

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：

美國；西元 2004 年 3 月 22 日；10/808,716

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：

本發明係與以下申請案相關：於 2003 年 9 月 30 日申請，標題為「計數指令執行及資料存取之方法及裝置」之申請案，申請案號為 10/675,777，律師登錄號 AUS920030477US1；於 2003 年 9 月 30 日申請，標題為「選擇性計數指令及資料存取之方法及裝置」之申請案，申請案號為 10/674,604，律師登錄號 AUS920030478US1；於 2003 年 9 月 30 日申請，標題為「在標記指令之執行及存取已標記之記憶體位置後產生中斷之方法及裝置」之申請案，申請案號為 10/675,831，律師登錄號 AUS920030479US1；於 2003 年 9 月 30 日申請，標題為「計數已超過一臨界值之資料存取及指令執行之方法及裝置」之申請案，申請案號為 10/675,776，律師登錄號 AUS920030481US1；於 2003 年 9 月 30 日申請，標題為「個別指令及記憶體位置之偵錯支援之方法及裝置」之申請案，申請案號為 10/675,751，律師登錄號 AUS920030482US1；於 2003 年 9 月 30 日申請，標題為「自主性選擇指令供選擇計數之方法及裝置」之申請案，申請案號為 10/675,721，律師登錄號 AUS920030483US1；於 2003 年 9 月 30 日申請，標題為「自主性計數應用程式之指令執行之方法及裝置」之申請案，申請案號為 10/674,642，律師登錄號 AUS920030484US1；於 2003 年 9 月 30 日申請，標題為「自主性接受所指定之指令之例外之方法及裝置」之申請案，申請案號為 10/674,606，律師登錄號 AUS920030485US1；於 2003 年 9 月 30 日申請，標題為「自主性概化應用程式之方法及裝置」之申請案，申請案號為 10/675,783，律師登錄號 AUS920030486US1；

師登錄號 AUS920040065US1；以上所有相關之申請案皆讓與相同之受讓人，在此併入本案供參考。

【發明所屬之技術領域】

本發明係關於一種改良的資料處理系統。具體而言，本發明提供一種方法及裝置，供取得一資料處理系統中的效能資料。更具體而言，本發明提供一種方法及裝置，以硬體協助軟體工具，取得一資料處理系統中的程式碼涵蓋資料。

【先前技術】

對於分析並增強一資料處理系統及執行於此資料處理系統內之應用程式之效能而言，知道資料處理系統內的哪一軟體模組，正在使用系統資源係有幫助的。資料處理系統之有效管理及改進，需要知道許多系統資源係如何的被使用，以及何時正在被使用。利用效能工具來監控及檢測一資料處理系統，以決定資料處理系統內所執行的許多軟體應用程式的資源耗損。舉例而言，一效能工具可辨識在一資料處理系統中，執行最頻繁的模組及指令，或者可辨識分配最大量的記憶體或運作最多 I/O 請求之模組。硬體效能工具可內建於系統中，或者隨後再增加。

習知的軟體效能工具之一為一追蹤工具。追蹤工具可使用不只一種的方法，來提供顯示一正在執行的程式之執行流的追蹤資訊。藉由隨著某些事件的發生，將其記錄下來，係追蹤指令的特定順序的一門技術。此技術稱為一事件導向的概要技術。舉例而言，追蹤工具可記錄每一個進

入或離開一模組、次常式、方法、功能、或系統元件的項目。或者，追蹤工具可記錄每一記憶體分配請求之請求者及所分配之記憶體大小。通常，每一事件會產生一時間戳記記錄。亦可使用類似項目離開記錄的相對應記錄對，追蹤程式碼的任何區段之執行、I/O 或資料傳輸之開始及結束、及許多其他事件。

為了改進許多系列的電腦所產生的程式碼之效能，通常需要決定處理器，在執行程式碼時，係將時間花在何處，在電腦處理技藝中此技術稱為找尋「熱點」。理想地，為了專注於最受益於程式碼之改進的部分，會想將此熱點區隔在程式碼之指令與/或原程式行階層。

另一種追蹤技術，涉及週期性地採樣一程式的執行流，以辨識在程式中，程式碼看似花費大量時間所在的地點。此技術係根據週期性地在固定時段，中斷應用程式或資料處理系統執行之觀念。此技術稱為採樣導向概要技術。在每一中斷，會在一既定時間內或一既定事件數記錄資訊。舉例而言，可在每一時段，記錄現正執行中的資料緒之程式計數器（即正在概化的一較大程式之可執行部分）。這些數值可在後處理階段，根據資料處理系統之一載量圖與符號表資訊解析，且可從此分析中取得時間花費所在的概要。

以當今可使用的軟體工具決定測試範圍係一困難的問題。當測試程式的程式碼時，現在沒有簡單的機制可決定哪些程式碼在被執行，哪些不在。此類型的辨識亦稱為程式碼涵蓋範圍（code coverage）。現有辨識程式碼涵蓋範圍之軟體技術及工具。然而，這些技術及工具通常涉及改變測試中的程式碼。由於此程序可能很冗長且費時，要

改變或調整程式碼係不理想的。其他類型的技術及工具係利用程式的靜態分析，產生測試情況。此類型的方法論並不能保證測試時的完整涵蓋範圍。

建立類似這類的工具，以找尋有關特定情況或問題的答案，可能要花很多精力，且由於其軟體工具本身在測試時亦影響系統，因此可能非常難標準化。本發明認為硬體協助的工具開發及問題分析可大量的減少所需花費在開發軟體效能工具的精力。再者，由於處理器的密度漸增，可利用硬體協助來提供額外的偵錯及分析功能。

因此，具有一種改進的方法、裝置、及電腦指令，提供效能工具分析資料處理系統中所執行的程式碼的涵蓋範圍之硬體協助係有益的。

【發明內容】

本發明提供一種方法、裝置、及電腦指令，供顯示程式碼的涵蓋資料。取得此涵蓋資料，其係包含與程式碼相關之指令存取指示符。辨識特定指令存取指示符，以形成已設指令存取指示符，其中特定指令存取指示符係由在資料處理系統中之處理器所設定之特定指令存取指示符，以回應該處理器執行該程式碼。每一已設指令存取指示符係與程式碼之一部份相關聯。產生此涵蓋資料之一顯示。此已設指令存取指示符係於其顯示中被辨識。其顯示可以是一圖像顯示。

【實施方式】

參考圖 1，顯示本發明可實施於其中之一資料處理系統的方塊圖。客戶端 100 係一電腦的範例，為實施本發明

之程序之程式碼或指令可能之所在。客戶端 100 實行一週邊元件互連(PCI)本區匯流排架構。雖然所描述的範例實行一 PCI 匯流排，亦可使用如圖形加速連接埠(AGP)以及工業標準架構(ISA)之其他匯流排架構。經 PCI 橋接器 108，處理器 102 以及主記憶體 104 與 PCI 本區匯流排 106 連結。PCI 橋接器 108 亦可包含供處理器 102 使用之一整合型記憶體控制器以及快取記憶體。經由元件直接互連或擴充卡，可額外的連結至 PCI 本區匯流排 106。

在描述的範例中，區域網路(LAN)配接器 110、小型電腦系統介面 SCSI 主匯流排配接器 112、以及擴充匯流排介面 114，以元件直接連結的方式，與 PCI 本區匯流排 106 連結。相反地，音效配接器 116、圖像配接器 118、以及音效/影像配接器 119，係藉由將擴充卡插入擴充槽的方式，與 PCI 本區匯流排 106 連結。擴充匯流排介面 114 提供鍵盤以及滑鼠配接器 120、數據機 122、以及額外的記憶體 124 的連結。SCSI 主匯流排配接器 112 提供硬碟機 126、磁帶機 128、以及光碟機 130 的連結。一般 PCI 本區匯流排實作會支援三或四個 PCI 擴充槽或擴充連結器。

一作業系統係執行於處理器 102，並用以協同及提供圖 1 中之資料處理系統 100 內許多元件的控制。作業系統可為一商用作業系統，例如微軟公司所提供之 Windows XP。如 Java 之一物件導向程式系統可與作業系統一同執行，並自執行於客戶端 100 之 Java 程式或應用程式，提供呼叫至作業系統。「Java」為昇陽公司之商標。作業系統、物件導向程式系統、以及應用軟體或程式的指令，係位於如硬碟機 126 之儲存裝置中，並且可載入主記憶體

104 供處理器 102 執行。

熟此技藝者皆知，圖 1 中的硬體，可能會因不同實作而有所改變。其他內建硬體或週邊裝置，例如快閃唯讀記憶體(ROM)、非揮發性記憶體類似者、或光碟機及其類似者，可額外使用，或代替圖 1 所描述的硬體。本發明之程序亦可應用於多處理器資料處理系統。

舉例而言，若選擇性地配置為一網路電腦，客戶端 100 可能不包含 SCSI 主匯流排配接器 112、硬碟機 126、磁帶機 128、以及光碟機 130。在此情況下，電腦(正確地稱為客戶端電腦)包含某種網路通訊介面，例如 LAN 配接器 110、數據機 122、或其類似者。如另一範例，客戶端 100 可為一單獨系統，不論是否包含某種網路通訊介面，配置客戶端 100 不需某種網路通訊介面即可啟動。如另一範例，客戶端 100 可能為一個人數位助理(PDA)，與 ROM 且/或快閃 ROM 配置，以供非揮發性記憶體，儲存作業系統檔且/或使用者所產生的資料。圖 1 中所描述的範例以及上述的範例並不暗意結構上的限制。

本發明之程序係經處理器 102，使用電腦實施的指令運作，其指令可能位於記憶體，如舉例而言，主記憶體 104、記憶體 124、或於一或多個週邊裝置 126-130。

參考圖 2，描述本發明之一較佳實施例，供處理資訊之一處理器系統之方塊圖。處理器 210 可如圖 1 中的處理器 102 般的實施。

在一較佳實施例中，處理器 210 為一單一積體電路超微處理器。因此，如下詳述，處理器 210 包含許多單元、暫存器、緩衝器、記憶體、以及其他區段，皆由積體電路所組成。在此較佳實施例中，處理器 210 亦根據精簡指令

集(「RISC」)運算技術運作。如圖 2 所示，系統匯流排 211 與處理器 210 之一匯流排介面單元(「BIU」)212 連結。BIU 212 控制處理器 210 及系統匯流排 211 間的資料轉移。

BIU 212 與處理器 210 之一指令快取 214 以及資料快取 216 連結。指令快取 214 輸出指令至順序單元 218。為回應指令快取 214 的此指令，順序單元 218 選擇性地輸出指令至處理器 210 的其他執行電路。

除順序單元 218 外，在此較佳實施例中，處理器 210 之執行電路包含許多用來執行指令的處理器單元。這些處理器單元亦稱為執行單元。這些執行單元包含，例如，一支單元 220、一定點單元 A(「FXUA」)222、一定點單元 B(「FXUB」)224、一複合式定點單元(「CFXU」)226、一載入/儲存單元(「LSU」)228；以及一浮點單元(「FPU」)230。FXUA 222、FXUB 224、CFXU 226、以及 LSU 228 自一般用結構暫存器(「GPRs」)232 以及定點更名緩衝器 234 輸入其來源運算元資訊。再者，FXUA 222 以及 FXUB 224 自一進位位元(「CA」)暫存器 239 輸入一「進位位元」。FXUA 222、FXUB 224、CFXU 226、以及 LSU 228 輸出其運算元之結果(目的運算元資訊)，以儲存所選項目定點更名緩衝器 234 內。CFXU 226 亦至且自特殊目的暫存處理單元(「SPR 單元」)237 輸入及輸出來源運算元資訊以及目的運算元資訊。

FPU 230 自浮點結構暫存器(「FPRs」)236 以及浮點更名緩衝器 238 輸入其來源運算元資訊。FPU 230 輸出其運算元之結果(目的運算元資訊)，以儲存所選項目於浮點更名緩衝器 238 中。

為回應一載入指令，LSU 228 自資料快取 216 輸入資

訊，並且複製此資訊至更名緩衝器 234 及 238 之一選取者。若此資訊不儲存於資料快取 216，則資料快取 216 自連結系統匯流排 211 的系統記憶體 260(經 BIU 222 及系統匯流排 211)輸入資訊。再者，資料快取 216 可自資料快取 216(經 BIU 222 及系統匯流排 211)輸出資料至連結系統匯流排 211 的系統記憶體 260。為回應一儲存指令，LSU 228 自 GRPRs 232 及 FPRs 236 中所選取者輸入資訊，並且複製此資訊至資料快取 216。

順序單元 218 至且自 GPRs 232 及 FPRs 236 輸入及輸出資訊。自順序單元 218，分支單元 220 輸入表示處理器 210 的一現狀之指令及信號。為回應此指令及信號，分支單元 220 輸出(至順序單元 218)信號，表示有適合處理器 210 執行之儲存一系列指令的記憶體地址。為回應來自分支單元 220 之信號，順序單元 218 自指令快取 214 輸入顯示的指令序列。若一或多個指令序列不儲存於指令快取 214，則指令快取 214 自連結至系統匯流排 211 的系統記憶體 239(經 BIU 212 及系統匯流排 211)輸入此指令。

為回應自指令快取 214 輸入的指令，順序單元 218 選擇性地發出指令至執行單元 220、222、224、226、228、及 230 之選取者。每一執行單元執行一特定指令類型的一或多個指令。舉例而言，FXUA 222 及 FXUB 224 以來源運算元(如加、減、AND、OR、XOR)，執行一第一類型的定點數學運算。CFXU 226 以來源運算元(如定點乘法及除法)，執行一第二類型的定點運算。FPU 230 以來源運算元(如浮點乘法及除法)，執行浮點運算。

當資訊儲存於一選取的更名緩衝器 234，此資訊係如選取之更名緩衝器所分配的指令所顯示，與一儲存位置

(如 GPRs 232 之一或進位位元(CA)暫存器 242)關聯。儲存於更名緩衝器 234 之一選取者的資訊被複製到與其相關的 GPRs 232(或 CA 暫存器 242)之一，以回應順序單元 218 的信號。順序單元 218 指導儲存於更名緩衝器 234 之一選取者的資訊之複製，以回應「完成」產生此資訊的指令。此複製稱為「回寫」(writeback)。

