

發明專利說明書

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：97115925

※ 申請日期：97.4.29

※IPC 分類：G06F 9/46

一、發明名稱：(中文/英文)

工作處理裝置

二、申請人：(共 1 人)

姓名或名稱：(中文/英文)

內特可力亞斯系統股份有限公司

NETCLEUS SYSTEMS CORPORATION

代表人：(中文/英文)

丸山 修孝 / MARUYAMA, NAOTAKA

住居所或營業所地址：(中文/英文)

日本神奈川縣橫濱市中區相生町 6-104 橫濱相生町大廈 9 樓

9F. Yokohama Aioicho Bldg., 6-104, Aioi-cho, Naka-ku, Yokohama-shi

Kanagawa 231-0012 Japan

國 籍：(中文/英文)

日本 / JAPAN

三、發明人：(共 1 人)

姓 名：(中文/英文)

丸山 修孝 / MARUYAMA, NAOTAKA

國 籍：(中文/英文)

日本 / JAPAN

四、聲明事項：

☐ 主張專利法第二十二條第二項 ☐ 第一款或 ☐ 第二款規定之事實，
其事實發生日期為： 年 月 日。

☒ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

☒ 有主張專利法第二十七條第一項國際優先權：

1. PCT、2007.08.16、PCT/JP2007/000880

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

九、發明說明：

【發明所屬之技術領域】

本發明係關於一種 OS(作業系統：Operating System)的功能，特別是關於中斷(interrupt)處理。

【先前技術】

不限於個人電腦等的汎用機器用 OS，行動電話等的專用機器用 OS 亦要求具有高度的功能。特別是，能以 1 個 CPU(Central Processing Unit)執行複數個工作的 OS(以下，將此種類型之 OS 稱為「多工 OS」)係裝載於大多電子機器。

多工 OS 將 CPU 之處理時間分割成單位時間(時間片段：Time Slice)，將時間片段依序分配至複數個工作。各工作僅能在被 OS 賦予時間片段時使用 CPU。在各時間片段執行一個工作。由於對使用者而言時間片段為非常短的時間，因此複數個工作感覺似乎是同時執行。根據此種處理方法，工作 A 成為輸入待機狀態而暫時不需要 CPU 之處理能力時，將執行權賦予另一個工作 B 以有效活用 CPU 之處理能力。此處，所謂執行權係與 CPU 之使用權同義。

將多工 OS 切換工作之執行權的動作稱為「工作切換」。工作切換，係在時間片段經過時、或在工作執行既定指令時產生。多工 OS 在工作切換之執行時序到達時，將執行中之工作之上下文資訊儲存於 TCB(工作控制區塊：Task Control Block)。上下文資訊，係工作執行時儲存於 CPU

之暫存器的資料與工作執行狀態相關的資料。TCB 係為了保持工作固有的資訊而由記憶體所確保的區域。多工 OS 將執行中之工作之上下文資訊儲存於 TCB 後，選擇下一個分配有執行權的工作，從該工作之 TCB 讀取上下文資訊，載入至 CPU 暫存器。以此方式，各工作以時間片段為單位逐漸執行本身的處理。

雖多工 OS 具有能高效率執行複數個工作的優點，但亦具有新產生上下文資訊之儲存、載入之酬載(Overhead)的缺點。一般而言，即使工作切換伴隨有酬載，多工 OS 仍具有許多優點。

專利文獻 1：日本特開平 11-272480 號公報

專利文獻 2：日本特開 2001-75820 號公報

非專利文獻 1：森久直、坂卷佳壽美、重松宏志「用於籍入式控制系統之即時 OS 的硬體化(Hardware implementation of a real-time operating system for embedded control system)」、東京都立產業技術研究所研究報告、日本、2005 年 8 月 4 日、p.55-58

【發明內容】

近年來，嚴格要求在一定時間內完成處理之即時 OS(以下，稱為「RTOS(Real-Time Operating System)」)以籍入式系統(Embedded System)為中心逐漸普及。在此種時間要求嚴格的 RTOS，工作切換時之酬載有可能對系統整體之效能造成大的影響。

又，RTOS 必須即時對應結束 DMA(直接記憶體存取：Direct Memory Access)傳輸、接收通訊封包、或按下鍵盤等不定期產生的各種外部因素。一般而言，接收表示產生外部因素的中斷要求訊號時，RTOS 啟動特別工作以執行「中斷處理」。在提升 RTOS 之效能之方面，此種外部因素對應時之酬載的抑制特別重要。

本發明係本發明人根據上述課題而完成者，其目的在於提供一種於多工處理用以更有效率地執行控制工作的技術，特別是提供一種用以實現中斷處理之高速化的技術。

本發明之形態為工作處理裝置。

此裝置具備處理暫存器，將資料從記憶體載入至處理暫存器、依處理暫存器之資料執行工作的執行控制電路，保持各工作之狀態資料的狀態暫存器，控制工作之執行狀態的工作切換電路，藉由既定選擇條件選擇工作的工作選擇電路，及處理來自外部之中斷要求訊號的中斷電路。

執行控制電路，當第 1 工作執行系統呼叫指令時，通知工作切換電路。工作選擇電路，從表示可執行待機狀態之 READY 狀態之工作之中選擇成為執行對像的工作。

工作切換電路，根據系統呼叫指令執行時之工作選擇電路之輸出選擇成為下一個執行對像的第 2 工作，使處理暫存器之資料保留於既定記憶區域，且設定變更第 1 工作之狀態資料，對選擇之第 2 工作，將已保留於記憶區域之資料載入至處理暫存器，且將其狀態資料從 READY 狀態變更成 RUN 狀態，以切換成為執行對像的工作。

另一方面，中斷電路，當接收來自外部的中斷要求訊號時，將與該中斷要求訊號相對應之中斷操作指令傳至工作切換電路。工作切換電路，根據該中斷操作指令設定變更工作之狀態資料，以對應中斷要求訊號。

載入至處理暫存器之資料，可為指令(instruction)與運算元(Operand)、不具運算元的指令、或程式計數器與堆疊指標等單純的資料。根據此種處理方法，由於藉由狀態暫存器管理工作的狀態，因此工作切換電路能根據工作選擇電路之輸出執行工作切換。此外，從外部接收中斷要求訊號時，工作切換電路根據與該中斷要求訊號相對應之中斷操作指令設定變更狀態暫存器的資料，因此能以硬體水準高速對應中斷要求訊號。

此外，以上構成元件之任意組合，或以方法、系統、記錄媒體、電腦程式表現本發明者，作為本發明之形態亦有效。

根據本發明，於多工處理，可實現更高效率之工作的執行控制。

【實施方式】

本實施例之工作處理裝置 100，係藉由電子電路實現多工 OS 之工作排程，再者，藉由將中斷處理演算法構裝為硬體邏輯，以提昇多工 OS 之處理效率。

首先，本實施例中，將藉由電子電路實現多工 OS 之工作排程之工作處理裝置 100 說明為「基本例」。之後，

將藉由中斷處理演算法之構裝來實現更高速化的方法說明為「改良例」。以下，「本實施例」之敘述原則上表示「基本例」及「改良例」之兩者。

(基本例)

本實施例所示之工作處理裝置 100，以電子電路實現多工 OS 之工作排程功能。在詳細說明工作處理裝置 100 前，首先以圖 1 說明工作的狀態遷移。此處，雖說明一般之多工 OS 之工作的狀態遷移，但工作處理裝置 100 之工作的狀態遷移亦相同。又，亦概要說明於工作處理裝置 100 所執行之系統呼叫。除了以圖 2 及圖 3 說明一般之多工 OS 的設計思想之外，亦以圖 4 至圖 10 詳細說明本實施例之工作處理裝置 100 的處理方法。再者，亦適當將旗號 (Semaphore)、互斥 (Mutex)、及事件等相關處理與一般技術比較來說明工作處理裝置 100 的特徵。

(工作的狀態遷移)

圖 1 係工作的狀態遷移圖。

在多工處理，各工作具有「狀態(State)」。各工作在後述複數個狀態間遷移，而恆成為任一狀態。狀態遷移之契機，係「系統呼叫之執行」與「中斷要求訊號之檢測」。系統呼叫係各工作執行之指令中的特別指令。中斷要求訊號係按下鍵盤、敲擊滑鼠、接收通訊資料等，從周邊機器接收既定資料時產生的訊號。當然，分配至各工作的時間片段經過後亦產生狀態遷移。

工作大致分為「一般工作」與「特殊工作」的 2 種類。

一般工作係以系統呼叫為契機所執行之通常的工作。特殊工作係以中斷要求訊號之檢測為契機所執行的工作。即所謂中斷處理程式(Interrupt Handler)。首先，說明各工作狀態後，再說明各種系統呼叫指令。

(1)STOP 狀態(停止狀態)

表示工作為停止狀態。一般工作與特殊工作皆有可能成為 STOP 狀態。以下，將處於 STOP 狀態之工作稱為「STOP—工作」。

1—1.一般工作

當工作執行指示另一個工作啟動之系統呼叫(以下，稱為「啟動系統呼叫」)時，處於 STOP 狀態之一般工作遷移至後述 READY 狀態。

1—2.特殊工作

特殊工作通常處於 STOP 狀態。藉由後述工作切換電路 210 檢測出中斷要求訊號時，特殊工作從 STOP 狀態遷移至後述 RUN 狀態。此時，原本處於 RUN 狀態之工作遷移至 READY 狀態。

(2)RUN 狀態(執行狀態)

表示工作正在執行中。亦即，將時間片段分配給工作而取得 CPU 之使用權的狀態。一般工作與特殊工作皆有可能成為 RUN 狀態。以下，將處於 RUN 狀態之工作稱為「RUN—工作」。複數個工作中，能成為 RUN 狀態之工作恆僅有 1 個，2 個工作無法同時成為 RUN 狀態。

2—1.一般工作

處於 RUN 狀態之一般工作，當執行既定系統呼叫時從 RUN 狀態遷移至 READY 狀態或後述 WAIT 狀態。處於 RUN 狀態之一般工作，在時間片段經過後，亦遷移至 READY 狀態。不論是何種情形，處於 READY 狀態之一般工作遷移至 RUN 狀態以替換原本處於 RUN 狀態之一般工作。當檢測出中斷要求訊號時，RUN—工作遷移至 READY 狀態。此時，處於 STOP 狀態之特殊工作遷移至 RUN 狀態。

當 RUN—工作執行使本身結束之系統呼叫(以下，稱為「結束系統呼叫」)時，RUN—工作遷移至 STOP 狀態。

2—2.特殊工作

因中斷要求訊號而從 STOP 狀態遷移至 RUN 狀態的特殊工作，當結束本身處理時回到 STOP 狀態。特殊工作有可能成為的狀態僅為 STOP 狀態與 RUN 狀態。

(3)READY 狀態(可執行狀態)

表示工作為可執行狀態。處於 READY 狀態之工作，被 OS 賦予執行權時皆可遷移至 RUN 狀態。僅一般工作有可能成為 READY 狀態。以下，將處於 READY 狀態之工作稱為「READY—工作」。

處於 RUN 狀態之一般工作藉由系統呼叫之執行而遷移至 RUN 狀態以外的狀態時，或處於 RUN 狀態之特殊工作結束本身的處理而遷移至 STOP 狀態時，READY—工作遷移至 RUN 狀態。一般工作僅從 READY 狀態遷移至 RUN 狀態。當處於 READY 狀態之工作有複數個時，根據上下文資訊之一部分的工作優先順序，其中一個 READY—工

作遷移至 RUN 狀態。當工作優先順序相同之 READY—工作有複數個時，最早移至 READY 狀態之工作遷移至 RUN 狀態。

(4)WAIT 狀態(待機狀態)

表示工作正在等待既定 WAIT 解除條件成立的狀態。當 WAIT 解除條件成立時，處於 WAIT 狀態之工作遷移至 READY 狀態。僅一般工作有可能成為 WAIT 狀態。以下，將處於 WAIT 狀態之工作稱為「WAIT—工作」。關於 WAIT 解除條件將在之後詳細說明。

綜合上述，各工作僅能在處於 RUN 狀態時使用 CPU 執行本身的處理。RTOS 一邊管理複數個工作的狀態，一邊適當切換 RUN—工作。據此，可實現 CPU 隨時執行任一工作的處理形態。

(系統呼叫)

接著，說明系統呼叫。系統呼叫大致分為「啟動系統」、「WAIT 系統」、「SET 系統」的 3 種類。

(1)啟動系統系統呼叫

STOP 狀態與 READY 狀態之間之遷移相關的系統呼叫。

1—1.啟動系統呼叫

RUN—工作之工作 A 使另一個一般工作 B 啟動的系統呼叫。此時，處於 STOP 狀態之一般工作 B 遷移至 READY 狀態。

1—2.結束系統呼叫

執行此系統呼叫之工作，結束本身的處理，從 RUN 狀態遷移至 STOP 狀態。結束系統呼叫亦可為某个工作使另一個工作結束的指令。

(2)WAIT 系統系統呼叫

RUN 狀態與 WAIT 狀態之間之遷移相關的系統呼叫。

2-1.待幾旗號系統呼叫

要求獲得旗號(後述)的系統呼叫。

2-2.待幾互斥系統呼叫

要求獲得互斥(後述)的系統呼叫。

2-3.待幾事件系統呼叫

等待事件(後述)成立的系統呼叫。除了事件 ID 之外，待幾旗標類型(後述)與旗標條件(後述)亦作為變數執行。

不論是何種情形，均藉由 WAIT 系統系統呼叫設定各種 WAIT 解除條件。當 WAIT 系統系統呼叫執行時，在 WAIT 解除條件已成立之狀況下，執行系統呼叫之 RUN-工作遷移至 READY 狀態。另一方面，當 WAIT 解除條件不成立時，RUN-工作遷移至等待 WAIT 解除條件成立的 WAIT 狀態。

(3)SET 系統系統呼叫

WAIT 狀態與 READY 狀態之間之遷移相關的系統呼叫。SET 系統系統呼叫之執行為 WAIT 解除條件的成立契機。

3-1.解除旗號系統呼叫

解除旗號的系統呼叫。

3-2.解除互斥系統呼叫

解除互斥的系統呼叫。

3-3.設定事件系統呼叫

設定事件之現行旗標類型(後述)的系統呼叫。

3-4.清除旗標系統呼叫

用以將現行旗標類型清除為零的系統呼叫。

於本實施例，以上述合計 9 種類之系統呼叫為對象進行說明，但除此以外當然亦可構裝各種系統呼叫。

(一般 RTOS 的設計思想)

圖 2 係一般 RTOS 的概念圖。

此 RTOS 係多工 OS。一般 RTOS 係實現為軟體。以將 RUN-工作從工作 A 切換至工作 B 之情形為例進行說明。由於工作 A 佔有 CPU，因此 RTOS 對 CPU 發出中斷，從工作 A 拿回 CPU 之使用權。此外，將工作 A 之上下文資訊儲存於 TCB。RTOS 選擇工作 B 作為下一個 RUN-工作，將上下文資訊從工作 B 之 TCB 載入至 CPU 的暫存器。當載入結束時，RTOS 將 CPU 之使用權交給工作 B。以此方式，藉由 RTOS 暫時取得 CPU 之使用權，執行工作 A 至工作 B 之工作切換。關於特殊工作的執行亦相同。此時，將 RUN-工作之上下文資訊儲存於 TCB 後，將 CPU 之使用權交給特殊工作以實現工作切換。

由於 RTOS 係實現為軟體，因此為了執行本身之處理必須要 CPU 之使用權。亦即，RTOS 與工作在 CPU 之使用上為競爭關係。以下，將以此方式藉由軟體實現之 RTOS

稱為「軟體 RTOS」。

圖 3 係軟體 RTOS 所執行之一般 CPU 的電路圖。

CPU84 包含統括地控制記憶體存取與指令之執行等的執行控制電路 90、儲存工作之上下文資訊等各種資料的處理暫存器 92、及執行運算的運算電路 94。處理暫存器 92 係複數種類之暫存器的集合，大致分為特殊暫存器 88 與汎用暫存器 86。特殊暫存器 88 係保持程式計數器、堆疊指標、旗標等的暫存器。汎用暫存器 86 係保持作業用之資料的暫存器，包含 R0~R15 的合計 16 個暫存器。雖特殊暫存器 88 有使用者用與系統用的二種，但汎用暫存器 86 僅有一種。以下，將儲存於處理暫存器 92 的資料稱為「處理資料」。

執行控制電路 90，藉由對輸出選擇器 98 的控制訊號 (CTRL)，將處理暫存器 92 中之所欲暫存器的處理資料輸出至運算電路 94。運算電路 94 依處理資料、亦即指令與變數執行運算。運算結果係輸出至輸入選擇器 96。執行控制電路 90，藉由對輸入選擇器 96 的控制訊號 (CTRL)，將運算結果輸入至處理暫存器 92 中之所欲暫存器。

又，執行控制電路 90，透過 CPU 資料匯流排從記憶體讀取資料，透過輸入選擇器 96 適當載入至處理暫存器 92。執行控制電路 90，同樣地透過 CPU 資料匯流排將處理資料適當紀錄於記憶體。執行控制電路 90 一邊更新特殊暫存器 88 的程式計數器一邊執行工作。

當產生工作切換時，執行控制電路 90 將處理資料儲存

於記憶體上之區域的 TCB。假設工作 A 執行系統呼叫，產生從工作 A 至工作 B 的工作切換。由於 RTOS 以系統呼叫執行為契機而取得 CPU 的使用權，因此 CPU84 暫時依 RTOS 用的程式執行動作。其處理過程如下述。

(工作 A 之上下文資訊的儲存)

1.執行控制電路 90，將特殊暫存器 88 從使用者用切換至系統用。RTOS 處理用之處理資料係載入至系統用的特殊暫存器 88。

2.執行控制電路 90，將汎用暫存器 86 之資料儲存於未圖示的堆疊。

3.執行控制電路 90，將 RTOS 用之處理資料從未圖示之記憶媒體、例如另一個暫存器載入至汎用暫存器 86。在此階段，處理暫存器 92 之處理資料完全替換成 RTOS 用之處理資料。

4. RTOS 自記憶體檢測工作 A 之 TCB，將儲存於堆疊的處理資料寫入 TCB。又，使用者用特殊暫存器 88 之處理資料亦作為上下文資訊的一部分而寫入 TCB。以此方式，將工作 A 之處理資料儲存於 TCB。RTOS 將工作 A 從「RUN」狀態遷移至「READY(或 WAIT)」的動作儲存於工作 A 之 TCB。

(工作 B 之上下文資訊的載入)

1. RTOS 自記憶體檢測工作 B 之 TCB，將 TCB 之上下文資訊寫入堆疊與使用者用特殊暫存器 88。RTOS 將工作 B 從「READY」狀態遷移至「RUN」的動作記錄於工作 B

之 TCB。

2. RTOS 將 RTOS 處理用之資料從汎用暫存器 86 儲存至未圖示之記錄媒體。

3. 執行控制電路 90，將堆疊之上下文資訊載入至汎用暫存器 86。執行控制電路 90，將特殊暫存器 88 從系統用切換至使用者用。以此方式，將工作 B 之處理資料載入至處理暫存器 92。

經由以上處理過程可實現工作切換。一般而言，由於汎用暫存器 86 僅有一種，因此為了切換工作用之處理資料與 RTOS 用之處理資料而使用堆疊。若汎用暫存器 86 亦具有二種，則不需透過堆疊儲存、載入，因此可執行更快速的工作切換。

於本實施例，藉由進一步依各工作設置保留暫存器 110，可實現更快速的工作切換。以圖 5 詳細說明使用保留暫存器 110 的工作切換。可知在以圖 3 說明之 CPU84 與一般軟體 RTOS 之情形，當工作切換時，頻繁產生對 TCB 的存取。在上述假設例，雖以從工作 A 工作切換至工作 B 為前題進行說明，但實際上，RTOS 為選擇下一個待執行之工作 B，亦須執行多數指令。此時，RTOS 頻繁地存取記憶體。本實施例之工作處理裝置 100，為了使後述工作控制電路 200 專用於工作選擇處理，實現更快速的工作切換。

(工作處理裝置 100 之 RTOS 的硬體化)

圖 4 係本實施例之 RTOS 的概念圖。

與一般軟體 RTOS 不同，本實施例之 RTOS 主要實現為與 CPU 不同之另一個硬體。以下，將藉由硬體實現之 RTOS 稱為「硬體 RTOS」。由於本實施例之 RTOS 主要為與 CPU 不同之另一個硬體，因此執行本身的處理時實質上幾乎不需要 CPU 的使用權。亦即，RTOS 與工作在 CPU 之使用上幾乎不會成為競爭關係。圖 2 所示之一般軟體 RTOS 之情形，CPU 為工作執行用電路亦為 RTOS 執行用電路。相對於此，本實施例之硬體 RTOS 之情形，CPU 明確地為工作執行用電路，工作排程功能能以後述儲存電路 120 與工作控制電路 200 為中心實現。

圖 5 係本實施例之工作處理裝置 100 的電路圖。

工作處理裝置 100 除了 CPU150 之外亦包含儲存電路 120 與工作控制電路 200。CPU150 係工作的執行主體，儲存電路 120 與工作控制電路 200 係具有圖 4 所示之 RTOS 之功能的電路。工作排程處理，係藉由工作控制電路 200 主導。

CPU150 包含執行控制電路 152、處理暫存器 154、及運算電路 160。CPU150 亦可為以圖 3 說明之一般 CPU。然而，本實施例之 CPU150 與圖 3 所示之 CPU84，在訊號線之連接方法等有一些變更。以下一個圖 6 詳細說明具體電路構成。

工作控制電路 200 包含工作切換電路 210、旗號表 212、事件表 214、工作選擇電路 230、及狀態記憶部 220。以圖 13 之後詳細說明旗號表 212 與事件表 214。狀態記憶部 220

係與各工作相對應的單元。以下，將與工作 A 對應之狀態記憶部 220 表記為「狀態記憶部 220_A」。各狀態記憶部 220 保持工作的狀態資料。狀態資料，係上下文資訊中，特別是表示工作優先順序與狀態等工作屬性的資訊。以圖 10 詳細說明具體資料的內容。所有工作之所有狀態資料係從各狀態記憶部 220 隨時輸出至工作選擇電路 230。工作選擇電路 230，係根據各工作的狀態資料執行 RUN—工作之選擇等各種工作選擇的電路。亦以圖 10 之後詳細說明工作選擇電路 230。工作切換電路 210，當檢測出從執行控制電路 152 接收之系統呼叫訊號(SC)與來自外部裝置之中斷要求訊號(INTR)時，執行工作切換。

執行控制電路 152，當系統呼叫執行時，將系統呼叫訊號(SC)傳至工作切換電路 210。又，當工作切換電路 210 檢測出中斷要求訊號(INTR)時，工作切換電路 210 對執行控制電路 152 傳送停止要求訊號(HR)。執行控制電路 152，當 CPU150 之動作停止時，將停止結束訊號(HC)傳至工作切換電路 210。藉由此等 3 種類的訊號，CPU150 與工作控制電路 200 產生連動動作。

