

I314701

98年6月17日修正本

第93102967 號專利案 98年4月修正

公告本

## 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：93102967

※ 申請日期：93年2月9日

※IPC 分類：G06F9/30  
(2006.01)

### 一、發明名稱：(中文/英文)

具有能加入條件行為至關聯程式的預測指令功能的處理資料設備、方法、及記錄相關指令的電腦程式產品

DATA PROCESSING APPARATUS AND METHOD WITH  
PREDICATION INSTRUCTION FUNCTION CAPABLE OF ADDING  
CONDITIONAL BEHAVIOR TO ASSOCIATED PROGRAMS, AND  
COMPUTER PROGRAM PRODUCT RECORDING RELATED  
INSTRUCTIONS

### 二、申請人：(共1人)

姓名或名稱：(中文/英文)

英商·ARM 股份有限公司

ARM LIMITED

代表人：(中文/英文)

方奈爾 SJ

FUNNELL, SJ

住居所或營業所地址：(中文/英文)

英國劍橋郡 CB1 9NJ 綺麗亨頓市富爾本路 110 號

110 Fulbourn Road, Cherry Hinton, Cambridge, CB1 9NJ, United Kingdom

國籍：(中文/英文)

英國/UNITED KINGDOM

### 三、發明人：(共2人)

姓名：(中文/英文)

- 1.希爾大衛詹姆士/SEAL, DAVID JAMES
- 2.瓦絲金維拉迪彌爾/VASEKIN, VLADIMIR

國 籍：(中文/英文)

- 1.英國/ UNITED KINGDOM
- 2.俄羅斯&英國/ RUSSIA & UNITED KINGDOM

#### 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：

英國；2003年6月5日；0312977.2

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。

- 1.希爾大衛詹姆士/SEAL, DAVID JAMES
- 2.瓦絲金維拉迪彌爾/VASEKIN, VLADIMIR

國 籍：(中文/英文)

- 1.英國/ UNITED KINGDOM
- 2.俄羅斯&英國/ RUSSIA & UNITED KINGDOM

#### 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：

英國；2003年6月5日；0312977.2

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。

## 玖、發明說明：

### 【發明所屬之技術領域】

本發明與資料處理系統的領域關聯。本發明尤指與在此類資料處理系統中程式指令之條件執行關聯。

### 【先前技術】

吾人已知提供資料處理系統以回應包含程式指令的指令集，其根據先前的處理操作結果，條件化執行或不執行。舉一示例，ARM 32 位元指令集在上述所有指令都徹底條件化，包括 4 位元條件碼欄位，其指定將禁止或允許執行的條件。

當徹底條件化的 32 位元 ARM 指令集在它的條件行為中功能強大時，對每一指令內條件碼提供的 4 位元，表示該指令集的位元編碼空間之重要部分。在許多情況中，不使用該些條件碼的全部能力，亦不使用對該些條件碼提供的 4 位元。

吾人亦知提供非條件化指令集，例如 ARM 處理器所支援的 Thumb 指令集。不對條件碼提供任何指令位元空間的方式，允許更小的 Thumb 指令，從而改進碼密度。然而，該方法的缺點是完成某些操作所需的指令數量也許更大，以及處理因而更慢，沒有能力利用 32 位元 ARM 指令集的條件碼。

吾人已知，從其他例如 PA-RISC 處理器之處理器，提供能夠根據它們的結果消除下一指令的指令。此類指令的

例子可能係一比較操作，其根據所比較的數量是否相等，禁止或允許下一後續指令執行。當此類消除下一次指令之類型有用時，它們限制了靈活性，因為它們僅用於依據自己的處理消除下一指令。

## 【發明內容】

本發明之一態樣提供用以處理資料之設備，該設備包含：

資料處理邏集，可操作以執行資料處理操作；及一指令解譯器，可操作以解譯程式指令，以控制該資料處理邏輯執行該些資料處理操作，其中：

依據用於處理資料集之該設備的一或多種條件狀態，藉由執行一或多種程式指令而非該預測指令，該指令解譯器回應一預測指令，控制該資料處理邏輯執行或不執行一或多種關聯的程式指令。

本發明確認，特別用途預測指令的供應可能極為有利，因為其能夠依據可較彈性指定之處理器的條件狀態，條件化獲得其他指令的執行。因此，程式員和編譯器更能利用此類預測指令產生更小和更快速的程式。

而該預測指令亦可能僅影響一其他指令，在較佳實施中，提供該預測指令以控制關聯的複數程式指令的執行。

當與部分或全部條件化關聯的程式指令一起使用該預測指令，本發明的較佳實施中，預測指令與非條件化之關聯程式指令一起使用。因此，該預測指令能加入條件化指

令行為至其他非條件化指令集。

當該預測指令以條件狀態為基礎時，它對關聯程式指令的控制能夠採取多種形式，在較佳實施中使用條件碼旗標，例如，一零旗標、一相等旗標、一攜帶旗標等等，以對該預測指令提供輸入。

由控制關聯的程式指令的預測指令衡量的條件狀態，可以在不同時間根據一特定系統的需要進行衡量。一旦執行該預測指令以及得到用以控制所有關聯程式指令的衡量結果，則可能衡量該些條件狀態。其有利於有可能在一指令管道(pipeline)早期徹底衡量該預測指令，以及因此以使快速和簡化的執行模式，自該指令管道移除該預測指令。在一替代性的實施例中，亦可以在每一關聯程式指令執行之前，衡量該些條件狀態，以使例如可令一些關聯的指令通過該條件碼，而其他關聯指令則不行。該方法具有以適用於許多一般型態之資料處理操作的方法，對條件化行為允許較佳程度之控制的優點，又可能需要以此類條件化行為部署該資料處理操作。

吾人將了解，該關聯指令可以具有相對於該預測指令之多種相對位置，在較佳實施中，該關聯的程式指令立即跟隨在該預測指令之後。這就部署和設計的用途而言是方便的。

亦有可能安排預測指令以多種方法控制關聯程式指令。在一可能的較佳實施中，把關聯的程式指令被安排至指令之兩區塊中，而其一區塊在一條件碼通過時執行，而

另一在條件碼失敗時執行。其可提供相似於「If...then...else」型態之指令。

在此類實施例中，不同區塊指令可以具有相同大小，具有由預測指令本身中的欄位所指定之不同大小尤佳。其允許彈性之區塊大小，便於控制該預測指令的分支及其他行為的區塊大小，以及改善碼密度。

