確保できた場合には、ステップ1243からステップ1244に進んで、当該の機能プログラムに、m個のサブプロセッサを確保したことを通知する。

30

【0663】

当該の機能プログラムは、ステップ1248で、このサブプロセッサ確保通知を受信し、後述のように自装置（情報処理装置1001）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

【0664】

一方、情報処理装置1001内のリソースマネージャは、ステップ1243で、ネットワークシステム全体の空いているサブプロセッサと、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、m個のサブプロセッサを確保できないと判断した場合には、ステップ1243からステップ1251に進んで、ネットワーク上に他の情報処理装置が接続されているか否かを判断する。

40

【0665】

そして、ネットワーク上に他の情報処理装置が接続されている場合には、情報処理装置1001内のリソースマネージャは、ステップ1251からステップ1252に進んで、当該他の情報処理装置（この場合は情報処理装置1002）内のリソースマネージャに、優先度の低いサブプロセッサプログラムを実行中のサブプロセッサの確保を要求する。この確保要求は、図29（A）に示したようなサブプロセッサ確保要求コマンドによって行う。

50

## 【 0 6 6 6 】

図 7 6 に示すように、この確保要求に対して、他の情報処理装置（この場合は情報処理装置 1 0 0 2 ）内のリソースマネージャは、ステップ 1 2 5 3 で、自装置内のサブプロセッサ管理テーブルから、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがあるか否かを判断し、ある場合には、その優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保して、その数および確保した各サブプロセッサのサブプロセッサ ID、サブプロセッサプログラム ID および発番したロックシーケンス番号を情報処理装置 1 0 0 1 内のリソースマネージャに通知し、自装置内に優先度の低いサブプロセッサプログラムを実行中のサブプロセッサがない場合には、その旨を情報処理装置 1 0 0 1 内のリソースマネージャに通知する。この通知は、図 2 9 ( B ) に示したようなサブプロセッサ確保返信コマンドによって行う。

10

## 【 0 6 6 7 】

なお、他の情報処理装置（この場合は情報処理装置 1 0 0 2 ）内のリソースマネージャは、ステップ 1 2 5 3 で、自装置内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを確保した場合には、その優先度の低いサブプロセッサプログラムおよび当該サブプロセッサプログラムと連携動作する機能プログラムに対してサブプロセッサが奪われたことを通知する。

## 【 0 6 6 8 】

他の情報処理装置（この場合は情報処理装置 1 0 0 2 ）内のリソースマネージャからの確保通知を受けて、情報処理装置 1 0 0 1 内のリソースマネージャは、図 7 6 に示すように、ステップ 1 2 5 4 で、ネットワークシステム全体の空いているサブプロセッサと、自装置（情報処理装置 1 0 0 1 ）内および他装置（情報処理装置 1 0 0 2 ）内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、m 個のサブプロセッサを確保できたか否かを判断し、確保できない場合には、図 7 5 に示すように、ステップ 1 2 5 4 からステップ 1 2 5 1 に進んで（戻って）、さらにネットワーク上に他の情報処理装置が接続されているか否かを判断する。

20

## 【 0 6 6 9 】

以下、情報処理装置 1 0 0 2 の場合と同様に、ネットワークシステム全体で m 個のサブプロセッサを確保できるまで、または、それ以上、ネットワーク上に情報処理装置が接続されていないことが検出されるまで、ステップ 1 2 5 2 , 1 2 5 3 , 1 2 5 4 および 1 2 5 1 の処理を繰り返す。

30

## 【 0 6 7 0 】

そして、空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせて、ネットワークシステム全体で m 個のサブプロセッサを確保できた場合には、情報処理装置 1 0 0 1 内のリソースマネージャは、ステップ 1 2 5 4 からステップ 1 2 5 5 に進んで、当該の機能プログラムに、m 個のサブプロセッサを確保したことを通知する。

## 【 0 6 7 1 】

当該の機能プログラムは、ステップ 1 2 5 8 で、このサブプロセッサ確保通知を受信し、以下のように自装置（情報処理装置 1 0 0 1 ）内のリソースマネージャに各サブプロセッサに対するサブプロセッサプログラムを送信する。

40

## 【 0 6 7 2 】

すなわち、当該の機能プログラムは、図 7 4 のステップ 1 2 3 8、図 7 5 のステップ 1 2 4 8、または図 7 6 のステップ 1 2 5 8 で、情報処理装置 1 0 0 1 内のリソースマネージャから、m 個のサブプロセッサを確保した旨の通知を受信したら、図 7 7 に示すように、ステップ 1 2 8 1 で、情報処理装置 1 0 0 1 内のリソースマネージャに、確保された各サブプロセッサに対するサブプロセッサプログラムを送信する。

## 【 0 6 7 3 】

図 7 7 に示すように、情報処理装置 1 0 0 1 内のリソースマネージャは、ステップ 1 2 8 2 で、当該の機能プログラムから送信されたサブプロセッサプログラムを受信して、自

50

装置（情報処理装置1001）分のサブプロセッサプログラムがある場合には、それを自装置内の確保されたサブプロセッサに送信し、さらにステップ1283に進んで、他の情報処理装置分のサブプロセッサプログラムがあるか否かを判断し、他の情報処理装置分のサブプロセッサプログラムがない場合には、ステップ1283からステップ1284に進んで、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

【0674】

一方、当該の機能プログラムから送信されたサブプロセッサプログラムに他の情報処理装置分のサブプロセッサプログラムがある場合には、情報処理装置1001内のリソースマネージャは、ステップ1283からステップ1291に進んで、他の情報処理装置1002, 1003のリソースマネージャに各情報処理装置分のサブプロセッサプログラムを送信する。

10

【0675】

他の情報処理装置1002, 1003のリソースマネージャは、ステップ1292で、それぞれ自装置分のサブプロセッサプログラムを、自装置内の確保されたサブプロセッサに送信し、さらにステップ1293に進んで、情報処理装置1001内のリソースマネージャに、確保されたサブプロセッサにサブプロセッサプログラムを送信したことを通知する。この通知を受けて、情報処理装置1001内のリソースマネージャは、ステップ1284で、当該の機能プログラムに、確保された各サブプロセッサにサブプロセッサプログラムを送信したことを通知する。

20

【0676】

なお、情報処理装置1001内のリソースマネージャ、および他の情報処理装置1002, 1003内のリソースマネージャは、図73のステップ1229, 1234、図74のステップ1242、または図76のステップ1253で、自装置内にサブプロセッサを確保した場合には、その確保したサブプロセッサにつき、自装置内のサブプロセッサ管理テーブルを書き替えるとともに、情報処理装置1001, 1002または1003内のリソースマネージャが発番したロックシーケンス番号をサブプロセッサ管理テーブルに記述し、図77のステップ1282および1292では、当該の機能プログラムから受信したサブプロセッサプログラムIDおよびロックシーケンス番号が自装置のサブプロセッサ管理テーブル内のそれらと同じであるか否かを判断して、異なる場合には、情報処理装置1001内のリソースマネージャが直接、または他の情報処理装置1002, 1003内のリソースマネージャが情報処理装置1001内のリソースマネージャを介して、当該の機能プログラムにエラーを通知するが、図74～図77では、この点を省略して、エラーを生じない場合のみを示した。

30

【0677】

図77に示すように、当該の機能プログラムは、ステップ1287で、情報処理装置1001内のリソースマネージャからの、確保された各サブプロセッサにサブプロセッサプログラムを送信した旨の通知を受信する。

【0678】

これによって、情報処理装置1001内の空いているサブプロセッサだけではm個のサブプロセッサを確保できないが、情報処理装置1001内の優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、情報処理装置1001内だけでm個のサブプロセッサを確保できる場合、あるいは、情報処理装置1001内の空いているサブプロセッサだけではm個のサブプロセッサを確保できないが、情報処理装置1001以外にもネットワーク上に情報処理装置が存在していて、ネットワークシステム全体で、空いているサブプロセッサだけで、または優先度の低いサブプロセッサプログラムを実行中のサブプロセッサを含めると、m個のサブプロセッサを確保できる場合の、サブプロセッサ割り当て処理を終了する。

40

【0679】

確保された各サブプロセッサで処理されるべきデータは、サブプロセッサプログラムの

50

送信と同時に、またはサブプロセッサプログラムの送信後、確保された各サブプロセッサに送信される。これによって、確保された各サブプロセッサでサブプロセッサプログラムを実行することができる。

【0680】

一方、情報処理装置1001内のリソースマネージャは、図75のステップ1251で、ネットワーク上に他の情報処理装置が接続されていないと判断した場合、すなわち、ネットワークシステム全体の空いているサブプロセッサと優先度の低いサブプロセッサプログラムを実行中のサブプロセッサとを合わせても、m個のサブプロセッサを確保できないと判断した場合には、図78に示すように、ステップ1251からステップ1295に進んで、当該の機能プログラムに、m個のサブプロセッサを確保できないことを通知する。当該の機能プログラムは、ステップ1297で、この通知を受信し、エラー処理を実行する。

10

【0681】

上述した例は、情報処理装置IDの置き換えられた数値の昇順に、すなわち情報処理装置1001 情報処理装置1002 情報処理装置1003の順に、サブプロセッサの確保を要求する場合であるが、例えば、各情報処理装置内のリソースマネージャは、実行されるべきサブプロセッサプログラムの優先度が高い場合には、上述した装置情報の一部であるメインプロセッサ動作周波数により、ネットワーク1009上に接続されている各情報処理装置の動作周波数を判断して、動作周波数が高い情報処理装置から順に、サブプロセッサの確保を要求するように、システムを構成してもよい。

20

【0682】

あるいはまた、装置情報の一部である情報処理装置種別IDにより、ネットワーク1009上に接続されている各情報処理装置の種別を判断することによって、ハードディスクレコーダのようにネットワーク1009から切断される可能性が低い情報処理装置から、PDAやポータブルCDプレーヤのようにネットワーク1009から切断される可能性が高い情報処理装置への順に、サブプロセッサの確保を要求するように、システムを構成してもよい。

【0683】

また、上述した例は、情報処理装置1001内の空いているサブプロセッサだけでは必要数のサブプロセッサを確保できない場合、情報処理装置1001内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、直ちにサブプロセッサの確保を要求する場合であるが、図60～図68に示して上述した例のように、情報処理装置1001内のリソースマネージャから、他の情報処理装置内のリソースマネージャに対して、まずサブプロセッサの空き状況を問い合わせ、その結果、ネットワークシステム全体で必要数のサブプロセッサを確保できる場合には、改めてサブプロセッサの確保を要求するように、システムを構成してもよい。

30

【図面の簡単な説明】

【0684】

【図1】この発明のネットワークシステムの一例を示す図である。

【図2】この発明の情報処理装置が備える情報処理コントローラの説明に供する図である。

40

【図3】ソフトウェアセルの一例を示す図である。

【図4】ソフトウェアセルのデータ領域の例を示す図である。

【図5】複数の情報処理装置が仮想的な1台の情報処理装置として動作する様子を示す図である。

【図6】情報処理装置のソフトウェア構成の一例を示す図である。

【図7】4台の情報処理装置が仮想的な1台の情報処理装置として動作する様子を示す図である。

【図8】図7のシステムにおける分散処理の例を示す図である。

【図9】各情報処理装置およびシステムの具体例を示す図である。

50

- 【図10】図9中のハードディスクレコーダのハードウェア構成を示す図である。
- 【図11】図9中のハードディスクレコーダのソフトウェア構成を示す図である。
- 【図12】図9中のPDAのハードウェア構成を示す図である。
- 【図13】図9中のPDAのソフトウェア構成を示す図である。
- 【図14】図9中のポータブルCDプレーヤのハードウェア構成を示す図である。
- 【図15】図9中のポータブルCDプレーヤのソフトウェア構成を示す図である。
- 【図16】サブプロセッサの割り当てによる分散処理が実行されるネットワークシステムの一例を示す図である。
- 【図17】情報処理装置のソフトウェア構成の一例を示す図である。
- 【図18】情報処理装置に用意されるサブプロセッサ管理テーブルの一例を示す図である 10
- 。
- 【図19】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図20】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図21】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図22】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図23】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図24】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図25】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図26】サブプロセッサ割り当て処理の一例の一部を示す図である。
- 【図27】サブプロセッサ割り当て処理の一例の一部を示す図である。 20
- 【図28】サブプロセッサ空き状況問い合わせコマンドおよびサブプロセッサ空き状況返信コマンドの一例を示す図である。
- 【図29】サブプロセッサ確保要求コマンドおよびサブプロセッサ確保返信コマンドの一例を示す図である。
- 【図30】サブプロセッサプログラム送信用のサブプロセッサプログラム送信コマンドの一例を示す図である。
- 【図31】サブプロセッサの割り当てによる分散処理が実行されるネットワークシステムの一例を示す図である。
- 【図32】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図33】サブプロセッサ割り当て処理の他の例の一部を示す図である。 30
- 【図34】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図35】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図36】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図37】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図38】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図39】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図40】サブプロセッサ割り当て処理の他の例の一部を示す図である。
- 【図41】この発明のネットワークシステムの他の例を示す図である。
- 【図42】この発明の情報処理装置が備える情報処理コントローラの説明に供する図である。 40
- 【図43】DMACの内部構造を示す図である。
- 【図44】情報処理装置内のコマンド/レスポンス構造を示す図である。
- 【図45】サブプロセッサからメインメモリへのアクセス時の手順を示す図である。
- 【図46】ワークメモリの内部構造を示す図である。
- 【図47】サブプロセッサの内部構造を示す図である。
- 【図48】コントロールレジスタの内部構造を示す図である。
- 【図49】複数の情報処理装置が仮想的な1台の情報処理装置として動作する様子を示す図である。
- 【図50】4台の情報処理装置が仮想的な1台の情報処理装置として動作する様子を示す図である。 50