儲存電路 120 包含載入選擇電路 112 及複數個保留暫存器 110。保留暫存器 110 亦為與各工作相對應的單元，為用以儲存處理暫存器 154 之處理資料的暫存器。是以，保留暫存器 110 具有與處理暫存器 154 相同、或處理暫存器 154 以上的資料容量。以下，將與工作 A 對應之保留暫存器 110 表記為「保留暫存器 110_A」。載入選擇電路 112，

當接受工作切換電路 210 的指示時，將任一個保留暫存器 110 的資料(以下，將保留暫存器 110 保持的資料稱為「儲存資料」)載入至處理暫存器 154。

各保留暫存器 110 將個別之儲存資料隨時輸出至載入選擇電路 112。當工作切換電路 210 將指定工作 ID 之工作選擇訊號(TS)輸入至載入選擇電路 112 時，載入選擇電路 112 將與指定之工作對應之保留暫存器 110 之儲存資料輸出至處理暫存器 154。再者，當工作切換電路 210 對處理暫存器 154 輸入寫入訊號(WT)時，該儲存資料實際上載入至處理暫存器 154。

另一方面，處理暫存器 154 之所有處理資料亦隨時輸出至所有保留暫存器 110。當工作切換電路 210 對所欲保留暫存器 110 傳送寫入訊號(WT)時，處理資料儲存於該保留暫存器 110。此處，連接處理暫存器 154 與各保留暫存器 110 之匯流排一次可傳送的位元數，係設定成可平行傳送處理資料。因此，工作切換電路 210 僅將寫入訊號傳至保留暫存器 110 一次，即可將處理資料一次寫入保留暫存器 110。又，連接保留暫存器 110 與載入選擇電路 112、及連接載入選擇電路 112 與 CPU150 之匯流排的位元數亦同樣地設定。

以下，分別對系統呼叫與中斷要求訊號說明工作切換的執行方法。

(1)系統呼叫執行

當 CPU150 之執行控制電路 152 執行系統呼叫時，執

行控制電路 152 使 CPU150 之時脈(以下，稱為「CPU 時脈 (CLK)」)停止。之後以圖 7 等詳細說明具體停止方法。執行控制電路 152，將表示系統呼叫之執行的系統呼叫訊號 (SC)傳至工作控制電路 200 的工作切換電路 210。又，當 CLK 之停止結束時，執行控制電路 152 將停止結束訊號(HC)傳至工作切換電路 210。

在 CPU150 與工作切換電路 210 之間連接有用以傳送系統呼叫訊號的 9 條訊號線。9 條訊號線與上述 9 種類的系統呼叫對應。執行控制電路 152，依執行之系統呼叫的種類，在任一條系統呼叫訊號線傳送數位脈衝。工作切換電路 210，能依從 9 條系統呼叫訊號線中哪一條訊號線檢測出數位脈衝，立即檢測出執行之系統呼叫的種類。工作切換電路 210，依系統呼叫的種類，從工作選擇電路 230 之輸出資料選擇必要的資料，執行系統呼叫所指示的處理。此處理係以 HC 被傳送為條件執行。以圖 10 詳細說明工作切換電路 210 與工作選擇電路 230 的關係。此外，系統呼叫之參數與回傳值，係寫入處理暫存器 154 中之既定汎用暫存器 158。工作切換電路 210 能對汎用暫存器 158 執行參數讀取與回傳值寫入。此處，假設 RUN—工作之工作 A 執行待機旗號系統呼叫。是以，首先，必需儲存工作 A 的處理資料。

(工作 A 之上下文資訊的儲存)

執行控制電路 152，將表示待機旗號系統呼叫之 SC 訊號輸入至工作切換電路 210。執行控制電路 152 使 CLK 停

止，停止結束時傳送 HC。工作切換電路 210，除了對工作選擇電路 230 所內建之各種選擇電路中之後述旗號選擇電路 234 輸出待機對象之旗號的旗號 ID 之外，亦選擇下一個待執行的工作 B。工作切換電路 210，對狀態記憶部 220_A 寫入既定資料。例如，將工作 A 之狀態從「RUN」變更設定成「READY」或「WAIT」。更具體而言，工作切換電路 210，除了將「WAIT」輸出至所有狀態記憶部 220 以作為表示狀態資料中之工作狀態的資料外，亦將寫入訊號 (WT_A) 僅輸入至狀態記憶部 220_A。以此方式，變更工作 A 之狀態設定。

接著，工作切換電路 210，對保留暫存器 110_A 輸出寫入訊號 (WT)。由於處理暫存器 154 之處理資料隨時輸出至各保留暫存器 110，因此藉由此寫入訊號 (WT) 儲存於工作 A 的保留暫存器 110_A。

(工作 B 之上下文資訊的載入)

當工作切換電路 210 結束工作 A 之狀態資料的變更、處理資料的儲存時，將指定工作 B 之工作選擇訊號 (TS_B) 輸出至載入選擇電路 112。據此，保留暫存器 110_B 之儲存資料係輸出至處理暫存器 154。當工作切換電路 210 將寫入訊號 (WT) 輸出至處理暫存器 154 時，工作 B 之儲存資料係載入至處理暫存器 154。又，工作切換電路 210，對工作 B 之狀態記憶部 220 寫入既定資料。例如，將工作 B 之狀態從「READY」變更設定成「RUN」。當結束以上處理時，執行控制電路 152 再開始 CPU 時脈。CPU150 係藉

由再開始之 CPU 時脈開始工作 B 的執行。之後以圖 8(b)說明處理方法之更進一步的細節。

(2) 中斷要求訊號的產生

工作切換電路 210 檢測來自周邊機器的中斷要求訊號 (INTR)。更具體而言，中斷要求訊號 (INTR)，係從未圖示之中斷控制器傳至工作切換電路 210。表示中斷要求訊號 (INTR) 之等級的參數，係紀錄於中斷控制器內建的暫存器。工作切換電路 210 將停止要求訊號 (HR) 傳至執行控制電路 152，執行控制電路 152 使 CPU 時脈停止。與系統呼叫執行時相同，工作切換電路 210 將 RUN—工作之處理資料儲存於保留暫存器 110。接著，工作切換電路 210 啟動特殊工作。不論中斷要求訊號之參數如何，啟動之特殊工作為 1 種類。特殊工作從中斷控制器的內建暫存器讀取 INTR 的參數，依參數執行處理。特殊工作執行之處理有可能為設定事件系統呼叫或設定旗號系統呼叫的執行，亦有可能為一般工作的啟動。依據參數，特殊工作有可能未執行特殊的處理即結束。依據 INTR 的參數，執行何種處理係取決於特殊工作的構裝。當特殊工作之執行結束時，從 READY—工作之中選擇下一個 RUN—工作。

工作切換電路 210 將與特殊工作對應之保留暫存器 110 的處理資料載入至 CPU150。此種從一般工作切換至特殊工作所需的時間，能依據工作控制電路 200 的動作時脈預測。將 HR 傳至執行控制電路 152 後，工作切換電路 210 之動作時脈經過既定時脈分時，工作切換電路 210 為了解

除 CPU 時脈的停止而停止傳送 HR。當執行控制電路 152 被停止傳送 HR 時，再開始 CPU 時脈。此時，結束工作切換電路 210 之從一般工作至特殊工作的工作切換。之後以圖 8(a)說明處理方法的細節。

不論是何種情形，

(A)處理資料之儲存、載入

(B)工作的狀態遷移及 RUN—工作的選擇

等工作切換的核心處理係藉由硬體實現。關於(A)及(B)，即使不需要存取記憶體上之 TCB 亦有助於工作切換的高速化。又，為實現工作處理裝置 100，對 CPU150 僅追加停止及再開始 CPU 時脈的功能即可。此外，此等功能皆藉由硬體實現，並未限定本發明的範圍。例如，藉由硬體實現(A)或(B)的主要功能，為了輔助硬體功能，亦能以軟體實現 RTOS 的一部分功能，此為本發明所屬技術領域中具有通常知識者所應理解之事。

圖 6 係圖 5 之 CPU150 的電路圖。

與圖 3 之 CPU84 不同，於處理暫存器 154，特殊暫存器 156 與汎用暫存器 158 皆僅有一種。於處理暫存器 154，分別追加來自載入選擇電路 112 的輸入匯流排、至保留暫存器 110 的輸出匯流排、及來自工作切換電路 210 之寫入訊號(WT)用的訊號線。執行控制電路 152，藉由對輸出選擇器 164 的控制訊號(CTRL)，將處理暫存器 92 中之所欲暫存器的資料輸入至運算電路 160。運算結果成為至輸入選擇器 162 的輸入。執行控制電路 152，藉由對輸入選擇

器 162 的控制訊號(CTRL)，將運算結果輸入至處理暫存器 154 中之所欲暫存器。執行控制電路 152 一邊更新特殊暫存器 156 的程式計數器一邊執行工作。

處理資料並非儲存於記憶體上之 TCB，而是儲存於保留暫存器 110。處理資料隨時從處理暫存器 154 輸出至各保留暫存器 110。實際上，處理資料在何時序儲存於何保留暫存器 110，如上述係藉由工作切換電路 210 所控制。

並非從記憶體上之 TCB，而是從保留暫存器 110 將儲存資料載入至處理暫存器 154。實際上，在何時序將何保留暫存器 110 的處理資料載入，如上述係藉由工作切換電路 210 所控制。

連接處理暫存器 154 與載入選擇電路 112、及連接處理暫存器 154 與保留暫存器 110 的匯流排，係能一次平行傳送處理資料之位元數的匯流排。因此，藉由工作切換電路 210 之寫入訊號(WT)，能一次讀取與寫入。一般軟體 RTOS，當工作切換時，必須暫時佔有處理暫存器 154，相對於此，本實施例之硬體 RTOS，不需將用於工作切換處理的特別處理資料載入至處理暫存器 154。當從工作 A 切換至工作 B 時，由於儲存工作 A 的處理資料後僅載入工作 B 的處理資料，因此不需要將處理暫存器 154 分成系統用與使用者用的 2 種、或執行透過堆疊之資料的替換處理。

圖 7 係顯示執行控制電路 152 使 CPU 時脈停止之構造的電路圖。

第 2AND 閘 174 的輸入係原時脈(CLK0)與第 1AND 閘

172 的輸出，後者係負邏輯。第 1AND 閘 172 的輸出係停止結束訊號(HC)。由於停止結束訊號(HC)通常為 0，因此第 2AND 閘 174 將輸入之原時脈(CLK0)直接輸出以作為 CPU 時脈(CLK)。CPU150 接收第 2AND 閘 174 輸出之 CPU 時脈而動作。當第 1AND 閘 172 的輸出為「1」時，亦即，當停止結束訊號(HC)=1 時，第 2AND 閘 174 之輸出固定為 0，CPU 時脈(CLK)停止。

第 1AND 閘 172 的輸入係 OR 閘 176 的輸出與 CPU 忙碌訊號(CBUSY)的輸出，後者係負邏輯。CBUSY，係從產生 CPU150 之內部周期的既知狀態機器輸出的訊號，當 CPU150 為可停止之狀態時為「1」的訊號。例如，運算電路 94 使執行中之單一指令或被鎖定之複數個指令的最後指令結束，當 CPU 為可停止之狀態時、或 CPU 時脈之供應已停止時為「0」。

OR 閘 176 的輸入，係指令解碼器 170 的輸出(SC_DETECT)與來自工作切換電路 210 的停止要求訊號(HR)。指令解碼器 170 內建有保持 SC_DETECT 的鎖存器電路。指令解碼器 170，以從 CPU150 讀取的資料(FD)為輸入，當 FD 為系統呼叫指令時輸出 SC_DETECT=1。藉由內建鎖存器電路，即使之後 FD 變化，指令解碼器 170 亦隨時輸出 SC_DETECT=1。工作切換電路 210 對處理暫存器 154 之寫入訊號(WT)亦輸入至指令解碼器 170。當 WT 從 0 變成 1 時，如上述執行將儲存資料載入至處理暫存器 154 的動作。此 WT 係在既定時間後從 1 回到 0 的脈衝訊

號。當 WT 從 1 變成 0 時，指令解碼器 170 的鎖存器電路被重設，指令解碼器 170 停止傳送 SC_DETECT。以圖 8(b) 詳細說明 SC_DETECT 與寫入訊號(WT)的關係。本實施例之指令解碼器 170，係為了判定執行對象指令是否為系統呼叫而專門設在執行控制電路 152 的裝置。作為變形例，指令解碼器 170 與擔當 CPU150 之解碼步驟的 CPU 解碼器共通化亦可。此時，指令解碼器 170，可藉由在 CPU 解碼器追加當解碼之資料為系統呼叫指令時輸出 SC_DETECT = 1 的功能來實現。

當產生中斷要求訊號(INTR)時，工作切換電路 210 將停止要求訊號(HR)傳至執行控制電路 152。亦即，當執行系統呼叫時、或傳送停止要求訊號(HR)時，OR 閘 176 的輸出為「1」。

綜上所述，當執行系統呼叫或產生中斷要求訊號，且 CPU 忙碌訊號為「0」時，第 1AND 閘 172 的輸出為「1」，CPU 時脈不會從第 2AND 閘 174 輸出。

圖 8(a)係顯示中斷要求訊號產生時之各種訊號之關係的時序圖。

圖 8(a)中，首先，在時刻 t_0 ，工作切換電路 210 檢測出來自外部的中斷要求訊號(INTR)。工作切換電路 210，為執行特殊工作，將停止要求訊號(HR)傳至執行控制電路 152。輸入時序 t_1 ，係與檢測時序 t_0 大致同時。在時刻 t_1 ，CPU150 之狀態機器係「工作執行中」、CBUSY=1。由於 HR=1，因此 OR 閘 176 輸出「1」，但因 CBUSY=1 故 CPU150

不停止。因此，即使輸入 $HR = 1$ ，CPU 時脈(CLK)亦暫時與原時脈(CLK)同步輸出。

時間經過後，在時刻 t_2 變化成 $CBUSY = 0$ 。由於已 $HR = 1$ ，因此第 1AND 閘 172 輸出 $HC = 1$ ，從第 2AND 閘 174 輸出之 CPU 時脈固定為 0。另一方面，工作切換電路 210，以 HC 被傳送為契機，開始從一般工作至特殊工作的工作切換。其詳細後述，但此工作切換所需時間以工作控制電路 200 的動作時脈需數次。從 HC 被傳送起，以工作控制電路 200 的動作時脈變化既定次數為條件(時刻 t_3)，工作控制電路 200 停止傳送停止要求訊號(HR)。由於 $HR = 0$ ，因此執行控制電路 152 使 CPU 時脈(CLK)再開始。當 CPU150 再開始處理時，CPU150 使 $CBUSY$ 從 0 變成 1(時刻 t_4)。以此方式，從 CPU 時脈停止的時刻 t_2 至時刻 t_3 之間，執行從一般工作至特殊工作的工作切換。

此外，另一個處理方法，取代工作控制電路 200 的動作時脈變化既定次數的條件，以工作控制電路 200 結束工作切換為條件，停止傳送 HR 亦可。又，執行控制電路 152，以 HR 被停止傳送為條件，停止傳送 HC 亦可。當 $HC = 0$ 時，執行控制電路 152 使 CPU 時脈(CLK)再開始。以此方式使工作的執行再開始亦可。

圖 8(b)係顯示系統呼叫執行時之各種訊號之關係的時序圖。

圖 8(b)中，首先，在時刻 t_0 ，指令解碼器 170 檢測出系統呼叫，使 SC_DETECT 從 0 變成 1。在時刻 t_0 ，CPU150

之狀態機器係「工作執行中」、 $CBUSY = 1$ 。由於 $SC_DETECT = 1$ ，因此 OR 閘 176 輸出「1」，但因 $CBUSY = 1$ 故 CPU150 不停止。因此，即使輸出 $SC_DETECT = 1$ ，CPU 時脈(CLK)亦暫時與原時脈(CLK0)同步輸出。

時間經過後，在時刻 $t1$ 變化成 $CBUSY = 0$ 。由於 $SC_DETECT = 1$ 且 $CBUSY = 1$ ，因此 HC 被停止傳送，CPU 時脈停止。工作切換電路 210，當輸入 $HC = 0$ 時，開始工作切換處理，將寫入訊號(WT)輸出至 CPU150。在 WT 從 0 變成 1 之時刻 $t2$ ，儲存資料係載入至處理暫存器 154。寫入訊號(WT)，因脈衝訊號在既定時間經過後的時刻 $t3$ 變成 $WT = 0$ 。藉由此 $WT: 1 \rightarrow 0$ 的下降檢測，鎖存於指令解碼器 170 之 SC_DETECT 被重設(時刻 $t4$)。此時， $CBUSY$ 從 0 變成 1。由於 $CBUSY = 1$ ，因此 $HC = 0$ ，再開始 CPU 時脈。從 CPU 時脈停止的時刻 $t1$ 至時刻 $t4$ 之間，執行工作切換。

此外，另一個處理方法，取代 $WT: 1 \rightarrow 0$ 的下降檢測條件，以工作控制電路 200 結束工作切換，停止傳送 HR 為條件，執行控制電路 152 停止傳送 HC 亦可。以 $HC = 0$ 為條件，重設 SC_DETECT 。執行控制電路 152 使 CPU 時脈(CLK)再開始， $CBUSY$ 從 0 變成 1。

不論是何種情形，CPU150 皆不需在 CPU 時脈停止期間辨識 RUN—工作之切換的執行。由於工作切換電路 210，在 CPU 時脈停止，亦即 CPU150 凍結(Freeze)的期間執行工作切換處理，因此 CPU150 之處理與工作控制電路 200

之處理在程序控制上係分離。

圖 9 係用以說明管線處理之 CPU 時脈之停止時序的示意圖。

CPU150 一邊將複數個指令從記憶體依序讀取至處理暫存器 154 一邊執行，以執行工作。此工作之執行單位的指令可分解成以下 4 個階段。

- 1.F(Fetch：讀取)：從記憶體取出指令。
- 2.D(Decode：解碼)：解釋指令。
- 3.E(Execution：執行)：執行指令。
- 4.WB(Write Back：寫入)：將執行結果寫入記憶體。

某个工作依序從指令 1 執行至指令 5 時，在執行指令 1 之 F 至 WB 之後，執行指令 2 之 F 亦可。然而，為了高效率執行，大多在指令 1 執行中即開始指令 2 的執行。此種處理方法稱為管線處理。例如，當指令 1 執行至 D 階段時，開始指令 2 之 F 階段。當指令 1 執行至 E 階段時，執行指令 2 之 D 階段、指令 3 之 F 階段。以此方式，藉由增加每單位時間所執行之指令數，可減少每個工作的執行時間。

再者，亦可將各階段細分成 2 個階段。例如，將 F 階段分離成 F1:F2 的 2 個階段。當指令 1 執行至 F2 階段時，開始指令 2 之 F1 階段。當指令 1 執行至 D1 階段時，執行指令 2 之 F2 階段、指令 3 之 F1 階段。藉由將階段細分，可更高效率地利用 CPU150 之計算資源。圖 9 係說明在將各階段細分成 2 個階段而進行之管線處理，產生系統呼叫

時的 CPU 時脈停止時序。

圖 9 中，指令 1 在 CPU 時脈「0」的時序開始處理。在 CPU 時脈「4」的時序，結束指令 1 的解碼。假設指令 1 為系統呼叫。指令解碼器 170 使 SC_DETECT 從 0 變成 1。接著，SC_DETECT 從 1 回到 0 的條件，係從工作切換電路 210 至處理暫存器 154 的寫入訊號(WT)從 1 變成 0。即使 SC_DETECT=1，由於指令 2~5 已執行中或已執行開始，因此 CBUSY=1。因此，第 2AND 閘 174 跟著 CPU 時脈開始輸出。然而，執行控制電路 152，當 SC_DETECT=1 時，暫時停止程式計數器的更新以使新的指令不會被讀取。是以，指令 6 以後不會從記憶體讀取被讀取。

雖在 CPU 時脈「8」的時序指令 1 執行結束，但由於指令 2~5 在執行中，因此 CPU 忙碌訊號維持「1」。至 CPU 時脈「12」的時序時指令 5 執行結束。此時，CPU 忙碌訊號為「0」。之後，依圖 8(b)之處理，停止 CPU 時脈的供應。工作切換電路 210，將指令 5 結束為止之階段的處理資料儲存於保留暫存器 110。依據此種停止方法，可不浪費系統呼叫執行後之指令的執行結果執行工作切換。當工作切換結束時，CPU 忙碌訊號再次被設定成「1」，指令解碼器 170 的處理亦再開始。以此方式，再次供應 CPU 時脈。

此外，另一個處理方法，在系統呼叫指令執行結束之時序，設 CPU 忙碌訊號為「0」，停止 CPU 時脈的供應亦可。此時，與系統呼叫指令同時執行之其他指令在執行途

中即停止。將中途停止之指令的中間處理結果記錄於處理暫存器 154 後，儲存於保留暫存器 110。此工作下一次成為 RUN—工作時，執行中途停止之指令的後續處理。例如，某個指令在結束讀取之階段中途停止時，將從記憶體讀取之指令或運算元儲存於保留暫存器 110。當工作再開始時，保留暫存器 110 的資料係載入至處理暫存器 154，從解碼步驟執行後續處理。

圖 10 係顯示狀態記憶部 220 與工作切換電路 210 之關係的電路圖。

狀態記憶部 220 包含狀態暫存器 250 與計時器 252，狀態記憶部 220 保持工作的狀態資料。又，計時器 252，係工作遷移至 READY 狀態或 WAIT 狀態時開始的計時器。將工作遷移至 READY 狀態後經過的時間稱為「READY 經過時間」，將工作遷移至 WAIT 狀態後經過的時間稱為「WAIT 經過時間」。計時器 252 之值係作為 TIM 訊號而隨時輸出。工作切換電路 210，當執行工作切換時，某個工作變成 READY 狀態或 WAIT 狀態時，驅動此工作之計時器 252 再開始時間測量。

狀態記憶部 220 係以下所示暫存器的集合。

(A)工作 ID 暫存器 254：保持工作 ID。表示工作 ID 之 ID 訊號係從工作 ID 暫存器 254 隨時輸出至工作選擇電路 230。以下，將從工作 A 之工作 ID 暫存器 254 輸出至工作選擇電路 230 的 ID 訊號表記為「ID_A 訊號」。從狀態記憶部 220 輸出之其他訊號亦相同。

(B)工作優先順序暫存器 256：保持工作優先順序。表示工作優先順序之 PR 訊號係從工作優先順序暫存器 256 隨時輸出。「0」為最高優先順序，值愈大表示工作優先順序愈低。

(C)工作狀態暫存器 258：表示工作狀態。STOP、READY、RUN、WAIT、IDLE 其中之一係作為 ST 訊號而隨時輸出。此外，IDLE 係工作初始化之前的狀態。

(D)工作啟動位址暫存器 260：表示記憶體中工作之 TCB 位址。輸出為 AD 訊號。

(E)待機理由暫存器 262：當工作為 WAIT 狀態時，表示 WAIT 解除條件之一部分的待機理由。待機理由為「旗號等待」、「事件等待」、「互斥等待」其中之一。輸出為 WR 訊號。