其亦可能對一特定集合之預定條件提供預測指令，在較佳實施中，預測指令包括一可變欄位，透過它可以指定該些預定的條件。

另一種賦予有益程度之彈性之預測指令型態，以及尤其能允許插入將能或將不能通過它們的條件碼指令，係包含一或多種欄位之預測指令，依據對一或多種條件狀態與一或多種預測狀態之比較，該一或多種欄位之每一者各自指定一關聯程式指令執行與否。

尤其在較佳實施中，預測指令能夠各自包含為每一關聯指令指定的一欄位，其指定是否一或多種條件狀態或該一或多種條件狀態之一組欲與預測狀態比較，以決定是否執行各自的關聯程式指令。因此，能夠指定個別的關聯程式指令，以執行"Then"或"Else"條件。

為了協助處理系統的控制，藉由一被測指令計數暫存器補足正常的程式計數暫存器，該預測指令計數暫存器可操作以儲存一計數值，其指示已執行屬於預測指令之一或多種關聯程式指令的多少。預測指令在其中有效地具有一編碼之觸及範圍(reach)，其指定它控制了多少關聯程式指

令，以及藉由該觸及範圍，系統需要追蹤多遠之已進行執行程序，並且何時超過了該觸及範圍。

該被測指令計數暫存器在處理異常和自異常返回的方面特別有用，據以當自異常返回時，可以恢復該被測指令計數暫存器，以及用它的值來決定是否所返回至的程式指令具有由一前面的預測指令所影響之行為。

亦有可能控制是否關聯程式指令依據預測指令執行，也可以用預測指令來修改當關聯指令，以使在它們執行時，它們有不同操作。它能夠視為該指令集之有效地延伸。

修改關聯程式指令之一特別的較佳方法，是當它們屬於含有一特別可程式化欄位的一特定預測指令時，分隔它們以防止改變一或多種條件狀態。因此，條件碼旗標更新能夠由一合適預測指令禁止。

本發明之另一態樣提供一種處理資料之方法，該方法包含：

以資料處理邏輯執行資料處理邏操作；及以一指令解譯器解譯程式指令，以控制該資料處理邏輯執行該些資料處理操作，其中：

依據用於處理資料集之該設備的一或多種條件狀態，藉由執行一或多種程式指令而非該預測指令，回應一預測指令，控制該資料處理邏輯執行或不執行一或多種關聯的程式指令。

本發明之更進一步態樣提供一種種式產品，其包含一種電腦程式產品，可操作以控制用以處理資料之一設備，



其具有資料處理邏輯，可操作以執行資料處理操作；以及一指令解譯器可操作以解譯程式指令，以控制該資料處理邏輯執行該些資料處理操作，該些電腦程式操作包含：

一預測指令，可操作以依據用於處理資料集之該設備的一或多種條件狀態，藉由執行一或多種程式指令而非該預測指令，控制該資料處理邏輯執行或不執行一或多種關聯的程式指令。

本發明之上述及其他物件、特徵及優點將清楚詳述於參照附圖說明之下列實施例中。

#### 【實施方式】

第 1 圖圖示一處理器核心型態之一資料處理設備，包含一程式暫存器組 (program register) 4、一乘法器 (multiplier) 6、一轉換器 (shifter) 8、一加法器 (adder) 10、一指令解譯器 (instruction decoder) 12、一指令管道 (instruction pipeline) 14、一預取單元 (prefetch unit) 16、一中斷控制器 (interrupt controller) 18、一程式計數暫存器 (program counter register) 20 及一預測指令計數暫存器 (predication instruction counter register) 22。吾人將了解處理器核心 2 通常包含許多其他電路元件，如同本項領域者所熟知者，但為了清楚說明之故，在第 1 圖中略去了那些電路元件。

在操作方面，從一記憶體 (未見於圖示) 取得程式指令，以及供給通過各種管道步驟之指令管道 14，例如，抓取、解譯和執行，如同本項領域者所熟知者。當從程式暫

寄存器組 4 讀取資料值以產生欲寫回程式暫存器組 4 之一結果值，來自指令管道 14 之解譯步驟之指令被提供給指令解譯器 12，其中使用它們產生設定及控制資料處理操作/控制之控制訊號，又該資料處理資料處理操作/控制係由乘法器 6、轉換器 8、加法器 10 及其他電路元件例如 ALU(未見於圖示)所執行。預取單元 6 用於控制指令位址產生，其用以取得程式指令 I 至指令管道 14。程式計數暫存器 20(也許在程式暫存器組 4 內是程式暫存器之一)供給執行目前指令的一位址值，或對該值具有一預定偏移量的一位址(例如，+4)。一環境暫存器 22 也向當從中斷返回時使用的預取單元 16 供給一值，如同中斷控制器 18 所控制般，因此即使因為異常的發生中斷了執行，一預測指令仍將影響屬於該預測指令的關聯程式指令的部分完成的序列。

第 2 圖圖示一第一實施例，其中一預測指令 IT 在一程式指令串流中饋送至指令譯碼器 12。當指令譯碼器 12 偵知此一預測指令時，將其解譯並有效地添加條件碼至指令串流內的預測指令之後的關聯程式指令。吾人將了解，該例中指令串流內的程式指令通常是非條件化指令，例如 16 位元 Thumb 指令通常沒有與它們關聯的條件碼。然而，支援 Thumb 指令集的處理器也支援徹底條件化的 32 位元 ARM 指令集。當來自 ARM 指令集指令被解譯時，則它們全部包括條件碼，其沿著指令管道 14 從指令譯碼器 12 向下傳遞，以影響該些指令爾後執行與否。因此，指令管道 14 已經知道條件碼且具有用於解讀和回應與程式指令

關聯的條件碼之架構和機制。在一縱觀的水平上，由此可見，預測指令在指令串流內在其後的一或多種關聯程式指令中添加條件碼，當其除了添加條件碼資訊外不提供處理功能，從該指令串流移走自己。

第 3 圖之圖示進一步詳述第 2 圖之預測指令 IT。尤其是，該指令指定一條件碼 "cond" 和的一遮罩值 "xyx" 以指示是否跟在 IT 指令從的 1、2、3 或 4 程式指令是屬於它的預測，以及是否因此可應用相反的條件碼。根據是否對應值 x, y, z 是否各自為其後之第二、其後之第三以及其後之第四關聯程式指令及旗標值，能夠應用該條件碼或該組條件碼。第 3 圖圖示該等值 x, y, z 如何形成以數值 "1" 結束之遮罩值。