【図 5 1】図 5 0 のシステムにおける分散処理の例を示す図である。  
 【図 5 2】各情報処理装置およびシステムの他の具体例を示す図である。  
 【図 5 3】図 5 2 中のハードディスクレコーダのハードウェア構成を示す図である。  
 【図 5 4】図 5 2 中のハードディスクレコーダのソフトウェア構成を示す図である。  
 【図 5 5】図 5 2 中の P D A のハードウェア構成を示す図である。  
 【図 5 6】図 5 2 中の P D A のソフトウェア構成を示す図である。  
 【図 5 7】図 5 2 中のポータブル C D プレーヤのハードウェア構成を示す図である。  
 【図 5 8】図 5 2 中のポータブル C D プレーヤのソフトウェア構成を示す図である。  
 【図 5 9】サブプロセッサの割り当てによる分散処理が実行されるネットワークシステムの他の例を示す図である。

10

【図 6 0】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 1】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 2】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 3】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 4】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 5】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 6】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 7】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 8】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 6 9】サブプロセッサの割り当てによる分散処理が実行されるネットワークシステムの他の例を示す図である。

20

【図 7 0】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 1】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 2】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 3】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 4】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 5】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 6】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 7】サブプロセッサ割り当て処理の他の例の一部を示す図である。  
 【図 7 8】サブプロセッサ割り当て処理の他の例の一部を示す図である。