當資訊儲存於一選取的更名緩衝器 238，此資訊與 FPRs 236 之一相關聯。儲存於更名緩衝器 238 之一選取者的資訊被複製到與其關聯的 FPRs 236，以回應順序單元 238 的信號。順序單元 218 指導儲存於更名緩衝器 238 之一選取者的資訊之複製，以回應「完成」產生此資訊的指令。

經由同時於許多不同的執行單元 220、222、224、226、228、及 230 處理多個指令，處理器 210 達成高效能。因此，每一指令以一階段順序處理，每一者可以平行於其他指令階段的方式執行。此技術稱為「管線式執行」(pipelining)。此描述的實施例之一重要面向為，一指令通常以六階段來處理，即擷取、解碼、發送、執行、完成、及回寫。

在擷取階段，順序單元 218 選擇性地輸入(自指令快取 214)來自一或多個記憶體地址之一或多個指令，此記憶體地址係用以儲存其後與分支單元 220 及順序單元 218 一同詳細敘述的指令順序。在解碼階段，順序單元 218 至多解碼四個已擷取的指令。

在發送階段，在保留已發送指令的結果(目的運算元資訊)的更名緩衝器項目後，順序單元 218 選擇性地發送至多四個已解碼的指令至執行單元 220、222、224、226、

228、及 230 之選取者(對應在解碼階段的解碼)。在發送階段中，運算元資訊提供已發送的指令給所選之執行單元。處理器 210 以其被程式化的順序發送指令。

在執行階段，執行單元執行其已發送的指令，且輸出其運算元之結果(目的運算元資訊)，以儲存於如上詳述之更名緩衝器 234 以及更名緩衝器 238 所選之項目中。依此方法，處理器 210 可執行相對無順序之程式化順序之指令。

在完成階段，順序單元 218 表示一指令係「完成」。處理器 210 以其程式化順序「完成」指令。

在寫回階段，順序單元 218 分別指導將資訊從更名緩衝器 234 及 238 複製到 GPRs 232 及 FPRs 236。順序單元 218 指導儲存於所選之更名緩衝器之資訊的複製。相同地，在一特定指令的寫回階段，處理器 210 更新其結構狀態，以回應特定指令。處理器 210 以其程式化順序，處理對應的指令「寫回」階段。處理器 210 在指定的狀態下，優勢地匯集一指令的完成階段及寫回階段。

在此描述的實施例中，每一指令需要一機器週期，完成每一指令處理的階段。然而，某些指令(例如，CFXU 226 所執行的複合式定點指令)可能需要大於一個週期。因此，一變數延遲可能於一特定指令的執行及完成階段間產生，以回應先前指令完成所需時間之變化。

完成緩衝器 248 提供於順序單元 218 內，追蹤於執行單元內執行的多個指令之完成。在顯示已成功地完成一指令或一指令群組之後，以一應用程式所指定的順序，完成緩衝器 248 可用來初始已完成指令的結果之轉移至相關聯的一般暫存器。

再者，處理器 210 亦包含效能監控單元 240，其係與指令快取 214 以及處理器 210 中的其他單元相連結。處理器 210 的運作可利用效能監控單元 240 來監控，在此描述的實施例中，效能監控單元 240 係一軟體可存取的機制，可提供詳細資訊描述指令執行資源以及儲存控制的使用性。雖然沒有描述於圖 2 中，效能監控單元 240 耦接處理器 210 的每一功能單元，以允許處理器 210 運作的全方面監控，包含例如，重建事件間的關係、辨識錯誤觸動、辨識效能瓶頸、監控管線阻塞、監控閒置處理器週期、決定發送效率、決定分支效率、決定錯誤資料存取排列的效能損失、辨識順序指令的執行頻率、辨識抑制的中斷、以及決定效能效率。相關的事件亦可包含，例如，指令解碼的時間、指令的執行、分支事件、未命中快取、以及命中快取。

效能監控單元 240 包含計數器 241-242 的一實施相關數(例如 2-8)，標示為 PMC1 及 PMC2，用來計算所選事件之發生。效能監控單元 240 更包含至少一監控模式控制暫存器(MMCR)。在此範例中，兩控制暫存器，MMCRs 243 及 244，指定計數器 241-242 的功能。計數器 241-242 以及 MMCRs 243-244 較佳地係以 SPRs 實施，可經 CFXU 226 所指執行的 MFSPR(自 SPR 移出)以及 MTSPR(移入 SPR)的指定，做讀取或寫入的存取。然而，在另一實施例中，計數器 241-242 以及 MMCRs 243-244 可能簡單地以 I/O 空間內的地址實施。在另一實施例中，控制暫存器以及計數器可能間接地經由一索引暫存器存取。此實施例實施於 IA-64 架構的 Intel 處理器。

再者，處理器 210 亦包含中斷單元 250，耦接指令快

取 214。再者，雖然圖 2 中無顯示，中斷單元 250 耦接處理器 210 內的其他功能單元。中斷單元 250 可接收來自其他功能單元的信號，以及初始一行動，例如，開始一錯誤處理或陷阱程序。在這些範例中，中斷單元 250 用來產生在一程式執行期間，可能會產生的中斷以及錯誤。

在一程式執行期間，本發明提供監控特定指令之執行，以及特定記憶體位置之存取的能力。這類型的決定亦稱為程式碼涵蓋範圍。更具體地，一備用欄位可用來存放一指示符，其係與一特定指令或記憶體位置相關聯。此指示符可為回應指令之執行或記憶體位置之存取而設定。這些指示符的檢驗可用來決定哪些程式碼已被執行，而哪些程式碼尚未被執行。一類似的辨識可用來決定在程式碼執行期間，哪些記憶體位置已被存取，而哪些記憶體位置尚未被存取。或者，此指示符可儲存於與其指令或記憶體位置相關聯之另一位置。

在指示符放置於指令中的情況下，通常會使用一備用欄位，但在其他情況，指令可能為了包含指示符所需的空間而延伸。在此情況下，處理器的架構可能需要改變。例如，一 64 位元的架構可能需要變成一 65 位元的架構，以包含此指示符。在資料存取方面，一指示符可能與資料或資料所位於的記憶體位置相關連。

在這些圖式範例中，這些辨識服亦稱為存取指示符。資料存取指示符與記憶體位置相關聯，而指令存取指示符係與指令相關聯。這些指示符通常在一程式執行前，係初始化為未設定。

當處理器執行一指令時，會設定與其指令相關聯之指令存取指示符。一記憶體位置之存取，導致與其記憶體

位置相關聯之一指示符被設定。藉由檢驗已設及未設的指示符，可以辨識程式碼涵蓋範圍。利用設計來測試程式碼涵蓋範圍的軟體工具中所實施的程序可實行此檢驗。依此方法，則不需要統計分析或調整程式碼，即可找出指令及記憶體位置之資料存取的程式碼涵蓋範圍。

參考圖 3，描述本發明之一較佳實施例中，用來處理與指示符相關之指令的元件。指令快取 300 接收套裝指令 (bundles) 302。指令快取 300 係圖 2 中指令快取 214 的一範例。一套裝為一指令群組。此類型的指令分組常見於 Intel 的 IA-64 處理器。指令快取 300 處理指令供執行。

如此指令處理的一部份，指令快取 300 遞送指令至不同執行單元供其執行。當發生此指令之執行時，會從緩衝器 306 接收信號 304。完成緩衝器 306 為一完成緩衝器之範例，如圖 2 中的完成緩衝器 248。信號 304 辨識其執行已完成的指令。此辨識可以不同型態產生。舉例而言，對應完成緩衝器 306 之項目的項數可用來辨識其指令已完成之指令。依此方法，指令快取 300 知道指定的指令常駐於其快取之何處。根據處理器之實作，可使用其他指示符來辨識哪些指令已完成，例如快取 300 中的狀態。

當接收信號 304 時，指令快取 300 設定與已執行之指令相關聯之指令存取指示符。回應接收信號 304，指令快取 300 會標記或設定指令存取指示符 308，其係與已完成執行之指令相關聯。依照特定實作，此指示符可在許多地點。舉例而言，指令快取指示符 308 可位於一影子記憶體的指令中一備用位元或欄位，或在一頁表中。

有關存取記憶體位置中的資料，其資料及指示符係由一資料快取（如圖 2 中的資料快取 216），而非由一指

令快取處理。或者，處理器可能具有一資料與指令結合之快取，在此情況，其指示符依此作適當的設定。

雖然此圖式範例顯示一信號的產生，顯示指令執行之完成，係由完成緩衝器 306 產生，此信號可根據特定處理器架構及設定，經由其他處理器單元產生。舉例而言，此單元可實際處理此指定之處理單元。再者，若硬體不支援不規則的作業，則將一指令送至一處理單元供其執行的派送，可作為一可接受的時間點，顯示此指令已被執行。換句話說，若一指令並非以投機地方式執行，則在處理器中的一遞送器可將其指令標記為已執行。

參考圖 4，一圖式根據本發明之一較佳實施例，顯示用來設定辨識已存取過的記憶體位置之指示符所使用的信號之資料流。當已執行一指令時，資料快取 400 接收來自完成緩衝器 404 之信號 402。信號 402 亦包含在一記憶體位置之資料，已被執行過的指令所存取的顯示。

為回應接收信號 402，設定資料存取指示符 406。此資料存取指示符係與已完成的指令所存取之記憶體位置相關連。在資料存取指示符不包含於其記憶體位置本身的情況，元資料 (metadata) 係用來映射指示符之關聯性與記憶體位置。在快取單元中，指令/資料狀態辨識指令/快取之實際記憶體中之位置。所有現今可使用的快取單元具有其能力。此快取單元亦知道至哪一位置更新其存取指示符。

參考圖 5，一圖式顯示根據本發明之一較佳實施例，將一存取指示符與一指令或記憶體位置相關聯之一機制。處理器 500 接收快取 502 之指令。在此範例中，其指示符不與其指令儲存在一起，或儲存於資料所找到的記憶

體位置。反而，指示符係儲存於一獨立的儲存範圍（快取指示符影子快取 504）。此儲存處可為任何儲存裝置，例如一系統記憶體、一快閃記憶體、一快取、或一磁碟。

當處理器 500 接收，且接著執行快取 502 的指令時，處理器 500 在存取指示符影子快取 504 中設定一指令存取指示符，對應至快取 502 的指令。包含資料之記憶體位置之存取運作類似的程序。在一實施例中，針對不影響實際資料區段之每一對應字組，會提供一全影子字組。換句話說，處理器 500 允許快取 502 的架構或設定維持不變。在這些範例中，在此所指的是逐字組的映射。然而，可使用某些其他類型的映射，例如，每一資料字組的一影子位元，其中在效能調整影子快取 504 中的一位元，對應資料的一個字組。

針對這類型的架構，編輯器使用此功能，以類似偵錯符號的方法，在資料範圍本身以外的一工作範圍，建立偵錯資訊。當載入一模組時，額外的資訊，效能指示符，由載入器準備，當指令載入快取 502 時，模組可包含於效能調整影子快取 504。這些快取範圍可相互交織，且如此標記，或為運作模式所理解作標記。當發生相關的資料存取及指令執行時，處理器 500 設定其存取指示符。在這些範例中，處理器經一偵錯器或一分析程式而程式化，以了解是否在處理器執行指令時，使用影子資訊。

參考圖 6，圖式描述本發明之一較佳實施例的一套裝指令。套裝指令 600 包含指令槽 602、指令 604、指令槽 606、以及範本 608。如圖示，套裝指令 600 包含 128 位元。每一指令槽包含 41 位元，且範本 608 包含 5 位元。範本 608 用來辨識現今套裝指令中的停止，以及映射指令

槽內的指令至不同類型的執行單元。

套裝指令 600 內的備用欄位，係用來存放本發明之指示符。例如，指示符 610、612、及 614 分別位於指令槽 602、604、606 內。根據特定的實施，這些指示符可為不同格式，並且可為不同大小。指示符可使用一單一位元、或可使用多個位元。在這些範例中，一單一位元可用來顯示已執行的指令。欄位的類似使用，可供標記資料或記憶體位置的指示符使用。當執行總是開始於套裝指令中之第一指令時（如同在 IA64 架構中），兩個位元可用來顯示最後執行的指令。僅有具有分支指令之範本有此需求。若基本塊已標有旗幟，則僅需標記分支指令及其隨後之分支。

參考圖 7，描述根據本發明之一較佳實施例，顯示產生元資料及存取指示符所使用之元件之方塊圖。其編輯器支援嵌入於來源的指示符，其指示符表明欲產生之元資料。此元資料係用來表明欲運作哪一作業，例如辨識哪一指令，或記憶體位置欲與特定存取指示符相關連。換句話說，元資料定義存取指示符至個別指令、指令群組、及記憶體位置之映射。針對記憶體位置，根據特定實作，元資料可能映射存取指示符至個別記憶體位置，或具有不同大小的不同群組的記憶體位置。這些存取指示符係由位於處理器中的硬體元件所使用，當存取一記憶體位置、或執行一指令時，會反映這些元件所作的更新。

編輯器 700 可產生指令 702 供執行，以及元資料及存取指示符 704 供監控程式碼涵蓋範圍。當指令或資料快取頁載入記憶體時，作業系統程式載入器/連結器及/或效能監控程式，讀取編輯器 700 所產生的元資料及存取指示

符 704，並且載入元資料及存取指示符 704 至記憶體。處理器 708 可接受編輯器 700 所產生之格式之元資料及存取指示符 704，並且分佈元資料及存取指示符 704 至影子記憶體 705。

在另一實施例中，效能監控程式，而非編輯器，產生其元資料及存取指示符。

在一實施例中，針對每一方塊或區域參數，格式單純的具有一元資料與/或一存取指示符影子快取項目，並且移動元資料及存取指示符 704 至其對應的影子項目。代替一影子快取，可修改快取本身的內部格式，以包含元資料及存取指示符 704。在修改指令流本身，以包含元資料的範例中，載入器會更新指令流，以包含適當的元資料與/或存取指示符及工作區，或者編輯器 700 已產生程式碼，以包含元資料及存取指示符 704。在任一情況下，在載入程式碼後，處理器接收元資料及存取指示符 704。