第 4 圖圖示此一指令之一示例，其中由預測指令預測四個關聯指令。認為該些指令之第一個可預測為 "Then" 形式，因此如果符合條件碼將執行它。依據該 "Then" 條件或該 "Else" 條件，能夠分別指定執行之第二、第三和第四個關聯指令。在該例中，該四個關聯程式指令具有 "Then, Then, Else, Then" 之形式。在遮罩值 24 內，第二、第三、及第四個關聯指令的遮罩值位元 x, y, z 分別是 0, 1, 0 以及該遮罩值係以一數值 "1" 結尾。

在該例中指定的條件碼是 "1010"。把該條件碼以它的正常形式應用於第一關聯指令。而後使遮罩值左移一位置並且把移出遮罩值之最重要位元移至欲應用至下一指令之條件碼的重不重要位置。因此，欲應用至下一指令之條件

碼能夠依據它的遮罩值中的最重要位元，在它的正常型態和它的補數型態間轉換，在該例中，可以了解在預測指令之後的第一、第二及第四指令屬於正常條件碼，而預測指令之後的第三指令屬於條件碼的補數。當遮罩值"1000"被偵知時，其指示已經達到最後的被測指令指令，並且預測指令的影響不會延伸至之後的程式指令。

指令之類型將在第 3 圖和第 4 圖中說明如下：

## IT

IF-Then(IT)指令使記憶體中的下一 N 指令條件化。在 IT 指令後和由其影響的 N 指令集稱作它的 "IT block"。

當它們在 IT block 內執行時，亦有可能重新定義一些 (或所有) 指令的功能。為例如，除了 CMP/CMN /TST 以外的 Thumb 16 位元指令可以不是旗標設定。在此一情況下，IT 指令能夠與 AL 條件並用以得到對自己之後的影響。

一額外的 PSR 位元 (或專屬的 IT 狀態暫存器) 能夠被用來保持 IT 狀態。它允許在 IT block 內的中斷。

在下面的例子中，定義 4 指令中具有最大 IT block 大小的 Thumb 16 位元 IT 指令。由相同條件或相反版本的條件 (即，EQ 或 NE 等等) 預測 IT block 內的所有指令。

### Thumb IT

#### 語法

```
IT{<x>{<y><z>}} <cond>
```

其中：

<x> 如果存在，是字母 "T" 和 "E" 之一，和表示一第二後續指令欲由 IT 指令所影響。如果 <x> 是 "T" ("Then")，附加至該指令的條件是 <cond>; 如果 <x> 是 "E"，其為 <cond> 的相反，係由邏輯反轉 <cond> 的最不重要位元獲得。如果 <x> 是 "omitted(省略)" (和因此 <y> 和 <z> 亦為省略)，則只有一後續指令係由 IT 指令所影響。

<y> 如果存在，是字母 "T" 和 "E" 之一，和表示一第三後續指令欲由 IT 指令所影響。如果 <y> 是 "T" ("Then")，附加至該指令的條件是 <cond>; 如果 <y> 是 "E"，其為 <cond> 的相反，係由邏輯反轉 <cond> 的最不重要位元獲得。如果 <x> 是 "omitted(省略)" (和因此 <z> 亦為省略)，則少於 3 個後續指令係由 IT 指令所影響。

<z> 如果存在，是字母 "T" 和 "E" 之一，和表示一第四後續指令欲由 IT 指令所影響。如果 <z> 是 "T" ("Then")，附加至該指令的條件是 <cond>; 如果 <z> 是 "E"，其為 <cond> 的相反，係由邏輯反轉 <cond> 的最不重要位元獲得。如果 <z> 是 "omitted(省略)"，則少於 4 個後續指令係由 IT 指令所影響。

<cond> 是欲應用至第一後續指令的條件，和如同 <x>、<y> 和 <z> 所指定般可能應用於更多後續指令。

<x>、<y>和<z>值根據下表，決定指令遮罩欄位的值。

<x>	<y>	<z>	Mask			
			[3]	[2]	[1]	[0]
Omitted	Omitted	Omitted	1	0	0	0
T	Omitted	Omitted	cond[0]	1	0	0
E	Omitted	Omitted	NOT cond[0]	1	0	0
T	T	Omitted	cond[0]	cond[0]	1	0
E	T	Omitted	NOT cond[0]	cond[0]	1	0
T	E	Omitted	cond[0]	NOT cond[0]	1	0
E	E	Omitted	NOT cond[0]	NOT cond[0]	1	0
T	T	T	cond[0]	cond[0]	cond[0]	1
E	T	T	NOT cond[0]	cond[0]	cond[0]	1
T	E	T	cond[0]	NOT cond[0]	cond[0]	1
E	E	T	NOT cond[0]	NOT cond[0]	cond[0]	1
T	T	E	cond[0]	cond[0]	NOT cond[0]	1
E	T	E	NOT cond[0]	cond[0]	NOT cond[0]	1
T	E	E	cond[0]	NOT cond[0]	NOT cond[0]	1
E	E	E	NOT cond[0]	NOT cond[0]	NOT cond[0]	1

註：該語法的基本概念是例如 "ITTET EQ" 說明第 1、第 2 及第 4 後續指令（由表中符號 "T" 之位置表示）具有 EQ 條件附加於它們，並且第 3 項（由 "E" 表示）具有一 NE 條件附加於它。

操作

```
if CPSR[26,25,15:10] != 0b000000000 then
```

```
    UNPREDICTABLE
```

```
else
```

```
    if mask == 0b00000 then
```

```
        UNPREDICTABLE
```

```
    else
```

```
        CPSR[15:12] = cond
```

```
        CPSR[11,10,26,25] = mask?
```

//而後指令逐一更新上述 CPSR 位元，導致 IT 指令的條件化執行作用發生。

8 IT 執行狀態位元編譯 256「指令間狀態(inter-instruction states)」之一，如下：

CPSR

狀態

2 2 1 1 1 1 1 1

6 5 5 4 3 2 1 0

P4	1	condbase	P1	P2	P3	欲條件化地以條件 (condbase,P1)·(condbase,P2) (condbase,P3)·(condbase,P4) 執行下 4 個指令
----	---	----------	----	----	----	---

1	0	condbase			P1	P2	P3	欲條件化地以條件 (condbase,P1)·(condbase,P2) (condbase,P3)執行下 3 個指 令
0	0	condbase			P1	P2	1	欲條件化地以條件 (condbase,P1)·(condbase,P2) 執行下 2 個指令
0	0	condbase			P1	1	0	欲條件化地以條件 (condbase,P1)執行下一個指 令
0	0	N(nonzero)			0		0	UNPREDICTABLE(不可預測)
0	0	0	0	0	0	0	0	Normal execution(正常執行)