(F)旗號 ID 暫存器 264：工作以旗號等待為理由處於 WAIT 狀態時，保持等待對象之旗號(以下，稱為「待機旗號」)的旗號 ID。輸出為 SID 訊號。

(G)互斥 ID 暫存器 265：工作以互斥等待為理由處於 WAIT 狀態時，保持等待對象之互斥(以下，稱為「待機互斥」)的互斥 ID。輸出為 MID 訊號。

(H)事件 ID 暫存器 266：工作以事件等待為理由處於 WAIT 狀態時，保持等待對象之事件(以下，稱為「待機事件」)的事件 ID。輸出為 EID 訊號。

(I)待機旗標暫存器 268：工作以事件等待為理由處於 WAIT 狀態時，保持待機旗標類型。輸出為 FL 訊號。

(J)旗標條件暫存器 270：工作以事件等待為理由處於 WAIT 狀態時，保持旗標條件。輸出為 FLC 訊號。關於待機旗標類型與旗標條件將於後述。

(K)旗標初始化暫存器 272：保持表示有無待機旗標類型的資料。輸出為 FLI 訊號。

(L)暫停計數器 274：於 WAIT 系系統呼叫暫停值被指定為變數。暫停計數器 274 保持暫停值。工作切換電路 210 定期減少各暫停計數器 274 的暫停值。輸出為 TO 訊號。取代工作切換電路 210 減少暫停值，暫停計數器 274 本身自律地定期減少本身的暫停值亦可。

工作選擇電路 230 根據從各狀態記憶部 220 輸出的各種訊號，執行工作的選擇。工作選擇電路 230 包含以下所示電路。

(A)執行選擇電路 232：當工作切換時，選擇下一個 RUN—工作。執行選擇電路 232，藉由從狀態記憶部 220 隨時輸出之狀態資料恆選擇任一個工作作為 RUN—工作。執行選擇電路 232 之輸入訊號為 ID、ST、PR、TIM 的 4 種類。輸出為下一個 RUN—工作的工作 ID。以圖 12 詳細說明詳細的電路構成。

(B)旗號選擇電路 234：藉由解除旗號系統呼叫的執行，選擇待從 WAIT 狀態遷移至 READY 狀態的工作。藉由解除旗號系統呼叫解除之旗號(以下，稱為「解除旗號」)的旗號 ID 係從工作切換電路 210 輸入。來自狀態記憶部 220 的輸入訊號為 ID、ST、WR、PR、SID、TIM 的 6 種類。

輸出訊號為從 WAIT 狀態遷移至 READY 狀態之工作的工作 ID。該工作不存在時，輸出 -1 等的既定值。以圖 13 詳細說明更具體的電路構成。

(C)事件選擇電路 236:藉由設定事件系統呼叫的執行，選擇從 WAIT 狀態遷移至 READY 狀態的工作。藉由設定事件系統呼叫設定之事件(以下，稱為「設定事件」)的事件 ID 係從工作切換電路 210 輸入。來自狀態記憶部 220 的輸入訊號為 ID、ST、WR、EID、FL、FLC 的 6 種類。輸出訊號為從 WAIT 狀態遷移至 READY 狀態之工作的工作 ID、及該工作的 FL、FLC。

(D)暫停檢測電路 238:檢測 WAIT 狀態之工作中、暫停計數器 274 之暫停值為 0 的工作。暫停檢測電路 238 每於更新暫停值時被驅動。暫停檢測電路 238 的輸入訊號為 ID、ST、TO 的 3 種類。輸出訊號為該工作的工作 ID。該工作不存在時，輸出 -1 等的既定值。

(E)互斥電路 240:藉由解除互斥系統呼叫的執行，選擇從 WAIT 狀態遷移至 READY 狀態的工作。藉由解除互斥系統呼叫解除之互斥(以下，稱為「解除互斥」)的互斥 ID 係從工作切換電路 210 輸入。來自狀態記憶部 220 的輸入訊號為 ID、ST、WR、PR、SID、TIM 的 6 種類。輸出訊號為從 WAIT 狀態遷移至 READY 狀態之工作的工作 ID。該工作不存在時，輸出 -1 等的既定值。

(F)檢測電路 242:當從工作切換電路 210 輸入工作 ID 時，輸出該工作的所有狀態資料。

以下，說明工作切換相關之 RUN—工作選擇、旗號、事件、互斥、暫停，特別是以工作選擇電路 230 之處理為中心，與一般技術比較並說明。

(RUN—工作選擇)

(1)一般軟體 RTOS 之 RUN—工作的選擇

圖 11 係顯示一般 RTOS 之 RUN—工作選擇時所利用之工作準備串列的圖。

工作準備串列，係形成於記憶體上、以指標連結各 READY—工作之 TCB 的串列。優先順序指標 280，係按各工作優先順序設置、表示該工作優先順序之工作之 TCB 的前端位址。圖 11 中之工作準備串列的情形，工作優先順序「0」之優先順序指標 280 指向工作 A 之 TCB 的位址，工作優先順序「1」之優先順序指標 280 指向工作 B 之 TCB 的位址。工作 A 之 TCB 進一步指向工作 D 之 TCB 的位址。

一般軟體 RTOS 一邊掃描此工作準備串列一邊選擇下一個 RUN—工作。此時，RTOS 執行下述 2 階段的處理。

A.將 RUN—工作從 RUN 狀態遷移至 READY。

B.選擇下一個 RUN—工作，將此工作之工作狀態從 READY 狀態遷移至 RUN。

若分解軟體 RTOS 之各處理則如下所示。

(RUN—工作之狀態遷移)

此處，RUN—工作係以工作 J 說明。

A1. RTOS 將 RUN—工作之工作 ID 保持於記憶體。根據此工作 ID 取得工作 J 之 TCB 的位址。

A2.存取 TCB，取得工作 J 之工作優先順序。假設工作優先順序為「0」。

A3.取得圖 11 所示之工作準備串列中、與工作 J 之工作優先順序對應的優先順序指標 280。

A4.檢測取得之優先順序指標 280 所表示的 TCB。此處檢測出工作 A 之 TCB。

A5.順著工作 A 之 TCB 具有的指標檢測最後端的 TCB。圖 11 中，工作 F 為最後端。

A6.將工作 F 之 TCB 的指標設定成指向工作 J 之 TCB 的位址。以此方式，工作 J 之 TCB 追加至工作準備串列。

A7.將工作 J 之 TCB 設定成「READY」。又，將處理資料複製至 TCB 的暫存器儲存區。

(READY－工作之狀態遷移)

B1.RTOS 檢測工作優先順序「0」之優先順序指標 280 表示哪一個 TCB。未找到 TCB 時，檢測工作優先順序「1」之優先順序指標 280 表示哪一個 TCB。在找到 TCB 之前，一邊降低工作優先順序一邊特定任一個工作。圖 11 中，特定工作 A。

B2.將工作 A 從工作準備串列移除。具體而言，將工作優先順序「0」之優先順序指標 280 覆寫成不是指向工作 A 而是指向工作 D 之 TCB 的位址。又，將工作 A 之指標設定成 NULL 以不指向工作 D 的位址。以此方式，將工作 A 之 TCB 從工作準備串列移除。

B3.將工作 A 之 TCB 設定成「RUN」。又，將儲存於

工作 A 之 TCB 之暫存器儲存區的處理資料載入至處理暫存器。

一般軟體 RTOS，藉由此種工作準備串列執行工作切換。亦即，RTOS 從複數個 READY－工作中選擇 RUN－工作的規則如下。

1. 為 READY－工作(第 1 條件)。
2. 為 READY－工作中工作優先順序最高的工作(第 2 條件)。
3. 當工作優先順序最高的工作存在複數個時，為變成 READY 狀態時間最早的工作(第 3 條件)。

將此等 3 個條件總稱為「RUN 工作選擇條件」。工作處理裝置 100 之執行選擇電路 232，藉由硬體實現此種 RTOS 之工作排程功能。

(2) 本實施例之硬體 RTOS 之 RUN－工作的選擇

圖 12 係執行選擇電路 232 的電路圖。

此處，說明從工作 0～工作 7 的 8 個工作選擇 RUN－工作。執行選擇電路 232 包含 4 個第 1 比較電路 290(290a～290d)、2 個第 2 比較電路 292(292a, 292b)、1 個第 3 比較電路 294。又，亦包含 8 個判定電路 296(296a～296h)。

判定電路 296 以表示工作狀態的 ST 訊號為輸入，輸出 READY 時表示「1」、READY 以外時表示「0」的 CID 訊號。判定電路 296 根據上述 RUN 工作選擇條件中的第 1 條件進行判定。第 1 比較電路 290 以 2 個工作的 ID、PR、TIM、及來自判定電路 296 的 CID 訊號為輸入。

著眼於第 1 比較電路 290a 來說明。第 1 比較電路 290a 比較工作 0 與工作 1，根據上述 RUN 工作選擇條件選擇較佳的工作。

第 1 判定：首先，比較分別從判定電路 296a 與判定電路 296b 輸出的 CID 訊號。其中一者為「1」時，亦即，僅其中一者之工作為 READY 狀態時，第 1 比較電路 290a 輸出該工作之 ID、PR、TIM。皆為「0」時，亦即，任一工作皆非 READY 狀態時，第 1 比較電路 290a 輸出 $ID = PR = TIM = NULL$ 。此係表示任一工作皆未被選擇。皆為「1」時，亦即，任一工作皆為 READY 狀態時，進行接下來的第 2 判定。

第 2 判定：比較工作 0 的 PR 訊號與工作 1 的 PR 訊號，選擇工作優先順序高的工作。例如，工作 0 的工作優先順序為「1」、工作 1 的工作優先順序為「2」時，輸出工作 0 之 ID、PR、TIM。藉由第 2 判定，可選擇工作優先順序高的工作作為 RUN—工作的候補。工作 0 與工作 1 之工作優先順序相同時，進行接下來的第 3 判定。

第 3 判定：比較工作 0 的 TIM 訊號與工作 1 的 TIM 訊號，選擇 READY 經過時間長的工作。READY 經過時間相同時，假設選擇工作 0。由於僅比較經過時間的大小即可判定，因此不需要工作準備串列般的 TCB 順序管理。

以此方式，藉由 RUN 工作選擇條件分別比較工作 0 與工作 1、工作 2 與工作 3、工作 4 與工作 5、工作 6 與工作 7。第 2 比較電路 292，藉由 2 個第 1 比較電路 290 的輸出

進一步篩選 RUN—工作的候補。第 2 比較電路 292a，藉由第 1 比較電路 290a 與第 1 比較電路 290b 的輸出執行工作選擇。因此，第 2 比較電路 292a 輸出工作 0～工作 3 中最適合 RUN 工作選擇條件之工作的 ID、PR、TIM。第 3 比較電路 294 亦相同，第 3 比較電路 294 輸出工作 0～工作 7 之任一工作的工作 ID。

依據此種處理方法，能以硬體實現 RUN 工作選擇條件。雖一般軟體 RTOS 一邊存取工作準備串列一邊選擇 RUN—工作，但本實施例之執行選擇電路 232，藉由從狀態記憶部 220 隨時輸出之狀態資料選擇 RUN—工作。綜合執行選擇電路 232 的處理則如下述。

(RUN—工作的狀態遷移)

此處，RUN—工作係以工作 J 說明。

A1.工作切換電路 210 將工作 J 之工作狀態暫存器 258 設定成「READY」。

A2.工作切換電路 210 設定工作 J 之計時器 252 開始測量 READY 經過時間。

以此方式，工作 J 從 RUN 狀態遷移至 READY。如上述，將處理資料儲存於工作 J 之保留暫存器 110。由於連接處理暫存器 154 與保留暫存器 110 的匯流排可平行傳送處理資料，因此可在 1 個時脈時間執行 A1 與 A2 的處理。

(READY—工作的狀態遷移)

B1.工作切換電路 210，當工作 J 之狀態遷移結束時，從執行選擇電路 232 輸出之工作 ID 特定 RUN—工作。將

此工作之工作狀態暫存器 258 設定成「RUN」。

以此方式，特定之工作從 READY 狀態遷移至 RUN。將特定之工作之處理資料從保留暫存器 110 載入至處理暫存器 154。由於連接保留暫存器 110 與處理暫存器 154 的匯流排亦為可平行傳送處理資料的位元數，因此可在 1 個時脈時間執行 B1 的處理。

軟體 RTOS，當工作切換時，由於對工作準備串列的存取，消費較多 CPU 的 CPU 時脈時間。相對於此，本實施例之工作控制電路 200，能以些微時間結束工作切換。由於狀態記憶部 220 將狀態資料隨時輸出至執行選擇電路 232，因此執行選擇電路 232 隨時輸出任一工作之工作 ID。並不是在產生工作切換後開始 RUN—工作的選擇處理，而是在產生工作切換時以執行選擇電路 232 之輸出執行 RUN—工作的選擇，此點亦有助於增加工作切換的速度。此處，雖說明工作為 8 個，但藉由增加比較電路的段數，亦可對應更多的工作。

(旗號處理)

圖 13 係顯示一般 RTOS 之旗號處理所利用之待機旗號串列的圖。

在說明待機旗號串列之前，先簡單說明旗號。在旗號表 212，旗號 ID 與旗號計數器係相對應紀錄。旗號計數器之初始值係設定成有限數。例如，假設設定成旗號 ID=4、旗號計數器=3。當任一工作以旗號 ID=4 之旗號作為待機旗號執行待機旗號系統呼叫時，工作切換電路 210 減少

待機旗號的旗號計數器。旗號計數器，係每當因待機旗號系統呼叫而被要求獲得時減少，變成 0 時則無法獲得。以旗號計數器為 0 之旗號作為待機旗號執行待機旗號系統呼叫的工作，狀態遷移至 WAIT 狀態。

另一方面，當任一工作以旗號 ID=4 之旗號作為解除旗號執行解除旗號系統呼叫時，工作切換電路 210 增加旗號表 212 的旗號計數器。綜合來說，

旗號計數器 > 0 時：執行待機旗號系統呼叫的工作從 RUN 遷移至 READY。此時，減少旗號計數器。

旗號計數器 $= 0$ 時：執行待機旗號系統呼叫的工作從 RUN 遷移至 WAIT。不減少旗號計數器。

執行待機旗號系統呼叫的工作為了從 WAIT 狀態遷移至 READY 狀態，另一個工作必須執行解除旗號系統呼叫。

(1) 一般軟體 RTOS 之旗號處理

一般軟體 RTOS 藉由待機旗號串列管理以旗號等待為理由而處於 WAIT 狀態之工作(以下，特別稱為「旗號等待工作」)的 TCB。待機旗號串列係與圖 11 之工作準備串列形狀相同的串列，形成於記憶體上。各旗號等待工作的 TCB 係藉由指標連結。優先順序指標 280 指向該工作優先順序之旗號等待工作之 TCB 的前端位址。

一般軟體 RTOS，當執行解除旗號系統呼叫時，一邊掃描此待機旗號串列一邊選擇待從 WAIT 狀態遷移至 READY 狀態的旗號等待工作。待機旗號系統呼叫及解除旗號系統呼叫執行時之 RTOS 的處理係如下述。

(待機旗號系統呼叫的執行)

此處，RUN—工作係以工作 J 說明。

A1.RTOS 將 RUN—工作的工作 ID 保持在記憶體。根據此工作 ID 取得工作 J 之 TCB 的位址。

A2.檢測於待機旗號系統呼叫指定之待機旗號的旗號計數器。以下，依旗號計數器的值處理產生分歧。

(旗號計數器 > 0 時)

A3.RTOS 減少待機旗號的旗號計數器。

A4.將工作 J 之 TCB 設定成「READY」。此時，工作 J 之 TCB 追加至工作準備串列。

(旗號計數器 $= 0$ 時)

A3.存取 TCB 以取得工作 J 的工作優先順序。假設工作優先順序為「0」。

A4.取得待機旗號串列中、與工作 J 的工作優先順序對應的優先順序指標。

A5.檢測取得之優先順序指標表示的 TCB。此處，檢測出工作 A 的 TCB。

A6.順著工作 A 之 TCB 具有的指標檢測最後端的 TCB。圖 13 中，工作 F 為最後端。

A7.將工作 F 之 TCB 的指標設定成指向工作 J 之 TCB 的位址。以此方式，工作 J 之 TCB 追加至待機旗號串列。

A8.將工作 J 之 TCB 設定成「WAIT」。又，亦設定待機旗號的旗號 ID。

(解除旗號系統的執行)

B1.RTOS 依序順著工作優先順序「0」的工作檢索以解除旗號為待機旗號的旗號等待工作。不存在時，以工作優先順序「1」的工作為檢索對象。依據是否檢測出以解除旗號為待機旗號的旗號等待工作，處理產生分歧。

(檢測出時)

B2.以檢測出之工作為工作 E 來說明。將工作 E 之 TCB 設定成「READY」。又，清除待機旗號的旗號 ID。

B3.將工作 E 之 TCB 從待機旗號串列移除。

B4.將解除旗號之工作的狀態從 RUN 狀態遷移至 READY。將此工作之 TCB 追加至工作準備串列。

(未檢測出時)

B2.增加旗號計數器。

B3.將解除旗號之工作的狀態從 RUN 狀態遷移至 READY。將此工作之 TCB 追加至工作準備串列。

一般軟體 RTOS，藉由管理此種待機旗號串列，執行旗號相關處理。當解除旗號時，RTOS 從複數個 WAIT－工作中選擇 READY－工作的規則如下。

1.為 WAIT－工作(第 1 條件)。

2.為 WAIT－工作中以解除旗號為待機旗號的工作(第 2 條件)。

3.當此種工作存在複數個時，為工作優先順序最高的工作(第 3 條件)。

4.當工作優先順序最高的工作存在複數個時，為變成 WAIT 狀態時間最早的工作(第 4 條件)。

將此等 4 個條件總稱為「旗號待機解除條件」。工作處理裝置 100 之旗號選擇電路 234，藉由硬體實現此種 RTOS 之工作排程功能。

(2) 本實施例之硬體 RTOS 之旗號處理

圖 14 係旗號選擇電路 234 的電路圖。

此處，亦說明工作 0～工作 7 的 8 個工作。旗號選擇電路 234 包含 4 個第 1 比較電路 300(300a～300d)、2 個第 2 比較電路 302(302a, 302b)、1 個第 3 比較電路 304。又，亦包含 8 個判定電路 306(306a～306h)。

判定電路 306，係以來自狀態記憶部 220 之 ST、WR、SID 訊號與來自工作切換電路 210 之表示旗號 ID 之訊號為輸入的電路。此處輸入之旗號 ID，係解除旗號的旗號 ID。判定電路 306 輸出為以解除旗號為待機旗號的旗號等待工作時表示「1」、不是以解除旗號為待機旗號的旗號等待工作時表示「0」的 CID 訊號。判定電路 306，係輸出上述旗號待機解除條件中之第 1 條件與第 2 條件相關之判定結果的電路。第 1 比較電路 300 以 2 個工作的 ID、PR、TIM 及來自判定電路 306 的 CID 訊號為輸入。

第 1 比較電路 300，係進行旗號待機解除條件中之第 3 條件與第 4 條件相關之判定的電路。第 2 比較電路 302 與第 3 比較電路 304 亦相同。如上述，RUN 工作選擇條件之第 2 條件及第 3 條件，與旗號待機解除條件之第 3 條件及第 4 條件相同。執行選擇電路 232 之各比較電路，係比較工作之狀態資料(PR、TIM)的電路。另一方面，旗號選擇

電路 234 之各比較電路亦為比較工作之狀態資料 (PR、TIM) 的電路。是以，執行選擇電路 232 之第 1 比較電路 290 與旗號選擇電路 234 之第 1 比較電路 300，係內建相同邏輯的電路，可加以共通化。各工作除了被判定電路 306 以第 1 條件與第 2 條件判定外，亦被傳至第 1 比較電路 300 的判定處理。之後，藉由與執行選擇電路 232 相同的判定處理，從第 3 比較電路 304 輸出任一工作 ID。待機旗號系統呼叫與解除旗號系統呼叫執行時之處理如下述。

(待機旗號系統呼叫的執行)

此處，RUN—工作係以工作 J 說明。

A1. 工作切換電路 210 從旗號表 212 檢測於待機旗號系統呼叫指定之旗號的旗號計數器。以下，依旗號計數器的值，處理產生分歧。

(旗號計數器 > 0 時)

A2. 工作切換電路 210 減少旗號表 212 的旗號計數器。

A3. 將工作 J 之工作狀態暫存器 258 設定成「READY」。

此時，工作切換電路 210 設定 RUN—工作之計時器 252，開始測量 READY 經過時間。

(旗號計數器 = 0 時)

A2. 工作切換電路 210，將工作 J 之工作狀態暫存器 258 設定成「WAIT」、待機理由暫存器 262 設定成「旗號等待」、旗號 ID 暫存器 264 設定成待機旗號的旗號 ID，設定計時器 252，開始測量 WAIT 經過時間。

以此方式，執行待機旗號系統呼叫的工作從 RUN 狀態

遷移至 READY 或 WAIT。

(解除旗號系統呼叫的執行)

B1.工作切換電路 210 將解除旗號的旗號 ID 輸入至各判定電路 306。各判定電路 306 以此旗號 ID 為對象判定旗號解除條件中之第 1 條件及第 2 條件是否成立。是以，各第 1 比較電路 300 根據第 3 條件及第 4 條件選擇工作。

(任一個判定電路 306 輸出「1」、第 3 比較電路 304 輸出任一個工作 ID 時)

B2.將檢測出之工作之工作狀態暫存器 258 設定成「READY」，清除待機理由暫存器 262 與旗號 ID 暫存器 264，以計時器 252 測量 READY 經過時間。

B3.將執行系統呼叫之工作之工作狀態暫存器 258 設定成「READY」，開始測量 READY 經過時間。

(任一個判定電路 306 皆未輸出「1」、第 3 比較電路 304 亦未輸出任一個工作 ID 時)

B2.工作切換電路 210 增加旗號表 212 的旗號計數器。

B3.將執行系統呼叫之工作之狀態從 RUN 狀態遷移至 READY。

由於狀態記憶部 220 將狀態資料隨時輸出至旗號選擇電路 234，因此當工作切換電路 210 將旗號 ID 輸入至判定電路 306 時，旗號選擇電路 234 可立即執行選擇處理。

(互斥處理)

互斥亦與旗號相同，利用工作間之同步處理。互斥與旗號在以下幾點不同。

1. 旗號計數器可設定成 1 以上的整數。相對於此，互斥係旗號計數器為 1 或 0 的特殊旗號。當旗號計數器為 2 以上時，2 個以上的工作可獲得相同旗號。然而，當互斥時，可獲得某個互斥的工作恆僅有 1 個。

2. 可藉由解除旗號系統呼叫解除旗號的工作，並不限於藉由待機旗號系統呼叫獲得旗號的工作。相對於此，可藉由解除互斥系統呼叫解除互斥的工作，僅為藉由待機互斥系統呼叫獲得互斥的工作。