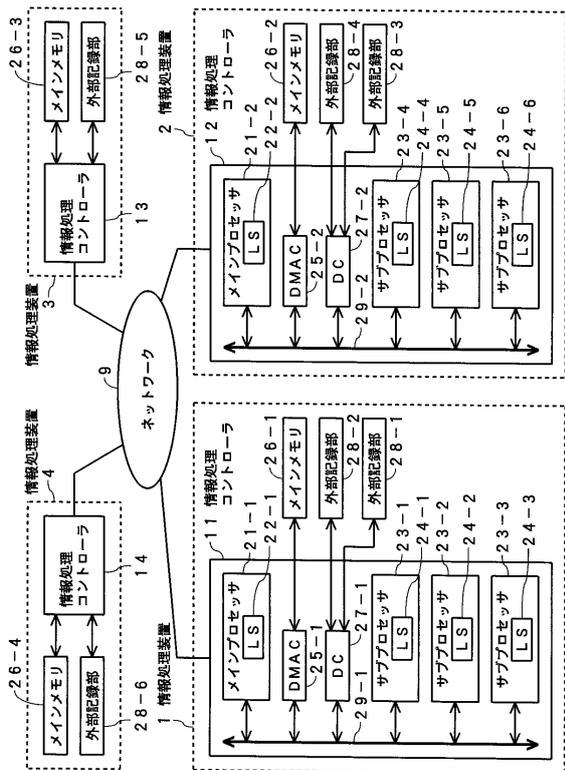
30

【符号の説明】

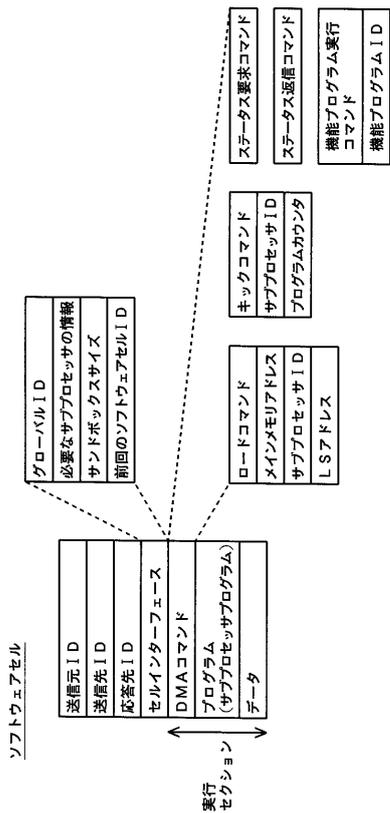
【 0 6 8 5 】

主要部については図中に全て記述したので、ここでは省略する。

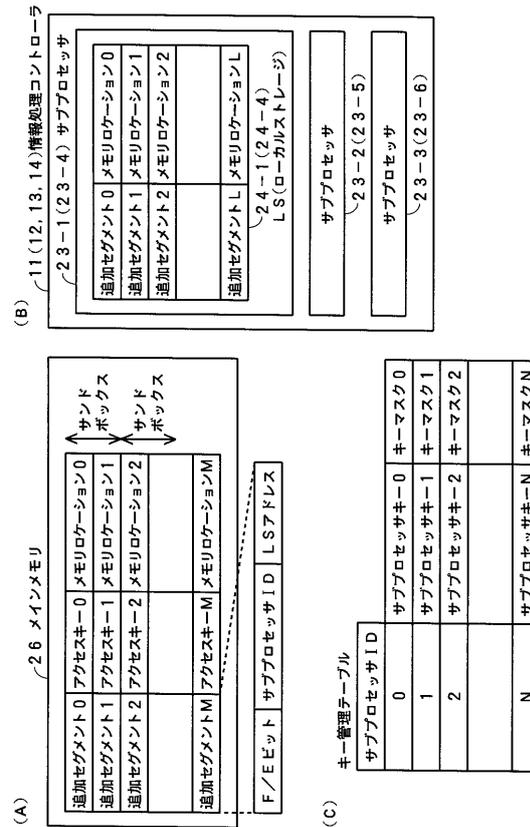
【図 1】



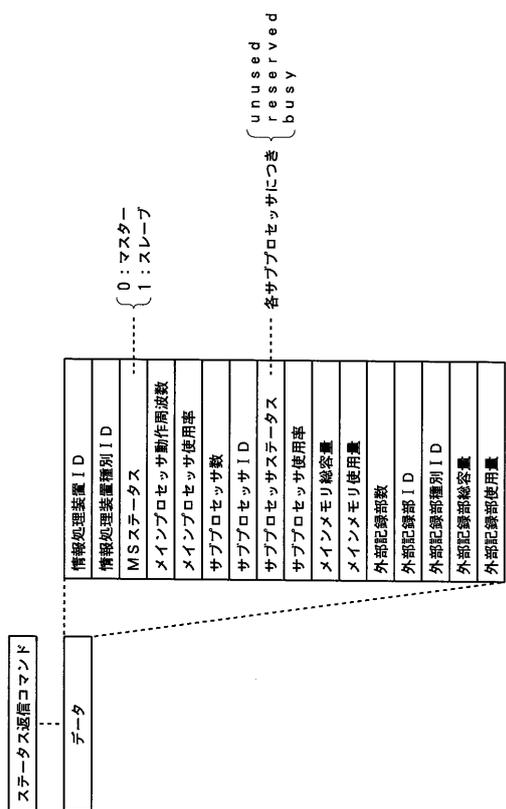
【図 3】



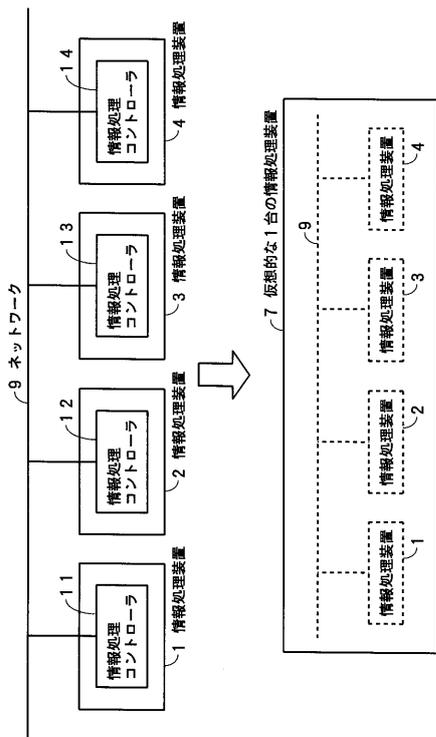
【図 2】



【図 4】

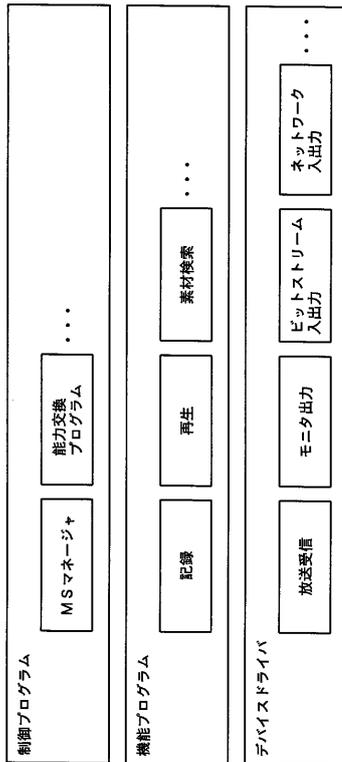


【図 5】

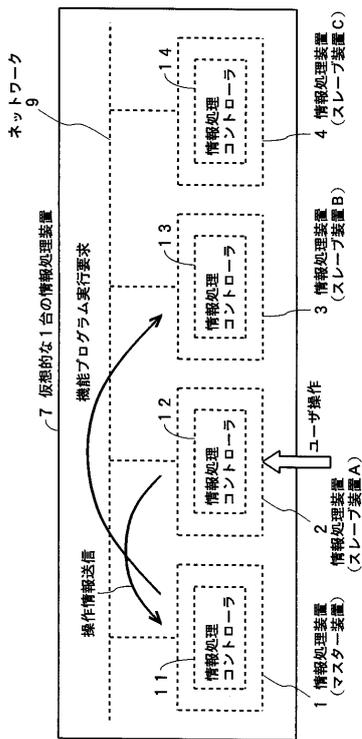


【図 6】

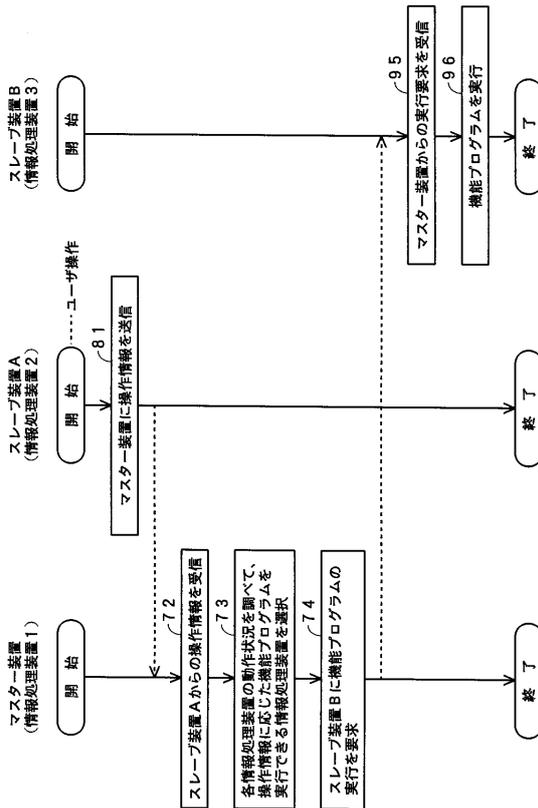
情報処理装置のソフトウェア構成



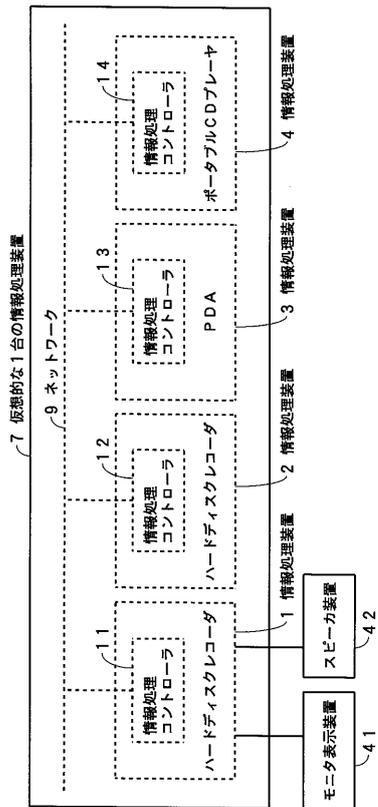
【図 7】



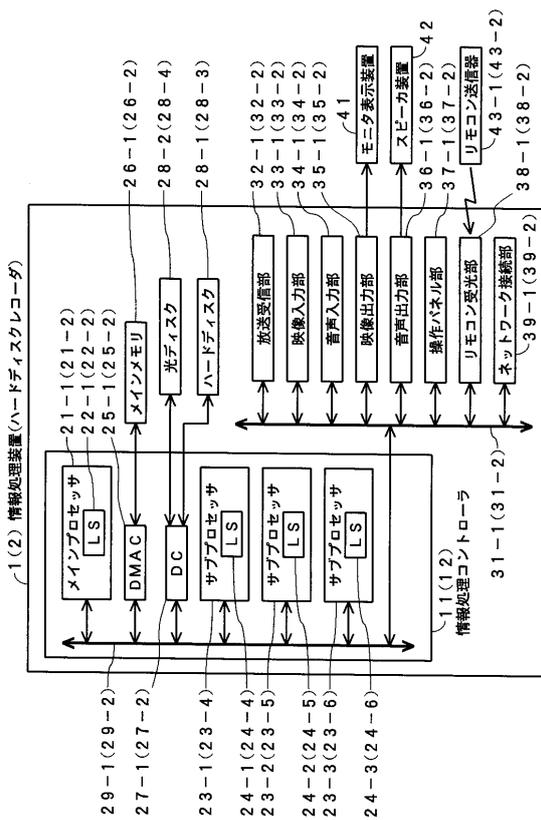
【図 8】



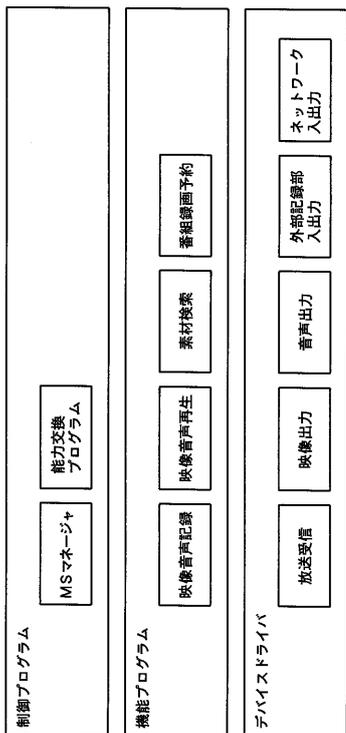
【図9】