在後續執行中上述位元之作用為：

- MRS 和 SRS 指令將正常讀取上述位元。MSR 指令將把它們正常寫入 SPSRs，但當寫入 CPSR 時將忽略寫入值。
- 在 ARM 和 JAVA 狀態執行期間，該些位元應該零（即，如果它們不是零，結果是 UNPREDICTABLE(無法預測)）。
- 異常帳目序列使該些位元自 CPSR 複製至 SPSR，然後在 CPSR 中清除（連同 J 及 T），處理器轉回至正常 ARM 執行。
- Thumb 狀態執行期間：



1. 當執行 IT 指令時，該些位元應該零。(即，IT 指令應禁止出現在其他 IT 指令所影響的指令區塊之中。)
2. 當執行一條條件化分支指令時，位元 [ 26、25、11、10 ] 應為零。(即，條件化分支指令應禁止出現在由 IT 指令所影響的指令區塊之中。然而，非條件化分支指令可能出現在那裡，並將導致被視為條件化：這規則基本上是為了避免在相同指令有兩條件時所引起的問題。)
3. 對於上述規則未決定為 UNPREDICTABLE 的所有指令，自現有的執行狀態位元產生一條件，如下：

```

if bits[26,25,11,10] == 0b0000 then
    /* 不是在一 IT 區塊 */
    condition is AL
else
    condition is bits[15:12], interpreting both 0b1110
        and 0b1111 as "always"
    /* 註：因為 "NV space" 已在 ARMv5 中恢復，該解讀
    已被標準化。
  
```

尤有甚者，非 CMP、CMN 或 TST 指令及通常設置條件旗標的 16 位元 Thumb 資料處理指令被修改如下：

```

if bits[26,25,11,10] != 0b0000 then
  
```

`/* 在一 IT 區塊中 */`

`instruction does not set condition flags`

(指令不設置條件旗標)

註：這對 IT 而言是重要的，有助於控制許多指令序列-例如，如果使用它條件化一 `LDR Rn; ADD Rn,Rn,#M; STR Rn` 序列，如果該 `ADD` 改變由 `STR` 指令所見的條件旗標，則它將是無用的。

註：可推論，藉由 1-4 後續指令，可用具有一 `AL` 條件(或一 `NV` 條件)之一 `IT` 指令禁止此類旗標設定。現有 Thumb 指令集之一不佳態樣為新插入之碼(例如，溢出/填滿碼)可能具有使條件碼崩潰的副作用。能夠以一單一額外的 16 位元指令處理它，可能非常有用。

4. 任何 PC 修改指令之成功執行(即，在任何條件檢查不失敗)將導致新的「執行狀態」變為：

`For exception return instructions (the ones which do SPSR->CPSR copies,`

`and RFE executed in a privileged mode):`

`Bits[26,25,15:10] of the value written to the CPSR.`

`Otherwise:`

`All zero.`

註：“otherwise”的情況是基本規則，其使得只有

IT 影響後續指令。

此一 PC 修改指令的不成功執行或非 PC 修改的任何執行(對論成功與否)，非 IT 指令將更新「執行狀態」位元如下：

現有狀態	條件	新狀態	註
26 25 15 14 13 12 11 10	針對狀態	26 25 15 14 13 12 11 10	
改變			

P4	1	condbase	P1	P2	P3	Always	1	0	condbase	P2	P3	P4	
1	0	condbase	P1	P2	P3	Always	0	0	condbase	P2	P3	1	
0	0	condbase	P1	P2	0	Always	0	0	condbase	P2	1	0	
0	0	condbase	P1	1	0	Always	0	0	0	0	0	0	A
0	0	Any	0	0	0	Always	0	0	0	0	0	0	B

A) 終結 IT 執行 - 但不需要特別的動作

B) 在 IT 區塊之外

註：

1. 分支(Branches)。除了另有說明者外，不允許 IT block 中的任何指令是任何分支的目標，不管是否是一分支指令或改變 PC 的一些其他指令的結果。若未能遵守該限制會導致 UNPREDICTABLE 行為。
2. 異常(Exceptions)。中斷及其他異常能夠發生於一指令及它的 IT 區塊間或 IT 區塊之中。一般而言，它們將導致適合的傳回資訊輸入異常管理器，置於適當模式之 R14 及 SPSR。為異常傳回所設計之指令一般可用於自

異常返回，以及將正確地繼續執行 IT 區塊。此為唯一的辦法，其中允許一 PC 修改指令在一 IT 區塊分支為一指令。

3. IT 區塊 (IT block)。在一 IT 區塊的指令不允許為：

- IT 指令
- 條件化分支指令

如果它們是，則結果是 UNPREDICTABLE。

第 5 圖圖示另一類型預測指令。在這種類型的預測指令中，有兩區塊的關聯程式指令 26、28 與與預測指令 30 關聯。如果在預測指令中指定的條件碼 30 通過並稱作「If 區塊 (If block)」，則執行上述區塊指令 26 之第一者。如果與預測指令 30 關聯之條件碼失敗並叫做「Else 區塊 (Else block)」，則另一區塊的執行指令 28。預測指令 30 指定上述兩區塊 26、28 的大小，其每一也許在長度上來自 1-4 指令。在第 5 圖的示例中，不是執行 If 區塊 26 就是執行 Else 區塊 28。該例中一旦已執行預測指令 30，則應用該衡量以預測所有關聯程式指令。因此，或這或交替著個整個的如果 26 執行區塊該整個 Else 區塊 28。當環境暫存器 22(MPC)指示已經觸及 If block 的末尾時，則強迫一分支進入在 Else 區塊 28 的最後指令之後的指令。當已觸及 Else 區塊 28 中的最後的指令時，則移除預測指令 30 的作用。

第 6 圖依據第 5 圖所示之操作類型，圖示系統行為的流程圖。在第 32 步驟中，系統等待確認一預測指令 (ITE

指令)。在第 34 步驟中，系統確認是否在目前程式狀態暫存器內儲存的環境計數值不是零。此一非零值指示已自一異常進行一傳回，並因此在第 36 步驟在目前程式狀態暫存器內，接受和儲存環境值，而處理進行至第 38 步驟，其中衡量在預測指令內所指定的條件碼。替代性地，如果第 34 步驟的測試沒有結果，則在第 40 步驟設定環境值為零。如果在第 38 步驟決定條件碼衡量為條件碼失敗，則表示應該執行 Else 區塊 28，以及因此在第 42 步驟，提昇程式計數值至 Else 區塊 28 中的第一指令，及之後以已終止於第 44 步驟之預測指令執行 Else 區塊 28。替代性地，如果在第 38 步驟衡量的條件碼通過，則第 46 步驟執行藉由用環境值加總的 PC 值所指向的指令，以及以指令位元組長度遞增環境值。