再者，元資料及存取指示符 704 可置放於與指令 702 關聯之影子記憶體 705。在一表格或偵錯資料區段，編輯器 700 產生資訊。效能監控程式將此資訊載入影子記憶體 705 中的影子資料範圍。或者，藉由作業系統或效能監控程式與處理器共同合作，可自動地分佈偵錯範圍。

指令 702 可接著由處理器 708 執行。編輯器 700 可設定一暫存器（如處理器 708 中的模式暫存器 710）。當設定此暫存器時，處理器 708 在執行指令 702 以決定元資料及存取指示符 704 是否與指令中正在執行中的指令相關時，會看影子記憶體 705 中的元資料及存取指示符 704。

若沒有設定模式暫存器 710，則當執行指令 702 時，會忽略元資料及存取指示符 704。在這些範例實施例中，

模式暫存器 710 係用來開啟或關閉標記指令之存取之功能。

除了定義與存取指示符相關之指令或指令群組外，在元資料及存取指示符 704 內的元資料可用來設定模式暫存器 710。當一存取指示符係在一指令階層有關聯性時，與一存取指示符相關聯之每一指令，在其指令執行時，會設定其相關之存取指示符。在這些範例中，當一存取指示符係與一指令群組相關連時，執行此群組中的任何指令，會設定此存取指示符。

類似程序可運作在記憶體位置 712 的資料。根據特定實作，元資料及存取指示符 704 可置放於指令中，或於資料內，而非影子記憶體 705。然而，藉由將元資料及存取指示符 704 置放於影子記憶體 705，當將元資料及存取指示符 704 置放影子記憶體 705 時，可動態地運作元資料及存取指示符 704 的產生。

此功能允許程式碼涵蓋範圍監控之指令的選擇與辨識的產生，而不需要修改程式。換句話說，在指令 702 已被編輯供處理器 708 執行後，編輯器 700 可能產生元資料及存取指示符 704。設定模式暫存器 710，導致處理器 708 找尋效能調整影子記憶體 705 中的元資料及存取指示符 704，而不需修改指令 702。在這些範例中，當於指令 702 中發生指令之執行時，與/或資料存取記憶體位置 712 中的記憶體位置時，與一指令或記憶體位置相關聯之一存取指示符之存在，會導致處理器 708 設定其存取指示符。

圖 8 顯示根據本發明之一示範實施例，使用一頁表，解析程式所指定的記憶體地址為一實體地址之資料流之方塊圖。如圖 8 所示，一程式地址 810（供資料或指令）

係藉由地址空間暫存器 820，使用許多指定活動中的地址空間之手段之一者，被解析為一虛擬地址。此形成的虛擬地址係由處理器使用，搜尋頁表 830，以找尋頁表 830 中與虛擬地址相匹配的一頁描述符。此匹配的頁描述符之內容通常包含實際地址以及與其虛擬頁相關聯之屬性。這些內容係用來解析其虛擬地址為一實際地址，且決定此頁之屬性（例如存取權）。

在本發明之另一實施例中，延伸此頁表，以包含額外的欄位，每一欄位用來儲存存取指示符（例如指令存取指示符及資料存取指示符）。當一程序存取頁表，以運作虛擬至實體頁地址之映射時，根據造成此頁表之存取之特定事件，可查詢這些額外的欄位、從這些欄位擷取數值、以及更新這些欄位的數值。

或者，為了避免效能的降低，在這些欄位中的存取指示符資訊可快取於類似於一轉換後備緩衝器「TLB」或一有效至真實地址解析緩衝器「ERAT」之處理器資源中。舉例而言，可提供一效能指示符後備緩衝器「PILAM」，其中上述頁表之欄位所提供之虛擬至真實地址解析資訊及效能指示符資訊可被快取。當接收一指令或資料地址存取要求時，程式或虛擬地址之找尋可在 PILAM 運作，以取得地址分析資訊及存取指示符資訊。若其程式及虛擬地址不存在於 PILAB，則可詢問頁表，以取得此資訊。

圖 9 描述本發明之一範例實施例之一範例頁表項目。如圖 9 所示，頁表項目 900 包含供儲存一虛擬頁地址之欄位 910、供儲存一實體頁地址之欄位 920、以及儲存決定程式碼涵蓋範圍相關之程式碼涵蓋欄位 930-950。測試或效能監控應用程式使用這些欄位，以決定測試中的一

電腦應用程式之程式涵蓋範圍。這些程式碼涵蓋欄位可包含，例如，一指令存取指示符、一資料存取指示符、或預先擷取資訊。這些額外欄位 930-950 之數值可根據測試或效能監控應用程式所提供這些單元之資訊，經由不同元件設定於處理器中。

舉例而言，與上述之存取指示符關聯指令與/或資料之某部分之類似方法，存取指示符可與這些指令與/或在頁表內之資料部分相關聯。因此，當決定一指令或資料部分是否具有一相關聯之存取指示符時，其指令或資料部分之虛擬地址可用來辨識頁表中的一項目，且可檢驗儲存於額外欄位 930 及 940 之數值，以查看是否有一存取指示符與實體頁或實體頁之一部份相關聯。即，若與虛擬地址相關聯之偏值落於欄位 930 所辨識之偏值範圍內是否具有一存取指示符儲存於其中。此指示符係設定於指令執行的事件中。

存取類型欄位 940 係用來辨識存取的類型，其存取係指一記憶體位置之存取。預先擷取資訊欄位 950 可包含例如指示器、偏移值、以及地址等資料。因此，舉例而言，當一指令或資料的某部分一定要從實體儲存處擷取時，會詢問頁表以辨識指令或資料的某部分之其實體儲存位置。同時，可詢問欄位 930-950，且可設定存取欄位 930 中的指示符，以表明此頁已從實體儲存處擷取，且載入記憶體或快取。

需要知道的是，圖 9 僅顯示供儲存一存取指示符的一單一欄位、供辨識一存取類型之一單一欄位、以及供儲存預先擷取資訊之一單一欄位，本發明並不限於此。反而，供儲存複數個存取指示符之任何欄位數，及其類似

者，與實體頁相關聯皆可使用，而不偏離本發明之精神及範圍。

參考圖 10，描述本發明之較佳實施例存取指令之一程序之流程圖。圖 10 中所描述的程序可實施於一指令快取（如圖 2 中的指令快取 214）中。

此程序開始於接收一套裝指令（步驟 1000）。接著，辨識在此套裝指令中供執行的一指令（步驟 1002）。接著，執行其指令（步驟 1004）。在此範例中，步驟 1004 的指令處理之一部份，係包含型其指令。接著，設定與此指令相關之一指示符，以表明此指令之執行（步驟 1006）。接著，決定是否在套裝指令中，有更多未處理的指令（步驟 1008）。

若在套裝指令中，有更多未處理的指令，則程序進行至步驟 1002，如上述。若在套裝指令中，沒有額外的指令未處理，則結束此程序。在另一實施例中，若此套裝指令不具有分支，則設定一位元，其位元表示當取得一分支時，最後執行的指令。

在此範例中，指令快取包含所有的資訊，且可直接存取包含欲更新之指令快取指示符的範圍，以反映指令執行時的改變。一完成緩衝器或存取此完成緩衝器之處理器單元會提供有關完成指令的資訊。

參考圖 11，描述本發明之較佳實施例，在指令快取之後的一程序，以設定指令快取指示符之流程圖。圖 11 中所描述的程序可實施於一指令快取（如圖 2 中的指令快取 214）。

此程序開始於接收表明指令執行已完成之信號（步驟 1100）。在這些範例中，信號是從一完成緩衝器（如圖

2 中的完成緩衝器 248) 接收。此完成緩衝器包含辨識其執行已完成之指令之資訊。接著，儲存儲存處中的更新資訊 (步驟 1102)，其後結束此程序。可將指令快取中的一項目儲存於儲存處，以回應此信號之接收。此項目係使用完成緩衝器的信號所接收之資訊來辨識。舉例而言，此信號可能表明對應項目號 7 的指令，已執行完成。此指令之資訊可位於指令快取所建立的暫存器中，以提供運作作業更快的途徑。將資訊儲存於快取中，此資訊係置放於另一位置 (例如一頁表或影子記憶體)。若其指令有與其相關聯之指示符，則此更新亦包含一指令存取指示符之設定。

因此，指令的程式碼涵蓋範圍可經處理器中的硬體元件而辨識。此硬體協助允許已執行的指令，經執行指令時所設定的指令存取指示符而辨識。上述範例提供細微的粒度，其每一執行過的指令皆有標記。若需要較大的粒度，則此程序可應用於程式碼的某部分 (例如程式碼區段或子程序)。

參考圖 12，描述本發明之較佳實施例，標示次常式存取之程序之流程圖。圖 12 所描述之程序可實施於一指令快取中 (例如圖 2 中的指令快取 214)。

此程序開始於辨識已執行的指令 (步驟 1200)。接著，決定此指令是否係位於供監控所選之一次常式之一起始位址 (步驟 1202)。若其指令不位於一次常式之一起始點，則決定先前是否在其次常式之一起始位址的指令設定一指令存取指示符 (步驟 1204)。若決定先前已在其次常式之一起址位址的指令設定一指令存取指示符，則決定所辨識之指令是否為次常式之一返回位址 (步驟 1206)。若其指令為一返回位址，則其返回位址之指令存取指示符係

設定於與其次常式相關聯之元資料中（步驟 1208），隨後結束此程序。

參考上述之步驟 1206 及 1204，若決定為否，則程序回到步驟 1200。於步驟 1202，若所辨識之指令係一次常式之起始點，則此程序進行至步驟 1208，如上述。

參考圖 13，描述本發明之較佳實施例，辨識程式碼區段之存取之程序之流程圖。圖 13 所描述之程序可實施於一指令快取中（如圖 2 中的指令快取 214）。

此程序開始於辨識欲執行的指令（步驟 1300）。接著，決定所辨識之指令是否為一分支指令（步驟 1302）。若此指令為一分支指令，則設定與其指令相關之指令存取指示符（步驟 1304），隨後結束此程序。如上述，此指示符可以許多不同方法實施。舉例而言，此指示符可能位於指令中、影子記憶體中、或頁表中，一額外的欄位、或者一尚未使用的欄位。

參考步驟 1302，若其指令並非一分支指令，則決定是否先前之指令為一分支指令（步驟 1306）。若先前指令係一分支指令，則程序進行至步驟 1304，以如上述標記一指令存取指示符。在步驟 1306，若先前指令並非一分支指令，則此程序結束。基本上，當發生一分支時，步驟 1306 與步驟 1304 標記一分支指令之後的一指令。

因此，本發明之機制提供已執行的程式碼區段之標記的硬體協助。藉由檢驗已標記的指令存取指示符，可辨識程式碼區段之執行，決定程式碼涵蓋範圍。

參考圖 14，描述本發明之較佳實施例，用以分析程式碼涵蓋資料之元件之方塊圖。根據本發明之硬體協助機制所產生的程式碼涵蓋資訊，可使用軟體工具，產生一報

告或圖像顯示，辨識尚未執行的程式碼範圍，以及已執行的程式碼範圍。此程式碼涵蓋資訊亦可包含在其程式碼執行期間所發生的記憶體位置之存取。根據此資訊，可初始化此程式碼之新測試，產生更新的報告或顯示，顯示程式碼涵蓋範圍的改變。

在此範例中，程式碼涵蓋應用程式 1400 取得程式碼 1404 之執行所產生的存取指示符資料 1402。此程式碼可為例如一程式、一應用程式、或一程式或應用程式之某部分的程式碼。存取指示符資料包含指令存取指示符與/或資料存取指示符。

當程式碼 1404 之執行開始時，最初是未設定這些指示符。在執行指令期間，當執行指令，或存取記憶體位置中的資料時，會設定這些指示符。在程式碼 1404 執行後，所有或部分的這些指示符會被設定。

程式碼涵蓋應用程式 1400 分析存取指示符資料 1402，辨識在程式碼 1404 執行期間，已涵蓋或尚未涵蓋之指令的路徑。這些執行資料係使用者圖像介面 (GUI) 1406 顯示。這些資料的顯示可包含強調、或使用不同顏色、或圖像指示符，表示已存取資料的記憶體位置。使用者亦可使用 GUI 1406，來設定測試程式碼 1404 所使用的新參數。當再次測試程式碼 1404 時，會接著分析新的存取或涵蓋資訊，且顯示於 GUI 1406，以允許使用者看到程式碼涵蓋範圍如何改變。

在這些範例中，程式碼涵蓋應用程式 1400 可為週期性執行，以採樣涵蓋範圍的虛擬應用程式，且可提供更新的報告或顯示。使用者可更改許多參數或可定義一組在執行程式碼 1404 時可使用的情況。

可收集且顯示程式碼涵蓋資料給使用者，供其分析。參考圖 15，描述本發明之較佳實施例，已執行與未執行指令的顯示之圖式。在此示範實施例中，程式碼 1500 係已執行的來源程式碼之範例，其使用硬體協助，標記處理器實際執行之指令的指令存取指示符。

在此範例中，已經執行區域 1502、1504、及 1506，而區域 1508、1510、1512、及 1514 係尚未執行的指令。可以使用一種顏色（如綠色）來呈現已執行的指令，以另一顏色（如紅色）呈現尚未執行的指令來呈現程式碼 1500。依此方法，使用者可簡單的辨認已執行及尚未執行的指令。在另一實施例中，僅顯示已執行或尚未執行的指令。再者，圖像指示符可與已執行的指令相關聯，或置放其鄰近處。可使用許多不同類型的顯示技術，以顯示程式碼涵蓋資訊。舉例而言，圖像使用者介面亦可使用螢光、不同字型、以及僅顯示所執行的指令於顯示器中。任何類型的顯示系統皆可用來允許使用者辨識程式碼中已執行及尚未執行的指令。

參考圖 16，描述本發明之較佳實施例收集程式碼涵蓋資料，以及顯示在一程式碼內，有涵蓋及未涵蓋之路徑的程序之流程圖。圖 16 中所描述的程序可實施於一程式碼涵蓋範圍或測試應用程式。圖 16 中所描述的程序係針對指令的執行作程式碼涵蓋資料的收集。

程序開始於接收程式所使用的測試參數（步驟 1600）。以下係可輸入的一範例指令：`Test -code-data Application`。在此範例中，`Test` 係測試程式，`Application` 係預測試之應用程式，`-code` 表示程式碼涵蓋範圍，以及 `-data` 表示資料涵蓋範圍。接著，執行此程式（步驟 1602）。