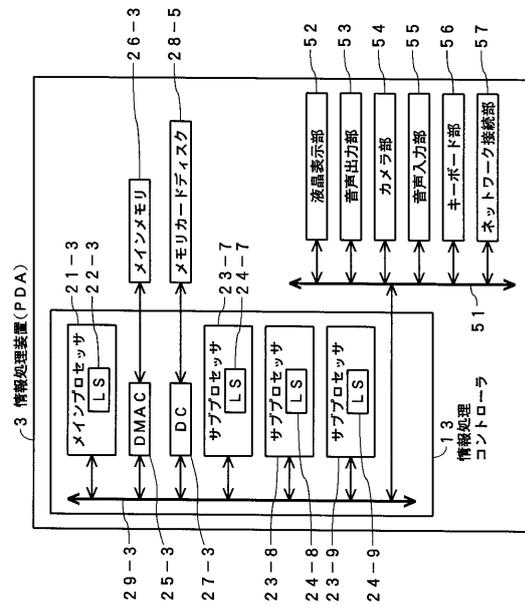
【図10】



【図11】

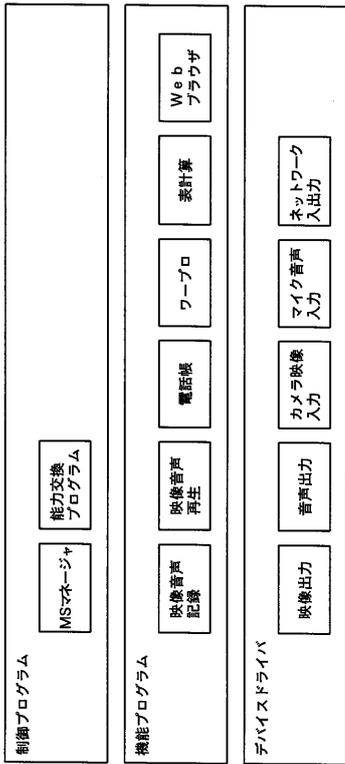


【図12】

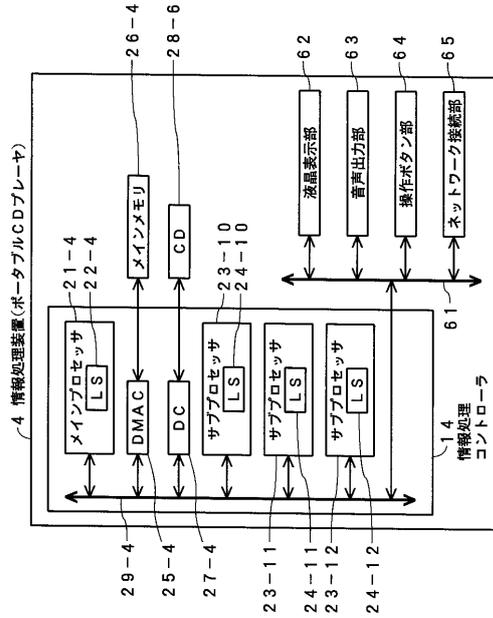


【図 13】

情報処理装置 3 のソフトウェア構成

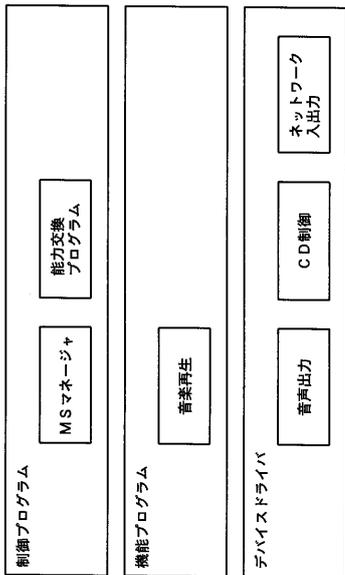


【図 14】

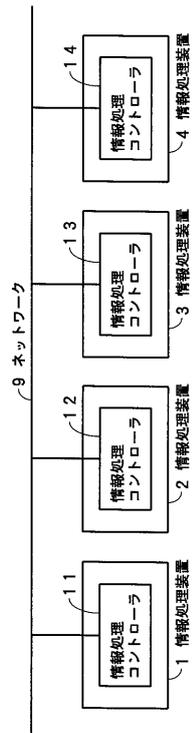


【図 15】

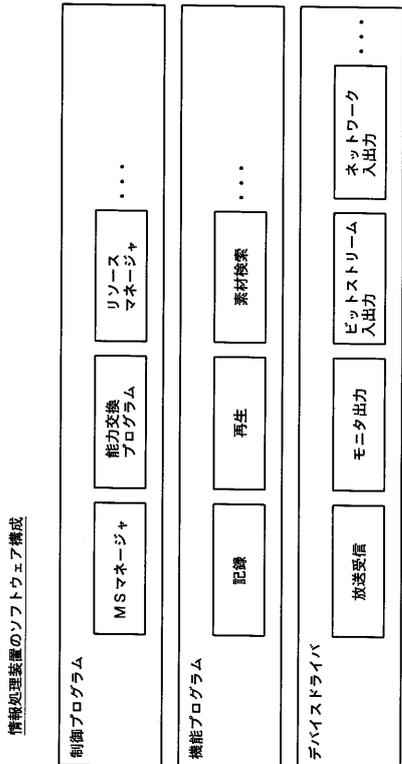
情報処理装置 4 のソフトウェア構成



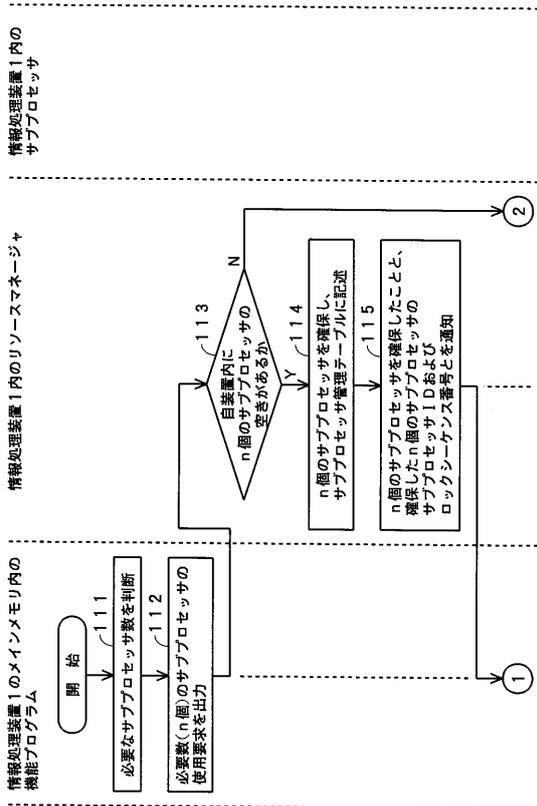
【図 16】



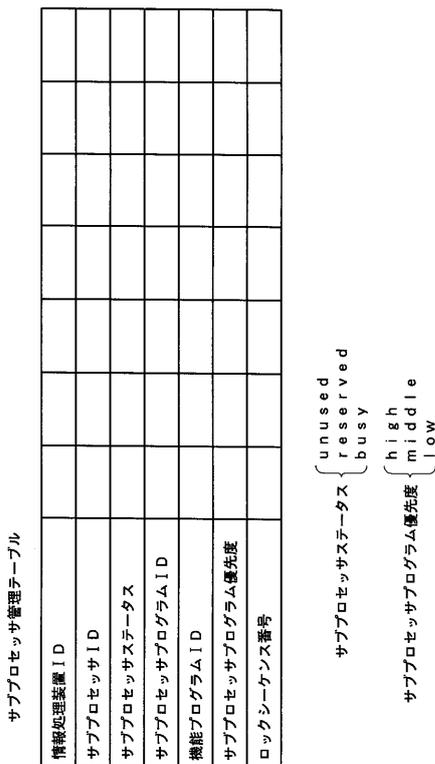
【図17】



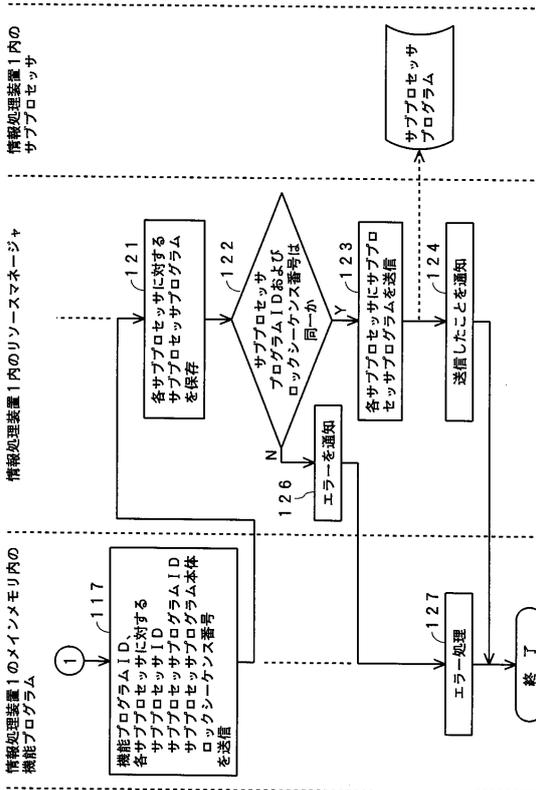
【図19】



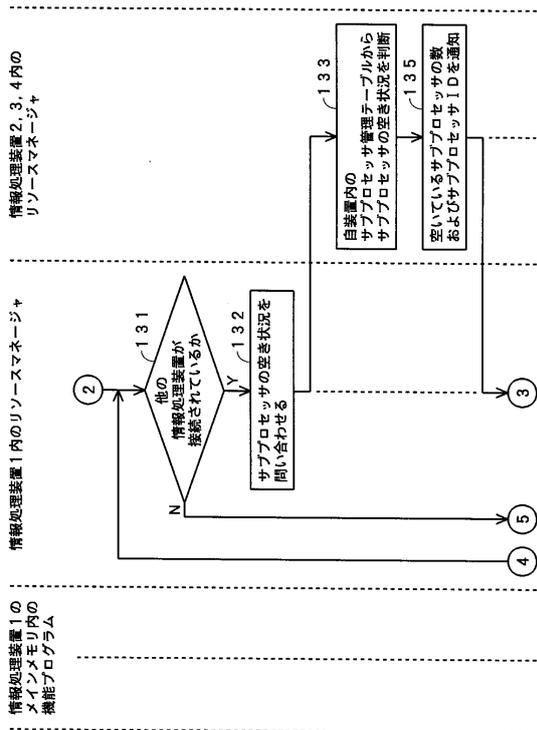
【図18】



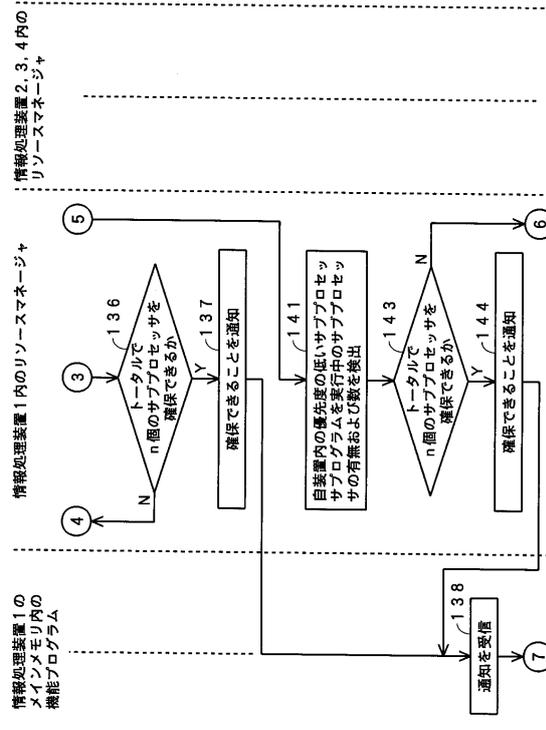
【図20】



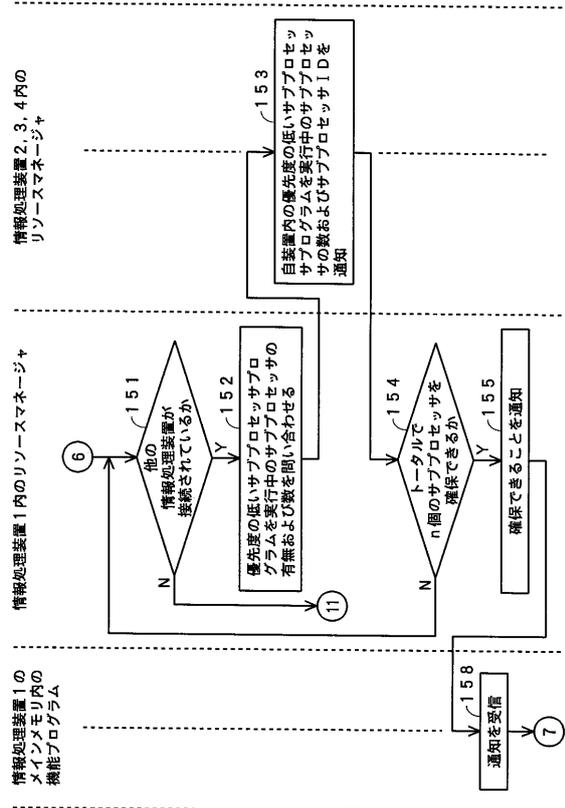