第 48 步驟檢查異常之發生以及如果此類異常存在，則第 50 步驟儲存目前程式狀態暫存器，包括環境值(預測指令計數值)，取得該異常，執行異常處理例式以及其後最終傳回至由程式計數值 PC 指示的指令，即預測指令。

第 52 步驟測試一分支的發生以及如果此一分支發生，則第 54 步驟改變分支目標的程式計數值，以及第 44 步驟終止預測指令的行動。

第 56 步驟測試以了解是否預測指令計數值已經觸及 If 區塊的大小，其指示已經觸及 If 區塊的末端。如果已經觸及 If 區塊的末端，則第 58 步驟把程式計數值移至 Else 區塊的末端之上，以及在第 44 步驟已經終止預測指令的行

動以後，繼續正常的處理。替代性地，處理回到第 46 步驟，其中執行在 If 區塊中的下一指令。

在第 5 圖及第 6 圖圖示之指令行動的另一描述，敘述如下：

### ITE

當條件通過時，If-Then-Else (ITE)指令允許指定指令區塊的大小(在指令之後，立即定位該區塊)，以及當條件失敗時，執行的指令區塊的大小(在 If 區塊之後，立即定位該區塊)。

應該在 PSR 或其他的特別暫存器保留一些 ITE 環境位元；我們將把該些位元稱作 PSR\_uPC。PSR\_uPC 位元的精確數量決定 If 區塊的最大可能大小。

### 語法

```
ITE<cond> #if_size, #else_size
```

### 操作

```
if PSR_uPC != 0 then
    // 自異常返回
    uPC = PSR_uPC
else
    uPC=0
if ConditionPassed(cond) then
    PC = PC + if_size
```

```
// 分支至 else 部分及終止 ITE
loop
  Instr = FetchInstmctionFrom( PC + uPC)
  Execute(Instr)
  if ExceptionO then
    PSR_uPC = uPC
    PC = ExceptionHandler
    // 取得異常
  if Branch( Instr) then
    PSR_uPC = 0
    PC = BranchTarget
    // 分支及終止 ITE
    uPC += InstructionLength( Instr )
  if uPC >= if_size then
    PSR_uPC = 0
    PC = PC + if_size + else_size
    // 在 else 部分之後分支及終止 ITE
endloop
```

註：

1. 分支(Branches)。不允許 ITE 的”if”部分中的任何指令是任何分支的目標，不管是否是一分支指令或改變 PC 的一些其他指令的結果。若未能遵守該限制會導致 UNPREDICTABLE 行為。

2. 異常 (Exceptions)。中斷及其他異常能夠發生於 If 區塊中的一指令。一般而言，它們將導致適合的傳回資訊輸入異常管理器，置於特殊異常模式之 R14 及 SPSR。為異常傳回所設計之指令一般可用於自異常返回，以及將正確地繼續執行 If 區塊。此為唯一的辦法，其中允許一 PC 修改指令在一 If 區塊分支為一指令。

使用如第 5 圖及第 6 圖所示之 If 區塊 26 及 Else 區塊 28 作為預測指令執行型式的替代性方法，即，其中一旦執行預測指令 30，則衡量條件碼，一替代性的方法是，為每一被預測指令 30 所影響之關聯程式指令衡量條件碼，當關聯指令執行時，可以由下列行為表示：

ITE Cond, # THEN, # ELSE

(0-3) (0-3)

CPSR holds 8 bits: 4 cond [3:0], 2 tlm [1:0], 2 else [1:0]

ITE docs

Cond, then, else = values from instruction

GENERK execution

IF CPSR Jhen == 0 THEN

EXECUTE INSTRUCTION NORMALLY

ELSE

EXECUTE INSTRUCTION WITH CONDITION CPSR\_cond



```

/* AND IN PARALLEL*/
IF CPSR_then>1THEN
    CPSRJhen - = 1
ELSE
    CPSRJhen = CPSR_else
    CPSR_else = 0
    CPSR_cond[0] =NOT (CPSR_cond[0]) /^FLIP COND */

```

		CPSR_cond	_then	_else	
e.g.	ITE	EQ,#3,#1	X	0	x
	Instr1	/*EQ*/	EQ	3	1
	Instr2	/*EQ*/	EQ	2	1
	Instr3	/*EQ*/	EQ	1	1
	Instr4	/*NE*/	NE	1	0
	Instr5	/*uncond*/	EQ	0	0

由上文可知，在示例中，預測指令指定包含三關聯程式指令之 If 區塊，以及包含一關聯程式指令之 Else 區塊。藉由 CPSR\_cond 值說明以該些程式指令之每一者有效地指定的條件碼。當已經退出 Else 區塊時，第五指令是不可預測的，且以與此一不可條件化的指令關聯之正常方法，不可條件化地執行。

雖然本文已經參照附圖，詳述本發明的實施例，應了解本發明並不侷限於上文所述之實施例，以及可能由那些熟知本項技藝者藉由不背離如下文申請專利範圍所定義之

本發明範圍及精神，所做之各種變化和修正。

## 【圖式簡單說明】

第 1 圖圖示一資料處理設備，可操作以執行包含預測指令之程式指令；

第 2 圖圖示回應一預測指令之一指令解譯器的行動；

第 3 圖及第 4 圖圖示一種形態之預測指令，其中依據一遮罩值及一條件碼，可控制是否執行個別的關聯程式指令；

第 5 圖圖示另一形態之預測指令，其中依據條件碼決定是否執行關聯指令之區塊；

第 6 圖係一流程圖，圖示第 5 圖型態之一預測指令之後的執行控制；

第 7 圖圖示用以部署上述技術之一普通用途電腦設備。

## 【元件代表符號簡單說明】

- 1 程式指令
- 2 資料處理系統
- 4 程式暫存器組
- 6 乘法器
- 8 轉換器
- 10 加法器

- 12 指令解譯器
- 14 指令管道
- 16 預取單元
- 18 中斷控制器
- 20 程式計數暫存器
- 22 環境暫存器
- 24 遮罩值
- 26 關聯程式指令
- 28 關聯程式指令
- 30 預測指令
- 32 預測指令 (ITE) ?
- 34 系統確認是否在目前程式狀態暫存器內儲存的環境計數值不是零
- 36 在目前程式狀態暫存器內，接受和儲存環境值
- 38 <Cond> 符合 ?
- 40 環境值為零
- 42 提昇程式計數值至 Else 區塊 28 中的第一指令
- 44 終止 ITE 及  $\mu PC=0$
- 46 執行  $PC+\mu PC$  及  $\mu PC=\mu PC+$ 指令位元組長度
- 48 異常是否發生 ?
- 50 SPSR  $\mu PC=\mu PC$ ，取得該異常以及傳回至由 PC 指示的指令
- 52 一支發生 ?
- 54 改變分支目標的程式計數值