步驟 1602 的程式執行，導致程式碼涵蓋範圍，以在處理器執行程式之指令期間所設定的存取指示符(如指令存取指示符)之形式而產生。

接著，使用硬體協助執行指令，決定是否完成程式碼涵蓋資料之收集(步驟 1604)。若尚未完成程式碼涵蓋資料之收集，則程序回到步驟 1602，繼續執行此程式。否則，辨識所執行的指令(步驟 1606)。在此示範例中，步驟 1606 係藉由辨識其指令存取指示符已被設定之指令來實施。接著，辨識尚未執行的指令(步驟 1608)。在此示範例中，步驟 1608 係藉由辨識其指令存取指示符尚未被設定之指令來實施。接著，顯示辨識執行所涵蓋的路徑及執行所未涵蓋的路徑之執行資料(步驟 1610)。在這些範例中，顯示於一圖像使用者介面，其中涵蓋的執行路徑及未涵蓋的執行路徑可以圖像顯示供使用者分析。

接著，決定是否有供執行此程式所使用的新參數存在(步驟 1612)。若沒有新參數，則結束程序。若有新參數，則程序接著進行至步驟 1600，如上所述。

在此描述範例中，涵蓋資料之顯示，係顯示在程式執行之後。根據其實作，資料的顯示可能在程式執行期間週期性的發生，且可能允許使用者在執行期間改變參數，以檢視程式碼涵蓋範圍如何改變。

本發明之機制亦提供以資料存取型態，決定涵蓋範圍之機制。指示符可與元資料儲存在一起，其中元資料選擇具體部分的資料作調整。資料是否已存取的辨識，可儲存在許多不同的地方。舉例而言，一個字組可延伸額外的位元、可提供一個影子快取範圍、以及可設計一頁表，包含此資訊。

為了支援資料存取涵蓋範圍，會更新編輯器，以支援產生一元資料偵錯部分（類似一符號部分），且作業系統與鏈結器/載入器與一效能監控或測試涵蓋裝置驅動程式提供專門的作業模式。這些技術模仿一額外位元或記憶體的逐字組模仿。以所指定的應用程式，使用不同機制，設定一位元，以表示此特定的資料範圍已被執行。表明此支援的方法有很多種。舉例而言，可提供一位元給每一資料位元，或每一位元給每一資料字組。編輯器所產生的元資料可提供此映射供硬體使用。此映射可具有資料類型依附性。當存取一資料欄位時，會設定適當的位元給一資料存取指示符。根據映射，相同的位元可設定給不同的資料存取。

參考圖 17，描述本發明之較佳實施例，標記存取包含資料之記憶體位置之程序之流程圖。在這些示範例中，每當資料存取發生時，初始此程序。圖 17 的程序可實施於一資料快取（如圖 2 中的資料快取 216）。

此程序開始於偵測記憶體位置中資料的存取（步驟 1700）。在這些範例中，資料快取所辨識的資料存取係藉由一完成緩衝器所產生的信號，以表示在執行一指令時，此資料已被存取。在一實施例中，此已存取的指令之辨識及其資料運算之瞭解，係藉由指令快取單元，接收來自完成緩衝器之信號，且傳送一信號至資料快取單元，以表示此指令的資料運算元已被存取。或者，完成緩衝器會直接傳送資訊已存取之辨識至資料快取。此方法會需要完成緩衝器知道其資料資訊。

接著，設定表示存取的資料存取指示符，其係與包含資料之記憶體位置相關聯（步驟 1702）。在這些示範例

中，資料存取指示符可位在一影子記憶體或一頁表中。當處理器處於一選取的或特殊的模式，導致具有一長度（如一位元、一字組、或一雙字組）的資料設定有一位元時，會產生指示符的設定。接著，處理記憶體位置存取（步驟1704），隨後結束此程序。

決定測試涵蓋範圍是一件困難的課題，因此瞭解資料存取及資料使用可能係有幫助的。換句話說，知道一分配的陣列已被使用，或一靜態資料已被利用係有幫助的。再者，測試資料邊界係測試涵蓋範圍的重要課題。舉例而言，決定是否已分配足夠的空間給一陣列係有用的。此決定可藉由決定是否產生陣列以外的存取來運作。

本發明之機制提供呼叫流樹的建立與維護，其呼叫流樹可藉由執行動態資料範圍涵蓋所使用的程式來存取。當要求資料的分配（例如 malloc）時，會呼叫一程式來建立一樹。決定呼叫堆疊的一種方法，係隨著堆疊決定在 malloc 時間點的呼叫順序。另一方法，係使用設定資料存取指示符所產生的硬體資訊。類似地技術描述於標題為「使用硬體協助程式緒堆疊追蹤及登記的符號資料來自動決定電腦程式流之方法及裝置」的美國專利申請案，律師登錄號為 AUS200305498US1，包含在此供參考。此技術係用來辨識呼叫順序；此後稱為呼叫堆疊。此樹係保存在記憶體中，且可經例如應用程式介面（API）呼叫讀取現今程式緒之呼叫堆疊資訊之一裝置驅動程式的呼叫來存取。保存硬體程式緒所保存的呼叫堆疊，且轉換地址為符號名稱的程序亦描述於標示為「使用硬體協助程式緒堆疊追蹤及登記的符號資料來自動決定電腦程式流之方法及裝置」的美國專利申請案，律師登錄號為

AUS200305498US1。裝置驅動程式所擷取出的呼叫堆疊傳送至弧流程式，其將堆疊帶入以程式緒為導向的樹。此程序描述於 1999 年 6 月 30 日申請之標示為「匯集事件基準資料即採樣資料為後處理追蹤輸出之方法及系統」的美國專利申請案，案號 09/343,438，律師登錄號

AT9-98-850。其一方法涉及表示開始範圍、結束範圍、及結束範圍外的一位元供標示。再者，其他統計亦可保存在此記憶體範圍。可保存使用其他硬體協助功能、存取數目、快取未擊中、週期等所涉及的其他統計範例。

參考圖 18，顯示本發明之較佳實施例，描述一資料快取後的一程序之流程圖。描述於圖 18 的程序可實施於一資料快取（如圖 2 中的資料快取 216）。

此程序開始於接收到表示已完成存取記憶體位置中的資料之指令執行的信號（步驟 1800）。此信號通常是來自一完成緩衝器。儲存更新資訊於儲存處（步驟 1802），隨後結束此程序。儲存於資料快取中的暫存器的資訊被儲存。此資訊係儲存於一頁表，以及設定一資料存取指示符於另一儲存處（例如在一影子快取）。儲存於頁表的資訊通常包含指令所存取的資料所作的任何改變。而儲存於另一儲存處的資訊涉及設定一資料存取指示符，表示此存取已在記憶體位置發生。

參考圖 19，描述本發明之較佳實施例，在執行期間，當發生一動態記憶體分配而取得呼叫堆疊資訊的程序之一流程圖。圖 19 所描述的程序可以軟體實施。此程序偵測一動態記憶體分配（步驟 1900）。在執行期間，當一指令要求記憶體的分配時，會要求此動態記憶體分配。接著，取得呼叫堆疊資訊（步驟 1902）。爾後，辨識所分配

的記憶體（步驟 1904）。接著，標示供報告的記憶體（步驟 1906），隨後結束此程序。記憶體標示供報告用，以使硬體（例如資料快取）知道何時標記位於所分配的記憶體內資料之存取。記憶體分配可使記憶體分配例行程式分配比所指定還多的記憶體，以允許檢測錯誤。舉例而言，記憶體分配例行程式可分配一額外位元，且標示開始地址、結束地址、及結束地址外的一位元。換句話說，若存取發生於這些標示的地址，則這些標示的部分將會有資料存取指示符。在這些範例中，會標示結束地址外的一位元，使得此記憶體位置的存取，會導致一資料存取指示符的設定。藉此，決定是否已分配足夠的記憶體供使用。若在一程式執行期間設定此資料存取指示符，則由於存取到分配以外的記憶體範圍，可能係之前所分配的記憶體不足所致。

參考圖 20，描述本發明之較佳實施例，在程式執行期間，當取消記憶體分配時，傳送有關資料存取資訊之程序的流程圖。圖 20 所描述的程序可以軟體實施。

此程序開始於偵測到取消記憶體分配（步驟 2000），例如一釋放要求。接著收集辨識所分配的記憶體中之資料的存取之資訊（步驟 2002）。爾後，傳送存取資料及呼叫堆疊資訊供處理（步驟 2004），隨後結束此程序。傳送此資料至一軟體工具供處理。步驟 2004 可涉及將資料放置於此軟體工具可取得的位置。

在這些範例中，資料係在取消記憶體分配時，即時傳送。其他可收集並傳送資料供處理的時間，包含一監控程序所運作的採樣。在其他情況，資訊是在程式完成執行後傳送。辨識資料存取的資訊可包含除了記憶體位置之辨

識以外的資訊。舉例而言，資訊亦可包含記憶體分配所要求的程式碼之辨識，以及所分配的記憶體大小。再者，記憶體位置的存取數量可使用描述於參考應用中的硬體協助機制來記錄。

參考圖 21，描述本發明之較佳實施例，產生一呼叫流樹，以辨識尚未被初始化或存取的範圍之程序之流程圖。圖 21 中所描述的程序可以用來辨識程式碼涵蓋範圍所使用的軟體工具中實作。此程序使用圖 19 及 20 中所收集的資訊，產生一呼叫流樹，以辨識尚未被初始化或存取的範圍。此呼叫流樹一包含已被存取的記憶體位置，以及尚未被存取之記憶體位置之資訊。

當接收到存取資料以及呼叫堆疊資訊時，開始此程序（步驟 2100）。接著，處理所接收的資料及呼叫堆疊資訊（步驟 2102）。爾後，產生辨識呼叫者及程式碼分配記憶體的呼叫流樹（步驟 2104）。接著，產生包含資料存取資訊的子節點辨識資料（步驟 2106），隨後結束此程序。資料範圍的存取資訊，係放置在一子節點，其係與所存取的資料所在之分配的記憶體之節點相關聯。

有關資料範圍存取的涵蓋範圍，本發明之機制提供尚未初始化、尚未存取、或已存取的資料範圍之資訊。在辨識這些資料範圍中，原始來源程式碼會被標記，或以表示哪些資料範圍已被存取，以及哪些資料範圍尚未被存取的方式來展現。針對資料的動態分配，發生動態資料分配的來源程式碼部分可標有旗幟，使得在執行期間所分配的記憶體範圍被存取時，會標示其資料存取識別符。再者，程式碼中的指令可能會被標示，使得當靜態資料結構被存取時，硬體設定指示符。

參考圖 22，描述本發明之較佳實施例，顯示資料涵蓋範圍之圖像報告的圖式。在此範例中，報告 2200 在範圍 2202 顯示資料存取涵蓋範圍。此範圍顯示靜態資料範圍。在範圍 2202 的 2204、2206、及 2208 部分，係資料存取所發生的記憶體位置。這些部分係藉由找出在程式碼執行期間已設定資料存取指示符來辨識。

呼叫流樹 2210 係一呼叫流樹的範例，其中節點 D 表示一部份的程式碼，係動態地分配記憶體，如範圍 2212 所示。在範圍 2212 的資訊係位於在呼叫流樹 2210 中與節點 D 相關聯的子節點中。範圍 2212 的部分 2214 顯示已存取的範圍 2212 部分。

參考圖 23，描述本發明之較佳實施例，產生顯示資訊之程式的流程圖。圖 23 所描述的程序可實作於一軟體工具，供決定資料存取的涵蓋範圍。

此程序開始於取得靜態及動態資料的資料存取資訊（步驟 2300）。此資訊包含取得分配供程式碼執行使用的資料存取指示符。其他取得的資訊包含上述的記憶體動態分配期間所產生的呼叫流樹。

接著，產生存取至靜態及動態資料的涵蓋地圖（步驟 2302）。爾後，顯示涵蓋地圖（步驟 2304），其後結束此程序。此涵蓋地圖與圖 22 中的報告 2200 類似。此報告程式可支援在報告結構中使用偏值。舉例而言，`c:\program files\appl.exe(func1:100)` 可用來表示 malloc 係從 func1 中的位置 100 所發佈。

處理器匯流排速度及記憶體存取比處理器速度要慢的多。資料處理系統的效能比較可能被記憶體存取，而非被處理器的速度所管制或限制。此現象的結果為多層快取

的使用，並且試圖預先擷取資料這些快取，以供較快的資料存取。然而，試圖預先擷取通常係程式碼或字首驅動，且不適合處理依附資料的動態存取。最常見的方法要求編輯器或程式設計者，在程式碼流中，建立預先擷取的暗示。一旦產生物件程式碼，這些暗示提供允許預先擷取的資訊。然而，此系統提供不大的機會修改程式碼。範例包含在 Java 物件中找尋收集垃圾的參考，或找尋指示符。

本發明之機制藉由硬體協助的使用來改進預先擷取。利用元資料，一預先擷取指示符可與資料相關聯。舉例而言，預先擷取指示符資料可表示此資料，係被視為欲預先擷取之指示符。預先擷取指示符本身可為資料結構的指示符。當資料載入快取時，處理器偵測到指示符預先快取的狀態，並預先擷取資料所指向的快取線。

參考圖 24，描述本發明之較佳實施例，使用元資料預先擷取資料至一快取之程序之流程圖。圖 24 中所描述的程序，在這些範例中，可實作於一載入/儲存單元。預先擷取的資料在這些範例中係一指示符。

此程序開始於載入一指令（步驟 2400）。接著，決定供預先擷取使用的元資料是否與指令相關聯（步驟 2402）。在這些範例中，元資料係與一指令相關聯的預先擷取指示符。

若供預先擷取使用的元資料係與指令相關聯，則決定硬體是否決定預先擷取資料（步驟 2404）。在一較佳實施例中，指令快取決定何時以及是否指令快取將會不預期的預先擷取。舉例而言，指令快取可決定沒有未解決的快取未擊中，且具有一段時間未使用的快取項目可被取代。然而，若指令快取具有尚未解決的未擊中，且/或沒有方