【図 2 1】



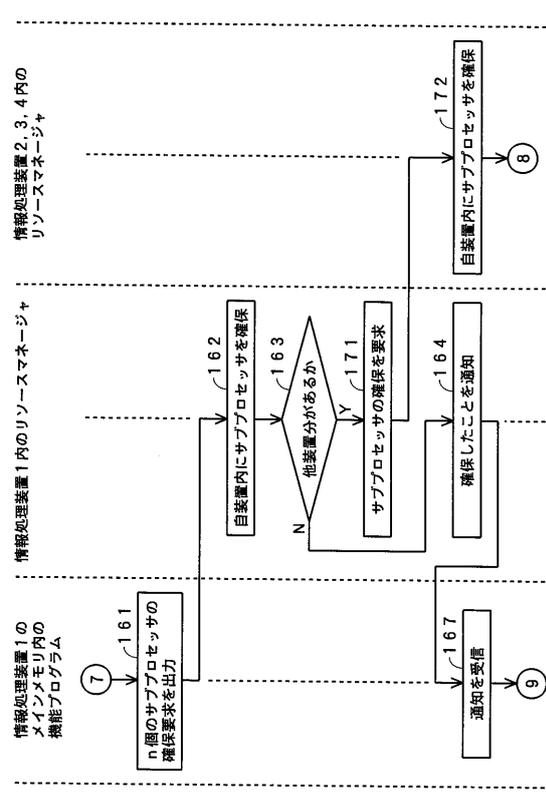
【図 2 2】



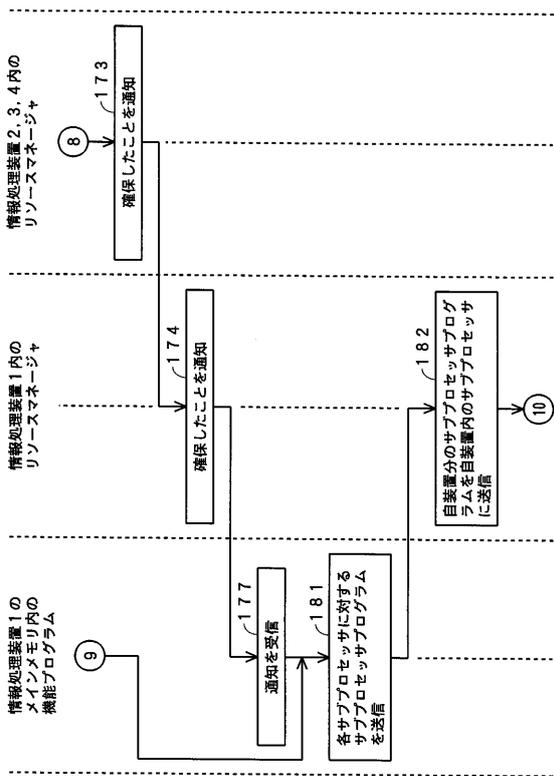
【図 2 3】



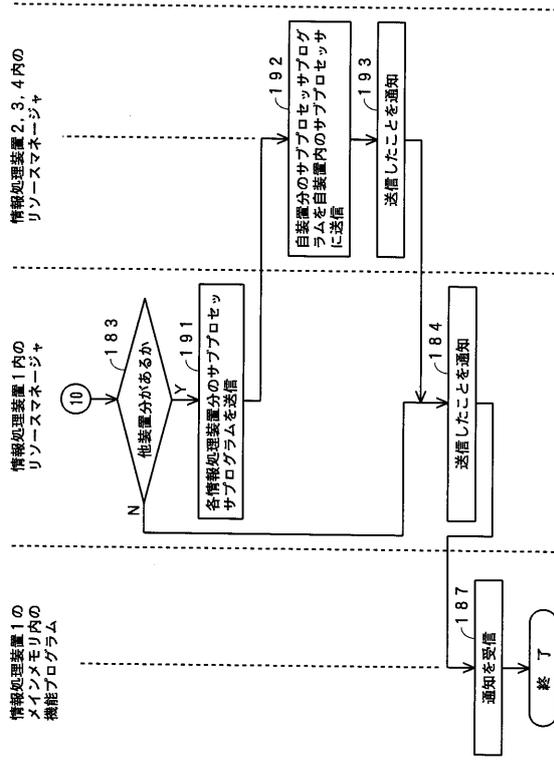
【図 2 4】



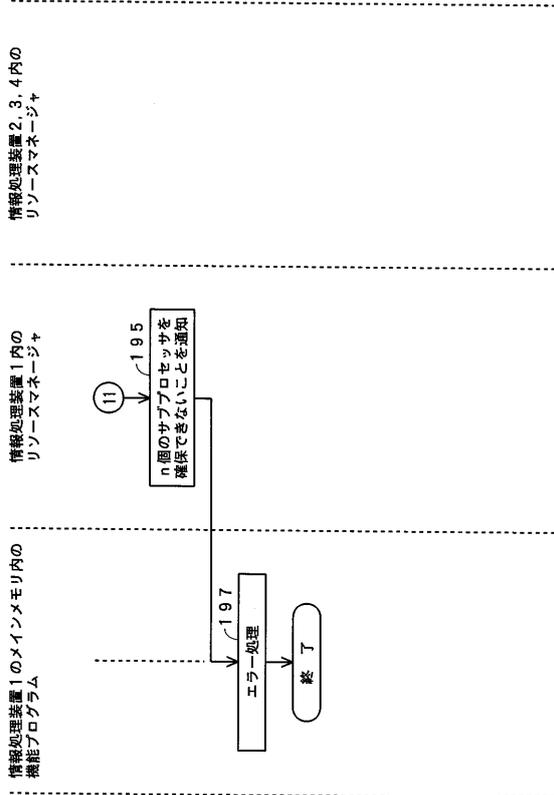
【図 25】



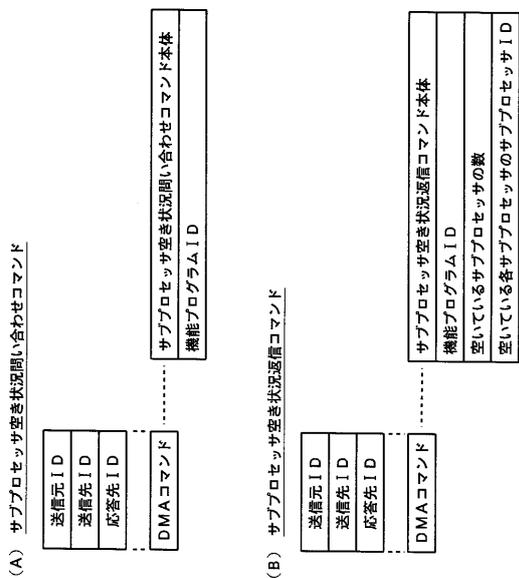
【図 26】



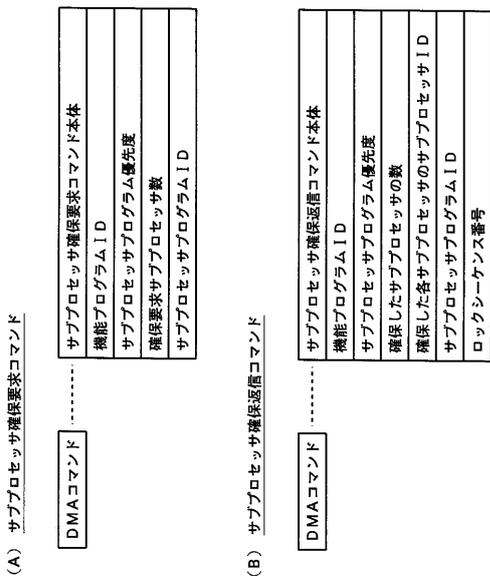
【図 27】



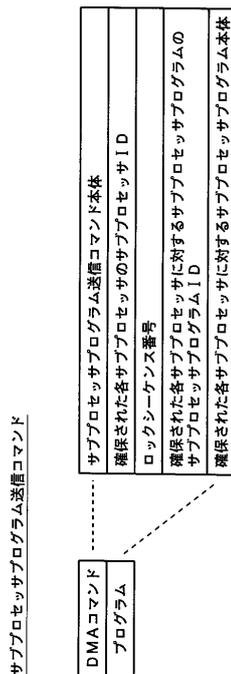
【図 28】



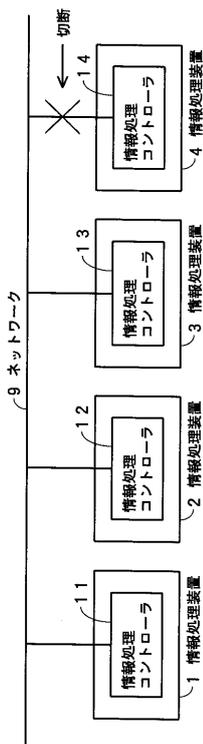
【図 29】



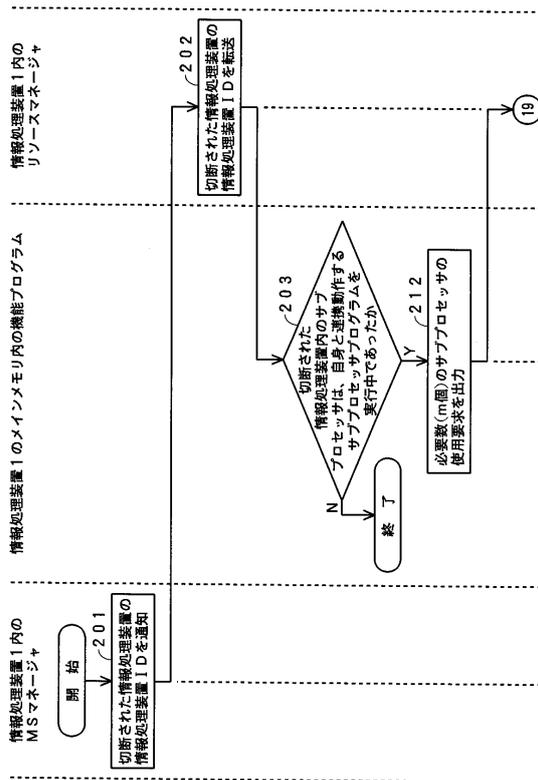
【図 30】



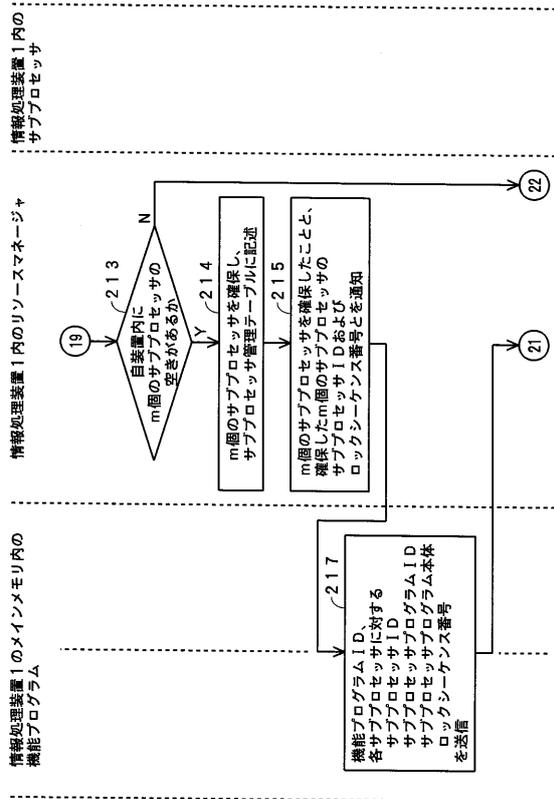
【図 31】



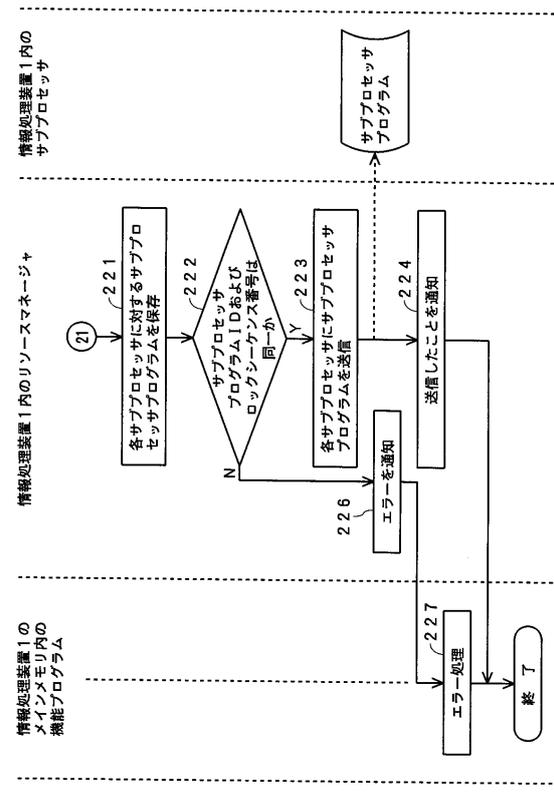
【図 32】



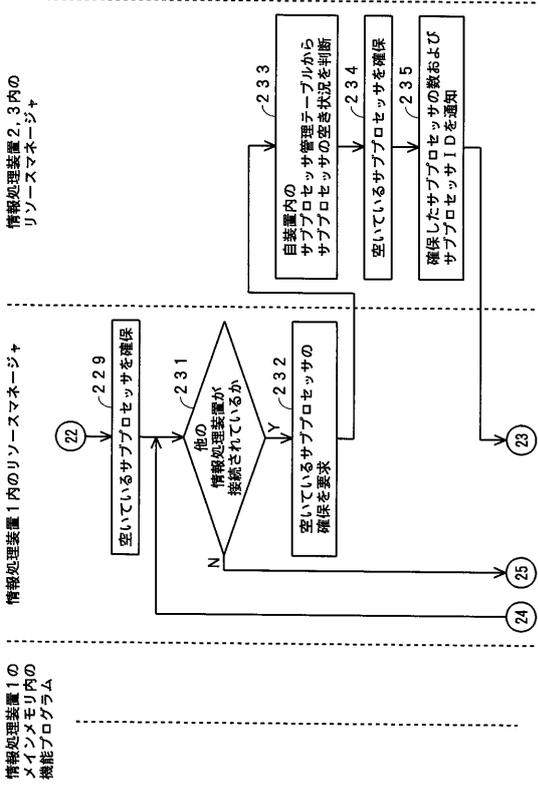