- 56 測試以了解是否預測指令計數值已經觸及 If 區塊的大小
- 58 把程式計數值移至 Else 區塊的末端之上
- 202 CPU
- 204 RAM
- 206 ROM
- 208 網路卡
- 210 硬碟
- 212 顯示驅動器
- 214 監視器
- 216 使用者輸入輸出
- 218 鍵盤
- 220 滑鼠
- 222 匯流排

伍、中文發明摘要：

在一資料處理系統 2 中，提供加入條件行為至關聯程式指令 26 及 28 之預測指令 30。

陸、英文發明摘要：

Within a data processing system 2 prediction instructions 30 are provided which add conditional behavior to a associated program instructions 26, 28.

## 拾、申請專利範圍：

1. 一種用以處理資料之設備，該設備包括：

一資料處理邏輯，可操作以執行資料處理操作；

一指令解譯器，可操作以解譯程式指令，以控制該資料處理邏輯，以執行該些資料處理操作；以及

一程式計數暫存器，可操作以儲存一位址，其指示執行之程式指令的一記憶體位置，其中

該指令解譯器，可依據用於處理資料集之該設備的一或多種條件狀態，藉由執行一或多種程式指令而非一預測指令回應該預測指令，控制該資料處理邏輯執行或不執行一或多種關聯的程式指令；其特徵在於：

一預測指令計數暫存器，可操作以儲存一計數值，其指示該一或多關聯程式指令中的多少屬於已執行之該預測指令，當執行屬於該預測指令之該一或多關聯程式指令時，該程式計數暫存器持續儲存與該預測指令對應的一位址；以及

一異常處理電路，在一異常發生時可操作以儲存該計數值，以及在完成該異常時，在由該計數值所指向的一程式指令處重新開始執行。

2. 如申請專利範圍第1項所述之設備，其中該預測指令控制複數關聯程式指令。

3. 如申請專利範圍第 1 項所述之設備，其中該一或多種關聯程式指令是非條件化程式指令。
4. 如申請專利範圍第 1 項所述之設備，其中該一或多種條件狀態包括一或多種條件碼旗標的值。
5. 如申請專利範圍第 1 項所述之設備，其中該一或多種條件狀態在下列之一種情況下衡量：
  - (i) 一旦執行該預測指令；及
  - (ii) 在執行每一關聯程式指令之前。
6. 如申請專利範圍第 1 項所述之設備，其中該一或多種關聯指令緊接在該預測指令之後。
7. 如申請專利範圍第 1 項所述之設備，其中該預測指令與下列關聯：
  - (I) 與欲執行的一或多種程式指令之區塊符合之一條件，如果該一或多種條件狀態符合一或多種預定條件；及
  - (II) 與欲執行的一或多種程式指令之區塊不符合之一條件，如果該一或多種條件狀態不符合該一或多種預定條件。

8. 如申請專利範圍第 7 項所述之設備，其中該預測指令指定下列之一或多種：
  - (I) 與一或多種程式指令之區塊符合的該條件之一長度；及
  - (II) 與一或多種程式指令之區塊不符合的該條件之一長度。
9. 如申請專利範圍第 7 項所述之設備，其中該預測指令指定該一或多種預定條件。
10. 如申請專利範圍第 1 項所述之設備，其中該預測指令包括一或多種欄位，根據該一或多種條件狀態與一或多種預定狀態的比較，該一或多種欄位之每一者指定是否欲執行一各自的關聯程式指令。
11. 如申請專利範圍第 10 項所述之設備，其中該預測指令包括一欄位，其為每一各別的關聯程式指令指定是否該一或多種條件狀態或該一或多種條件狀態之一補數與該一或多種預定狀態比較，以決定是否執行該各自的關聯程式指令。
12. 如申請專利範圍第 1 項所述之設備，其中當屬於該預測指令時，該一或多種關聯指令指定不同的資料處理操作。



13. 如申請專利範圍第 12 項所述之設備，其中依據在該預測指令內之一可程式化欄位，該一或多種關聯指令被分隔開來，防止對該一或多種條件狀態進行任何改變。

14. 一種處理資料的方法，該方法包含下列步驟：

以資料處理邏輯執行資料處理操作；

以一指令解譯器解譯程式指令，以控制該資料處理邏輯，以執行該些資料處理操作，以及

儲存一位址，其指示執行於一程式計數暫存器內之程式指令的一記憶體位置；其中

回應一預測指令，依據用於處理資料集之一設備的一或多種條件狀態，藉由執行一或多種程式指令而非該預測指令，該指令解譯器控制該資料處理邏輯執行或不執行一或多種關聯的程式指令；其特徵在於以下步驟：

儲存一計數值於一預測指令計數暫存器內，其指示該一或多關聯程式指令中的多少屬於已執行之該預測指令；以及

當執行該一或多關聯程式指令時，在該程式計數暫存器內持續儲存與該預測指令對應的一位址；以及

在一異常發生時儲存該計數值，以及在完成該異常時，在由該計數值所指向的一程式指令處重新開始

執行。

15. 如申請專利範圍第 14 項所述之方法，其中該預測指令控制複數關聯程式指令。

16. 如申請專利範圍第 14 項所述之方法，其中該一或多種關聯程式指令是非條件化程式指令。

17. 如申請專利範圍第 14 項所述之方法，其中該一或多種條件狀態包括一或多種條件碼旗標的值。

18. 如申請專利範圍第 14 項所述之方法，其中該一或多種條件狀態在下列之一種情況下衡量：

- (i) 一旦執行該預測指令；及
- (ii) 在執行每一關聯程式指令之前。

19. 如申請專利範圍第 14 項所述之方法，其中該一或多種關聯指令緊接在該預測指令之後。

20. 如申請專利範圍第 14 項所述之方法，其中該預測指令與下列關聯：

- (I) 與欲執行的一或多種程式指令之區塊符合之一條件，如果該一或多種條件狀態符合一或多種預定條

件；及

(II)與欲執行的一或多種程式指令之區塊不符合之一條件，如果該一或多種條件狀態不符合該一或多種預定條件。

21. 如申請專利範圍第 20 項所述之方法，其中該預測指令指定下列之一或多種：

(I)與一或多種程式指令之區塊符合的該條件之一長度；及

(II)與一或多種程式指令之區塊不符合的該條件之一長度。

22. 如申請專利範圍第 20 項所述之方法，其中該預測指令指定該一或多種預定條件。

23. 如申請專利範圍第 14 項所述之方法，其中該預測指令包括一或多種欄位，根據該一或多種條件狀態與一或多種預定狀態的比較，該欄位之每一者指定是否欲執行一各自的關聯程式指令。