便的快取線來代替，則可能不會不預期的預先擷取。一臨
界值可設定快取未擊中的數量及/或可代替的快取線數
量，以決定當有一預先擷取指示符存在時，指令快取是否
會預先擷取資料。舉例而言，若未解決的快取未擊中比所
選定的臨界值小時，則可發生一預先擷取。若快取線的代
替係有需要的，且若選擇代替的快取線數量，係大於所選
定的臨界值，則預先擷取亦可在這些範例中產生。載入儲
存單元可發佈要求，以不預期的載入快取單元，但由快取
單元決定是否將會實際作預先擷取。

或者，一載入儲存單元可根據硬體實作來決定是否
不預期的載入快取單元。若硬體決定預先擷取資料，則會
使用與指令相關聯的元資料，來將資料預先擷取至一快取
(步驟 2406)。在這些範例中，資料係一指示符。此指示
符為資料結構或資料塊的指示符。接著，傳送指令至一單
元供處理(步驟 2408)，隨後結束此程序。

參考步驟 2404，若硬體決定不預先擷取資料，則程
序回到步驟 2408，如上述。在步驟 2302，若供預先擷取
的元資料不與指令相關聯，則程序進行到步驟 2408，如
上述。在這些範例中，硬體可傳送指令至一單元，供在硬
體發佈一預先擷取應發生的辨識之前、同時、或之後處理。

除了預先擷取資料結構或資料塊的指示符外，本發
明之機制提供從資料結構內的記憶體位置，預先擷取資料
的功能。與指示符一起的字首或偏值可用來辨識欲預先擷
取的資料。或者，可使用一地址來辨識所需資料的位置。

參考圖 25，描述本發明之較佳實施例，描述包含欲
從起始點預先擷取之資料之辨識的元資料之程序之流程
圖。圖 25 中的程序可實作於一載入儲存單元(如圖 2 中

的載入儲存單元 228)。

此程序開始於偵測表示一預先快取之元資料 (步驟 2500)。元資料可包含來自於指示符所辨識之資料結構欲預先擷取之資料的辨識。此元資料可包含一指示符型態的預先擷取指示符。藉由一偏值或地址的使用，元資料亦可包含資料本身之辨識。根據快取單元之狀態決定是否應該產生一預先擷取 (步驟 2502)。根據快取單元的狀態，這些範例的預先擷取，對於選擇性運作的程序而言，可係不預期的。舉例而言，若快取未擊中的數目超過某一數值，則不發生預先快取，且結束此程序。

在步驟 2502 中，若將發生預先擷取，則定址記憶體中的一選取點，供預先擷取 (步驟 2504)。此選取點記憶體可使用一指示符或一偏值或字首，或一記憶體地址來辨識。接著，資料從記憶體中的一選取點，預先擷取至快取 (步驟 2506)，隨後結束程序。參考步驟 2502，若沒有發生預先擷取，則結束程序。

因此，本發明提供一種改進的方法、裝置、及電腦指令，以提供程式碼涵蓋範圍的硬體協助。本發明之機制提供載一程式執行期間，監控指令執行以及存取記憶體位置的功能。存取指示符係提供來表示指令執行或記憶體位置內資料的存取。這些指示符的設定是藉由硬體 (例如處理器中的不同元件) 來運作。藉此，軟體工具可取得並分析較佳涵蓋範圍的涵蓋資料，並預防改變測試中的程式碼。

再者，針對硬體提供決定程式執行中的涵蓋範圍之協助，上述流程圖中的步驟提供程序的邏輯流程圖，其可由硬體中不同的元件執行。更具體地，這些步驟可以處理

器中的微程式碼或硬體邏輯實作。在這些範例中，微程式碼係最低階指令，直接控制一處理器。一單一機器語言指令通成解析為許多微程式碼指令。微程式碼可為硬體或軟體的型態。在許多現代處理器中，微程式碼係直接寫入處理器中。

需要注意的是，雖然本發明以一全功能資料處理系統的情況描述，熟此技藝者皆知，本發明之程序可以一電腦可讀取媒介之指令的形式及許多不同型態分佈，且不論所使用來實施散佈的信號負載媒體係屬任何特定類型，本發明可均等應用。電腦可讀取媒體的範例包含可讀取類型媒體（如磁碟片、硬碟機、RAM、光碟機、DVD-ROM），以及傳輸類型媒體（如數位及類比通訊鏈結、使用例如無線電頻率或光波傳輸型態的有線或無線通訊鏈結）。電腦可讀取媒體可為程式碼格式的型態，在特定資料處理系統中解碼供實際使用。

本發明之描述已呈現供示範及描述之用，且並非意在限定或消耗本發明所揭露的形式。熟此技藝者當可作許多潤飾及修改。舉例而言，需要注意的是，不同處理器架構可散佈此說明書中所指的功能於不同單元，使用不同機制作跨單元的資訊通訊。再者，雖然特定處理器單元已再本發明之功能實作中描述，這些不同功能可與硬體內其他處理器結合。再者，圖式中所描述的步驟可以硬體電路或以處理器內的不同處理器單元所特定使用的指令（如一執行單元所使用的微程式碼）實作。

在此所選擇及描述的實施例，僅係用來將本發明之原理、實務應用作最佳的解釋，且使熟此技藝者瞭解本發明亦適用於其他特定使用所作的潤飾之實施例。

【圖式簡單說明】

本發明之新穎特徵係以下列專利申請範圍界定。然而，發明本身、較佳使用模式、其他面向及其優勢，藉由伴隨圖式，參考以下實施例之詳細說明，可較易明白本發明，其中：

圖 1 係本發明可實施於其中之一資料處理系統之一方塊圖；

圖 2 係本發明之一較佳實施例用以處理資訊之一處理器系統之一方塊圖；

圖 3 係一圖式，描述本發明之一實施例在處理與指示符相關聯之指令所使用之元件；

圖 4 係一圖式，描述本發明之一實施例用來設定辨識已存取過的記憶體位置所使用的信號之資料流；

圖 5 係一圖式，描述本發明之一實施例用以將一存取指示符與一指令或記憶體位置相關聯之一機制；

圖 6 係一圖式，描述本發明之一實施例之一程式捆；

圖 7 係本發明之一實施例用以產生元資料（如指示符）之元件之一方塊圖；

圖 8 係本發明之一實施例解析程式所指令的記憶體位址成為一實體位址所使用之一頁表中之資料流之一範例方塊圖；

圖 9 係本發明之一範例實施例之一範例頁表項目；

圖 10 係本發明之較佳實施例標記指令之存取之一程序之一流程圖；

圖 11 係本發明之較佳實施例在一指令快取至已設指令存取指示符後之一程序之一流程圖；

圖 12 係本發明之較佳實施例顯示標記一次常式之存取之一程序之一流程圖；

圖 13 係本發明之較佳實施例辨識程式碼區段之存取之一程序之一流程圖；

圖 14 係顯示本發明之一較佳實施例分析程式碼涵蓋資料所使用之元件之一方塊圖；

圖 15 係本發明之較佳實施例收集程式碼涵蓋資料及顯示被涵蓋及不被涵蓋之路徑之一程序之一流程圖；

圖 16 係本發明之較佳實施例標記包含資料之記憶體位置之存取之一程序之一流程圖；

圖 17 係本發明之較佳實施例顯示在一資料快取後之一程序之一流程圖；

圖 18 係本發明之較佳實施例當一動態記憶體分配發生在執行期間時取得呼叫堆疊資訊之一程序之一流程圖；

圖 19 係本發明之較佳實施例當記憶體被取消分配時，傳送與在一程式執行期間所產生之資料存取相關之資訊之一程序之一流程圖；

圖 20 係本發明施較加實施例產生一呼叫流數以辨識尚未初始化或存取之範圍之一程序之一流程圖；

圖 21 係本發明之一較佳實施例顯示一圖像報表供顯示資料涵蓋範圍之一圖式；

圖 22 係本發明之較佳實施例產生顯示資訊之一程序之一流程圖；

圖 23 係本發明之較佳實施例預先擷取資料至一快取之一程序之一流程圖；

圖 24 係本發明之較佳實施例顯示包含欲從起點預先擷取哪些資料之一辨識之元資料之一程序之一流程圖；以

及

圖 25 係本發明之較佳實施例描述包含欲從起始點預先擷取之資料之辨識的元資料之程序之流程圖。

【主要元件符號說明】

100 客戶端	102 處理器
104 主記憶體	106 PCI 本區匯流排
108 PCI 橋接器	110 區域網路配接器
112 SCSI 主匯流排配接器	
114 擴充匯流排介面	116 音效配接器
118 圖像配接器	119 音效/影像配接器
120 鍵盤以及滑鼠配接器	
122 數據機	124 記憶體
126 硬碟機	128 磁帶機
130 光碟機	210 處理器
211 系統匯流排	212 匯流排介面單元
214 指令快取	216 資料快取
218 順序單元	220 分支單元
222、224 定點單元	226 複合式定點單元
228 載入/儲存單元	230 浮點單元
232 一般用結構暫存器	234 定點更名緩衝器
236 浮點結構暫存器	238 浮點更名緩衝器
240 效能監控單元	241、242 計數器
243、244 監控模式控制暫存器	
248 完成緩衝器	250 中斷單元
260 系統記憶體	300 指令快取
302 套裝指令	304 信號

- 306 完成緩衝器
- 400 資料快取
- 404 完成緩衝器
- 500 處理器
- 504 快取指示符影子快取
- 600 套裝指令
- 604 指令
- 608 範本
- 700 編輯器
- 704 元資料及存取指示符
- 705 影子記憶體
- 710 模式暫存器
- 810 程式地址
- 830 搜尋頁表
- 910 虛擬頁地址欄位
- 930-950 程式碼涵蓋欄位
- 1400 程式碼涵蓋應用程式
- 1402 存取指示符資料
- 1404 程式碼
- 1406 係使用者圖像介面
- 1500 程式碼
- 1502、1504、1506 已執行區域
- 1508、1510、1512、1514 未執行區域
- 2200 報告
- 2202、2212 範圍
- 2204、2206、2208、2212 已存取之記憶體位置
- 2210 呼叫流樹
- 308 指令存取指示符
- 402 信號
- 406 資料存取指示符
- 502 快取
- 602 指令槽
- 606 指令槽
- 610、612、614 指示符
- 702 指令
- 708 處理器
- 712 記憶體位置
- 820 地址空間暫存器
- 900 頁表項目
- 920 實體頁地址欄位

五、中文發明摘要：

一種顯示程式碼之涵蓋資料之方法、裝置、及電腦指令。取得包含與此程式碼相關之指令存取指示符的涵蓋資料。辨識特定指令存取指示符，以形成已設指令存取指示符，其中特定指令存取指示符係由在資料處理系統中之處理器所設定之特定指令存取指示符，以回應該處理器執行該程式碼。每一已設指令存取指示符係與程式碼之一部份相關聯。產生此涵蓋資料之一顯示。此已設指令存取指示符係於其顯示中被辨識。其顯示可以是例如一圖像顯示。

六、英文發明摘要：

A method, apparatus, and computer instructions for presenting coverage data for code. The coverage data containing instruction access indicators associated with the code is obtained. Particular instruction access indicators that have been set by a processor in the data processing system in response to execution of the code by the processor are identified to form set instruction access indicators. Each set instruction access indicator is associated with a portion of the code. A presentation for the coverage data is generated. The set instruction access indicators are identified in the presentation. This presentation may be, for example, a graphical presentation.