當解除互斥時，從複數個 WAIT—工作中選擇 READY—工作的規則如下。

1. 為 WAIT—工作(第 1 條件)。
2. 為 WAIT—工作中以解除互斥為待機互斥的工作(第 2 條件)。
3. 當此種工作存在複數個時，為工作優先順序最高的工作(第 3 條件)。
4. 當工作優先順序最高的工作存在複數個時，為變成 WAIT 狀態時間最早的工作(第 4 條件)。

將此等 4 個條件總稱為「互斥待機解除條件」。

是以，待機互斥系統呼叫及解除互斥系統呼叫執行時之本實施例之硬體 RTOS 的處理如下述。在旗號表 212，互斥 ID 與表示該互斥是否佔有任一個工作的佔有狀態資料係相對應保持。佔有狀態資料，當未佔有時為「0」、當佔有時為佔有互斥之工作的工作 ID。

(待機互斥系統呼叫的執行)

此處，RUN—工作係以工作 J 說明。

A1.工作切換電路 210 檢測是否佔有於待機互斥系統呼叫指定的互斥。以下，依互斥的佔有狀態，處理產生分歧。

(未佔有互斥時)

A2.工作切換電路 210 記錄執行系統呼叫之工作的工作 ID 以作為互斥的佔有資料。

A3.將工作 J 之工作狀態暫存器 258 設定成「READY」。此時，工作切換電路 210 設定 RUN—工作之計時器 252，開始測量 READY 經過時間。

(佔有互斥時)

A2.工作切換電路 210，將工作 J 之工作狀態暫存器 258 設定成「WAIT」、待機理由暫存器 262 設定成「互斥等待」、互斥 ID 暫存器 265 設定成待機互斥的互斥 ID，設定計時器 252，開始測量 WAIT 經過時間。

(解除互斥系統呼叫的執行)

B1.工作切換電路 210，以執行系統呼叫之工作佔有解除互斥為條件，將解除旗號 ID 輸入至互斥電路 240。互斥電路 240 亦與圖 14 相同，包含多段連接的比較電路與判定互斥待機解除條件中之第 1 條件及第 2 條件是否成立的判定電路。此判定電路，以此互斥為對象，僅於互斥待機解除條件中之第 1 條件及第 2 條件皆成立時輸出「1」。此外，當未佔有解除互斥之工作執行解除互斥系統呼叫時，將該工作之狀態從 RUN 狀態遷移至 READY。

(任一個判定電路輸出「1」、互斥電路 240 輸出任一

個工作 ID 時)

B2.將檢測出之工作之工作狀態暫存器 258 設定成「READY」，清除待機理由暫存器 262 與互斥 ID 暫存器 265，以計時器 252 測量 READY 經過時間。

B3.將執行系統呼叫之工作之工作狀態暫存器 258 設定成「READY」，開始測量 READY 經過時間。

(任一個判定電路皆未輸出「1」、互斥電路 240 亦未輸出任一個工作 ID 時)

B2.工作切換電路 210 於旗號表 212，將互斥設定成非佔有狀態。

B3.將執行系統呼叫之工作之狀態從 RUN 狀態遷移至 READY。

(事件處理)

簡單說明本實施例之事件管理。在事件表 214，事件 ID 與旗標類型(以下，稱為「現行旗標類型」)係相對應記錄。旗標類型係 8 位元的位元類型。

設定事件系統呼叫，係變更現行旗標類型之設定的系統呼叫，以事件 ID 與旗標類型(以下，稱為「設定旗標類型」)為參數。當執行設定事件系統呼叫時，就該事件，現行旗標類型變更成與設定旗標類型的邏輯和。例如，當現行旗標類型為「00001100」、設定旗標類型為「00000101」時，現行旗標類型變成「00001101」。以下，對各旗標類型，從左依序稱為第 0 位元、第 1 位元、...、第 7 位元。

待機事件系統呼叫，係用以等待待機事件之現行旗標

類型滿足既定條件的系統呼叫，以事件 ID、旗標類型(以下，稱為「待機旗標類型」、旗標條件為參數。當執行待機事件系統呼叫時，判定在現行旗標類型與待機旗標類型之間旗標條件是否成立。旗標條件係邏輯和(OR)或邏輯積(AND)。當旗標條件為邏輯積(AND)時，WAIT 解除條件，係就待機旗標類型為「1」的所有位元，現行旗標類型之該位元亦皆為「1」。當旗標條件為邏輯和(OR)時，WAIT 解除條件，係就待機旗標類型為「1」的任一位元，現行旗標類型之該位元為「1」。例如，當現行旗標類型為「00001101」、待機旗標類型為「00000011」、旗標條件為「邏輯和(OR)」時，由於待機旗標類型之第 6 位元及第 7 位元中，現行旗標類型之第 7 位元為「1」，因此，此時待機事件系統呼叫之 WAIT 解除條件成立。另一方面，當旗標條件為「邏輯積(AND)」時，由於現行旗標類型之第 6 位元為「0」，因此 WAIT 解除條件不成立。

(1)一般軟體 RTOS 之事件處理

待機事件系統呼叫及設定事件系統呼叫執行時之一般 RTOS 的處理如下述。一般 RTOS 為了管理事件，將事件表保持在記憶體上。於此事件表，不僅事件 ID、現行旗標類型，通常，以此事件為待機事件之處於 WAIT 狀態之工作(以下，稱為「事件等待工作」)的工作 ID、待機旗標類型、旗標條件係相對應保持。

(待機事件系統呼叫的執行)

A1. RTOS 從事件表讀取於系統呼叫指定之事件的現行

旗標類型。

A2.依旗標條件比較現行旗標類型與待機旗標類型，判定 WAIT 解除條件是否成立。

(WAIT 解除條件成立時)

A3.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷移至 READY。

(WAIT 解除條件不成立時)

A3.在事件表記錄執行系統呼叫之工作的工作 ID。

A4.在事件表記錄待機旗標類型。

A5.在事件表記錄旗標條件。

A6.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷移至 WAIT。

(設定事件系統呼叫的執行)

B1.RTOS 就系統呼叫指定之設定事件，從事件表讀取現行旗標類型、工作 ID、待機旗標類型、旗標條件。

B2.將現行旗標類型與設定旗標類型的邏輯和記錄成新的現行旗標類型。

(設定事件不存在事件等待工作時，或即使存在、依據待機旗標類型與旗標條件 WAIT 解除條件亦不成立時)

B3.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷移至 READY。

(設定事件存在事件等待工作、WAIT 解除條件成立時)

B3.將待機事件之工作之工作狀態從 WAIT 狀態遷移至 READY。

B4.清除事件表之待機工作 ID、待機旗標類型、旗標條件。

B5.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷移至 READY。且執行 RUN—工作的選擇。

當執行設定事件系統呼叫時，從複數個 WAIT—工作中選擇 READY—工作的規則如下。

1.為 WAIT—工作(第 1 條件)。

2.為 WAIT—工作中以設定事件為待機事件的工作(第 2 條件)。

3.為比較待機旗標類型、現行旗標類型、旗標條件後，WAIT 解除條件成立的工作(第 3 條件)。

將此等 3 個條件總稱為「事件待機解除條件」。

(2)本實施例之硬體 RTOS 之事件處理

工作處理裝置 100 之待機事件系統呼叫及設定事件系統呼叫執行時之處理如下述。於工作處理裝置 100 內建之旗號表 212，事件 ID 與現行旗標類型係相對應。待機工作 ID 與待機旗標類型等的資訊係儲存於狀態記憶部 220。

(待機事件系統呼叫的執行)

A1.工作切換電路 210 從事件表 214 讀取現行旗標類型。

A2.工作切換電路 210 依旗標條件比較現行旗標類型與待機旗標類型，判定 WAIT 解除條件是否成立。

(WAIT 解除條件成立時)

A3.將執行系統呼叫之工作之工作狀態暫存器 258 設定

成「READY」。

(WAIT 解除條件不成立時)

A3.工作切換電路 210，分別將執行系統呼叫之工作之工作狀態暫存器 258 設定成「WAIT」、待機理由暫存器 262 設定成「事件等待」、事件 ID 暫存器 266 設定成待機事件的事件 ID、待機旗標暫存器 268 設定成待機旗標類型、旗標條件暫存器 270 設定成旗標條件。

(設定事件系統呼叫的執行)

B1.工作切換電路 210 從事件表 214 讀取現行旗標類型，且將於系統呼叫指定之設定事件的事件 ID 輸入至事件選擇電路 236。

B2.工作切換電路 210，將設定旗標類型邏輯累加至事件表 214 的現行旗標類型。

B3.事件選擇電路 236，就輸入之事件 ID，選擇事件待機條件成立的工作。此時，不論工作優先順序與 WAIT 經過時間如何，選擇複數個工作亦可。

(滿足事件待機解除條件的工作存在時)

B4.將事件等待工作之工作狀態暫存器 258 設定成「READY」，清除事件 ID 暫存器 266、待機旗標暫存器 268、旗標條件暫存器 270。

B5.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷移至 READY。

(滿足事件待機解除條件的工作不存在時)

B4.將執行系統呼叫之工作之工作狀態從 RUN 狀態遷

移至 READY。

(暫停處理)

移至 WAIT 狀態之工作，當 WAIT 解除條件成立時遷移至 READY 狀態。然而，由於某些外部因素或應用程式的錯誤(Bug)，而妨礙 WAIT 解除條件成立時，工作無法離開 WAIT 狀態。因此，一般而言，將工作遷移至 WAIT 狀態時設定暫停值。暫停值係定期減少，當成為 0 時，即使 WAIT 解除條件不成立，工作亦可從 WAIT 狀態遷移至 READY 狀態。亦即，可防止工作停止在 WAIT 狀態暫停值以上的時間。

(1)一般軟體 RTOS 之暫停處理

軟體構成之一般 RTOS 之情形，在 WAIT 狀態之工作的 TCB 設定暫停值，此暫停值係定期減少。RTOS 周期性對 CPU 之處理發出中斷，檢查所有 TCB，檢測出暫停值成為 0 的 WAIT—工作。當檢測出此種工作時，RTOS 將該工作之工作狀態從 WAIT 狀態遷移至 READY。

(2)本實施例之硬體 RTOS 之暫停處理

另一方面，本實施例之情形，工作切換電路 210 定期減少各暫停計數器 274 的暫停值。暫停值，當執行 WAIT 系系統呼叫時係設定為參數，工作切換電路 210 在執行該系統呼叫之工作的暫停計數器 274 設定暫停值。

由於 CPU150 不介入暫停值的減少處理，因此工作切換電路 210 能與工作執行處理獨立，更新暫停值。因此，即使 CPU150 正執行工作時，在工作控制電路 200 亦自律

地執行暫停值的更新。由於狀態資料係隨時輸入至暫停檢測電路 238，因此暫停檢測電路 238，能以與暫停值更新時序大致相同的時序，檢測暫停值成為 0 的工作。暫停檢測電路 238 輸出此種工作的工作 ID。工作切換電路 210，當從暫停檢測電路 238 輸入工作 ID 時，認知產生暫停，傳送 HC 以停止供應 CPU 時脈。工作控制電路 200 將產生暫停之 WAIT—工作遷移至 READY 狀態，且將 RUN—工作遷移至 READY 狀態。工作切換電路 210 從 READY—工作之中選擇下一個待執行工作。又，工作切換電路 210 再啟動產生暫停之工作的計時器 252，測量 READY 經過時間。

依據此種處理方法，當工作執行中，亦即，當 CPU 時脈動作中產生暫停時，可即時對 CPU150 發出中斷，執行工作切換。又，在工作執行中，工作切換電路 210 不需借助 CPU150 的處理能力即可獨立執行暫停值的更新處理。

(有限狀態機器之工作切換電路 210)

圖 15 係工作切換電路 210 的狀態遷移圖。

於初始化處理(A1)，所有工作處於 IDLE 狀態。當初始化處理結束時(S10)，任一個工作成為 RUN—工作、而成為工作執行狀態(A2)。當檢測出中斷要求訊號時(S12)，特殊工作成為 RUN—工作、執行中斷處理(A3)。當中斷處理結束時(S14)，工作切換電路 210 從一般工作選擇 RUN—工作，遷移至 A2。

又，於工作執行中(A2)，當執行系統呼叫時(S16)，執

行系統呼叫處理(A4)。當未產生工作切換、亦即 RUN—工作的切換時(S18)，返回 A2。另一方面，當藉由系統呼叫處理(A4)產生工作切換時(S20)，工作切換電路 210 根據執行選擇電路 232 的輸出執行 RUN—工作的選擇(A5)。當工作切換結束時(S22)，處理狀態移至 A2。

最後，說明僅構裝工作處理裝置 100 之主要元件之儲存電路 120 與工作控制電路 200 之其中一者的情形。

(未裝載工作控制電路 200 之類型的工作處理裝置 100)

圖 16 係圖 5 之工作處理裝置 100 中，未裝載工作控制電路 200 之工作處理裝置 100 的電路圖。

替代未裝載工作控制電路 200，追加暫存器切換控制電路 322 與處理資料保持部 320。由於未裝載工作控制電路 200，因此工作排程功能係藉由軟體 RTOS 實現。是以，當工作切換時，RTOS 必須暫時取得 CPU150 的使用權。一般而言，處理資料保持部 320 保持 RTOS 用的處理資料。當 RTOS 取得 CPU150 的使用權時，處理資料保持部 320 替換處理資料保持部 320 之 RTOS 用的處理資料與特殊暫存器 156 之工作用的處理資料。以下，以從工作 A 至工作 B 的工作切換說明其處理過程。

A1.當工作 A 執行系統呼叫時，將系統呼叫之變數與系統呼叫之 ID 記錄於汎用暫存器 158 的一部分。

A2.暫存器切換控制電路 322 將工作 A 之處理資料移至處理資料保持部 320，將處理資料保持部 320 之 RTOS 用處理資料載入至處理暫存器 154。在此階段，RTOS 取得

CPU150 的使用權。

A3.暫存器切換控制電路 322 將寫入訊號輸入至保留暫存器 110a，將處理資料保持部 320 之工作 A 用處理資料儲存至保留暫存器 110。

A4.RTOS 根據記錄於汎用暫存器 158 之系統呼叫的變數與 ID，執行對應於系統呼叫的處理。又，將工作 A 之 TCB 之工作狀態資料設定成「READY」，在工作準備串列追加工作 A 之 TCB。

B1.接著，RTOS 依據上述 RUN 工作選擇條件選擇 RUN 一工作，此處，選擇工作 B。

B2.RTOS 指示暫存器切換控制電路 322，將指定工作 B 之工作選擇訊號輸入至載入選擇電路 112。將處理資料從保留暫存器 110b 移至處理資料保持部 320。

B3.暫存器切換控制電路 322 替換處理資料保持部 320 之工作 B 用處理資料與處理暫存器 154 之 RTOS 用處理資料。據此，工作 B 取得 CPU150 的使用權。

依據此種處理方法，相較於裝載工作控制電路 200 之圖 5 的工作處理裝置 100，可縮小工作處理裝置 100 整體的電路尺寸。雖 RTOS 實現為軟體，但能以來自暫存器切換控制電路 322 之訊號硬體地控制處理資料的載入、儲存。將分別連接處理暫存器 154、處理資料保持部 320、載入選擇電路 112、保留暫存器 110 之匯流排設定成可平行傳送處理資料的位元數時，相較於將處理資料儲存至 TCB 及從 TCB 載入處理資料，可實現高速的工作切換。

(未裝載儲存電路 120 之類型的工作處理裝置 100)

圖 17 係圖 5 之工作處理裝置 100 中，未裝載儲存電路 120 之工作處理裝置 100 的電路圖。

替代未裝載儲存電路 120，追加中斷介面電路 324。由於未裝載儲存電路 120，處理資料係儲存於記憶體的 TCB。處理資料的載入、儲存係藉由軟體庫的 RTOS 實現。是以，當工作切換時，RTOS 必須暫時取得 CPU150 的使用權。以下，以從工作 A 至工作 B 的工作切換說明其處理過程。

當藉由執行系統呼叫產生工作切換時，首先，軟體 RTOS 將工作 A 的處理資料儲存於工作 A 的 TCB。接著，將 RTOS 用的處理資料載入至處理暫存器 154。此時的處理方法，與以圖 3 說明之內容相同。

軟體 RTOS 將系統呼叫之參數寫入中斷介面電路 324。執行控制電路 152 停止 CPU150 之 CPU 時脈。中斷介面電路 324 使工作控制電路 200 執行工作切換。首先，工作切換電路 210 將工作 A 之工作狀態暫存器 258 設定成 READY，藉由來自工作選擇電路 230 的輸出選擇下一個 RUN—工作之工作 B。工作切換電路 210 指示中斷介面電路 324 以載入工作 B 的處理資料。此處，中斷介面電路 324 使執行控制電路 152 再開始 CPU 時脈。又，中斷介面電路 324 通知軟體 RTOS 工作 B 已被選擇。軟體 RTOS 存取工作 B 的 TCB，將工作 B 的處理資料載入至處理暫存器 154。

依據此種處理方法，相較於裝載儲存電路 120 之圖 5 的工作處理裝置 100，可縮小工作處理裝置 100 整體的電

路尺寸。雖 RTOS 之功能的一部分係實現為軟體，但工作選擇處理能藉由工作控制電路 200 實現。

相較於圖 2 及圖 3 所說明之軟體 RTOS，圖 16 及圖 17 之工作處理裝置 100，皆能使 RTOS 之功能的一部分硬體化。如圖 16 所說明，由於存在儲存電路 120，因此處理資料的儲存、載入不需存取 TCB。因此，能藉由暫存器切換控制電路 322 執行處理資料的儲存、載入處理。又，如圖 17 所說明，由於存在工作控制電路 200，因此軟體 RTOS 能將工作選擇功能讓給工作控制電路 200。

如圖 5 所說明，裝載儲存電路 120 與工作控制電路 200 之工作處理裝置 100 的情形，可將 RTOS 之工作排程功能完全硬體化。由於當工作切換時不需存取記憶體的 TCB，因此工作切換處理可更快速。依據本發明人的實驗，相較於圖 3 等所說明之一般軟體 RTOS，確認本實施例之工作處理裝置 100 能以大約 100 倍的速度動作。

(改良例)

接著，作為改良例，說明藉由硬體邏輯實現中斷處理的工作處理裝置 100。

圖 18 係一般軟體 OS 之中斷處理的時序圖。

軟體 OS，當從中斷控制器(未圖示)接收中斷要求訊號時，啟動中斷處理程式，亦即，啟動基本例之特殊工作。按下鍵盤、接收通訊封包、結束 DMA 傳輸、或單純經過既定時間等各種因素成為中斷要求訊號的產生契機。特殊工作係藉由軟體構裝之工作，對應中斷因素執行各種中斷

處理。

同圖中，首先，在一般工作執行中檢測中斷要求訊號 INTR。為待即時對應中斷要求訊號時，中斷執行中的一般工作，執行權移至 OS(S100)。OS 使一般工作之上下文資訊保留於 TCB(S102)，啟動特殊工作(S104)。

特殊工作分析中斷因素(S106)。由於因中斷要求訊號而執行對中斷因素暫存器(未圖示)的各種寫入，因此藉由調查中斷因素暫存器可特定中斷因素。特殊工作對應中斷因素判定待執行中斷處理，開始中斷處理。於中斷處理之過程中執行各種系統呼叫指令。執行系統呼叫指令時，執行權再次移至 OS(S108)。OS 執行指定之系統呼叫(S110)。系統呼叫執行後，執行權再次移至特殊工作(S112)。由於中斷處理為優先度高的處理，因此一般而言只要特殊工作的處理未結束，執行權不會移至一般工作。

特殊工作繼續中斷處理(S114)，當再次執行系統呼叫指令時，執行權移至 OS(S116)。如此，特殊工作與 OS 交互取得執行權，且最後之執行權移至特殊工作(S118)，特殊工作結束中斷處理(S120)。結束中斷處理後，執行權移至 OS(S122)，執行從特殊工作至一般工作之工作切換(S124)。以此方式，再次開始一般工作之通常處理(S126)。

基本例所示之工作處理裝置 100，在藉由硬體邏輯實現 RTOS 之功能之點與軟體 OS 不同，但中斷處理之基本流程則與軟體 OS 大致相同。然而，如基本例所說明，S102 與 S124 之工作切換、或 S110 等之系統呼叫執行，相較於

軟體 OS 速度快許多。

基本例之工作處理裝置 100 之情形，在 S100、S108、S116、S122 除了停止 CPU 時脈(CLK)之外，亦執行 RTOS 之處理。又，藉由 S104、S112、S118、S126 再次開始 CPU 時脈(CLK)，並執行特殊工作或一般工作的處理。特殊工作係工作優先順序特別高的工作，但在為根據 CPU 時脈(CLK)動作之上下文為基之工作之點，與一般工作相同。

於改良例，藉由硬體邏輯實現此種中斷處理，具體而言為特殊工作之功能的一部分，使中斷處理更加高速化。

雖中斷處理有各種內容，但根據中斷求訊號亦有很多單純定型的處理內容。例如，假設某個一般工作 A 開始 DMA 傳輸，等待 DMA 傳輸結束之狀況。一般工作 A 在 DMA 傳輸開始時，執行待機事件系統呼叫，移至 WAIT 狀態。DMA 傳輸結束時，既定中斷要求訊號輸入至(基本例之)工作切換電路 210。此時啟動之特殊工作，執行設定事件系統呼叫，將表示 DMA 傳輸結束之旗標類型設定於事件表 214。藉由改變事件表 214 之現行旗標類型，一般工作 A 之 WAIT 解除條件成立，一般工作 A 移至 READY 狀態。如此，DMA 傳輸結束所伴隨執行之中斷處理的內容較單純。

因此，改良例之工作處理裝置 100，中斷處理的內容較單純，較佳為將產生頻率高的中斷要求訊號事前標記為「高速中斷要求訊號 INTR(H)」。改良例之中斷要求訊號大致分為高速中斷要求訊號 INTR(H)與「通常中斷要求訊

號 INTR(N)」的 2 種。

圖 19 係改良例之工作處理裝置 100 的電路圖。

改良例之工作處理裝置 100，亦具備 CPU150、儲存電路 120、工作控制電路 200。改良例之工作處理裝置 100 進一步追加有中斷電路 400。

高速中斷要求訊號 INTR(H)係輸入至中斷電路 400。中斷電路 400 之構成及中斷電路 400 之高速中斷要求訊號之處理方法將於後述。通常中斷要求訊號 INTR(N)，與基本例相同，直接輸入至工作切換電路 210，藉由特殊工作進行中斷處理。高速中斷要求訊號之中斷處理(以下，稱為「高速中斷處理」)比通常中斷要求訊號之中斷處理(以下，稱為「通常中斷處理」)更快速。另一方面，通常中斷處理之情形，具有可彈性定義詳細之處理內容以作為軟體之特殊工作的優點。改良例之工作處理裝置 100，藉由併用高速中斷要求訊號與通常中斷要求訊號，可維持 RTOS 之汎用性並謀求高速化。

圖 20 係中斷電路 400 的電路圖。

中斷電路 400 具備訊號選擇電路 402、操作電路 404、及記憶部 406。INTR(H)₀~INTR(H)_{i-1} 之 i 個高速中斷要求訊號，係不定期地輸入至訊號選擇電路 402。有短期間內複數個高速中斷要求訊號 INTR(H)輸入的情形，亦有同時複數個高速中斷要求訊號 INTR(H)輸入的情形。訊號選擇電路 402 能暫時儲存接收之高速中斷要求訊號 INTR(H)。