24. 如申請專利範圍第 23 項所述之方法，其中該預測指令包括一欄位，其為每一各別的關聯程式指令指定是否該一或多種條件狀態或該一或多種條件狀態之一補數

與該一或多種預定狀態比較，以決定是否執行該各自的關聯程式指令。

25. 如申請專利範圍第 14 項所述之方法，其中當屬於該預測指令時，該一或多種關聯指令指定不同的資料處理操作。

26. 如申請專利範圍第 25 項所述之方法，其中依據在該預測指令內之一可程式化欄位，該一或多種關聯指令被分隔開來，防止對該一或多種條件狀態進行任何改變。

27. 一種電腦程式產品，其包含一電腦程式可操作以控制一資料處理設備，該設備具有資料處理邏輯其係可操作以執行資料處理操作，一指令解譯器其係可操作以解譯程式指令以控制該資料處理邏輯執行該些資料處理操作，及一程式計數暫存器其係可操作以儲存一位址，該位址指示執行之程式指令的一記憶體位置，該電腦程式包含：

一預測指令，依據用於處理資料集之該設備的一或多種條件狀態，藉由執行一或多種程式指令而非該預測指令，該預測指令可操作以控制該資料處理邏輯執行或不執行一或多種關聯的程式指令；其特徵在於：

儲存一計數值於一預測指令計數暫存器內，其指示該一或多種關聯程式指令中的多少屬於已執行之該預

測指令；以及

當執行屬於該預測指令之該一或多關聯程式指令時，在該程式計數暫存器內持續儲存與該預測指令對應的一位址；以及

在一異常發生時儲存該計數值，以及在完成該異常時，在由該計數值所指向的一程式指令處重新開始執行。

28. 如申請專利範圍第 27 項所述之電腦程式產品，其中該預測指令控制複數關聯程式指令。

29. 如申請專利範圍第 27 項所述之電腦程式產品，其中該一或多種關聯程式指令是非條件化程式指令。

30. 如申請專利範圍第 27 項所述之電腦程式產品，其中該一或多種條件狀態包括一或多種條件碼旗標的值。

31. 如申請專利範圍第 27 項所述之電腦程式產品，其中該一或多種條件狀態在下列之一種情況下衡量：

(i) 一旦執行該預測指令；及

(ii) 在執行每一關聯程式指令之前。

32. 如申請專利範圍第 27 項所述之電腦程式產品，其中該

一或多種關聯指令緊接在該預測指令之後。

33. 如申請專利範圍第 27 項所述之電腦程式產品，其中該預測指令與下列關聯：

(I) 與欲執行的一或多種程式指令之區塊符合之一條件，如果該一或多種條件狀態符合一或多種預定條件；及

(II) 與欲執行的一或多種程式指令之區塊不符合之一條件，如果該一或多種條件狀態不符合該一或多種預定條件。

34. 如申請專利範圍第 33 項所述之電腦程式產品，其中該預測指令指定下列之一或多種：

(I) 與一或多種程式指令之區塊符合的該條件之一長度；及

(II) 與一或多種程式指令之區塊不符合的該條件之一長度。

35. 如申請專利範圍第 33 項所述之電腦程式產品，其中該預測指令指定該一或多種預定條件。

36. 如申請專利範圍第 27 項所述之電腦程式產品，其中該預測指令包括一或多種欄位，根據該一或多種條件狀

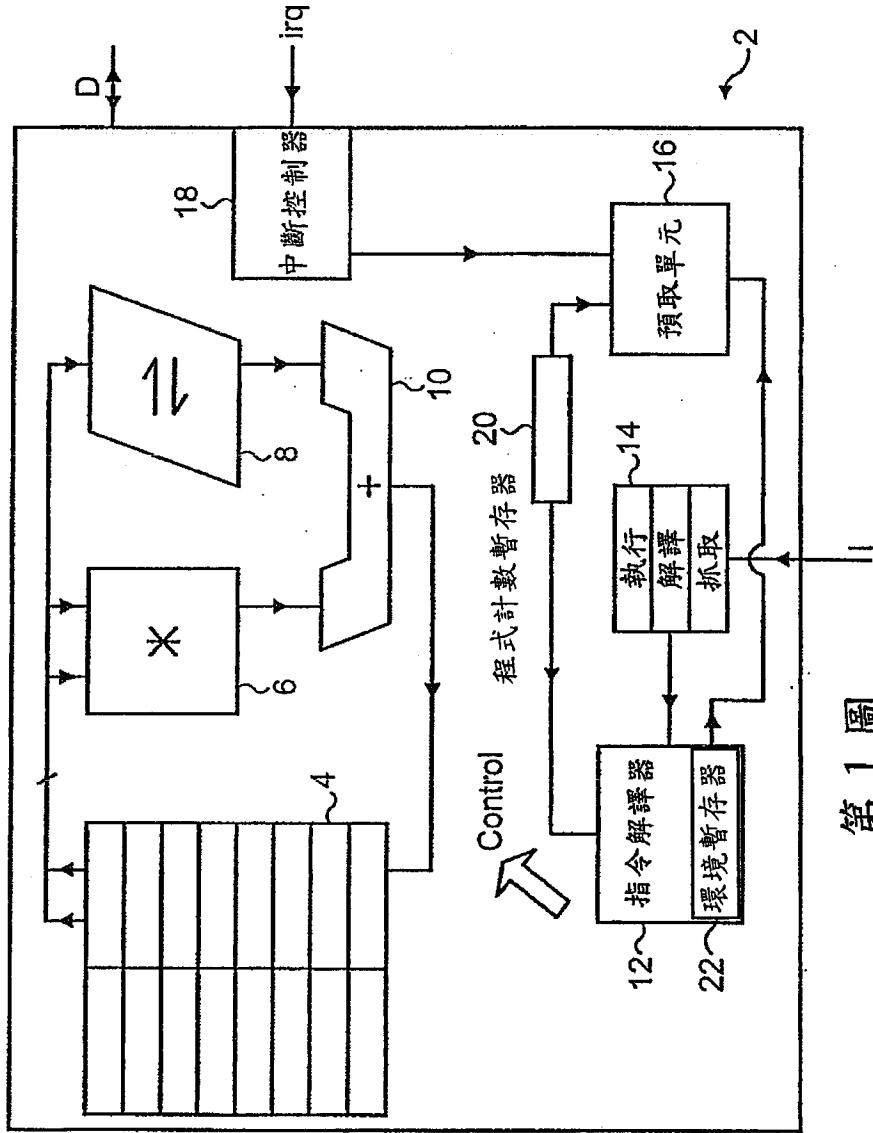
態與一或多種預定狀態的比較，該一或多種欄位之每一者指定是否欲執行一各自的關聯程式指令。