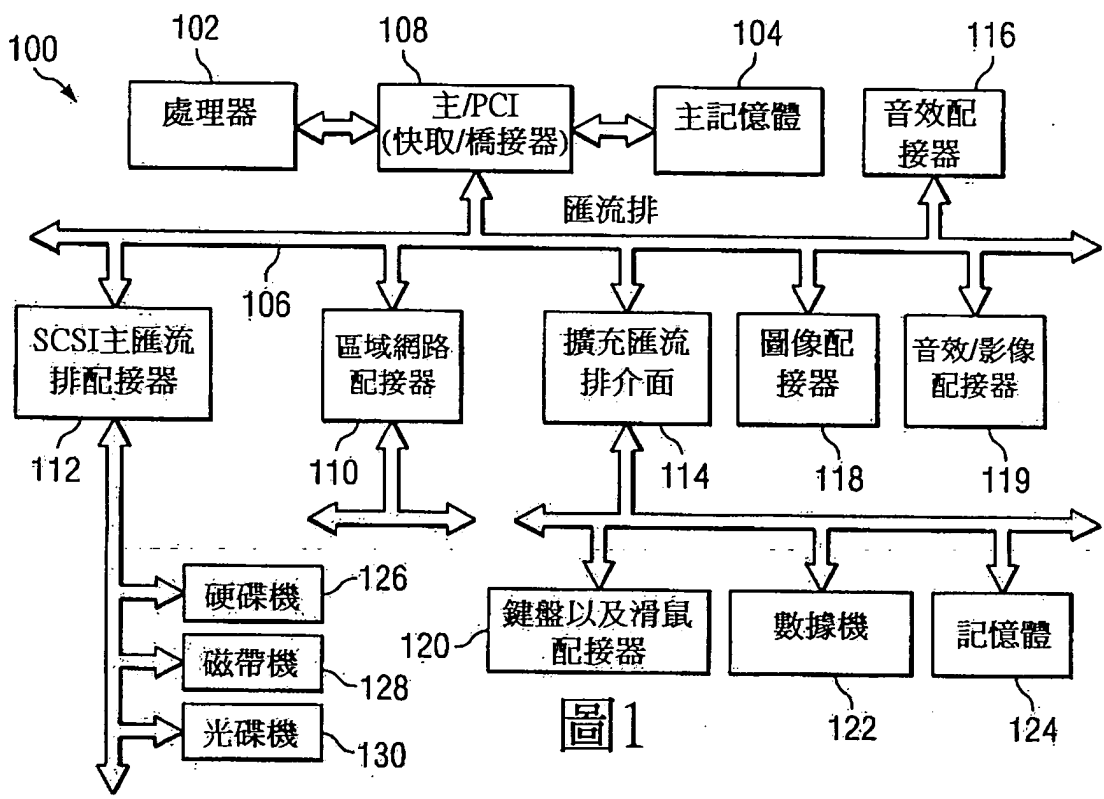


圖 1

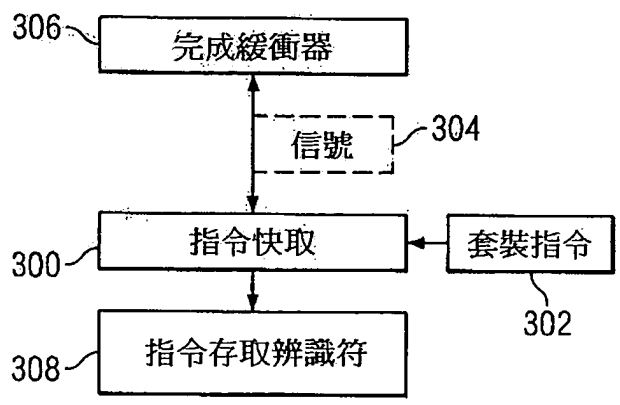


圖 3

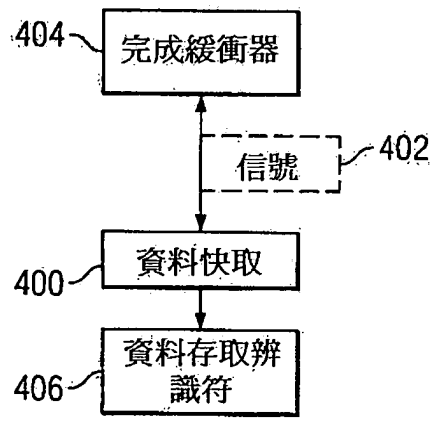


圖 4

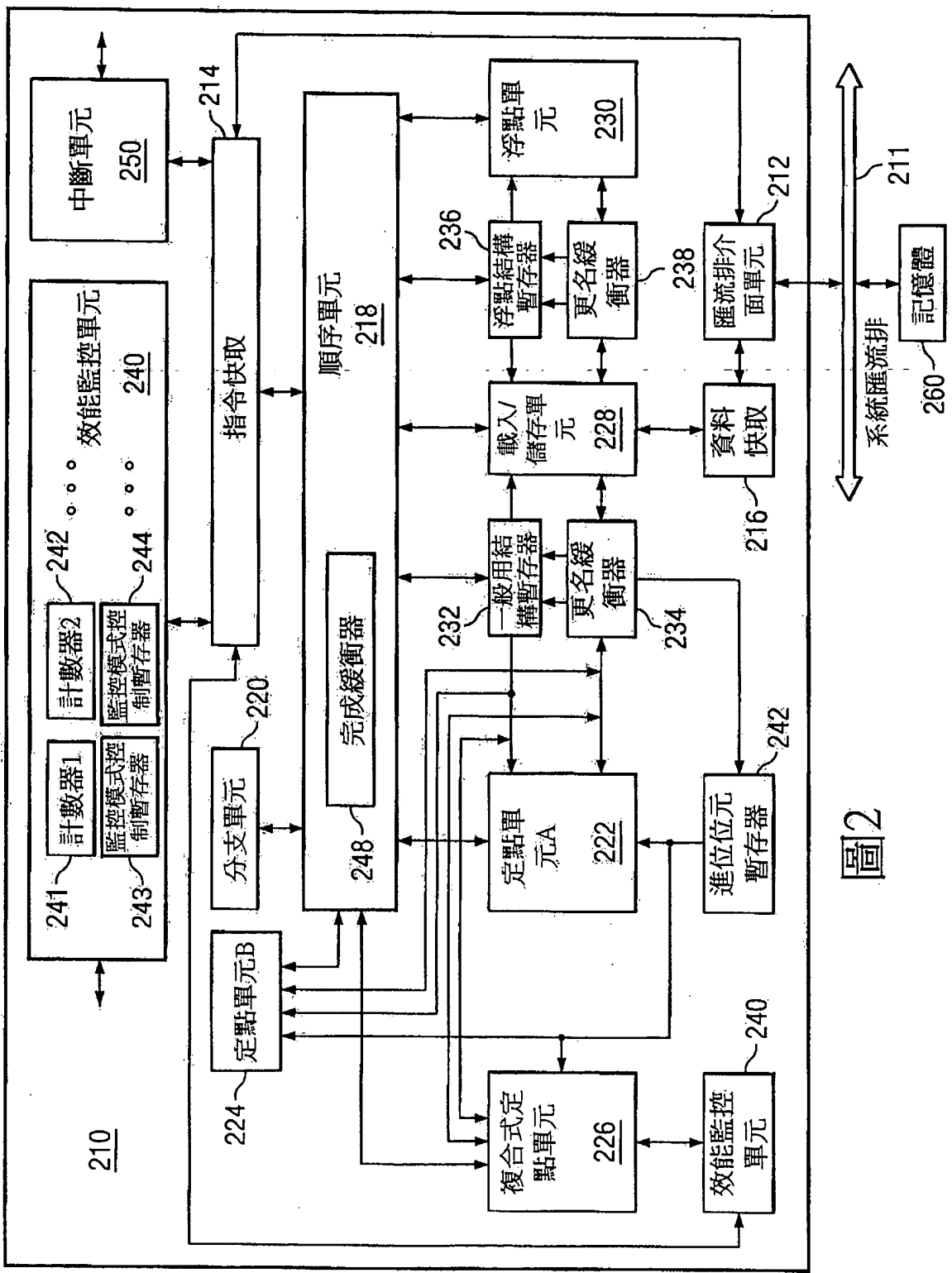


圖2

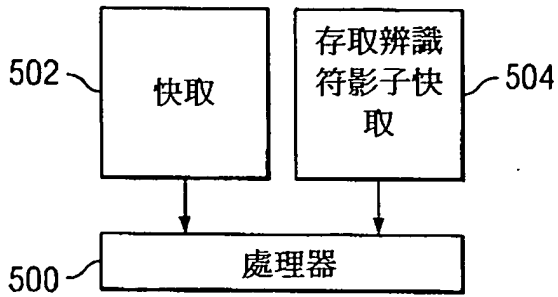


圖5

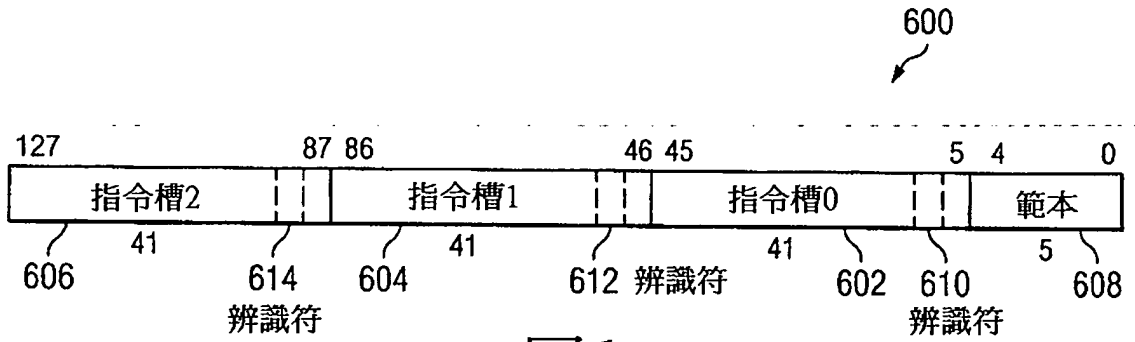


圖6

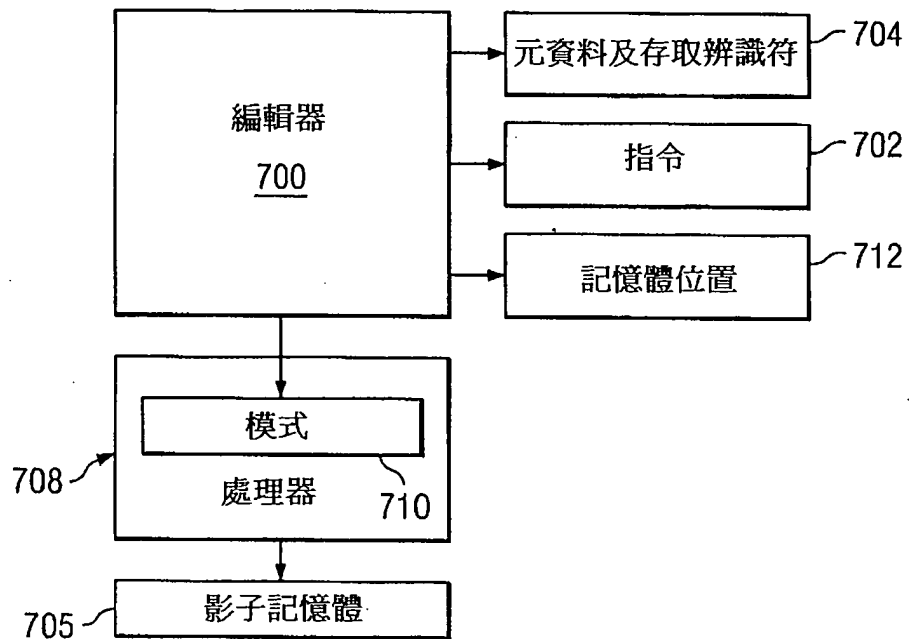


圖7

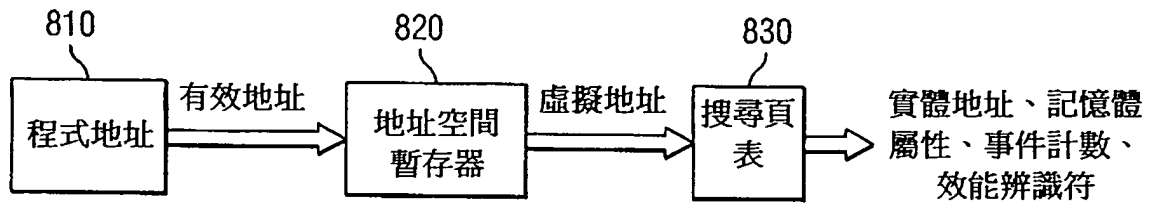


圖8

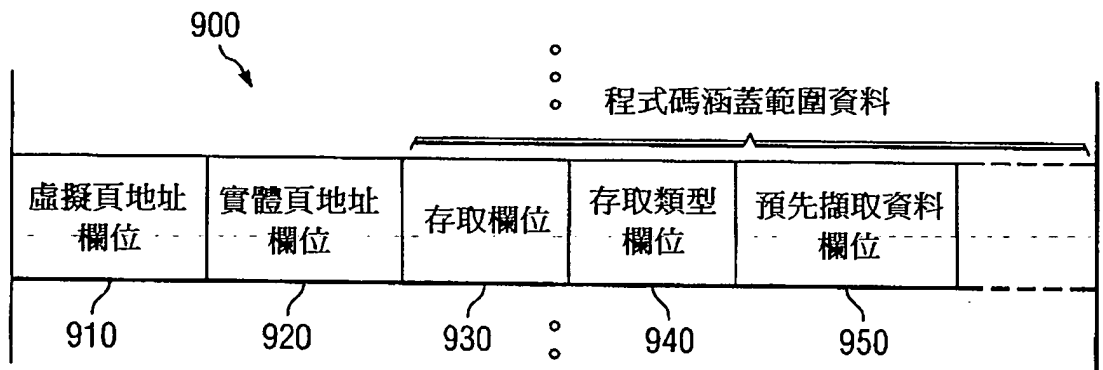


圖9

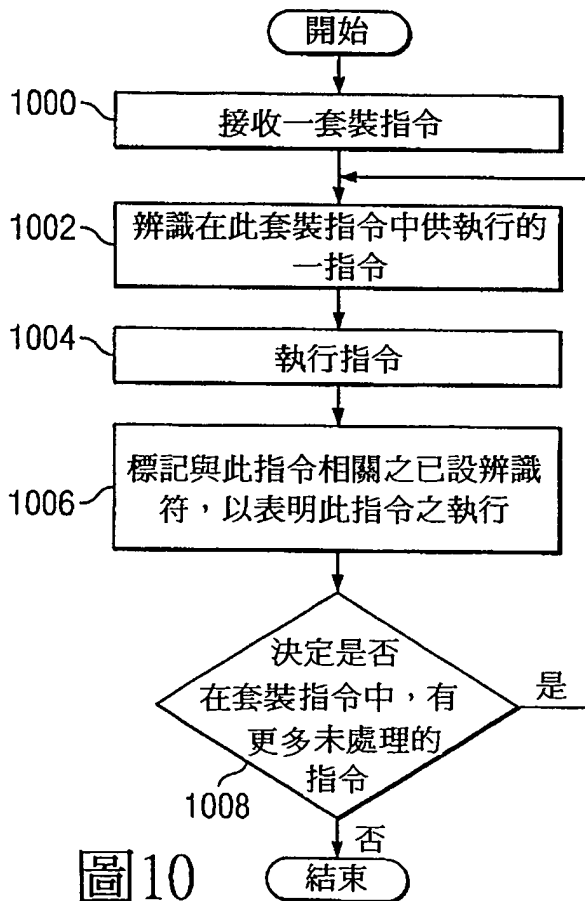


圖10

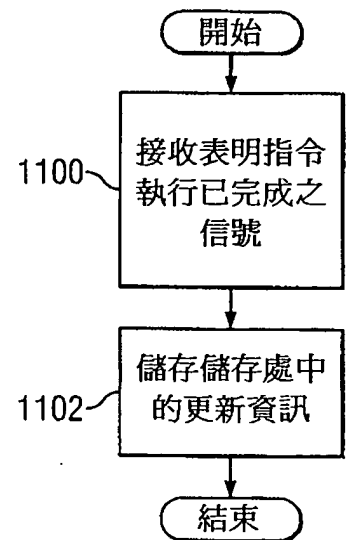


圖11

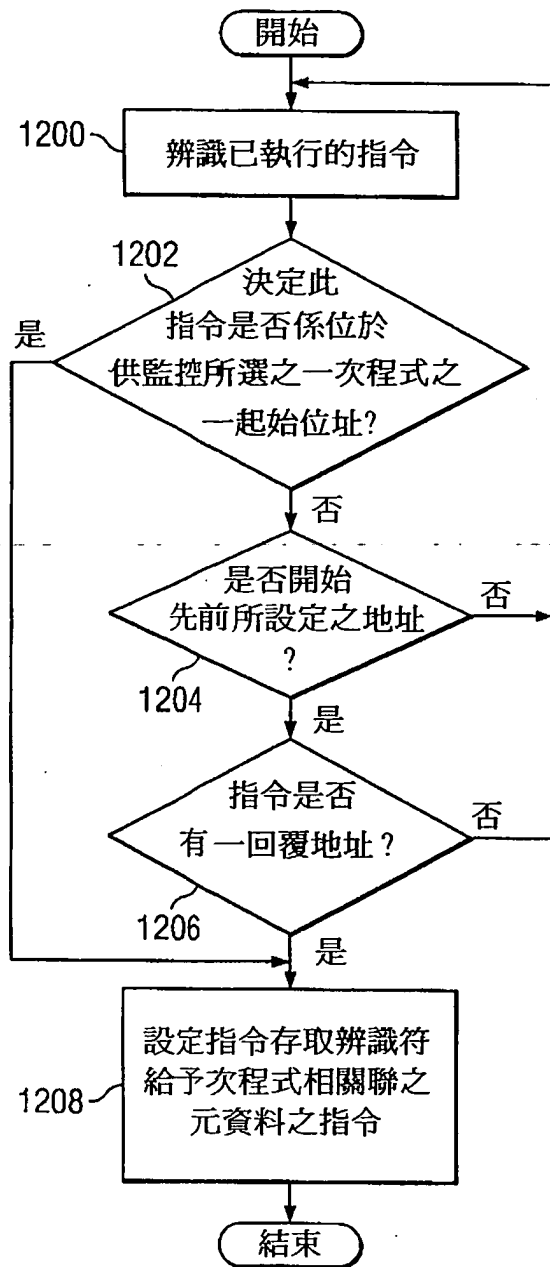


圖12

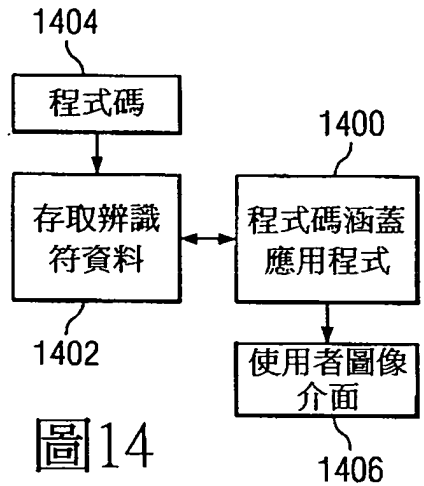


圖14

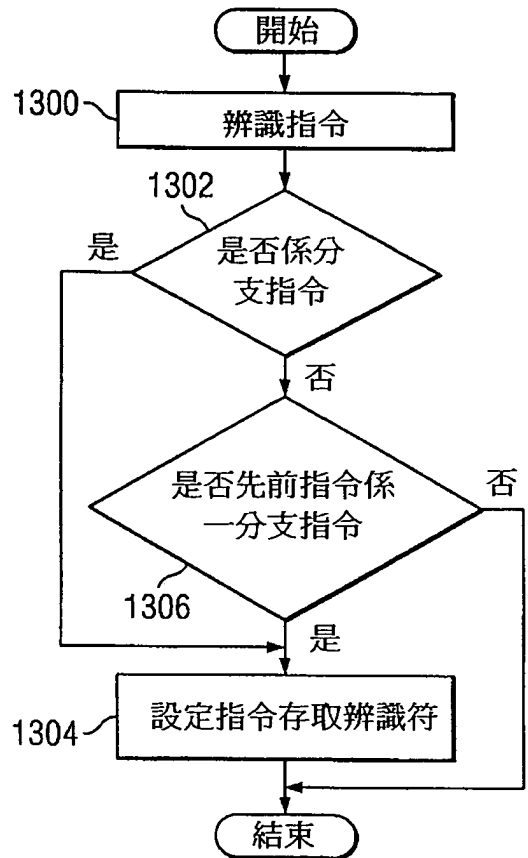


圖13

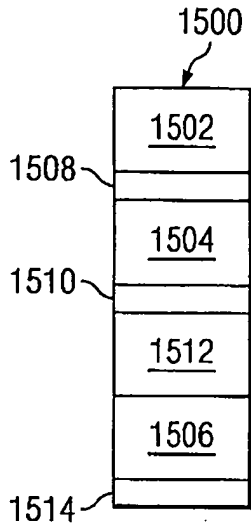


圖15

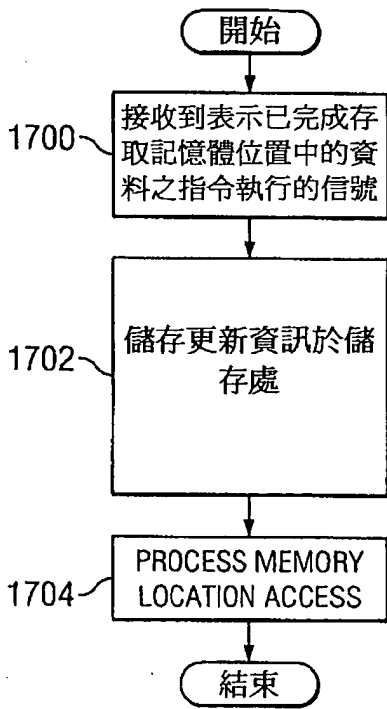


圖17

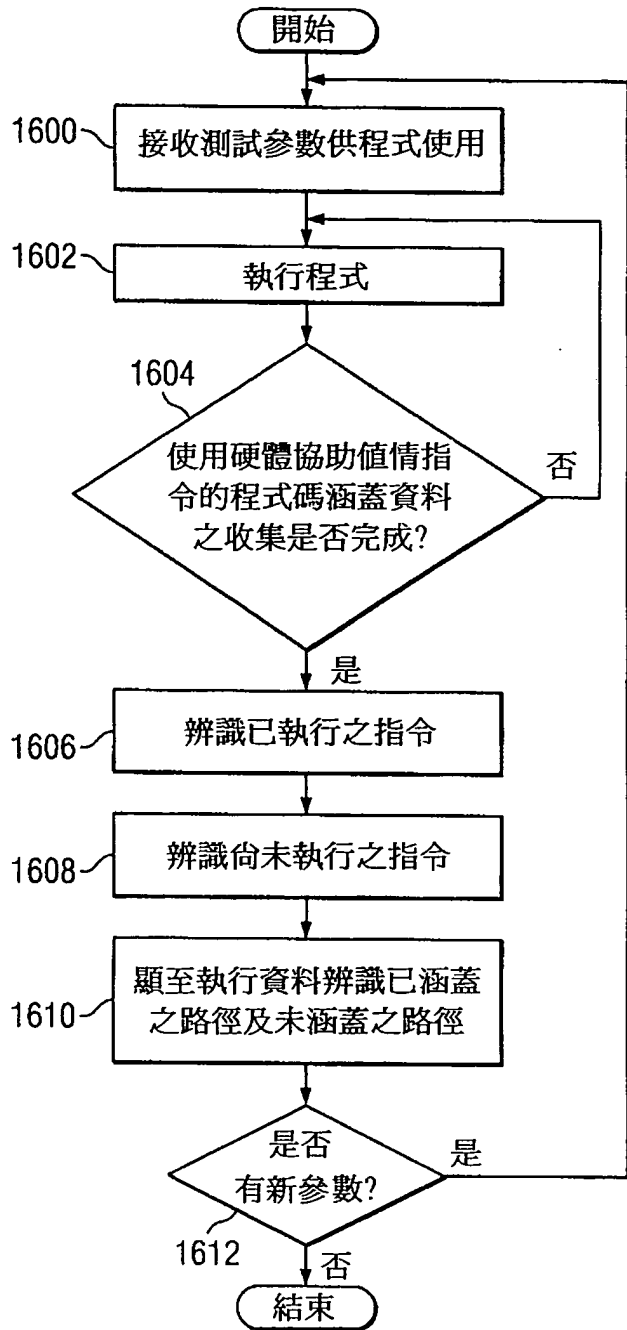


圖16

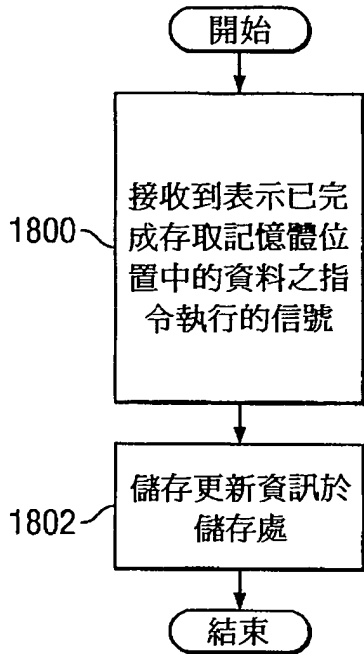


圖18

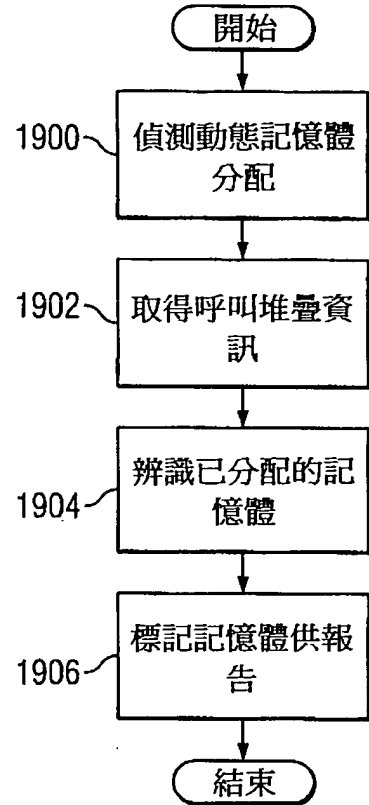


圖19

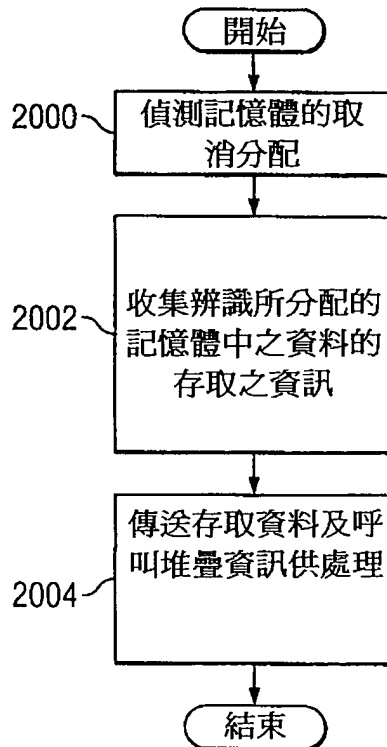


圖20

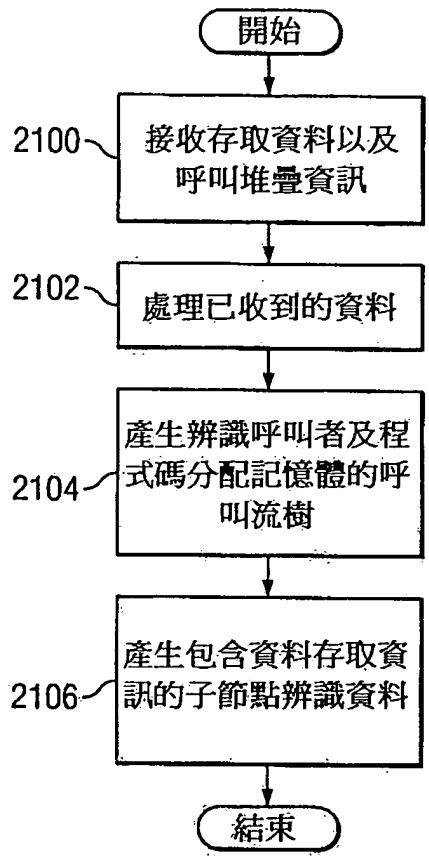


圖21

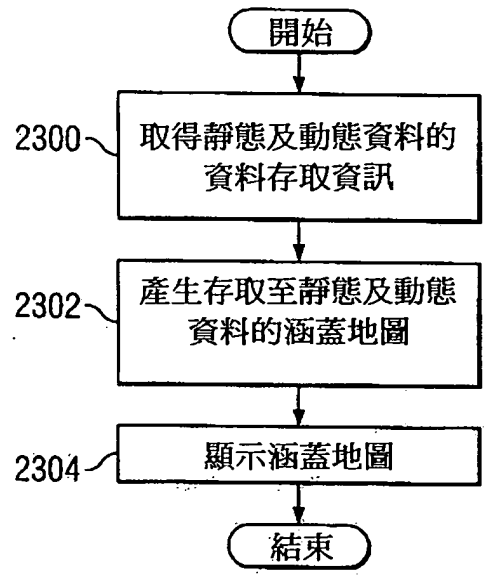


圖23

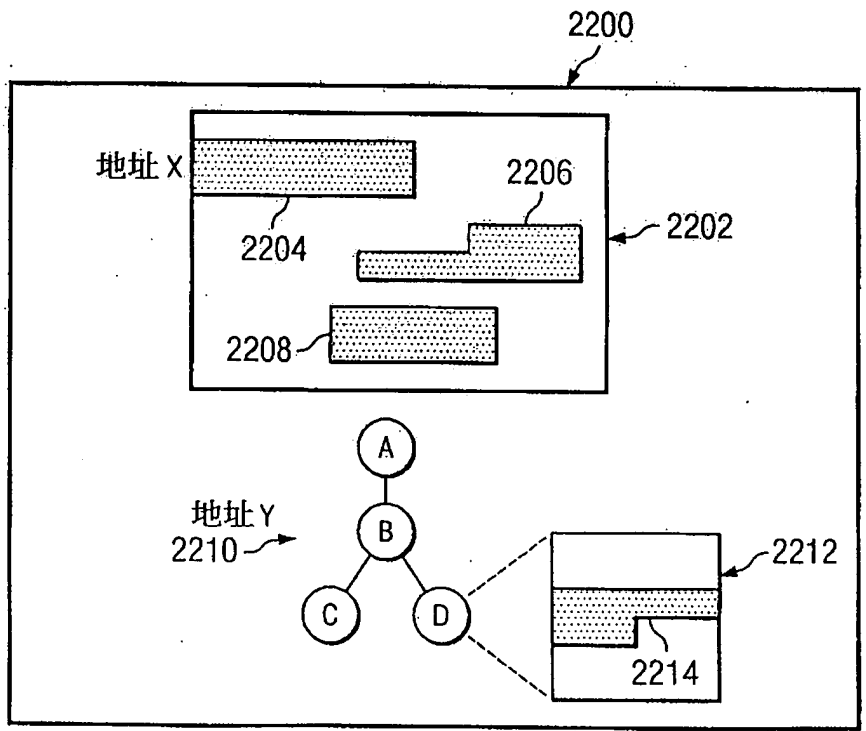


圖22

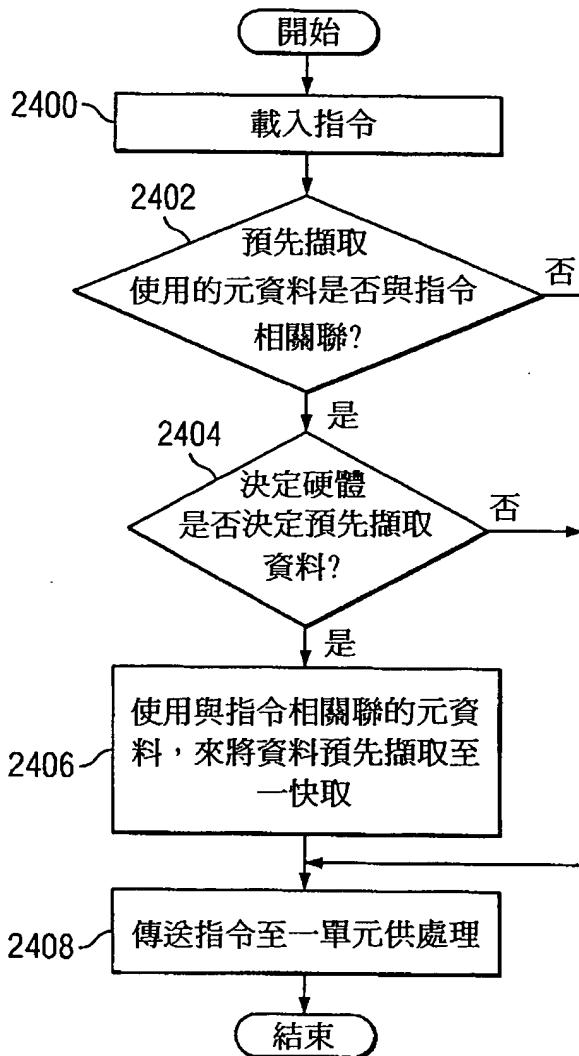


圖24

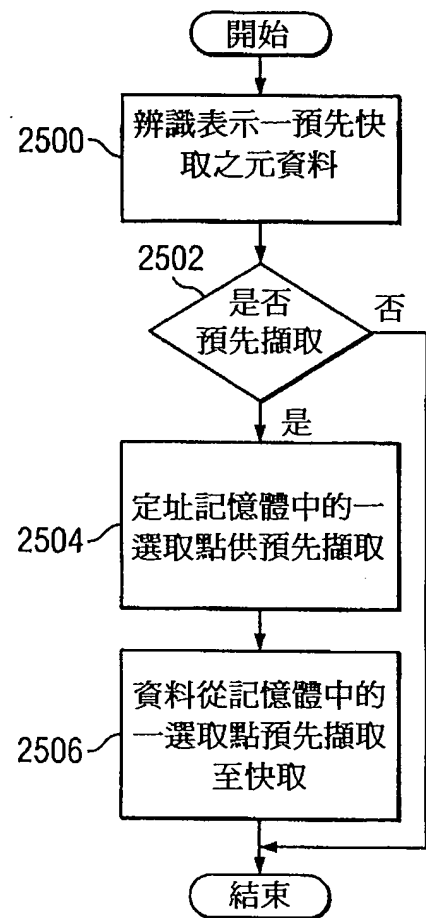


圖25

七、指定代表圖：

(一)本案指定代表圖為：圖 2。

(二)本代表圖之元件符號簡單說明：

210 處理器	211 系統匯流排
212 匯流排介面單元	214 指令快取
216 資料快取	218 順序單元
220 分支單元	222、224 定點單元
226 複合式定點單元	228 載入/儲存單元
230 浮點單元	232 一般用結構暫存器
234 定點更名緩衝器	236 浮點結構暫存器
238 浮點更名緩衝器	240 效能監控單元
241、242 計數器	
243、244 監控模式控制暫存器	
248 完成緩衝器	250 中斷單元
260 系統記憶體	

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

發明專利說明書

公告本

(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：94107739

※ 申請日期：94年3月14日

※ IPC 分類：G06F15/177 (2006.01)

G11B11/00 (2006.01)

一、發明名稱：(中文/英文)

顯示程式碼之涵蓋資料之方法系統及電腦程式產品

METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT

FOR PRESENTING COVERAGE DATA FOR CODE

二、申請人：(共1人)

姓名或名稱：(中文/英文)

萬國商業機器公司

INTERNATIONAL BUSINESS MACHINES CORPORATION