【 図 3 3 】



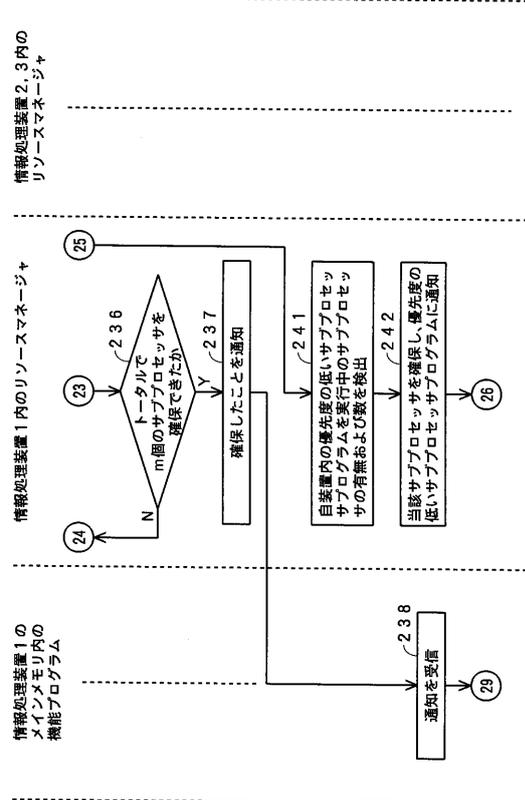
【 図 3 4 】



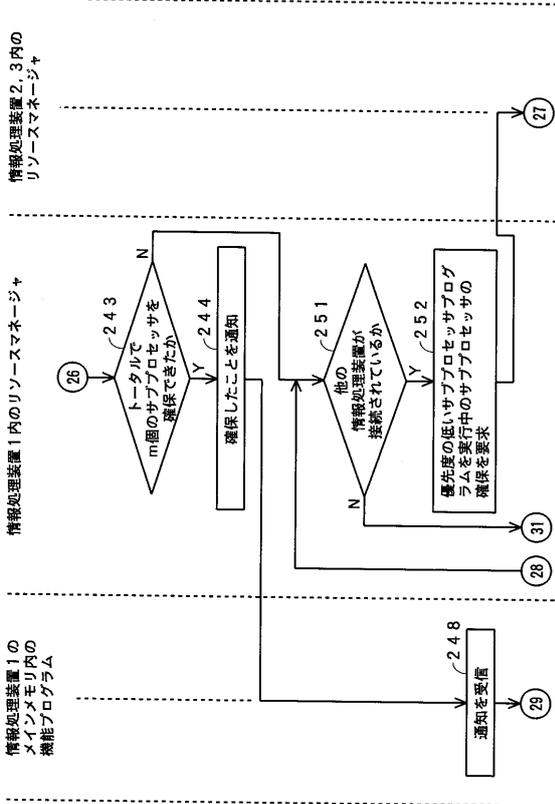
【 図 3 5 】



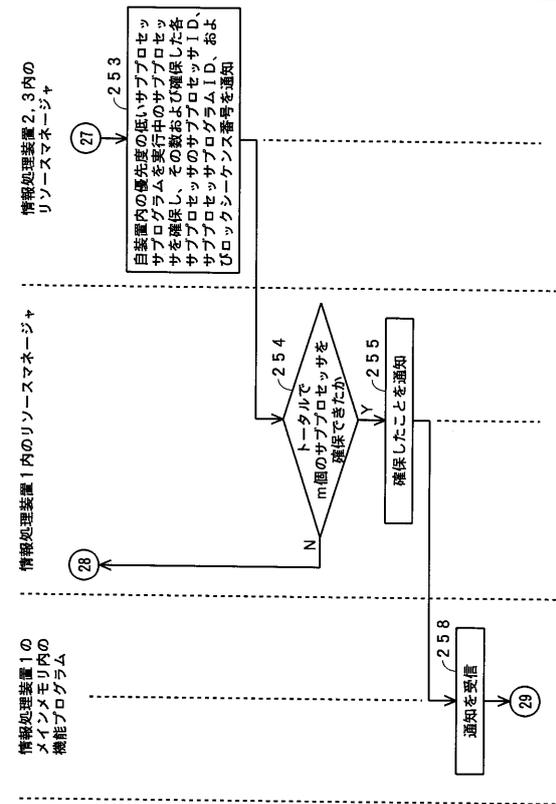
【 図 3 6 】



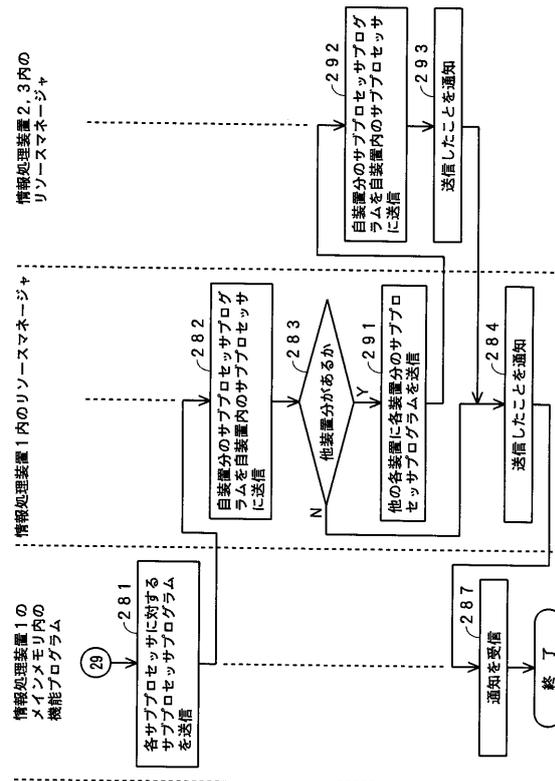
【 図 37 】



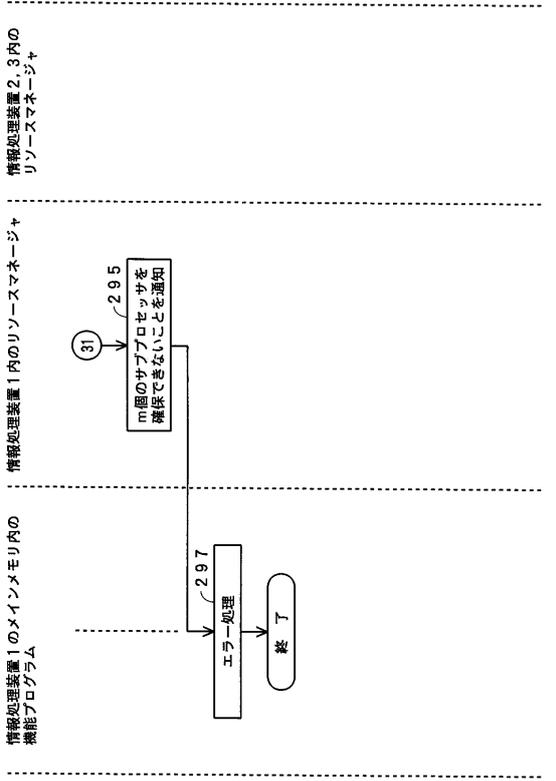
【 図 38 】



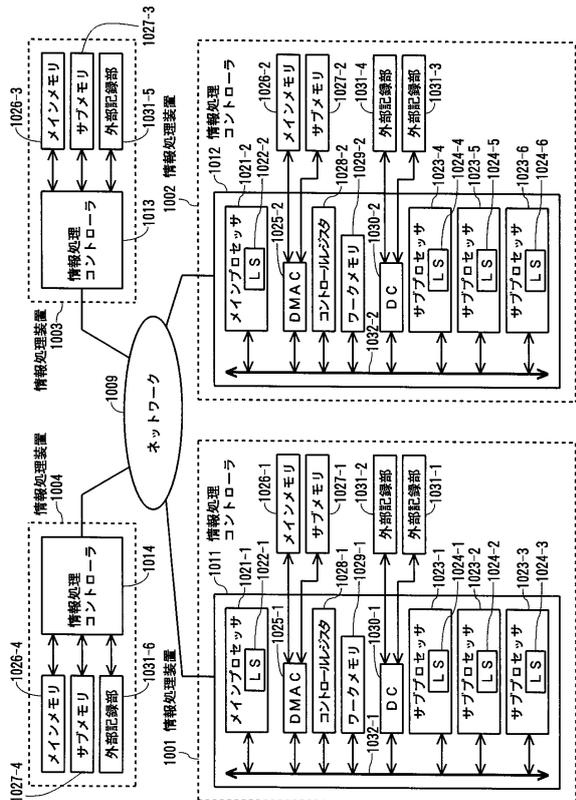
【 図 39 】



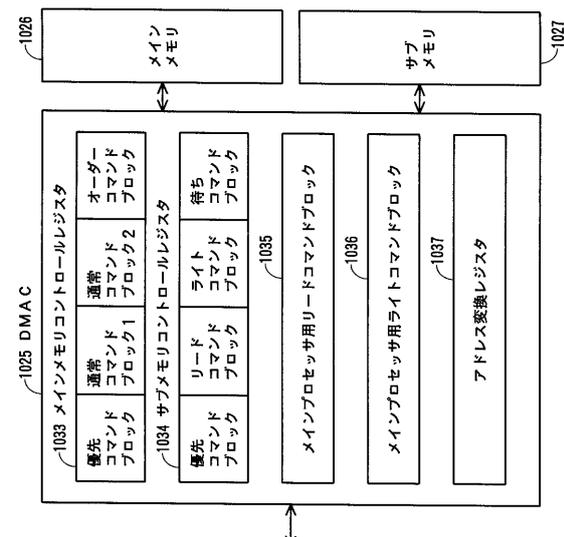
【 図 40 】



【 4 1 】



【 4 3 】

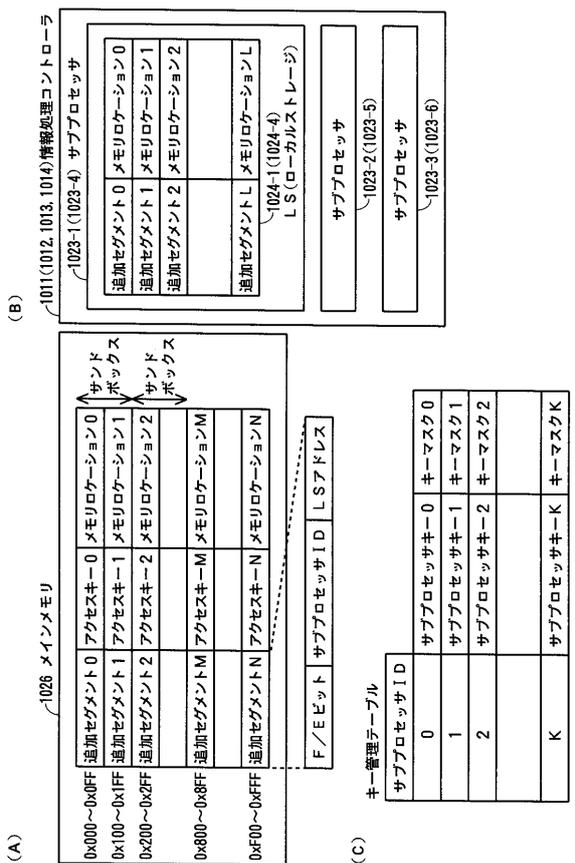


【 4 4 】

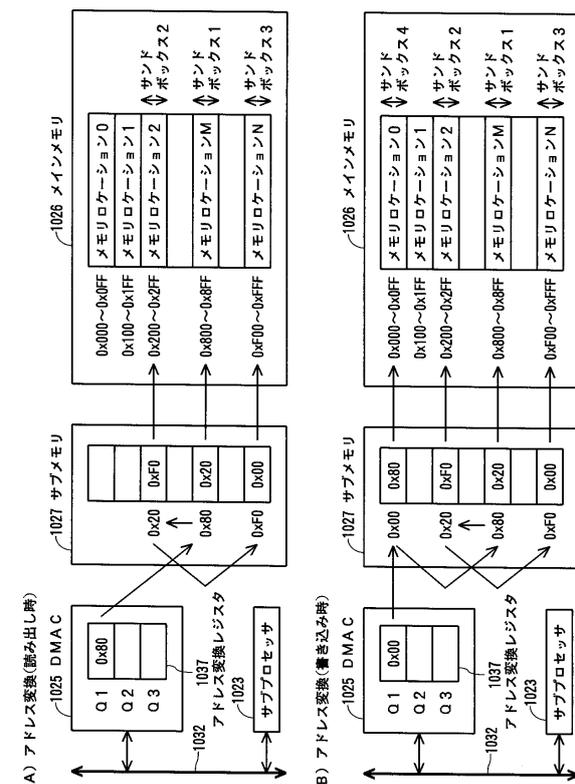
コマンド/レスポンス構造

コマンドタイプ
優先コマンド識別子
通常コマンド識別子
チェーンコマンド識別子
アドレス
サブプロセッサ識別子
アドレスレジスタ識別子
プレセッシングレジスタ識別子
OK/NG
データ