訊號選擇電路 402，根據既定選擇規則從所緩衝之高速中斷要求訊號 INTR(H)之中選擇 1 個高速中斷要求訊號 INTR(H)。此處所謂選擇規則，只要根據設計要件任意決定即可。例如，預先在各高速中斷要求訊號 INTR(H)設定優先順序，當緩衝有複數個高速中斷要求訊號 INTR(H)時，選擇其中優先順序最高的高速中斷要求訊號 INTR(H)即可。或，選擇對訊號選擇電路 402 之輸入時序最早的高速中斷要求訊號 INTR(H)亦可，隨機選擇亦可。當選擇高速中斷要求訊號 INTR(H)_n(n 為 $0 \sim i-1$ 之間之整數)時，訊號選擇電路 402 傳送對應訊號 QINT_n。

操作電路 404，當傳送 QINT_n 時，傳送 ISROP 訊號。藉由 ISROP 訊號之傳送，將高速中斷處理中之事傳至訊號選擇電路 402。當傳送 ISROP 訊號時，在停止傳送 ISROP 訊號之前訊號選擇電路 402 不會傳送下一個 QINT。當停止傳送 ISROP 訊號時，訊號選擇電路 402 可選擇成為下一個處理對象之高速中斷要求訊號 INTR(H)。

當傳送 QINT_n 時，操作電路 404 為了對工作切換電路 210 要求高速中斷處理之執行，亦傳送 ISR_RQ。當傳送 ISR_RQ 時，工作切換電路 210 停止供應 CPU 時脈 (CLK)。以此方式，停止一般工作之執行。

操作電路 404，根據 QINT_n，亦即，根據所選擇之高速中斷要求訊號 INTR(H)_n，在 DSC_ADD[k-1:0]設定既定位址 ADD[n]。藉由 DSC_ADD[k-1:0] = ADD[n]之輸入，記憶部 406 之位址 ADD[n]所儲存之中斷操作指令 p0

傳至工作切換電路 210 以作為 $ISR_DT[31:0]$ 。工作切換電路 210，根據接收之中斷操作指令 p0 執行高速中斷處理。

本實施例之中斷操作指令，如之後圖 22 所述般，被格式化成 32 位元大小，藉由其最上位位元 $ISR_DT[31]$ ，表示後續中斷操作指令是否存在。此外，中斷操作指令，亦能以 64 位元、128 位元等 32 位元以外之任意大小來格式化。當最上位位元 $ISR_DT[31]=1$ 時，工作切換電路 210 傳送 ISR_NX ，對操作電路 404 要求後續中斷操作指令 p1。 ISR_NX 成為後續要求訊號。操作電路 404，將上述位址 $ADD[n]$ 累加 1 字組(本實施例為 32 位元)後之位址 $ADD[n]+1$ 設定在 $DSC_ADD[k-1:0]$ 。位址 $ADD[n]+1$ 所儲存之中斷操作指令 p1 傳至工作切換電路 210 以作為 $ISR_DT[31:0]$ 。工作切換電路 210，根據接收之中斷操作指令 p1 執行高速中斷處理。

當最上位位元 $ISR_DT[31]=0$ 時，工作切換電路 210 傳送 ISR_END ，將高速中斷處理結束通知操作電路 404。 ISR_END 成為操作結束訊號。操作電路 404 停止傳送 $ISROP$ 。當停止傳送 $ISROP$ 時，訊號選擇電路 402 可選擇另一個高速中斷要求訊號 $INTR(H)$ ，傳送新的 $QINT$ 。

此外，於本改良例， ISR_END 訊號係輸入至操作電路 404，藉由操作電路 404 停止傳送 $ISROP$ ，控制訊號選擇電路 402。作為變形例，工作切換電路 210，當高速中斷處理結束時，直接將 ISR_END 訊號傳至訊號選擇電路 402，控制訊號選擇電路 402 亦可。

圖 21 係記憶部 406 的資料結構圖。

本改良例之記憶部 406 係記憶體。在位址「0x000～0x0FF」儲存有對應高速中斷要求訊號 INTR(H)_0 之中斷操作指令群。同樣地，在位址「0x100～0x1FF」儲存有對應高速中斷要求訊號 INTR(H)_1 之中斷操作指令群。本改良例之中斷操作指令係系統呼叫指令。藉由中斷要求訊號與中斷操作指令之對應形成「中斷表」。例如，訊號選擇電路 402 選擇高速中斷要求訊號 INTR(H)_0 時，操作電路 404 將高速中斷要求訊號 INTR(H)_0 之最初位址「0x000」指定於 DSC_ADD[k - 1 : 0]。該中斷操作指令「systemcall_00」從記憶部 406 傳至工作切換電路 210。由於中斷操作指令「systemcall_00」存在有後續中斷操作指令「systemcall_01」，因此中斷操作指令「systemcall_00」之最上位位元設定成「1」。工作切換電路 210，在執行中斷操作指令「systemcall_00」後，為了要求後續中斷操作指令而傳送 ISR_NX。

操作電路 404，將「0x000」累加 1 字組後之位址「0x001」重新指定在 DSC_ADD[k - 1 : 0]。後續中斷操作指令「systemcall_01」從記憶部 406 傳至工作切換電路 210。「systemcall_01」之最上位位元設定成「0」，工作切換電路 210，對於高速中斷要求訊號 INTR(H)_0 之高速中斷處理，可辨識「systemcall_01」為最後之中斷操作指令。工作切換電路 210，當中斷操作指令「systemcall_01」之執行結束後，傳送 ISR_END，通知高速中斷處理之結束。

圖 22 係中斷操作指令的資料結構。

如上述，ISR_DT[31]表示有無後續中斷操作指令。藉由 ISR_DT[30:24]表示系統呼叫之種類。ISR_DT[30:24] = 0000001 表示「設定事件系統呼叫」。因此，中斷操作指令 410a 係設定事件系統呼叫之執行指令。ISR_DT[23:8]表示設定旗標類型，ISR_DT[7:0]表示事件 ID。工作切換電路 210，當接收中斷操作指令 410a 時，以與基本例同樣之處理方法設定事件表 214。

ISR_DT[30:24] = 0000010 表示「解除旗號系統呼叫」。因此，中斷操作指令 410b 係解除旗號系統呼叫之執行指令。ISR_DT[7:0]表示解除旗號之旗號 ID。ISR_DT[30:24] = 0000011 表示「解除待機系統呼叫」。因此，中斷操作指令 410c 係解除待機系統呼叫之執行指令。ISR_DT[4:0]表示待從 WAIT 狀態解除之工作的工作 ID。ISR_DT[30:24] = 0000100 表示「喚醒工作系統呼叫」。因此，中斷操作指令 410d 係喚醒工作系統呼叫之執行指令。ISR_DT[4:0]表示待從 WAIT 狀態解除之工作的工作 ID。ISR_DT[30:24] = 0000101 表示「啟動系統呼叫」。因此，中斷操作指令 410e 係啟動系統呼叫之執行指令。ISR_DT[4:0]表示待啟動工作的工作 ID。除此之外，亦可將各種系統呼叫儲存於記憶部 406 以作為中斷操作指令。

記憶部 406 可為 ROM(Read Only Memory)，亦可為可讀寫之 RAM(Random Access Memory)。若可藉由應用程式覆寫記憶部 406 的內容，則能藉由軟體設定變更高速中斷

處理的內容。

圖 23 係顯示高速中斷處理之過程的程序控制圖。

首先，訊號選擇電路 402，選擇成為處理對像之高速中斷要求訊號 $\text{INTR(H)}_n(\text{S130})$ ，傳送 $\text{QINT}_n(\text{S132})$ 。操作電路 404，傳送 ISROP ，將正在進行高速中斷處理通知訊號選擇電路 402(S134)。訊號選擇電路 402，當傳送 ISROP 時，即使接收新的高速中斷要求訊號，亦預先緩衝，不會傳送 QINT 。

另一方面，操作電路 404，當傳送 QINT_n 時，傳送 ISR_RQ ，對工作切換電路 210 要求高速中斷處理之開始(S136)。此時，工作切換電路 210 停止 CPU 時脈(CLK)，準備高速中斷處理之開始。操作電路 404，將對應 QINT_n 之位址 $\text{ADD}[n]$ 指定在 DSC_ADD ，從記憶部 406 讀取中斷操作指令 $\text{p0}(\text{S138})$ 。中斷操作指令 p0 ，係作為 $\text{ISR_DT}[31:0]$ 而傳至工作切換電路 210(S140)。

工作切換電路 210 根據接收之中斷操作指令 p0 ，變更旗號表 212、事件表 214、狀態記憶部 220 的設定內容。更具體而言，工作切換電路 210，藉由執行與一般工作發出解除旗號系統呼叫(傳訊旗號系統呼叫)或設定事件系統呼叫(設定旗標系統呼叫)時同等的處理程序，變更旗號表 212、事件表 214、狀態記憶部 220 的設定內容。中斷操作指令之處理內容，與基本例所示之系統呼叫指令之處理內容相同。只要中斷操作指令之最上位位元設定成「1」，則工作切換電路 210 傳送 ISR_NX ，對操作電路 404 要求

下一個中斷操作指令 p1(S144)。操作電路 404，將下一個中斷操作指令 p1 載入(S146)，將中斷操作指令 p1 傳至工作切換電路 210(S148)。

當工作切換電路 210 執行最後之中斷操作指令 px 時，亦即，執行最上位位元設定成「0」之中斷操作指令 px 時，工作切換電路 210 傳送 ISR_END(S152)。操作電路 404，辨識高速中斷處理之結束，停止傳送 ISROP(S154)。以此方式，訊號選擇電路 402 能選擇下一個高速中斷要求訊號。

圖 24 係改良例之工作切換電路 210 的狀態遷移圖。

於改良例，除了圖 15 所示之狀態遷移圖外，追加高速中斷處理(A6)。又，通常中斷處理(A3)與圖 15 所示之中斷處理(A3)相同。除此之外，與基本例相同符號之構成表示相同處理內容。

於工作執行中(A2)，當檢測高速中斷要求訊號 INTR(H)時(S24)，傳送 ISR_RQ，執行高速中斷處理(A6)。當中斷電路 400 將中斷操作指令傳至工作切換電路 210 時(S26)，工作切換電路 210 執行對應之系統呼叫處理(A4)。當系統呼叫指令之執行結束後，狀態遷移至高速中斷處理(A6)(S28)。接著，若殘留有待處理之中斷操作指令，則結束高速中斷處理(S30)，選擇下一個待執行之一般工作(A5)。

圖 25 係顯示改良例之工作處理裝置 100 之高速中斷處理之處理過程的時序圖。

同圖中，首先，在一般工作執行中檢測中斷要求訊號

INTR。為待即時對應之中斷要求訊號時，停止 CPU 時脈 (CLK) 以中斷執行中之一般工作，開始中斷電路 400 與工作切換電路 210 之高速中斷處理 (S160)。

中斷電路 400 適當地讀取中斷操作指令，工作切換電路 210 執行指定為中斷操作指令之系統呼叫指令。以此方式，當藉由一連串之系統呼叫所實現之高速中斷處理結束後 (S162)，工作切換電路 210 選擇下一個 RUN－工作 (S164)。選擇後，再次開始 CPU 時脈 (CLK)，以再次開始一般工作之通常處理 (S166)。

根據改良例所示之工作處理裝置 100，藉由中斷電路 400 與工作切換電路 210 之共同運作，能以硬體水準實現高速中斷處理。根據本發明人之實驗，確認了相較於基本例之工作處理裝置 100，改良例之工作處理裝置 100 能以約 4 倍的速度動作。藉由將記憶部 406 形成為可寫入的記憶體，即使是高速中斷處理，亦能某種程度彈性地設定其處理內容。

以往，為了高速執行包含 OS 之軟體，嘗試各種方法。上述處理大多為 CPU 時脈數的增加、記憶體容量或暫存器容量的增加、或複數個 CPU 之平行處理、再者為網路分散處理等大規模化、複雜化者。

相對於此，本實施例所示之工作處理裝置 100 之情形，藉由在既有之 CPU150 外追加儲存電路 120 或工作控制電路 200，大幅提升處理效率。軟體 OS 之情形，為了實現原本欲執行之工作切換等的處理，必須執行 TCB 或工作準備

串列之管理等附加處理。相對於此，工作處理裝置 100，由於藉由硬體邏輯實現工作切換、工作之狀態管理、中斷處理，因此可解除此種軟體特有的酬載。因此，可構成抑制耗電與成本增加、且實現高速化的理想系統。如改良例所示，藉由電子電路實現高速中斷處理的演算法，以進一步實現高速化。

以上，根據實施例說明本發明。實施形態係例示，此等各構成元件與各處理步驟的組合能有各種變形例，且此等變形例亦在本發明的範圍內，此為本發明所屬技術領域中具有通常知識者應當理解之事。

依據本發明，可實現於多工處理更高效率之工作的執行控制。

【圖式簡單說明】

圖 1 係工作的狀態遷移圖。

圖 2 係一般 RTOS 的概念圖。

圖 3 係軟體 RTOS 所執行之一般 CPU 的電路圖。

圖 4 係本實施例之 RTOS 的概念圖。

圖 5 係本實施例之工作處理裝置的電路圖。

圖 6 係圖 5 之 CPU 的電路圖。

圖 7 係顯示執行控制電路 152 使 CPU 時脈停止之構造的電路圖。

圖 8(a)係顯示中斷要求訊號產生時之各種訊號之關係的時序圖。

圖 8(b)係顯示系統呼叫執行時之各種訊號之關係的時序圖。

圖 9 係用以說明管線處理之 CPU 時脈之停止時序的示意圖。

圖 10 係顯示狀態記憶部與工作切換電路之關係的電路圖。

圖 11 係顯示一般 RTOS 之 RUN—工作選擇時所利用之工作準備串列的圖。

圖 12 係執行選擇電路的電路圖。

圖 13 係顯示一般 RTOS 之旗號處理所利用之待機旗號串列的圖。

圖 14 係旗號選擇電路的電路圖。

圖 15 係工作切換電路的狀態遷移圖。

圖 16 係圖 5 之工作處理裝置中，未裝載工作控制電路之工作處理裝置的電路圖。

圖 17 係圖 5 之工作處理裝置中，未裝載儲存電路之工作處理裝置的電路圖。

圖 18 係一般軟體 OS 之中斷處理的時序圖。

圖 19 係改良例之工作處理裝置的電路圖。

圖 20 係中斷電路的電路圖。

圖 21 係記憶部的資料結構圖。

圖 22 係中斷操作指令的資料結構。

圖 23 係顯示高速中斷處理之過程的程序控制圖。

圖 24 係改良例之工作切換電路的狀態遷移圖。

圖 25 係顯示改良例之工作處理裝置之高速中斷處理之處理過程的時序圖。

【主要元件符號說明】

84	CPU
86	汎用暫存器
88	特殊暫存器
90	執行控制電路
92	處理暫存器
94	運算電路
96	輸入選擇器
98	輸出選擇器
100	工作處理裝置
110	保留暫存器
110_0~110_n	保留暫存器
112	載入選擇電路
120	儲存電路
150	CPU
152	執行控制電路
154	處理暫存器
156	特殊暫存器
158	汎用暫存器
160	運算電路
162	輸入選擇器

164	輸出選擇器
170	指令解碼器
172	第 1AND 閘
174	第 2AND 閘
176	OR 閘
200	工作控制電路
210	工作切換電路
212	旗號表
214	事件表
220	狀態記憶部
220_0 ~ 220_n	狀態記憶部
230	工作選擇電路
232	執行選擇電路
234	旗號選擇電路
236	事件選擇電路
238	暫停檢測電路
240	互斥電路
242	檢測電路
250	狀態暫存器
252	計時器
254	工作 ID 暫存器
256	工作優先順序暫存器
258	工作狀態暫存器
260	工作啟動位址暫存器

262	待機理由暫存器
264	旗號 ID 暫存器
265	互斥 ID 暫存器
266	事件 ID 暫存器
268	待機旗標暫存器
270	旗標條件暫存器
272	旗標初始化暫存器
274	暫停計數器
280	優先順序指標
290a~290d	第 1 比較電路
292a, 292b	第 2 比較電路
294	第 3 比較電路
296a~296h	判定電路
300a~300d	第 1 比較電路
302a, 302b	第 2 比較電路
304	第 3 比較電路
306a~306h	判定電路
320	處理資料保持部
322	暫存器切換控制電路
324	中斷介面電路
400	中斷電路
402	訊號選擇電路
404	操作電路
406	記憶部

200910207

410a～410e

中斷操作指令

五、中文發明摘要：

工作處理裝置，具備 CPU、儲存電路、及工作控制電路。工作控制電路 200，具備與工作選擇電路及各工作相對應的狀態記憶部。CPU，當執行既定系統呼叫指令時，通知工作控制電路。工作控制電路，當被通知執行系統呼叫指令時，根據工作選擇電路的輸出選擇下一個待執行的工作。當中斷電路 400 接收高速中斷要求訊號時，工作切換電路 210 執行中斷電路 400 所指定之中斷操作指令，以控制工作的狀態遷移。

六、英文發明摘要：

(無)

十、申請專利範圍：

1.一種工作處理裝置，其特徵在於，具備：

處理暫存器，暫時保持用以執行工作的資料；

執行控制電路，將指令及運算元從記憶體載入至該處理暫存器，依該處理暫存器之指令及運算元執行工作；

狀態暫存器，保持表示工作之執行狀態的狀態資料；

工作切換電路，控制工作的執行狀態；

工作選擇電路，根據該狀態資料，藉由既定選擇條件選擇工作；以及

中斷電路，處理來自外部之中斷要求訊號；

該執行控制電路，當執行中之第 1 工作執行既定系統呼叫指令時，將既定系統呼叫訊號傳至該工作切換電路；

該工作選擇電路，從表示可執行待機狀態之 READY 狀態之工作之中選擇成為下一個執行對像的工作；

該工作切換電路，根據接收該系統呼叫訊號時之該工作選擇電路之輸出選擇成為下一個執行對像的第 2 工作，使該處理暫存器之資料保留於既定記憶區域，且將該第 1 工作之狀態資料從表示工作執行中之 RUN 狀態設定變更成其他狀態，對該第 2 工作，將已保留於該記憶區域之資料載入至該處理暫存器，且將該第 2 工作之狀態資料從 READY 狀態變更成 RUN 狀態，以切換成為執行對像的工作；

該中斷電路，當接收該中斷要求訊號時，將與該中斷要求訊號相對應之中斷操作指令傳至該工作切換電路；

該工作切換電路，根據該中斷操作指令變更工作之狀態資料。

2.如申請專利範圍第 1 項之工作處理裝置，其中，該中斷電路，具備：

訊號選擇電路，從外部接收中斷要求訊號並暫時保持，從所保持之 1 個以上之中斷要求訊號之中，選擇任一個中斷要求訊號作為處理對像；以及

操作電路，將對應該選擇之中斷要求訊號之中斷操作指令傳至該工作切換電路；

該工作切換電路，當該中斷操作指令之執行結束時，對該中斷電路傳送操作結束訊號；

該訊號選擇電路，以接收到該操作結束訊號為條件，將成為下一個處理對像之中斷要求訊號傳至該操作電路。

3.如申請專利範圍第 2 項之工作處理裝置，其中，該工作切換電路，當該中斷操作指令之執行結束時，將該操作結束訊號傳至該中斷電路之中之該操作電路；

該操作電路，當接收該操作結束訊號時，將可選擇訊號傳至該訊號選擇電路；

該訊號選擇電路，以接收到該可選擇訊號為條件，將成為下一個處理對像之中斷要求訊號傳至該操作電路。

4.如申請專利範圍第 2 項之工作處理裝置，其中，該中斷電路進一步具備保持與中斷求訊號及中斷操作指令相對應之中斷表的記憶體；

該操作電路，參照該中斷表，特定對應該選擇之中斷

要求訊號之中斷操作指令，將該特定之中斷操作指令傳至該工作切換電路。

5.如申請專利範圍第 4 項之工作處理裝置，其中，該記憶體保持該中斷表，以作為一個中斷要求訊號可對應複數個中斷操作指令的表；

該工作切換電路，當從該中斷電路接收之中斷操作指令中指定後續之中斷操作指令存在時，將後續要求訊號傳至該操作電路；

該操作電路，當接收該後續要求訊號時，參照該中斷表，將與該中斷操作指令對應相同中斷要求訊號之後續的中斷操作指令傳至該工作切換電路；

該工作切換電路，當從該中斷電路接收之中斷操作指令中指定後續之中斷操作指令不存在時，在該中斷操作指令之執行結束時，將該操作結束訊號傳至該中斷電路。

6.如申請專利範圍第 1 項之工作處理裝置，其中，該中斷電路，將複數種之中斷要求訊號之中之一部分之高速中斷要求訊號的中斷操作指令傳至該工作切換電路；

該工作切換電路，當接收該高速中斷要求訊號以外之通常中斷要求訊號時，除了將用以執行既定中斷工作之資料載入至該處理暫存器以外，亦藉由該中斷工作執行對應該通常中斷要求訊號的處理。

7.如申請專利範圍第 1 項之工作處理裝置，其進一步具備用以保留該處理暫存器之資料，與複數個工作分別相對應的複數個保留暫存器；

該工作切換電路，當接收該系統呼叫訊號時，將該處理暫存器之資料保留於與該第 1 工作相對應之保留暫存器，將與該第 2 工作相對應之保留暫存器之資料載入至該處理暫存器。

十一、圖式：

如次頁。

該工作切換電路，當接收該系統呼叫訊號時，將該處理暫存器之資料保留於與該第 1 工作相對應之保留暫存器，將與該第 2 工作相對應之保留暫存器之資料載入至該處理暫存器。