代表人：(中文/英文) 傑羅 羅森梭/ROSENTHAL, GERALD

住居所或營業所地址：(中文/英文)

美國紐約州 10504 亞芒克市新奧爾察德路

New Orchard Road, Armonk, NY 10504, U.S.A.

國籍：(中文/英文) 美國/US

三、發明人：(共3人)

姓名：(中文/英文)

1. 羅勃特 陶德 迪西/DIMPSEY, ROBERT TOD

2. 法蘭克 伊立特 藍威/LEVINE, FRANK ELIOT

3. 羅伯特 約翰 優哈特/URQUHART, ROBERT JOHN

國籍：(中文/英文)

1.-3. 均為美國/US

於 2003 年 9 月 30 日申請，標題為「計數指令及記憶體位置範圍之方法及裝置」之申請案，申請案號為 10/675,872，律師登錄號 AUS920030487US1；於 2004 年 1 月 14 日申請，標題為「維持業表中的效能監控結構供一電腦程式之監控效能使用之方法及裝置」之申請案，申請案號為 10/757,250，律師登錄號 AUS920030488US1；於 2004 年 1 月 14 日申請，標題為「提供記錄事件之前後操作符之方法及裝置」之申請案，申請案號為 10/757,192，律師登錄號 AUS920030543US1；標題為「使用硬體協助之程式緒堆疊追蹤及歸類符號資料以自主地決定電腦程式流之方法及裝置」之申請案，律師登錄號 AUS920030548US1；於 2004 年 1 月 14 日申請，標題為「偵測到錯誤快取線分享時自主地移動快取項目至專屬儲存處之方法及裝置」之申請案，申請案號為 10/757,227，律師登錄號 AUS920030555US1；於 2004 年 1 月 14 日申請，標題為「利用具有效能指示符及計數資訊的註解追蹤資訊將程式碼執行最佳化之方法及裝置」之申請案，申請案號為 10/757,197，律師登錄號 AUS920030556US1；標題為「資料存取涵蓋範圍之硬體協助之方法及裝置」之申請案，律師登錄號 AUS920040061US1；標題為「提供動態分配資料之資料存取涵蓋範圍之硬體協助之方法及裝置」之申請案，律師登錄號 AUS920040062US1；標題為「使用資料涵蓋範圍之硬體協助自主測示範例回饋之方法及裝置」之申請案，律師登錄號 AUS920040063US1；於 2004 年 1 月 14 日申請，標題為「預先擷取資料之硬體協助之方法及裝置」之申請案，律師登錄號 AUS920040064US1；標題為「從資料結構預先擷取資料之方法及裝置」之申請案，律

十、申請專利範圍：

1. 一種在一資料處理系統中顯示程式碼之涵蓋資料之方法，包含：

由一指令快取接收執行程式碼；

回應正由一執行單元執行之該程式碼，由一完成緩衝器發送一訊號至已執行該程式碼之指令快取，其中該完成緩衝器包含識別該執行已完成之該程式碼的資訊；

回應接收到來自該完成緩衝器之該訊號，由該指令快取設定一指令存取指示符(indicator)於元資料(meta data)中，其中該元資料包含形式為一指示器的一預先擷取指示符，識別該指示器將自一資料結構預先擷取的資料並藉由使用一偏移值或一位址識別該資料，其中每個指令存取指示符關聯至該程式碼之不同部分，且每個指令存取指示符在它所關聯之部分程式碼執行前係初始化為未設定，

取得涵蓋資料，該涵蓋資料係包含與該程式碼相關之指令存取指示符；

產生該涵蓋資料之一顯示，其中每個已設指令存取指示符係於該顯示中被辨識；及

在該執行單元之該程式碼執行期間，辨識尚未經設定之未設指令存取指示符(unset instruction access indicators)，其中每個未設指令存取指示符係關聯至該程式碼之一未執行部分，且每個未設指令存取指示符係辨識於該顯示中。

2. 如請求項1所述之方法，其中該已設指令存取指示符係

利用一第一顏色，辨識於該顯示中，而其中該未設指令存取指示符係利用一第二顏色辨識於該顯示中。

3. 如請求項 1 所述之方法，其中該已設指令存取指示符係利用一圖像指示符，辨識於該顯示中，而其中該未設指令存取指示符係利用該圖像指示符，辨識於該顯示中。
4. 如請求項 1 所述之方法，其中該產生步驟之執行係為回應一事件。
5. 如請求項 4 所述之方法，其中該事件係該程式碼之該執行之一完成、一時間之逾期、及該程式碼中一所選類型之指令之該執行之至少之一。