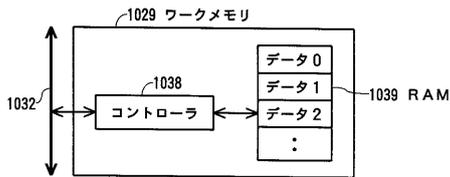
【 4 2 】



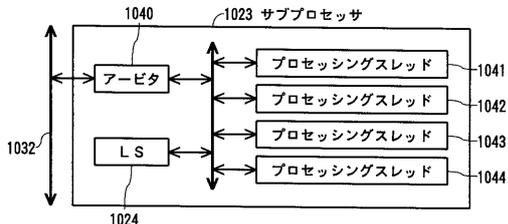
【 4 5 】



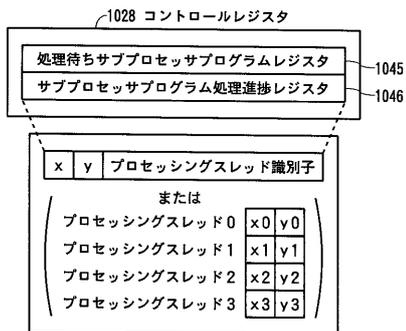
【図46】



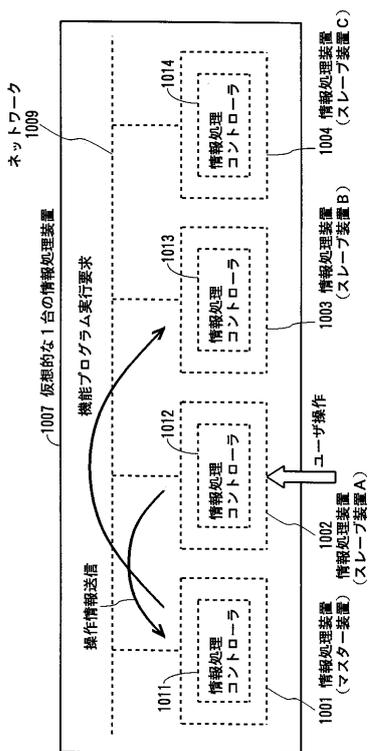
【図47】



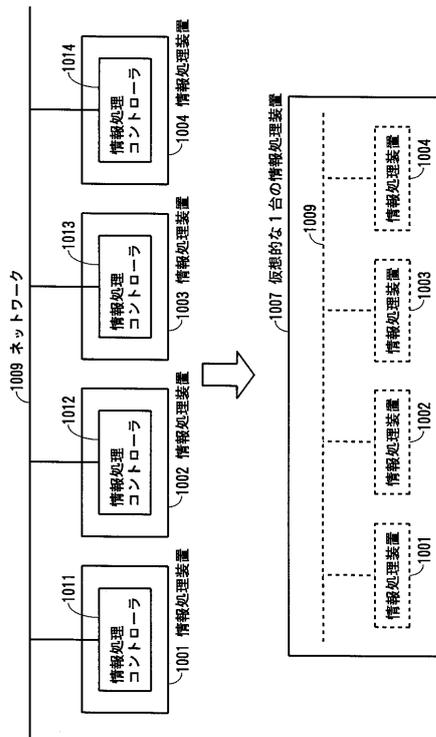
【図48】



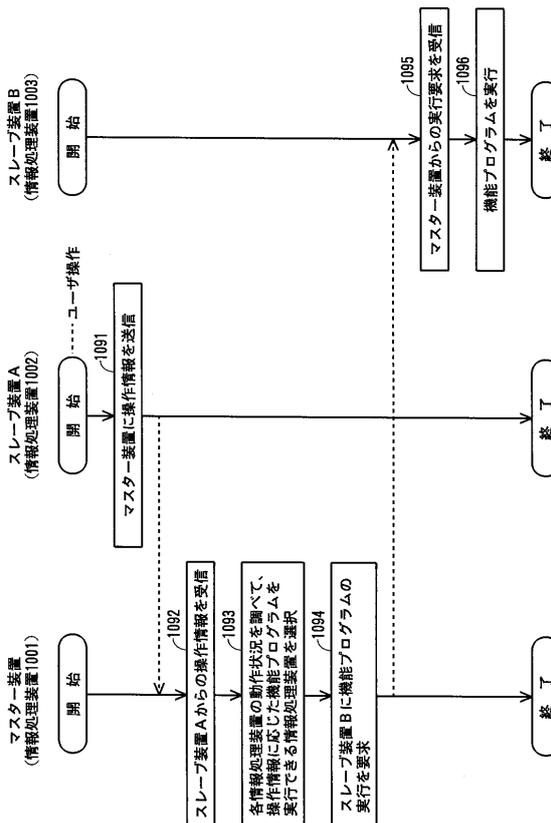
【図50】



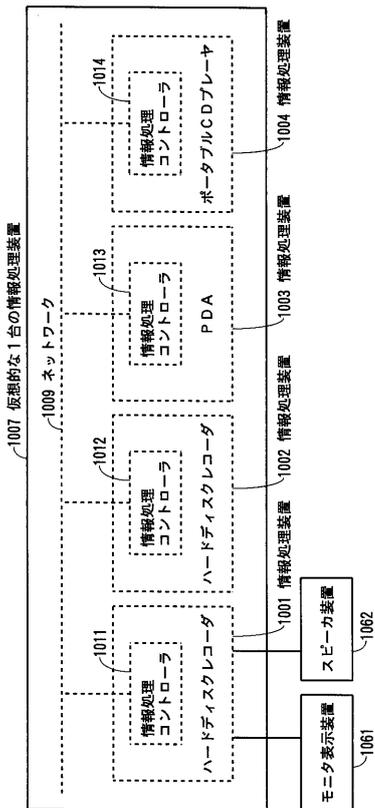
【図49】



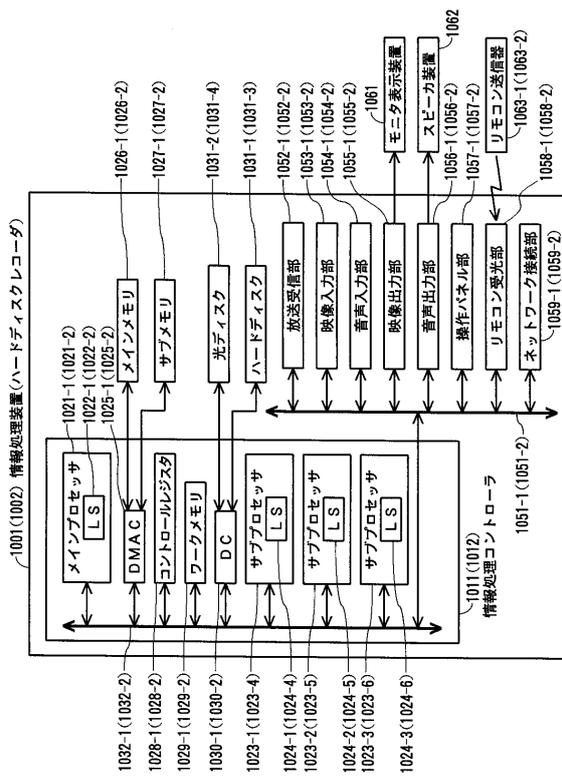
【図51】



【図 5 2】

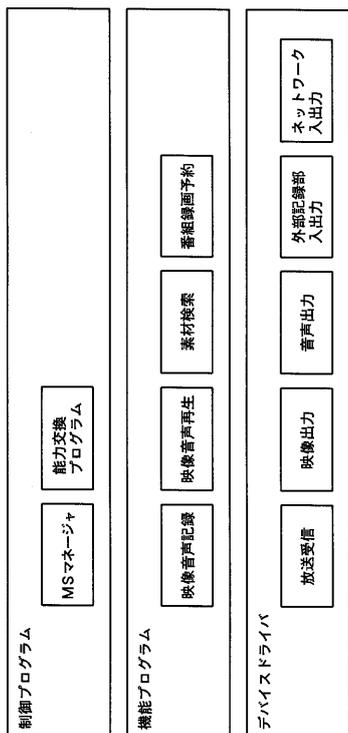


【図 5 3】

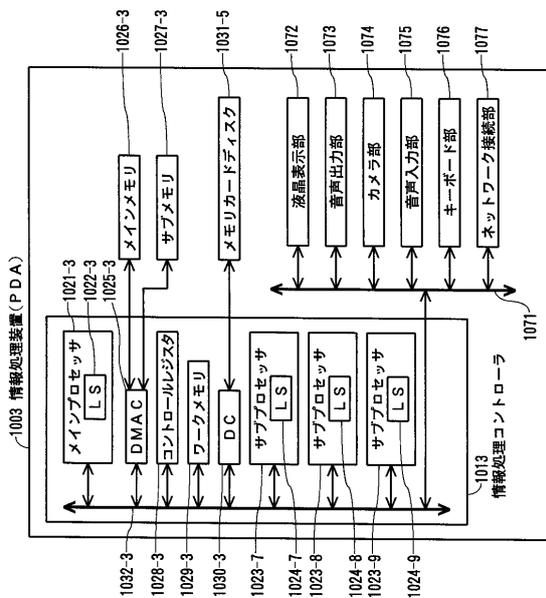


【図 5 4】

情報処理コントローラ1011、1012(情報処理装置1001、1002)のソフトウェア構成

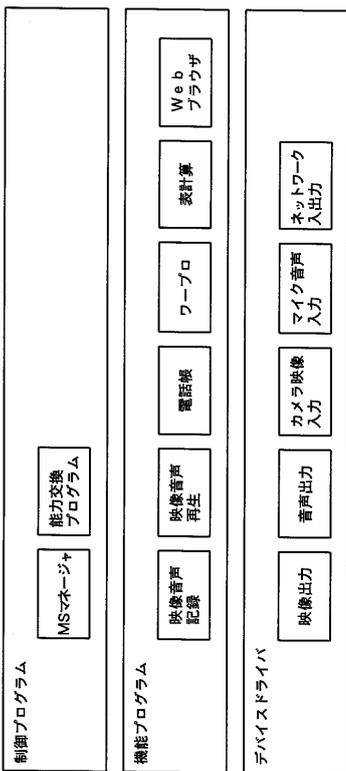


【図 5 5】

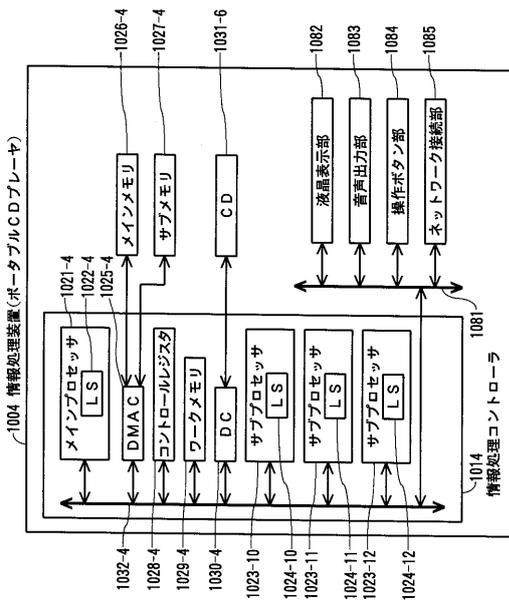


【図56】

情報処理コントローラ1013(情報処理装置1003)のソフトウェア構成

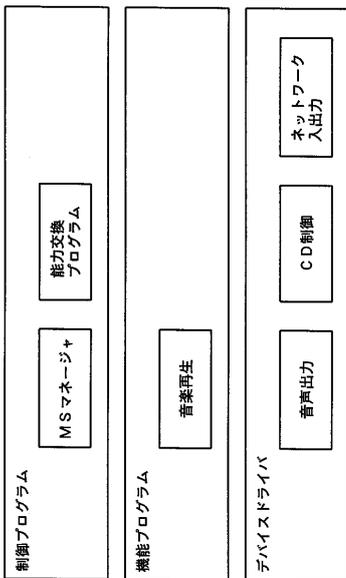


【図57】

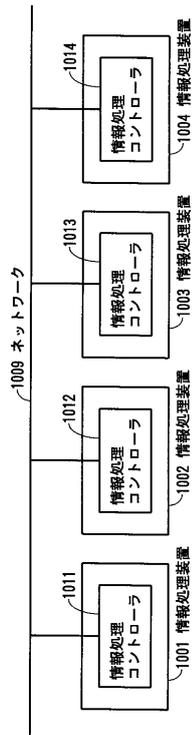


【図58】

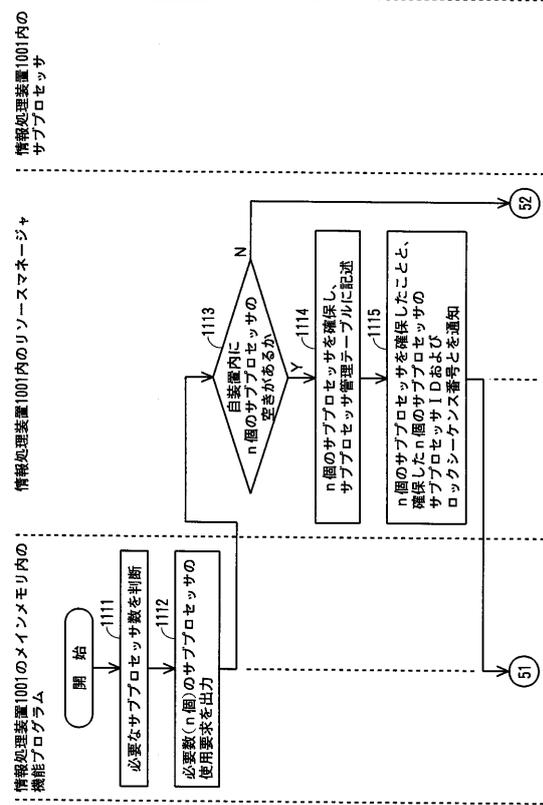
情報処理コントローラ1014(情報処理装置1004)のソフトウェア構成



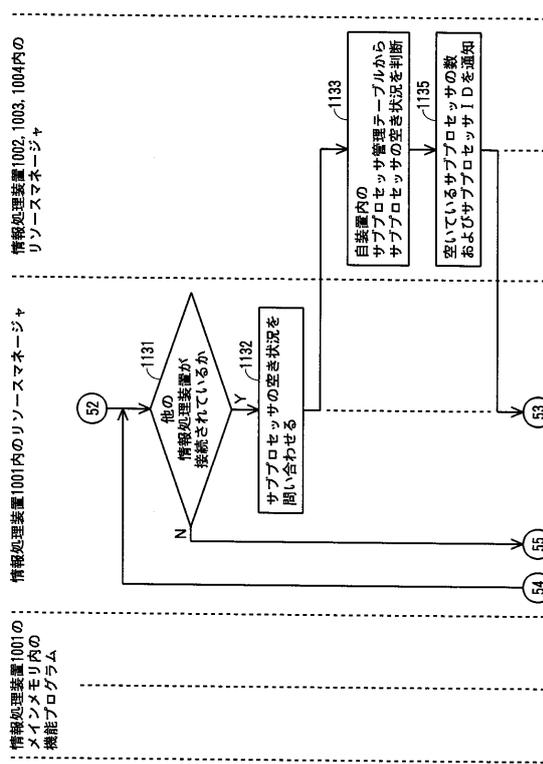
【図59】



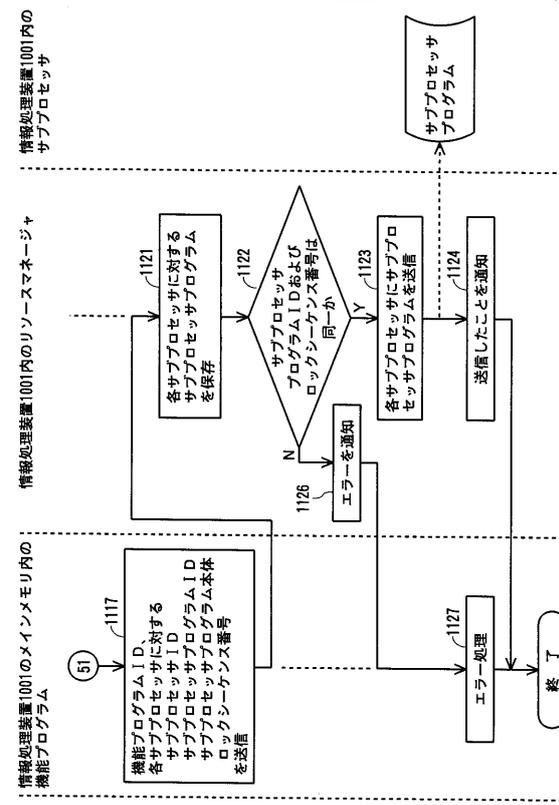
【図 60】



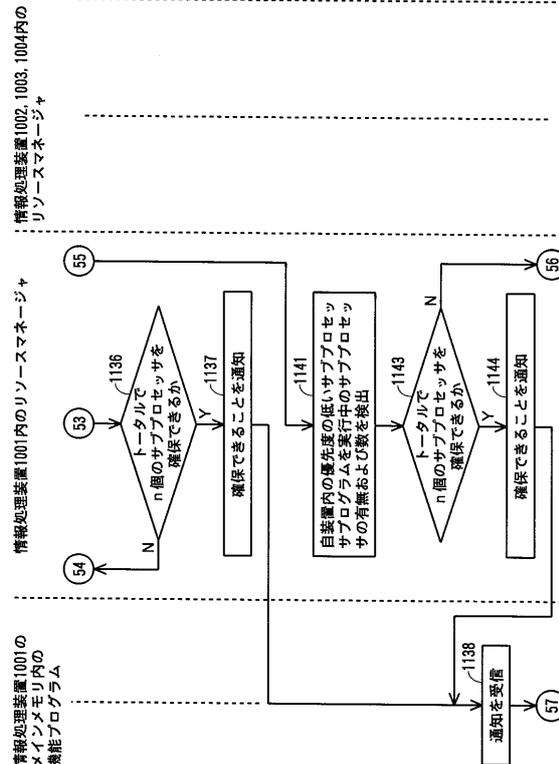
【図 62】