十一、圖式：

如次頁。

圖 1

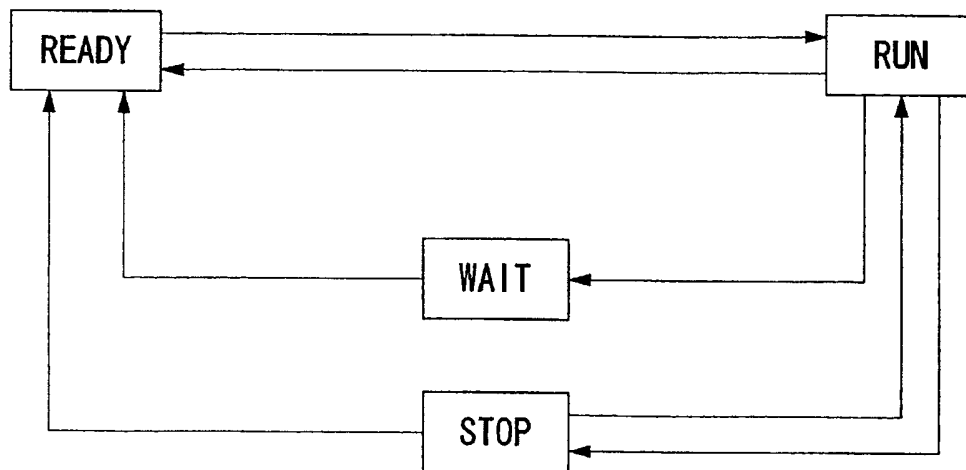


圖 2

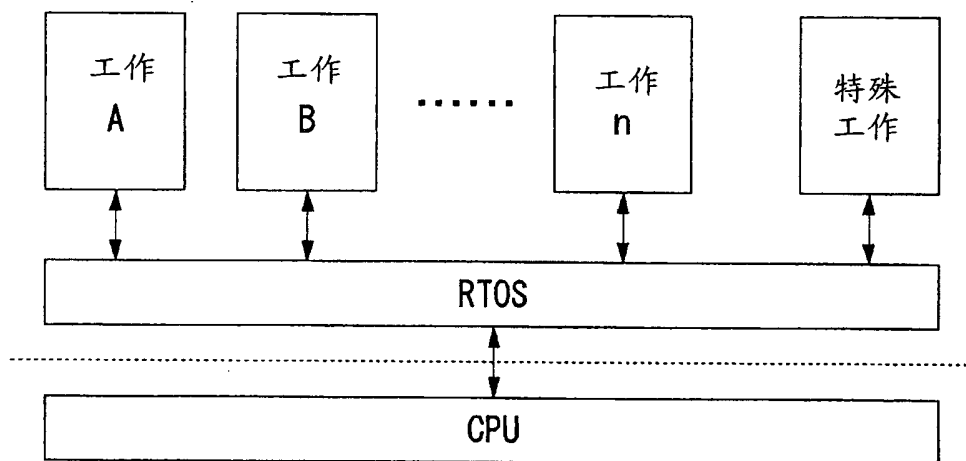


圖3

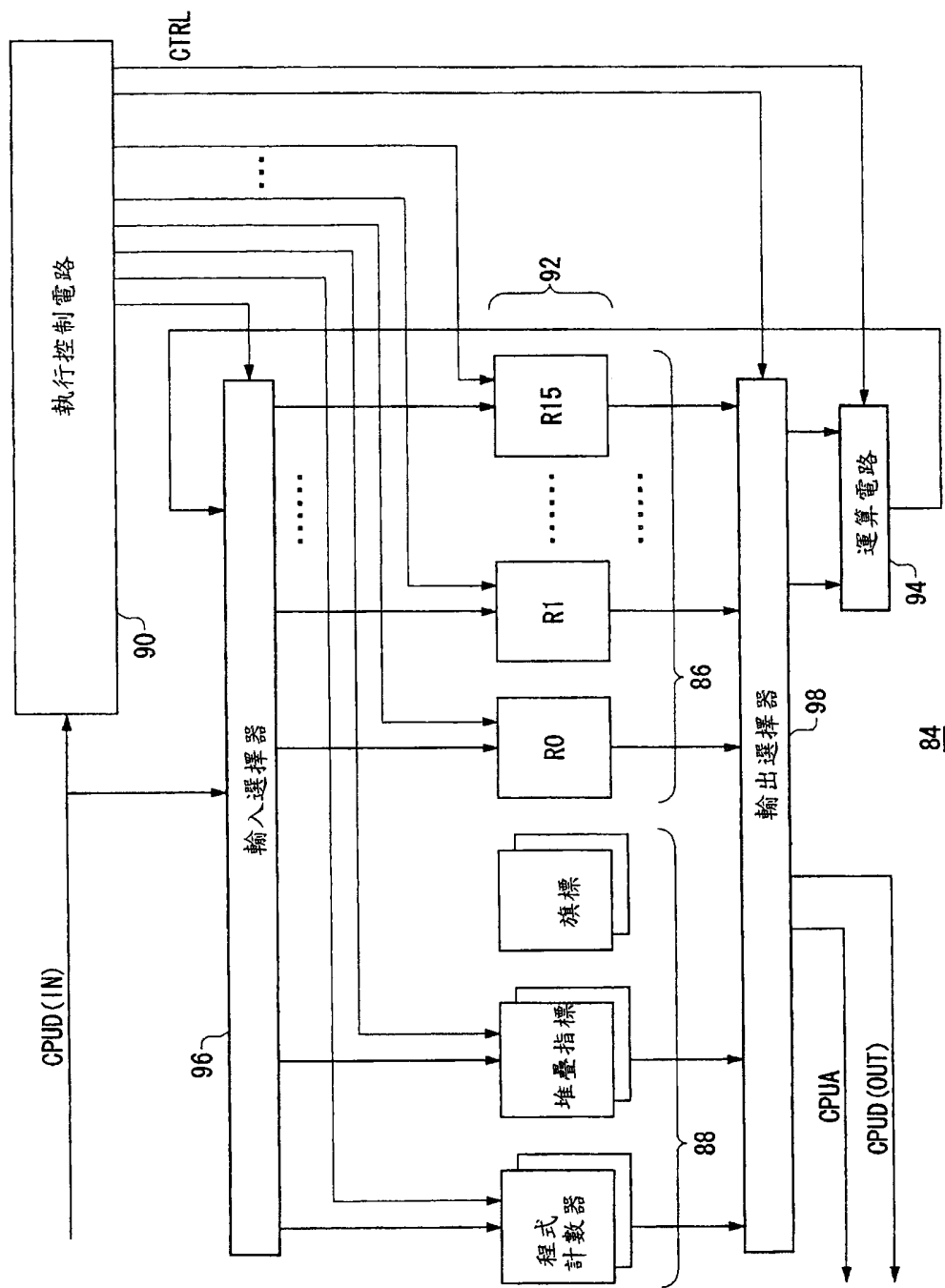


圖4

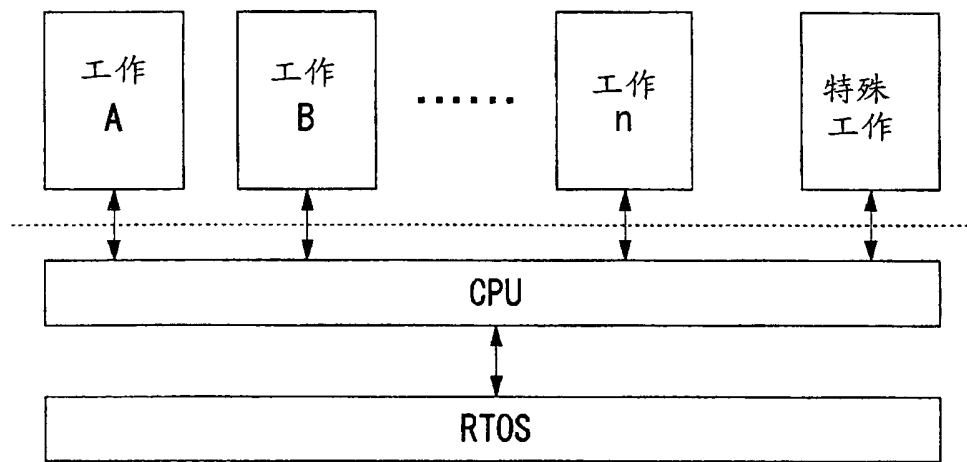


圖5

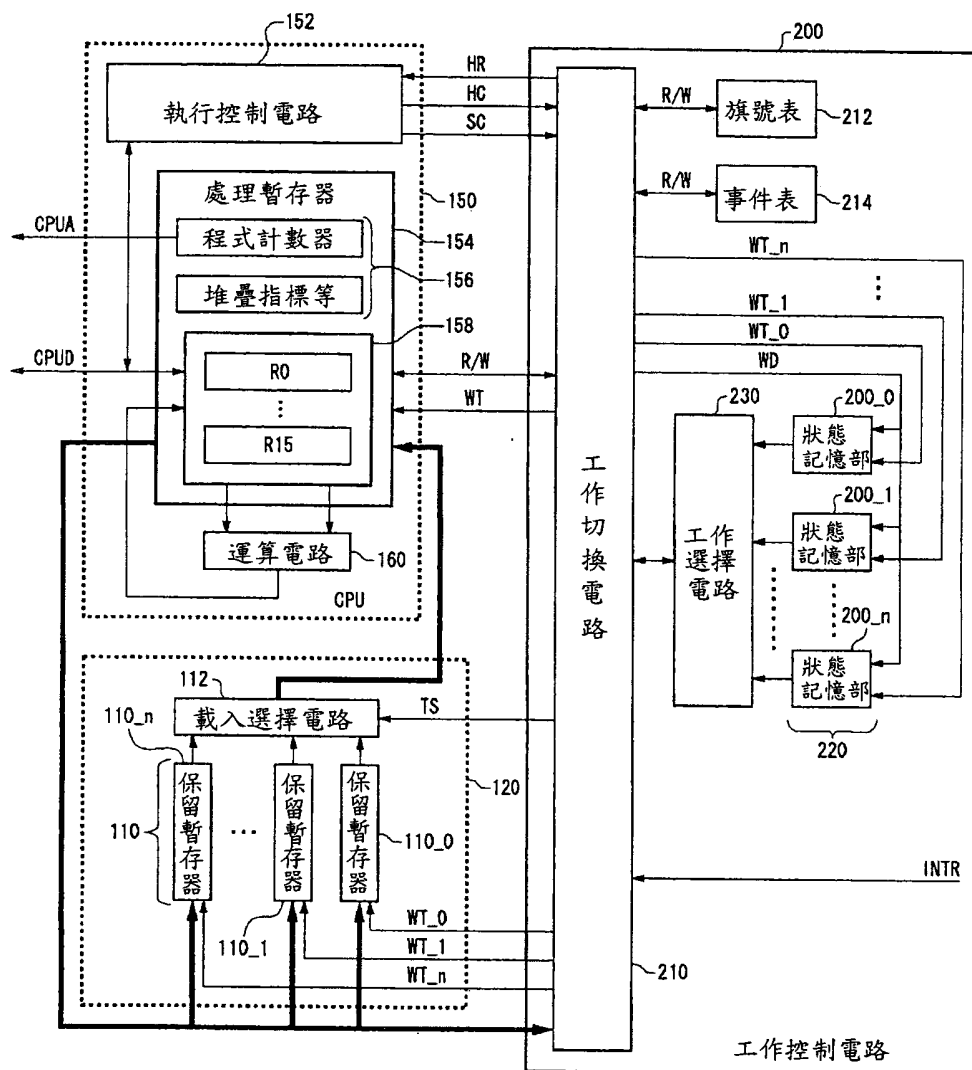


圖6

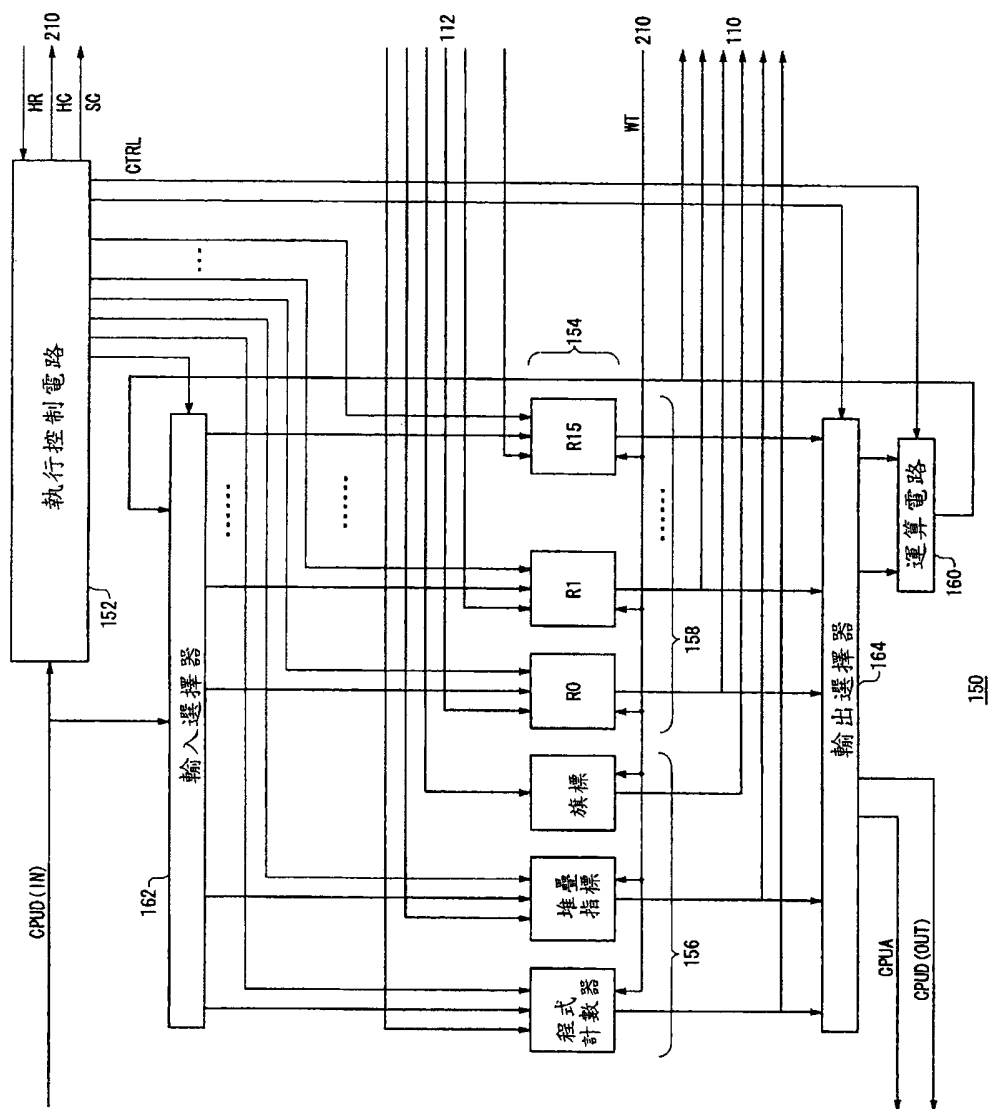


圖 7

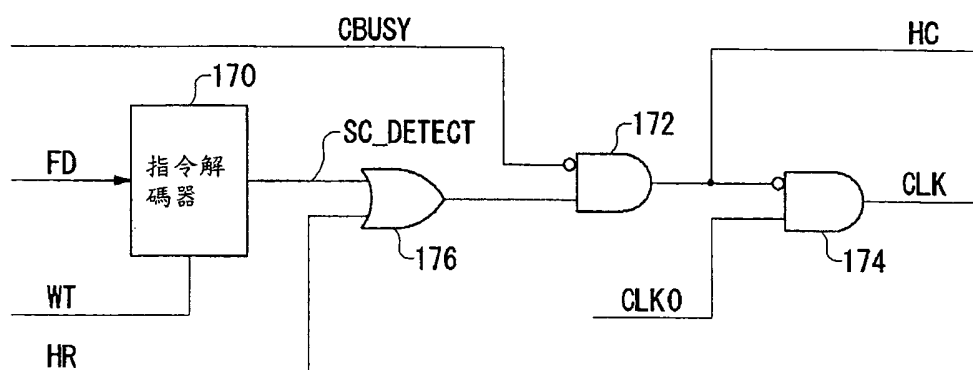


圖8(a)

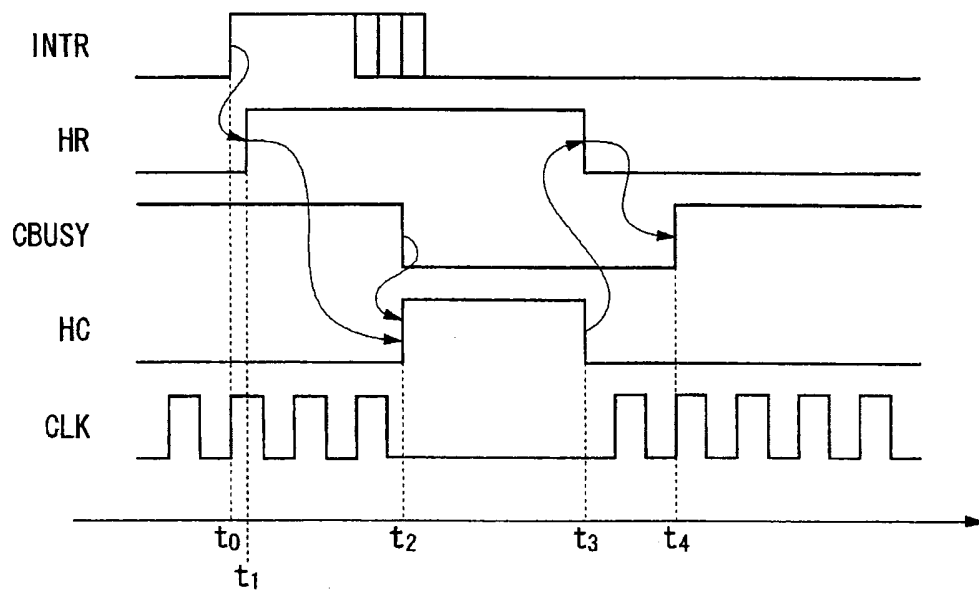


圖8(b)