6. 如請求項 1 所述之方法，其中該部分程式碼係該程式碼中之一單一指令，且其中該程式碼中之每一指令係與一不同的指令存取指示符相關聯。
7. 如請求項 1 所述之方法，其中該部分程式碼係該程式碼中之一次常式 (subroutine)。
8. 如請求項 1 所述之方法，其中該部分程式碼係該程式碼中之一分支指令。
9. 一種顯示程式碼之涵蓋資料之資料處理系統，包含：
接收裝置供由一指令快取接收執行程式碼；
發送裝置供回應正由一執行單元執行之該程式

碼，由一完成緩衝器發送一訊號至已執行該程式碼之指令快取，其中該完成緩衝器包含識別該執行已完成之該程式碼的資訊；

發送裝置供回應接收到來自該完成緩衝器之該訊號，由該指令快取設定一指令存取指示符(indicator)於元資料(meta data)中，其中該元資料包含形式為一指示器的一預先擷取指示符，識別該指示器將自一資料結構預先擷取的資料並藉由使用一偏移值或一位址識別該資料，其中每個指令存取指示符關聯至該程式碼之不同部分，且每個指令存取指示符在它所關聯之部分程式碼執行前係初始化為未設定；

取得裝置，供取得涵蓋資料，該涵蓋資料係包含與該程式碼相關之指令存取指示符；

產生裝置，供產生該涵蓋資料之一顯示，其中每個已設指令存取指示符係於該顯示中被辨識；及

辨識裝置，供在該執行單元之該程式碼執行期間，辨識尚未經設定之未設指令存取指示符(unset instruction access indicators)，其中每個未設指令存取指示符係關聯至該程式碼之一未執行部分，且每個未設指令存取指示符係辨識於該顯示中。

10. 如請求項9之資料處理系統，其中該已設指令存取指示符係利用一第一顏色，辨識於該顯示中，而其中該未設指令存取指示符係利用一第二顏色，辨識於該顯示中。
11. 如請求項9之資料處理系統，其中該已設指令存取指示符係利用一圖像指示符，辨識於該顯示中，而其中該未

設指令存取指示符係利用該圖像指示符，辨識於該顯示中。

12. 如請求項 9 之資料處理系統，其中該產生裝置之執行係為回應一事件。
13. 如請求項 12 之資料處理系統，其中該事件係該程式碼之該執行之一完成、一時間之逾期、及該程式碼中一所選類型之指令之該執行之至少之一。
14. 如請求項 9 之資料處理系統，其中該部分程式碼係該程式碼中之一單一指令，且其中該程式碼中之每一指令係與一不同的指令存取指示符相關聯。
15. 如請求項 9 之資料處理系統，其中該部分程式碼係該程式碼中之一次常式。
16. 如請求項 9 之資料處理系統，其中該部分程式碼係該程式碼中之一分支指令。
17. 一種記載可以執行顯示程式碼之涵蓋資料之電腦程式產品，該電腦程式包含：
 - 第一指令，供接收裝置供由一指令快取接收執行程式碼；
 - 第二指令，供回應正由一執行單元執行之該程式碼，由一完成緩衝器發送一訊號至已執行該程式碼之指令快取，其中該完成緩衝器包含識別該執行已完成

之該程式碼的資訊；

發送裝置供回應接收到來自該完成緩衝器之該訊號，由該指令快取設定一指令存取指示符(indicator)於元資料(meta data)中，其中該元資料包含形式為一指示器的一預先擷取指示符，識別該指示器將自一資料結構預先擷取的資料並藉由使用一偏移值或一位址識別該資料，其中每個指令存取指示符關聯至該程式碼之不同部分，且每個指令存取指示符在它所關聯之部分程式碼執行前係初始化為未設定；

第四指令，供取得涵蓋資料，該涵蓋資料係包含與該程式碼相關之指令存取指示符；

第五指令，供產生該涵蓋資料之一顯示，其中每個已設指令存取指示符係於該顯示中被辨識；及

第六指令，供在該執行單元之該程式碼執行期間，辨識尚未經設定之未設指令存取指示符 (unset instruction access indicators)，其中每個未設指令存取指示符係關聯至該程式碼之一未執行部分，且每個未設指令存取指示符係辨識於該顯示中。

18. 如請求項 17 之電腦程式產品，其中該已設指令存取指示符係利用一第一顏色，辨識於該顯示中，而其中該未設指令存取指示符係利用一第二顏色，辨識於該顯示中。

19. 如請求項 17 之電腦程式產品，其中該已設指令存取指示符係利用一圖像指示符，辨識於該顯示中，而其中該未設指令存取指示符係利用該圖像指示符，辨識於該顯

示中。

20. 如請求項 17 之電腦程式產品，其中該第三指令之執行係為回應一事件。
21. 如請求項 20 之電腦程式產品，其中該事件係該程式碼之該執行之一完成、一時間之逾期、及該程式碼中一所選類型之指令之該執行之至少之一。