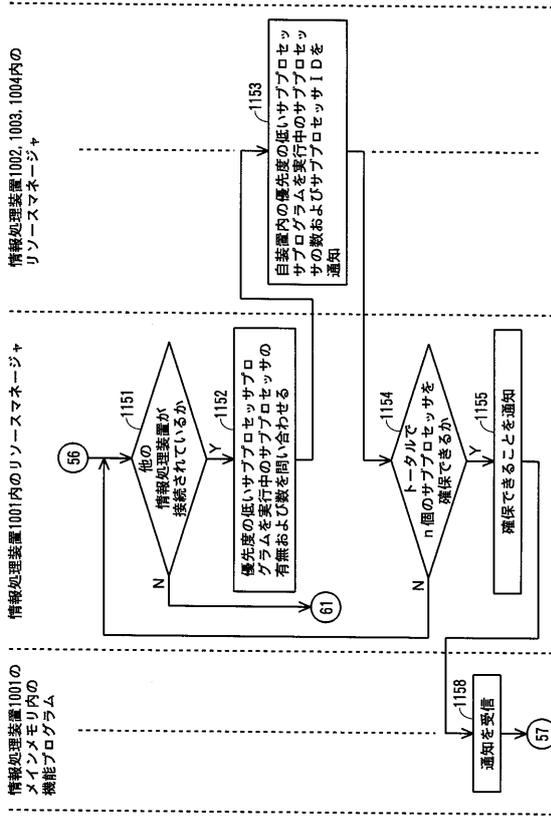
【図 61】



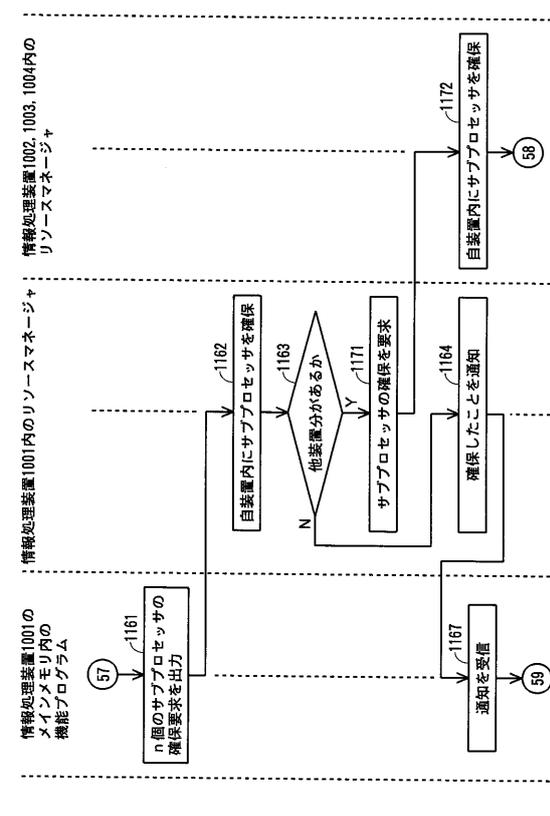
【図 63】



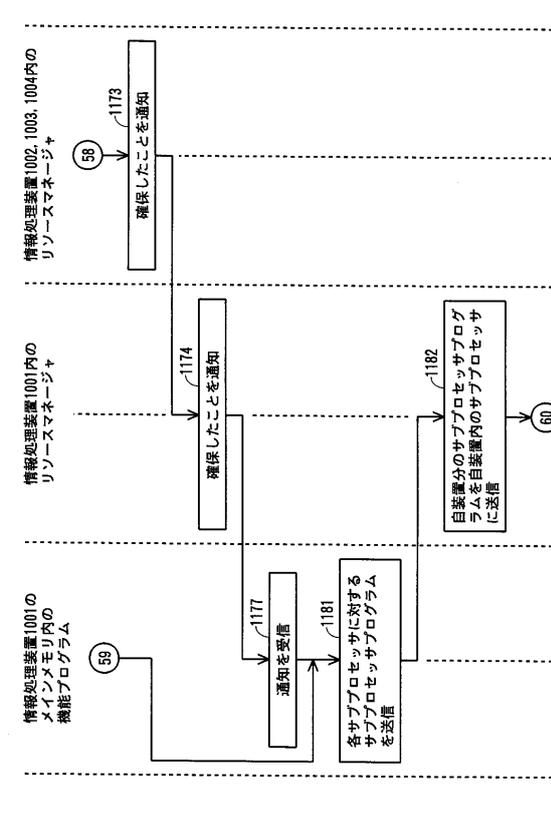
【 図 6 4 】



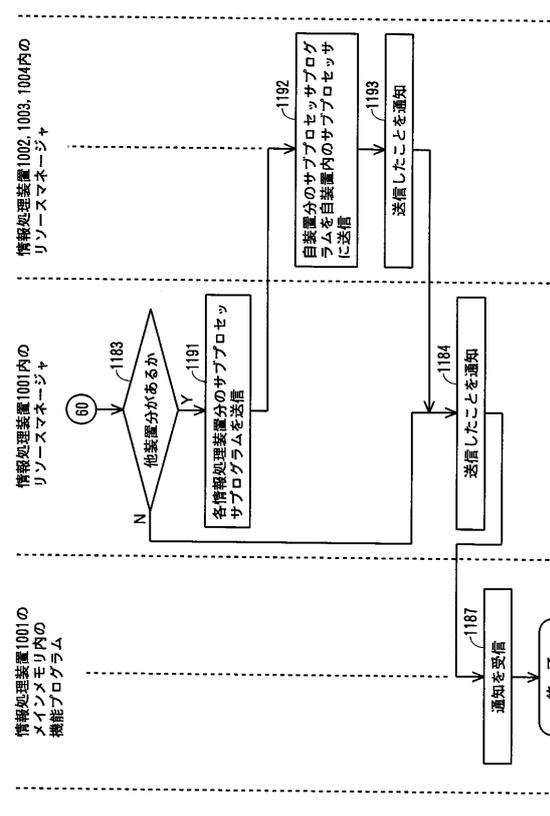
【 図 6 5 】



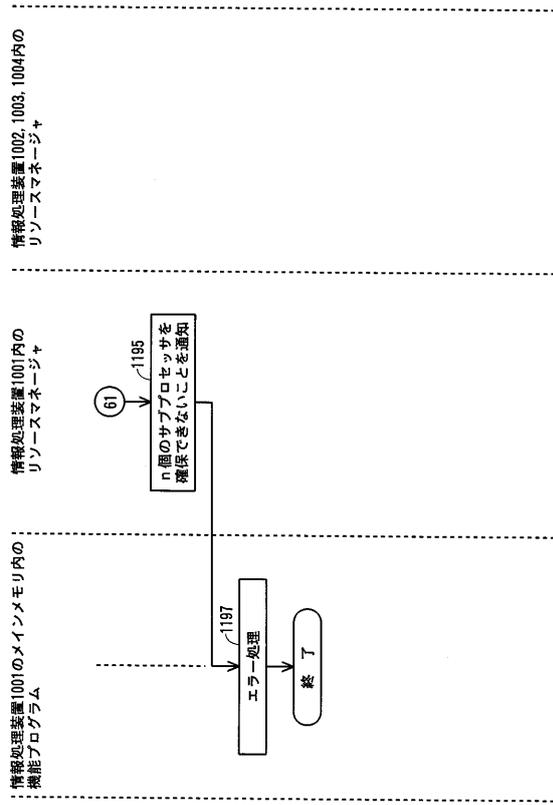
【 図 6 6 】



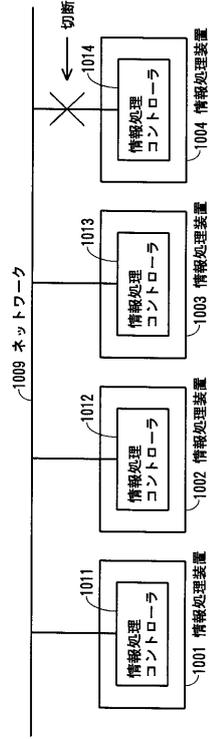
【 図 6 7 】



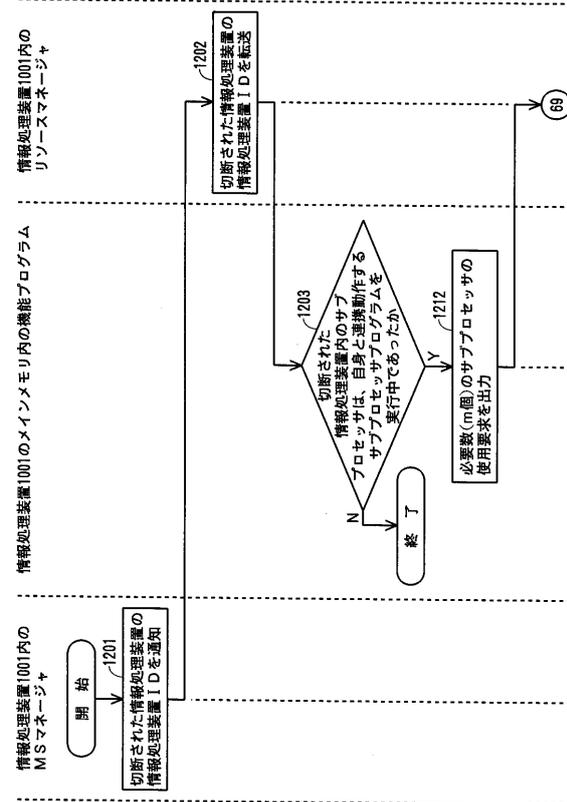
【図68】



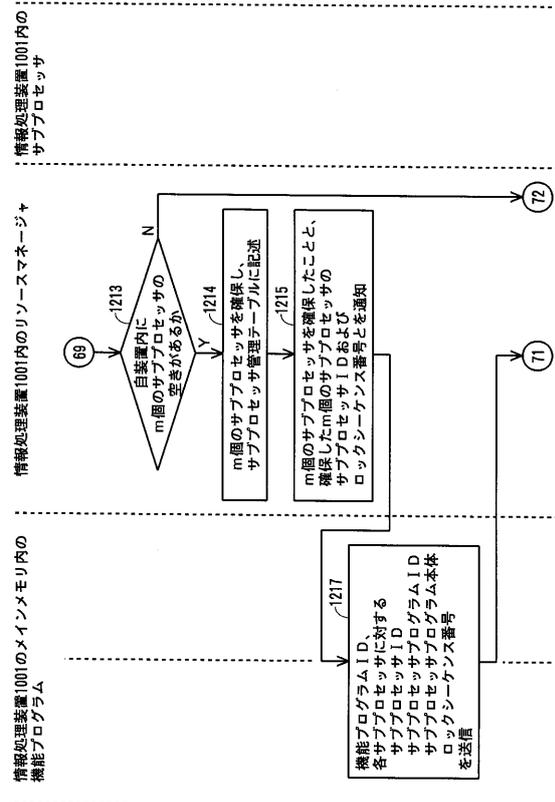
【図69】



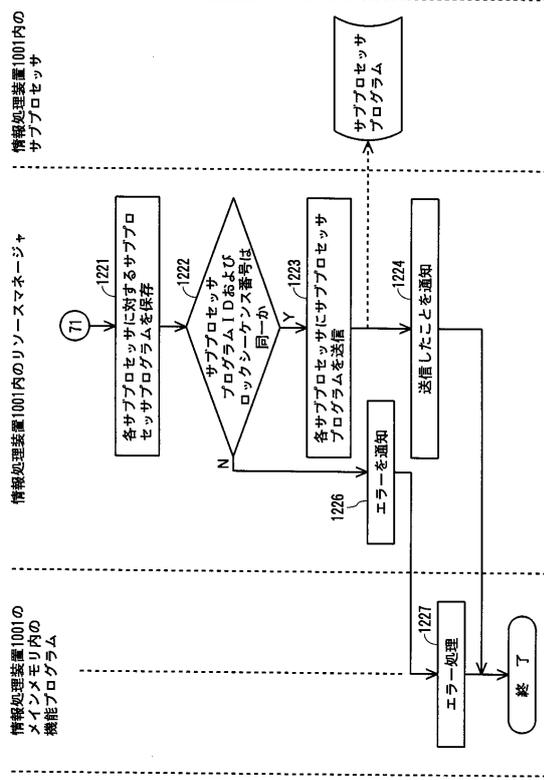
【図70】



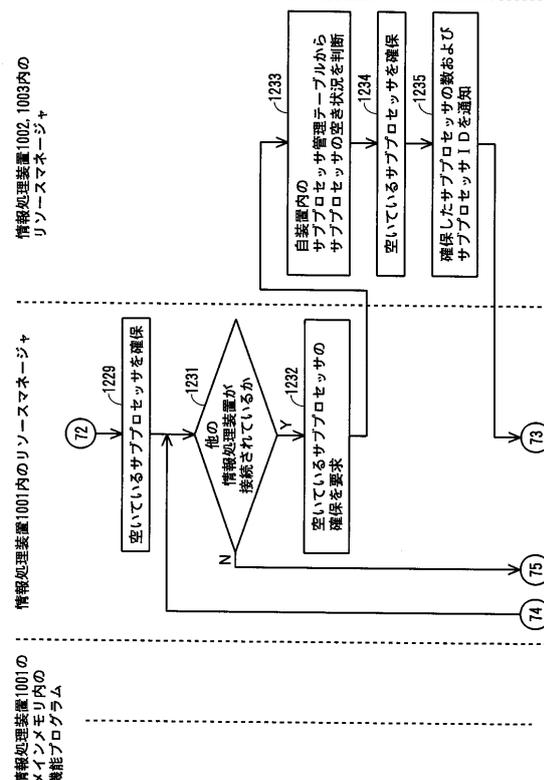
【図71】



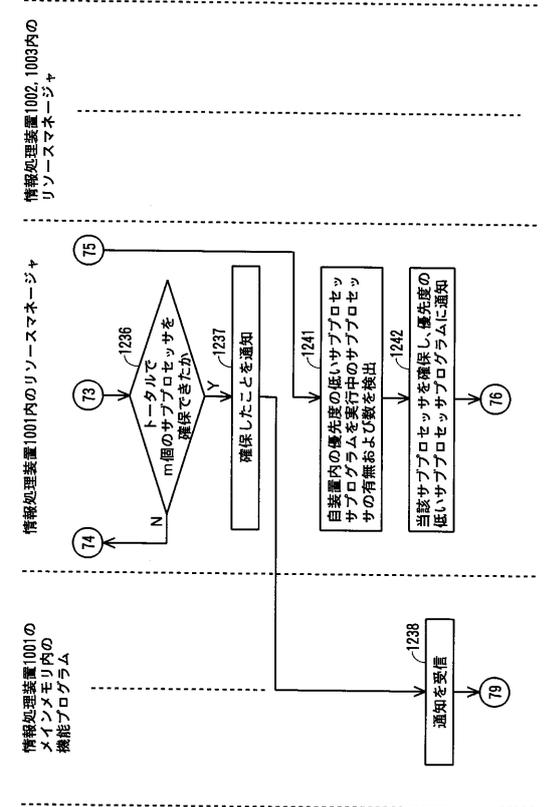
【 図 7 2 】



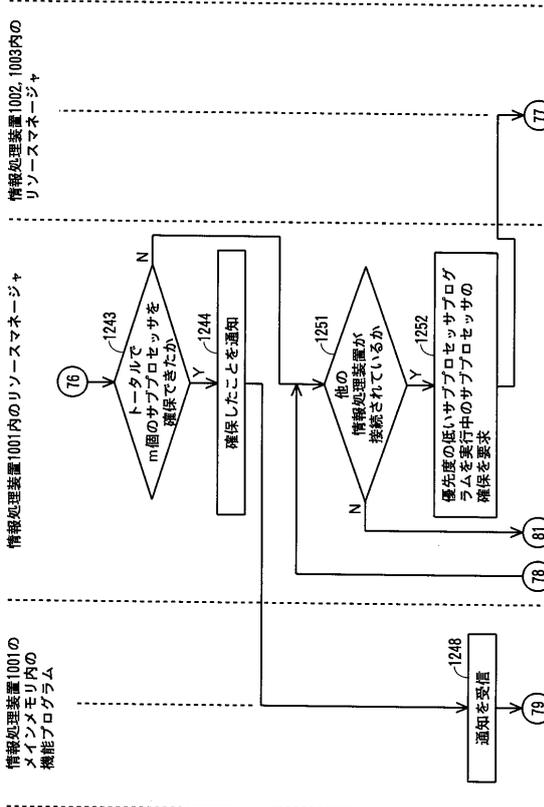
【 図 7 3 】



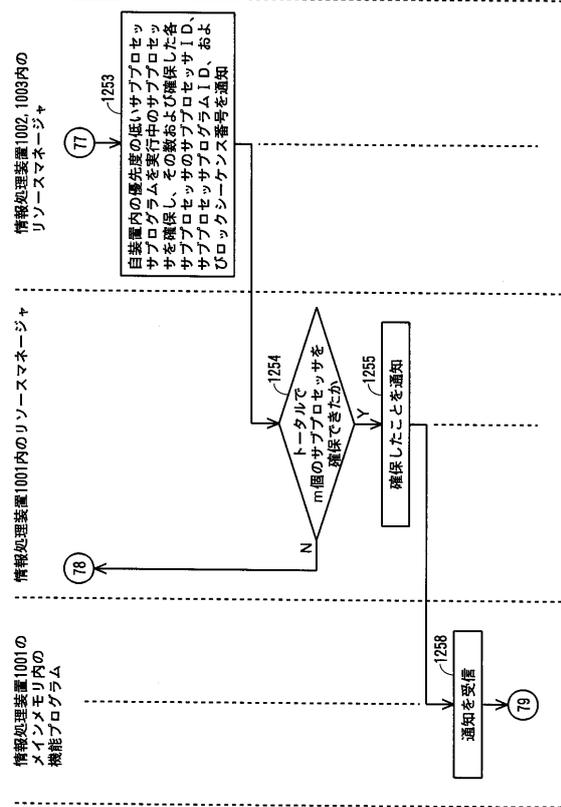
【 図 7 4 】



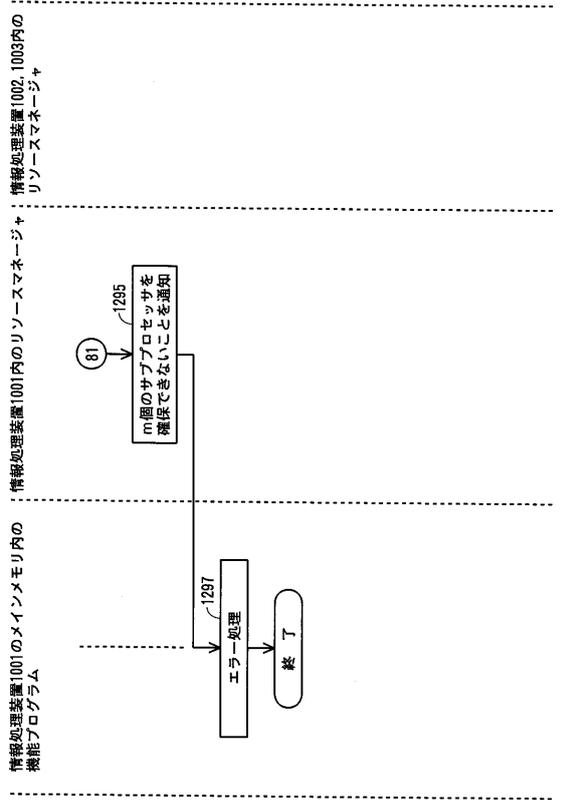
【 図 7 5 】



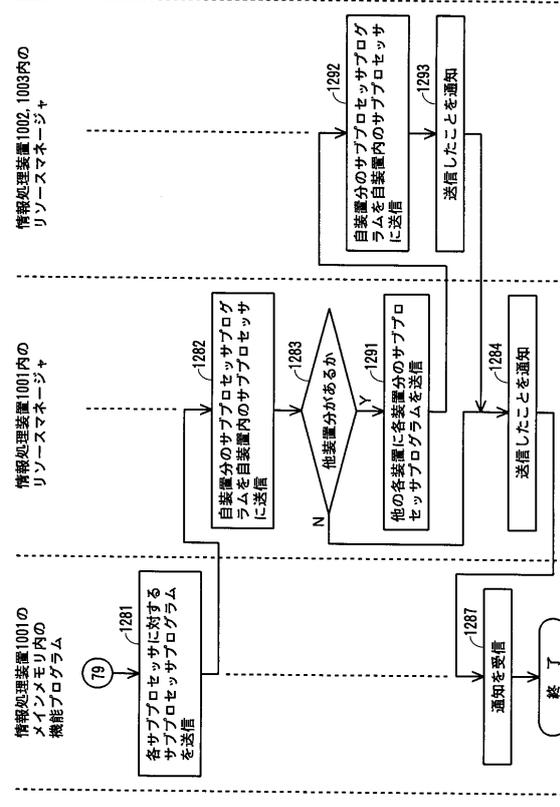
【 図 76 】



【 図 78 】



【 図 77 】



【 図 79 】



---

フロントページの続き

(72)発明者 廣瀬 幸由  
東京都品川区北品川6丁目7番35号 ソニー株式会社内

審査官 漆原 孝治

(56)参考文献 特開平08-077026(JP,A)  
特開2000-194674(JP,A)  
特開平11-120014(JP,A)  
特開2001-195268(JP,A)  
特開平08-221372(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G06F 9/46 - 9/54