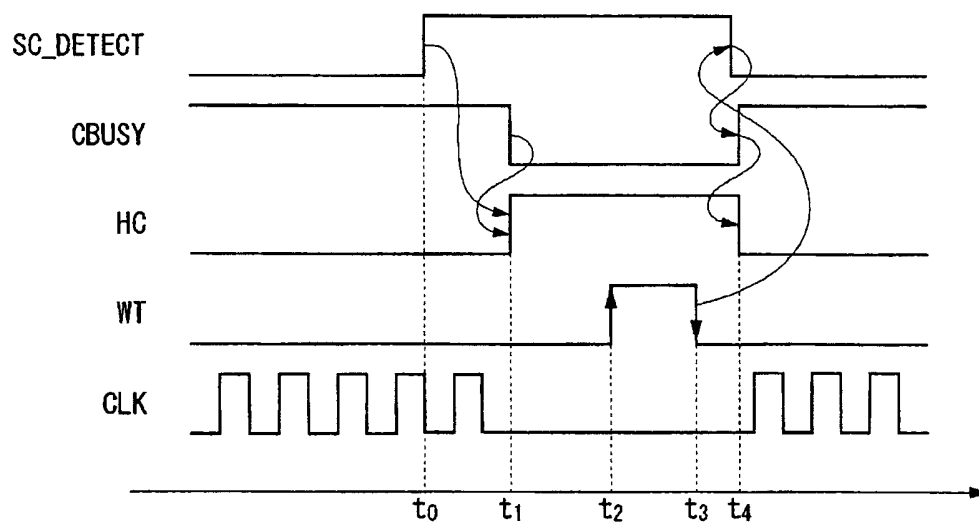


圖10

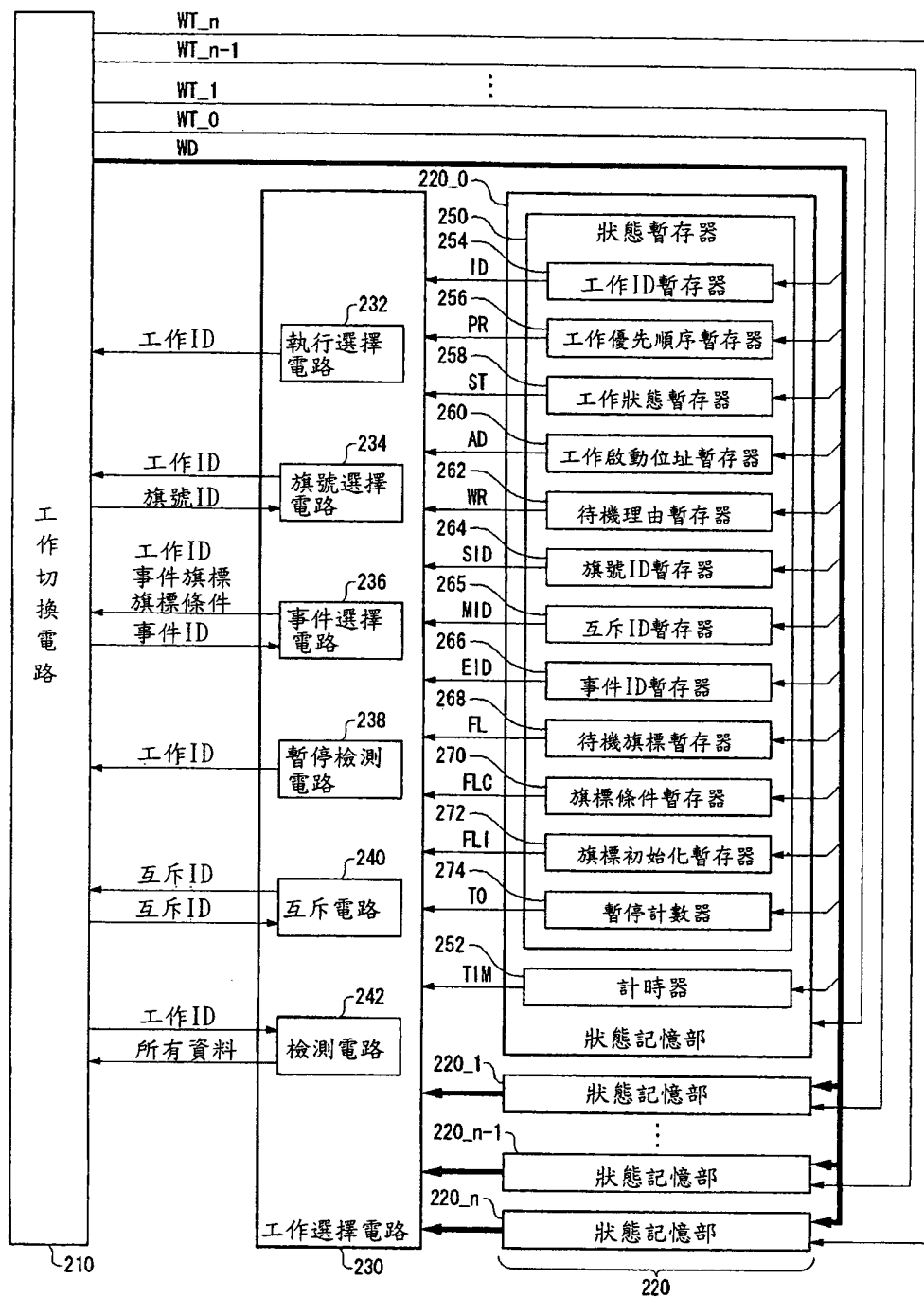


圖11

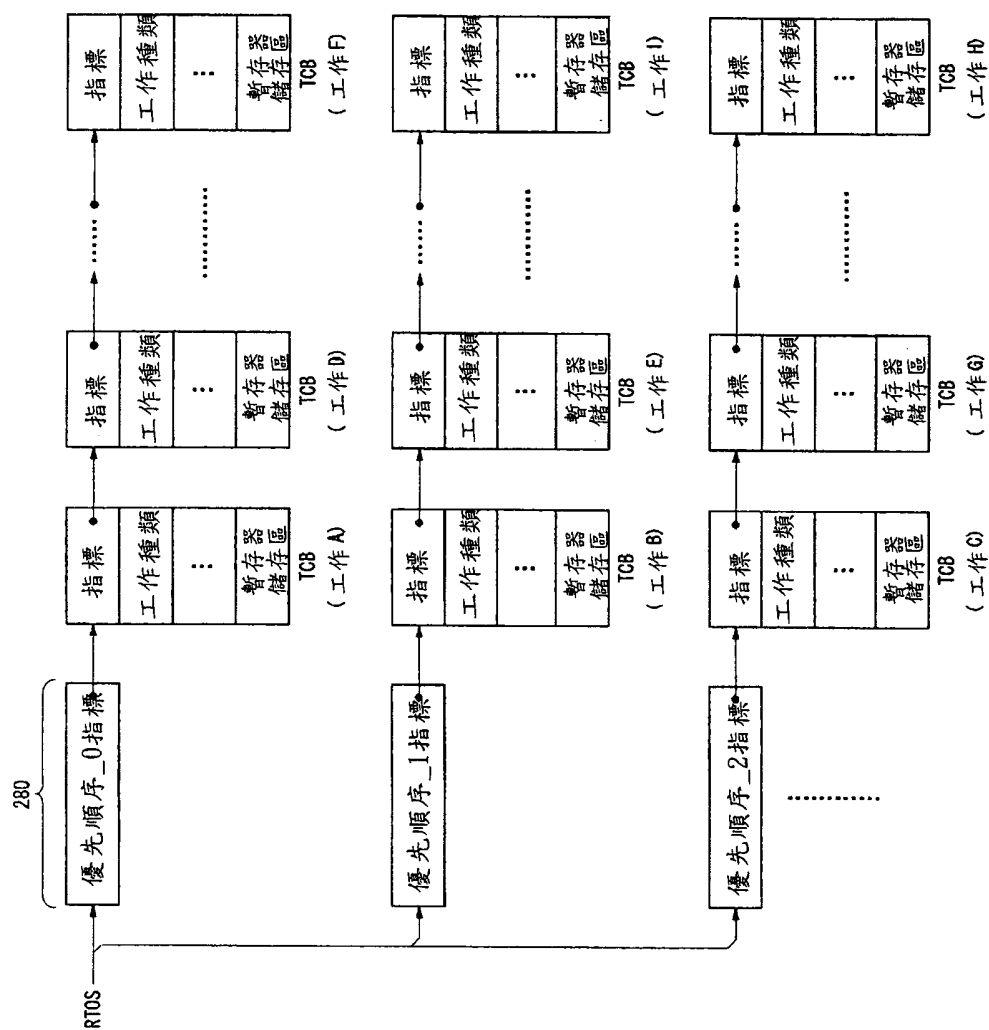


圖12

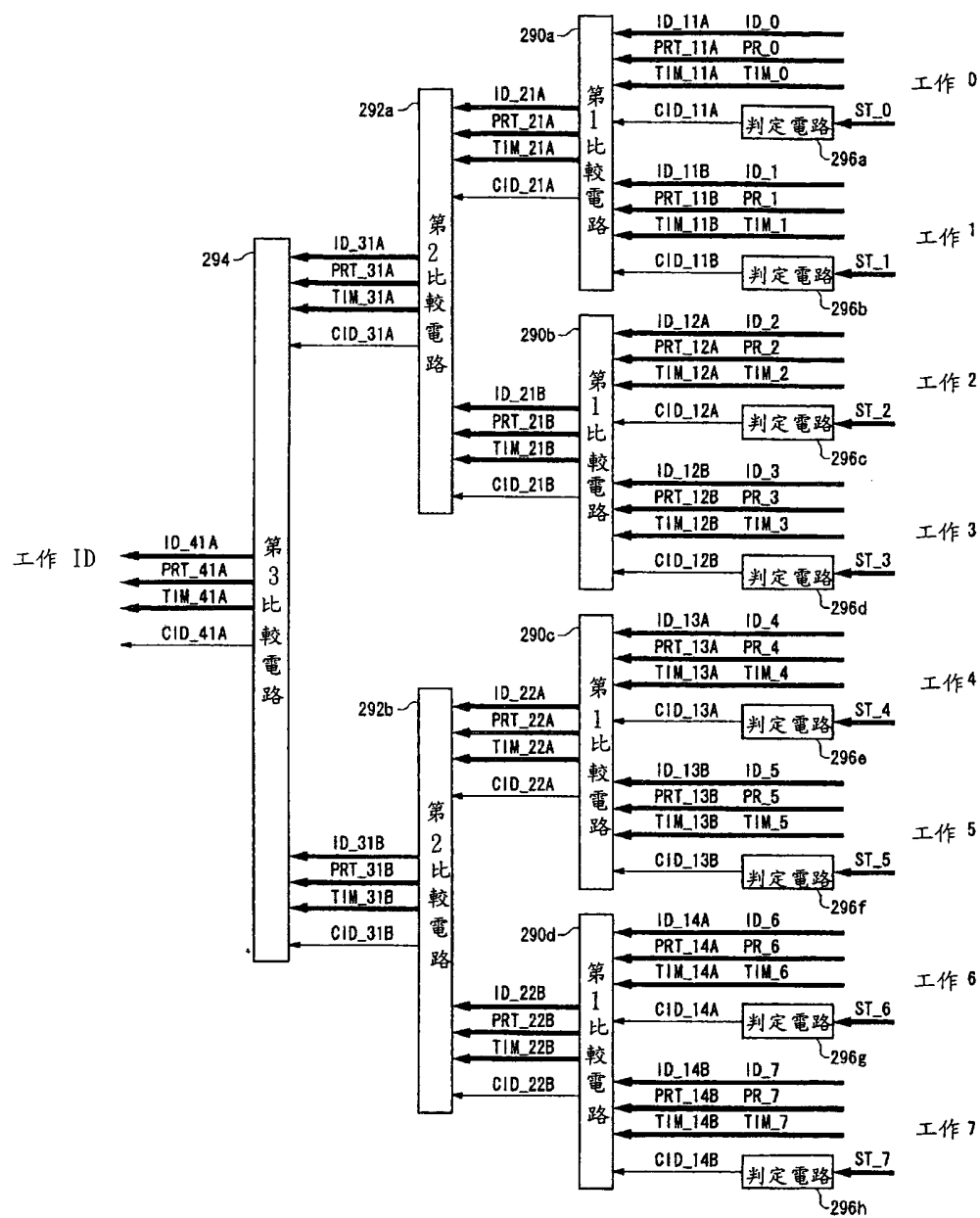


圖13

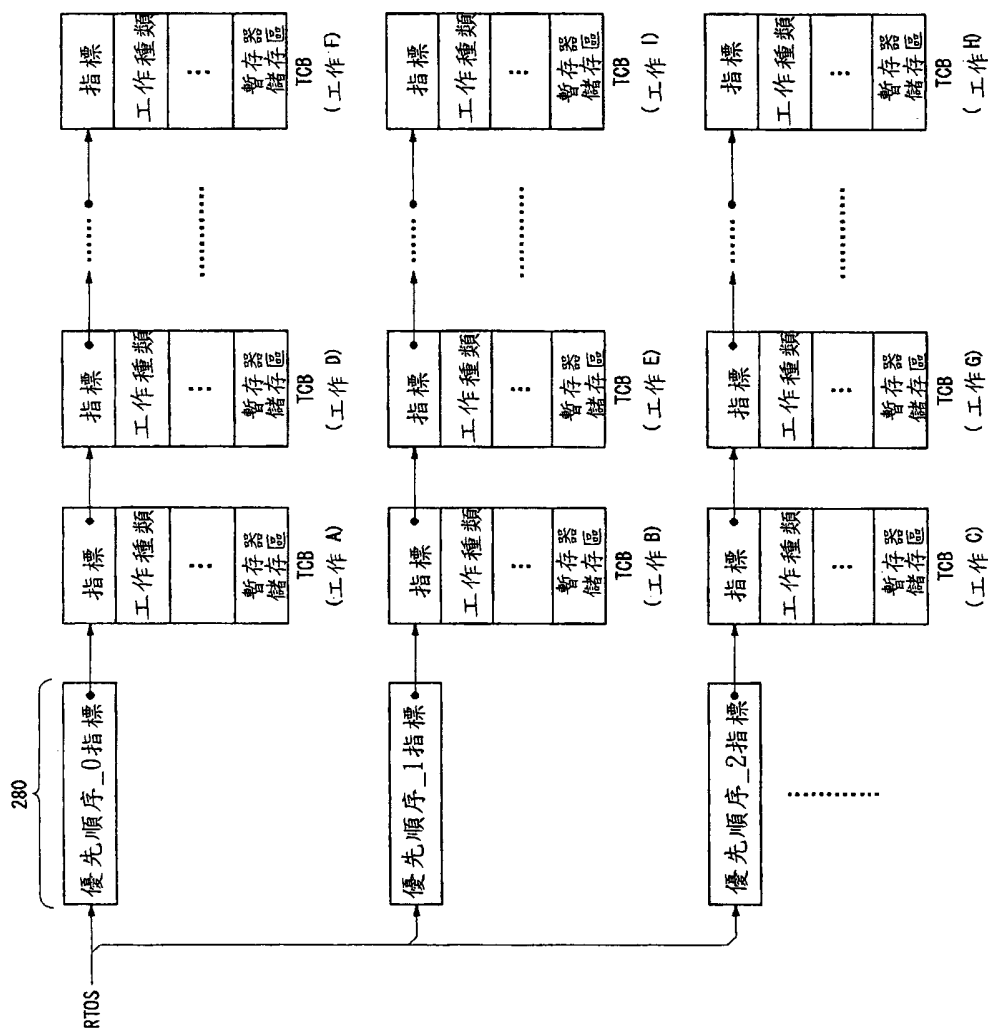


圖 14

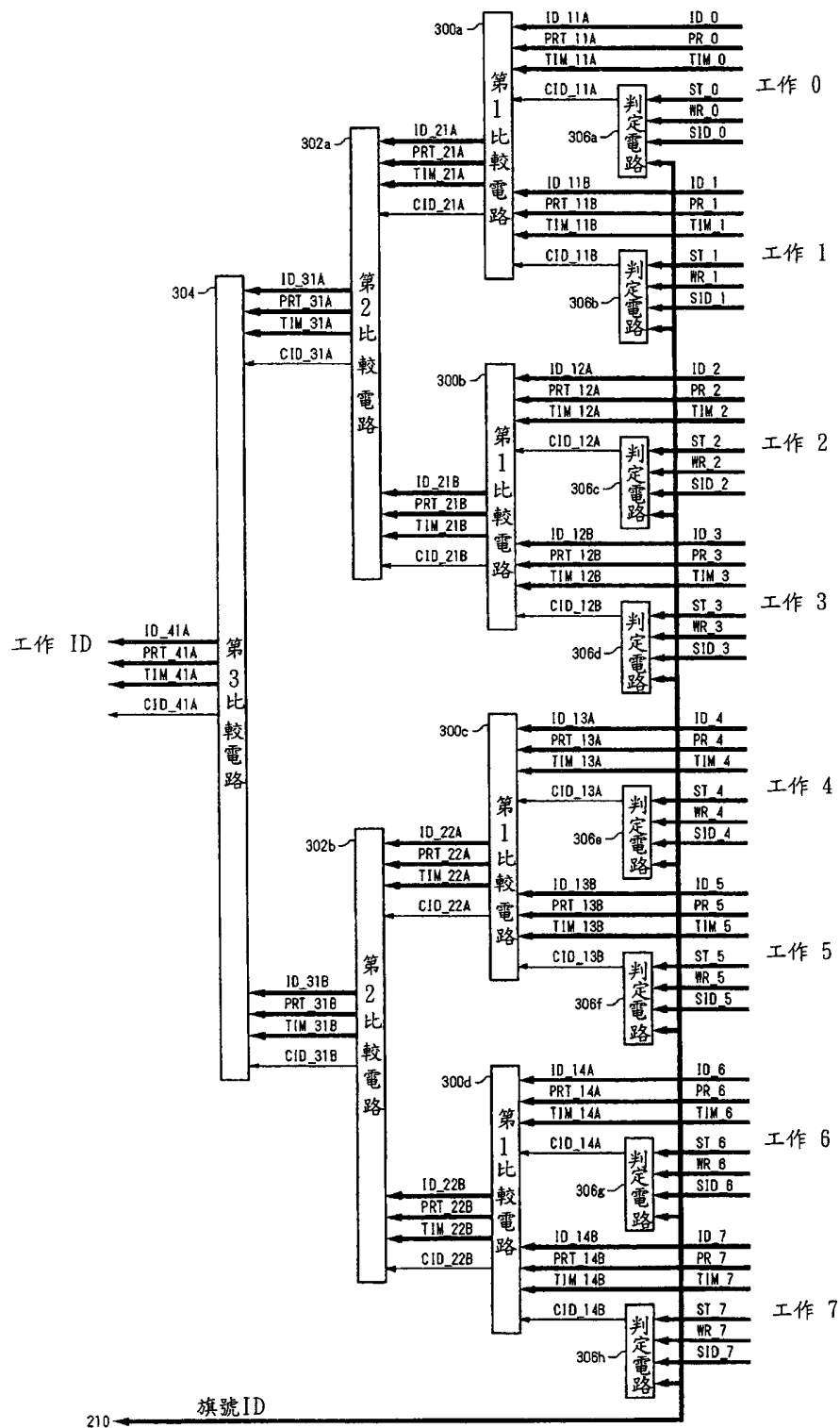


圖15

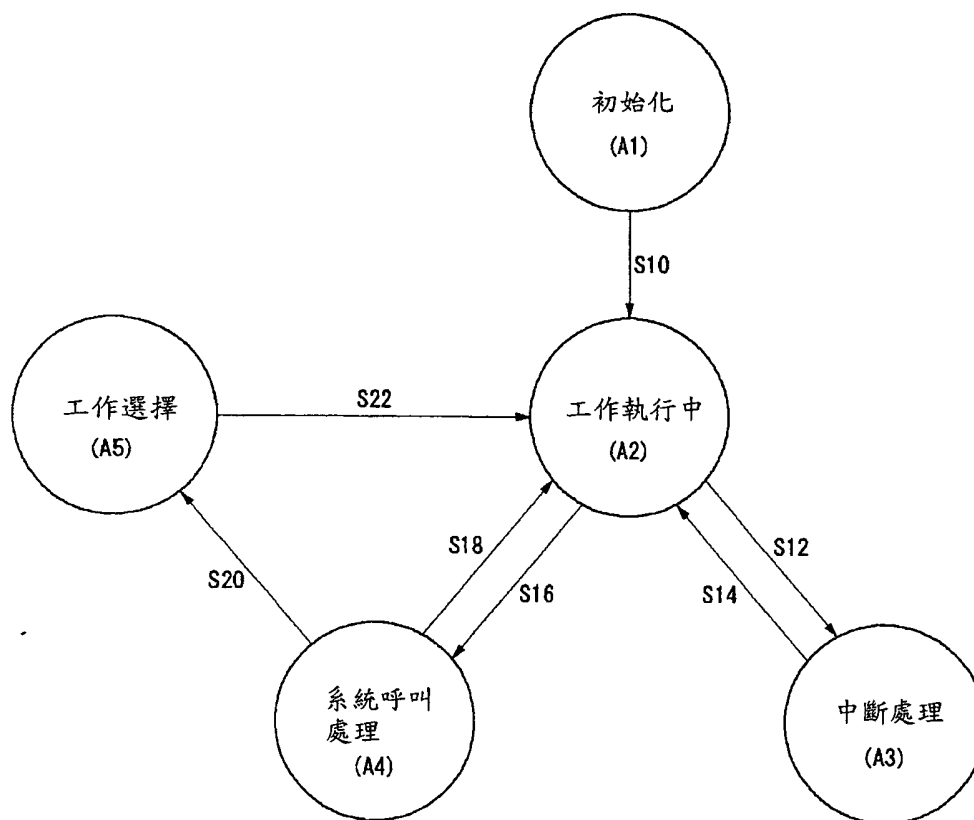


圖16

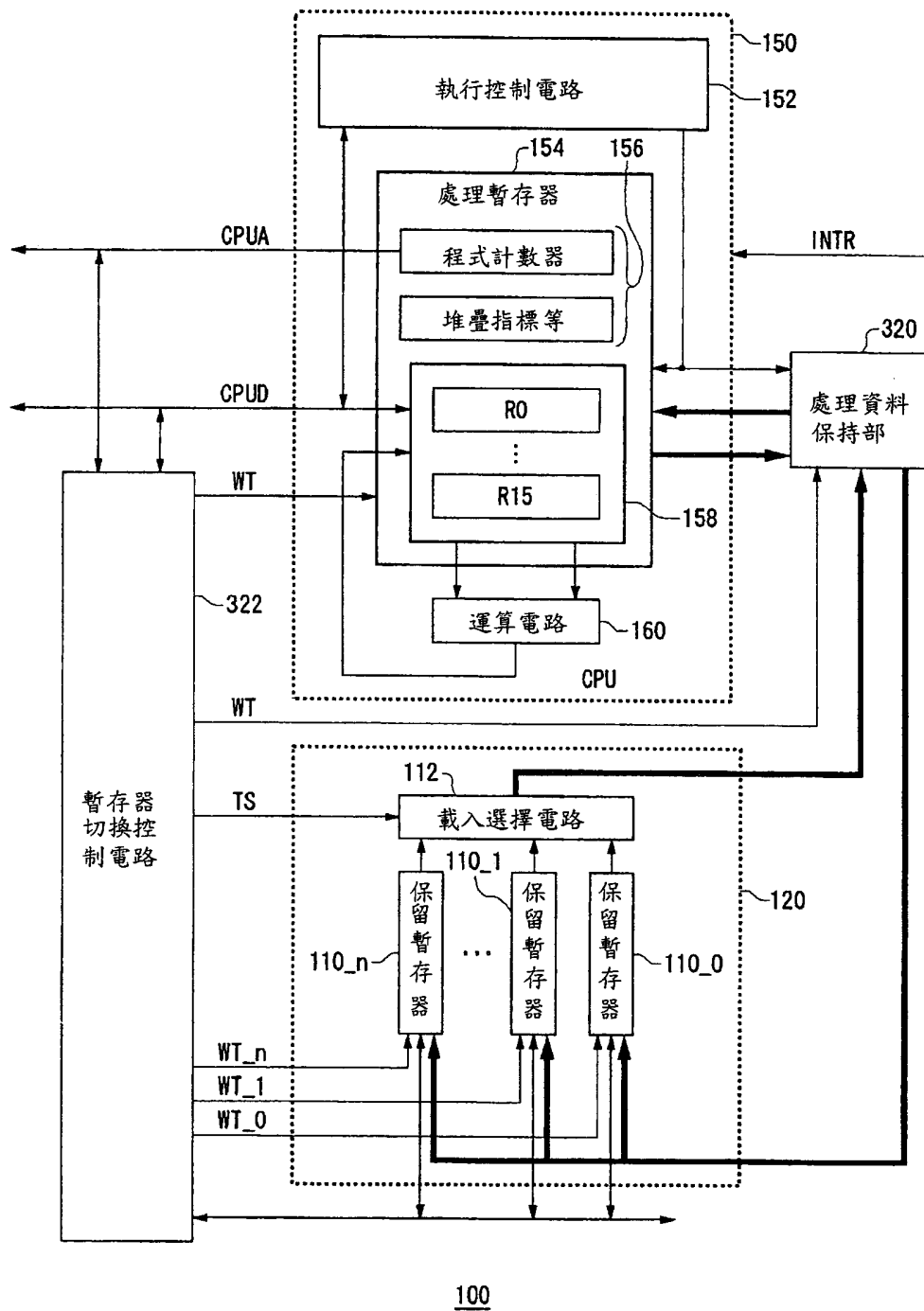


圖17

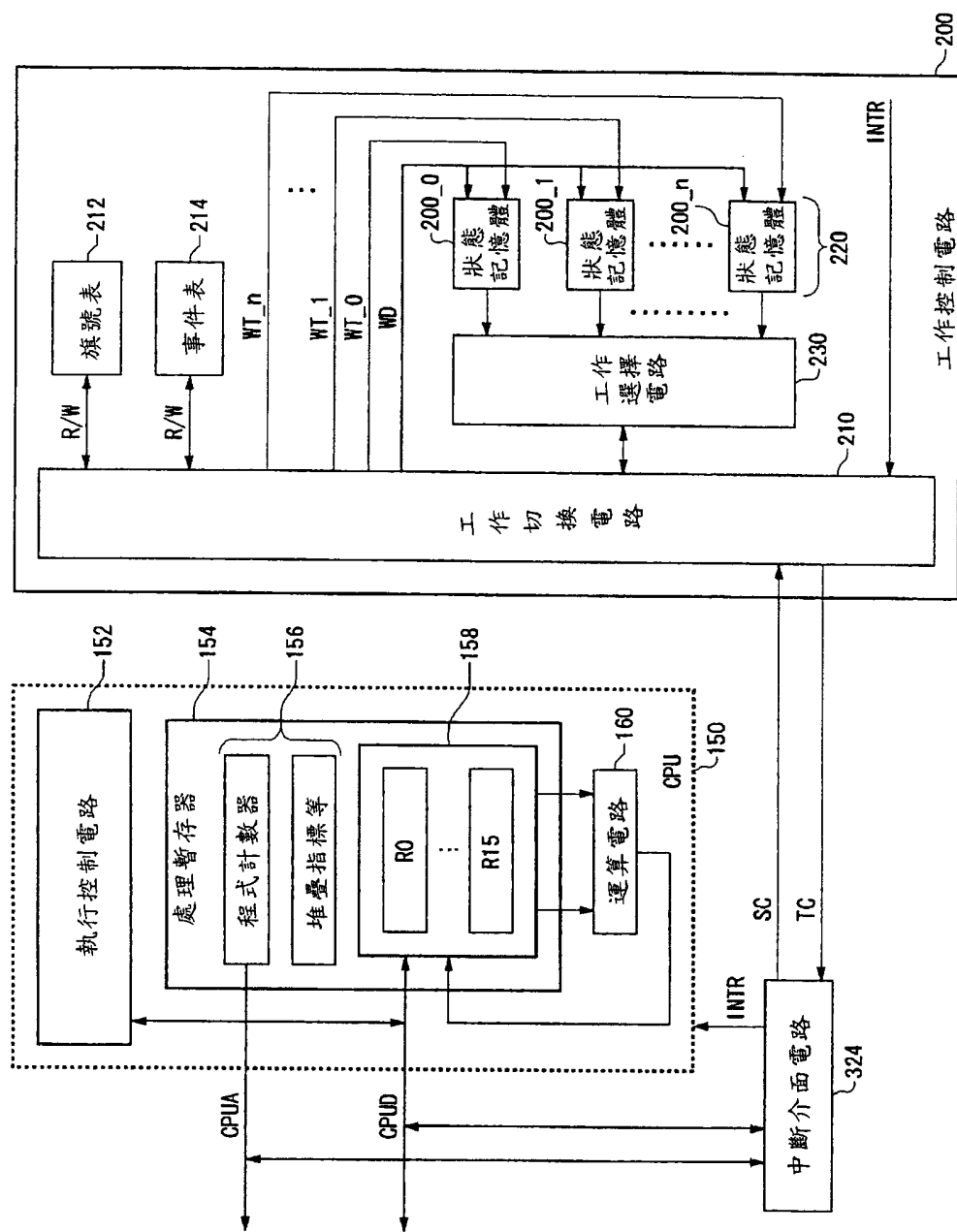


圖 18

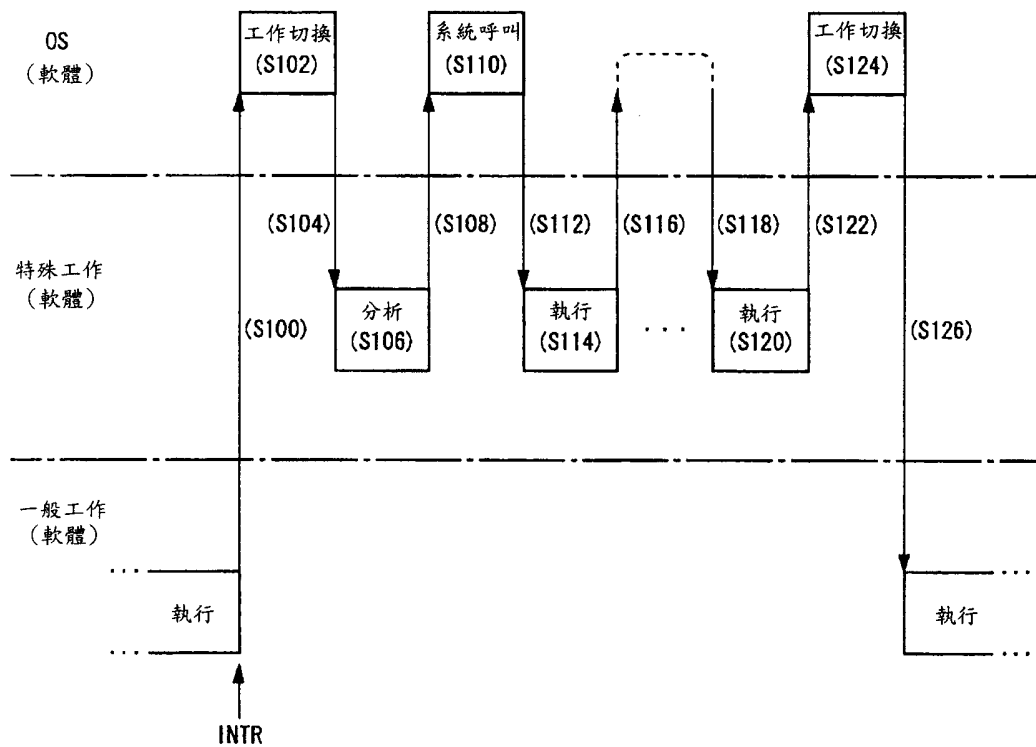


圖 19

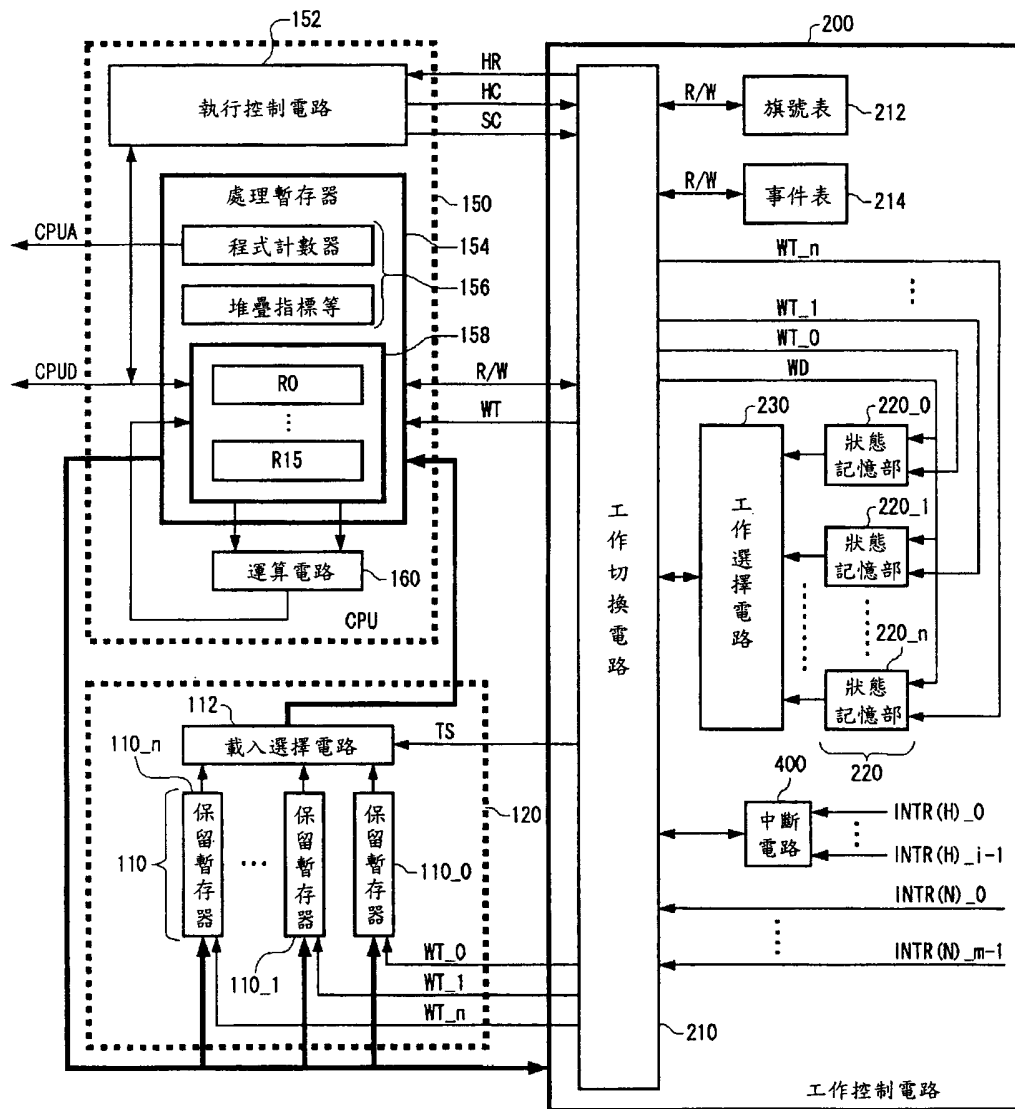


圖20

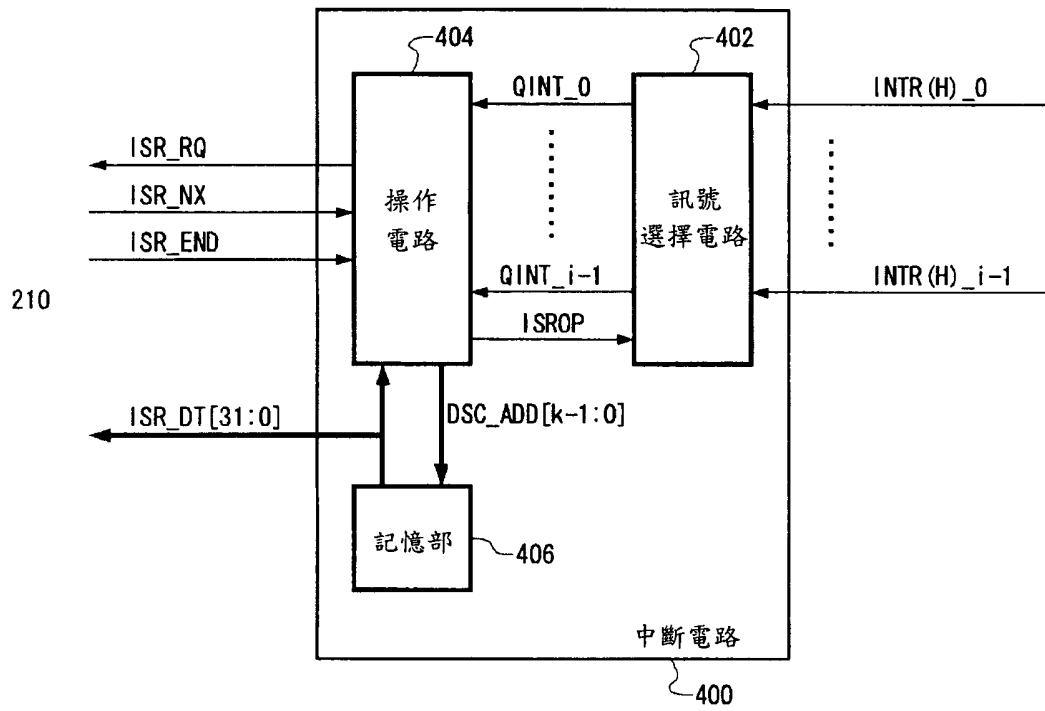


圖 21

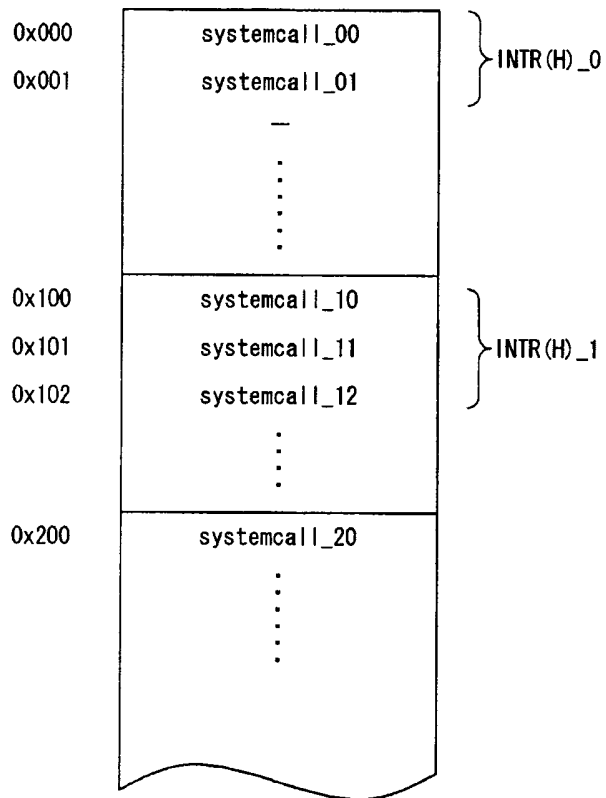


圖22

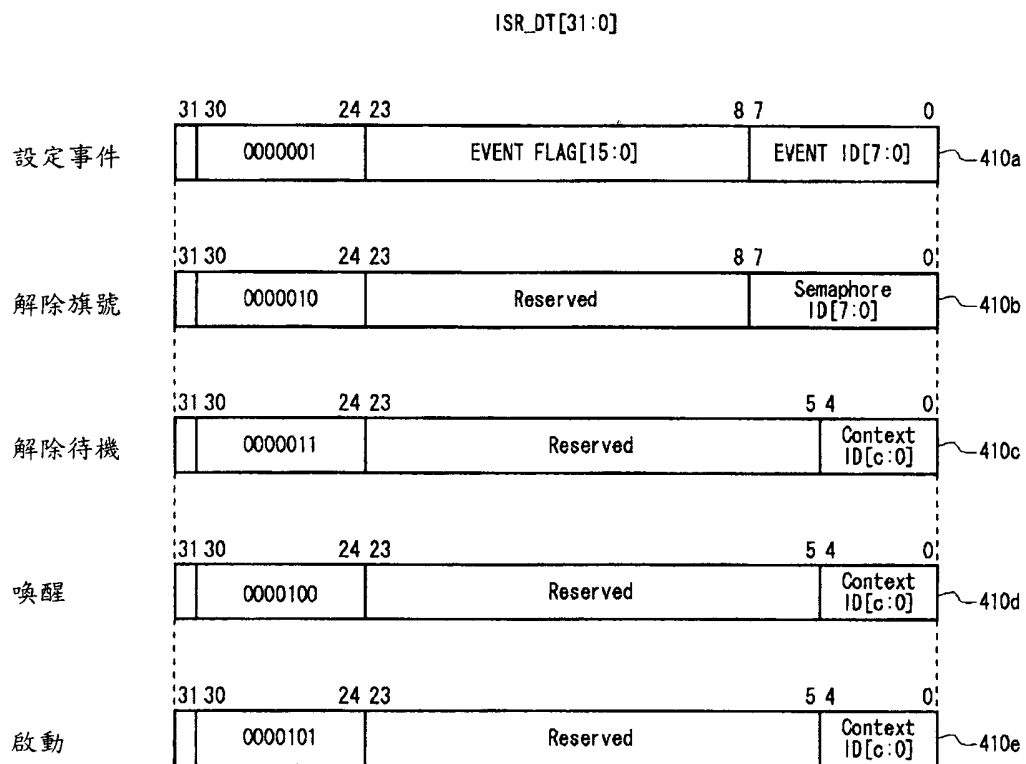


圖 23

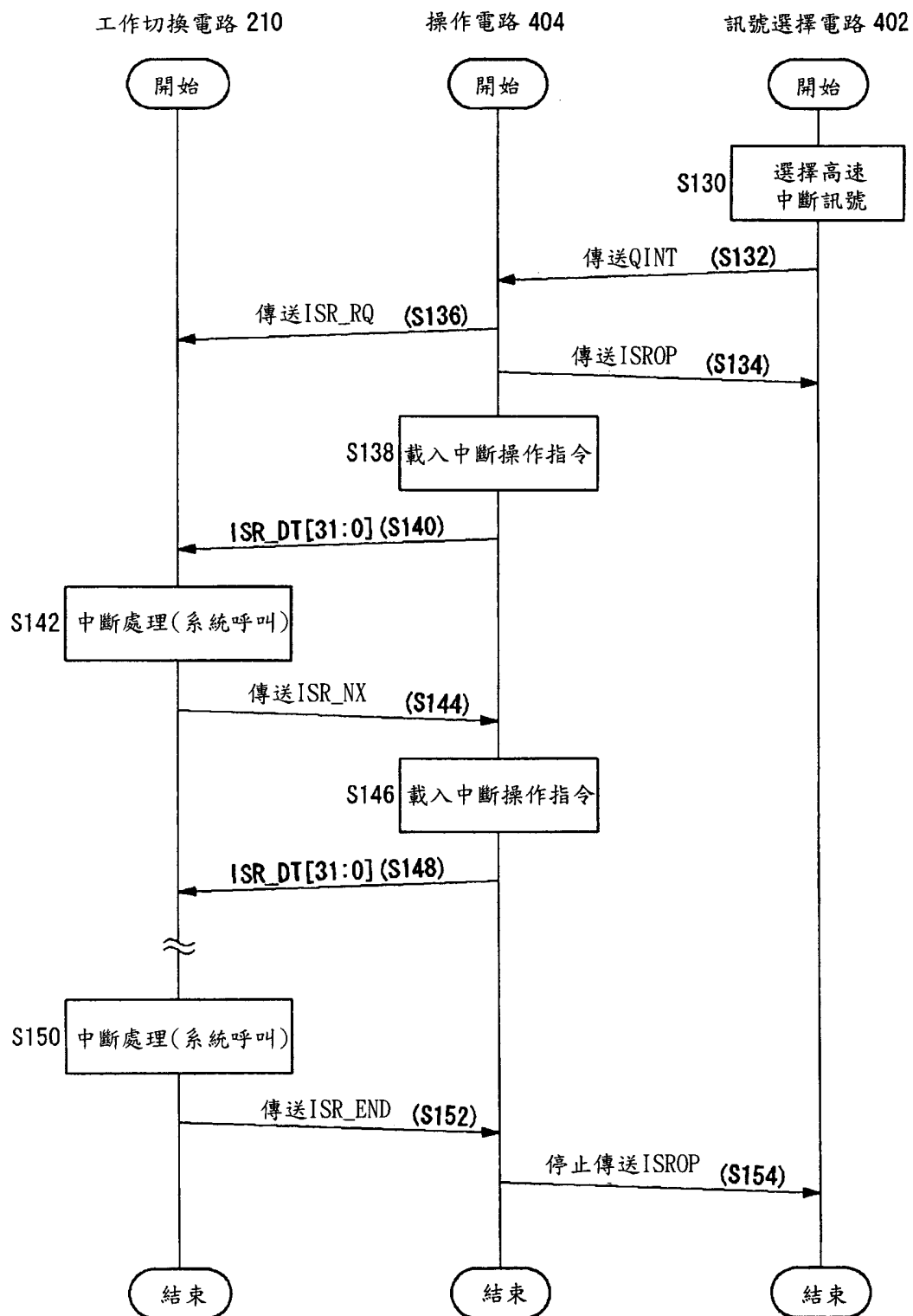


圖 24

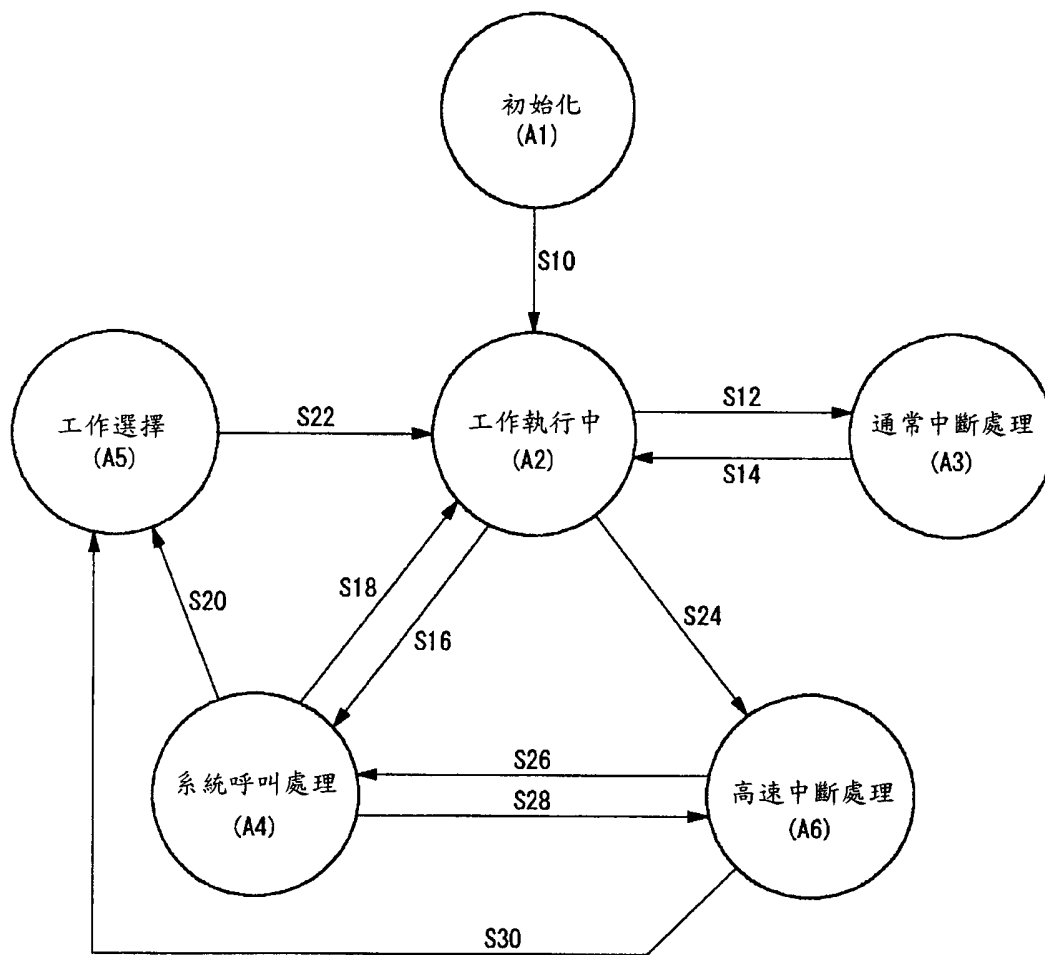
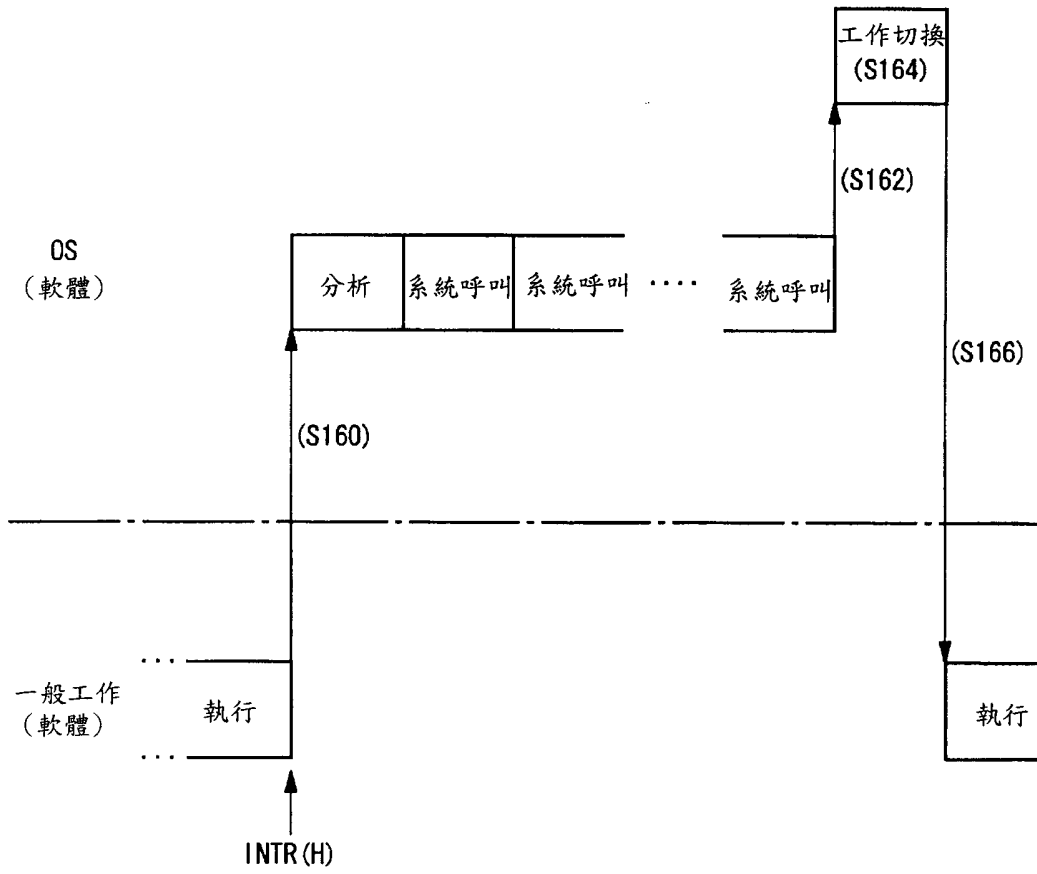


圖25



七、指定代表圖：

(一)本案指定代表圖為：第 (20) 圖。

(二)本代表圖之元件符號簡單說明：

210 工作切換電路

400 中斷電路

402 訊號選擇電路

404 操作電路

406 記憶部

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

(無)