37. 如申請專利範圍第 36 項所述之電腦程式產品，其中該預測指令包括一欄位，其為每一各別的關聯程式指令指定是否該一或多種條件狀態或該一或多種條件狀態之一補數與該一或多種預定狀態比較，以決定是否執行該各自的關聯程式指令。

38. 如申請專利範圍第 27 項所述之電腦程式產品，其中當屬於該預測指令時，該一或多種關聯指令指定不同的資料處理操作。

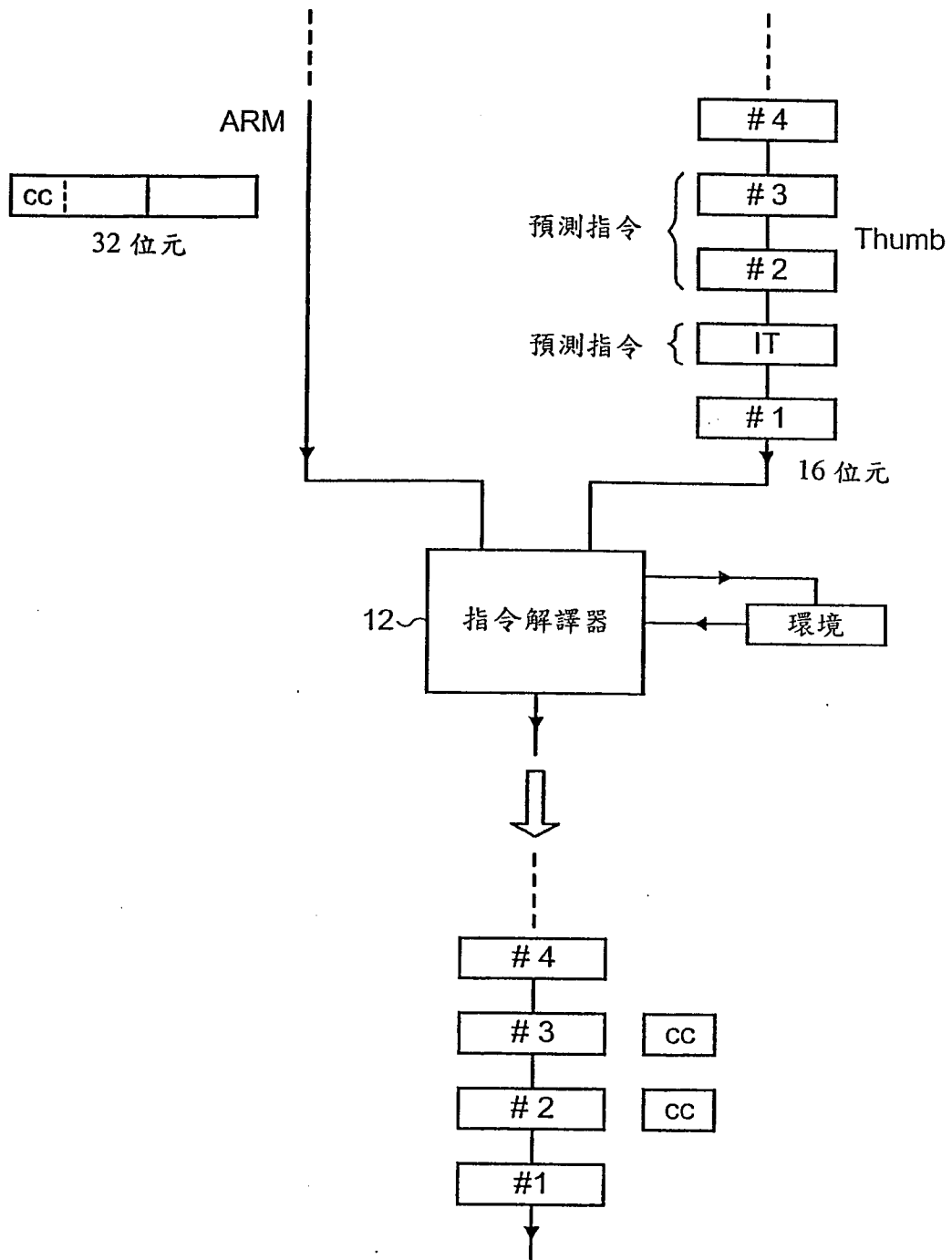
39. 如申請專利範圍第 38 項所述之電腦程式產品，其中依據在該預測指令內之一可程式化欄位，該一或多種關聯指令被分隔開來，防止對該一或多種條件狀態進行任何改變。

98年4月22日修(受)正替換頁

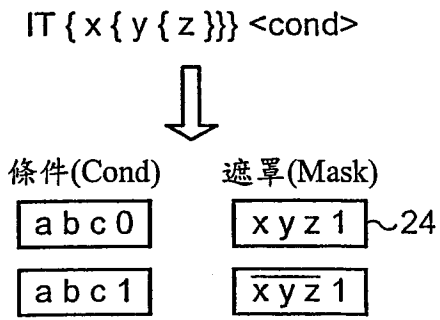


第 1 圖

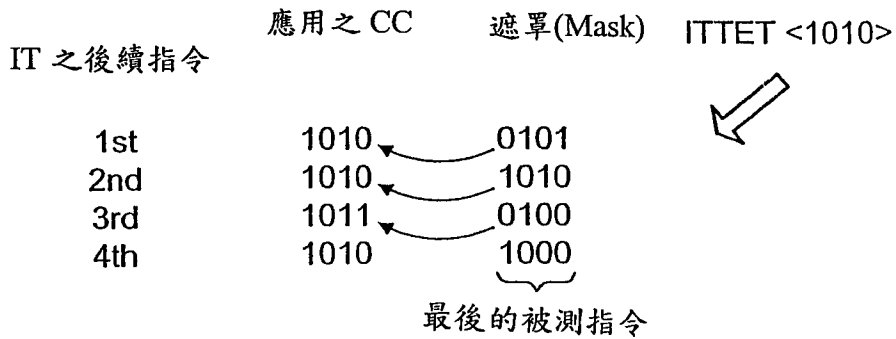




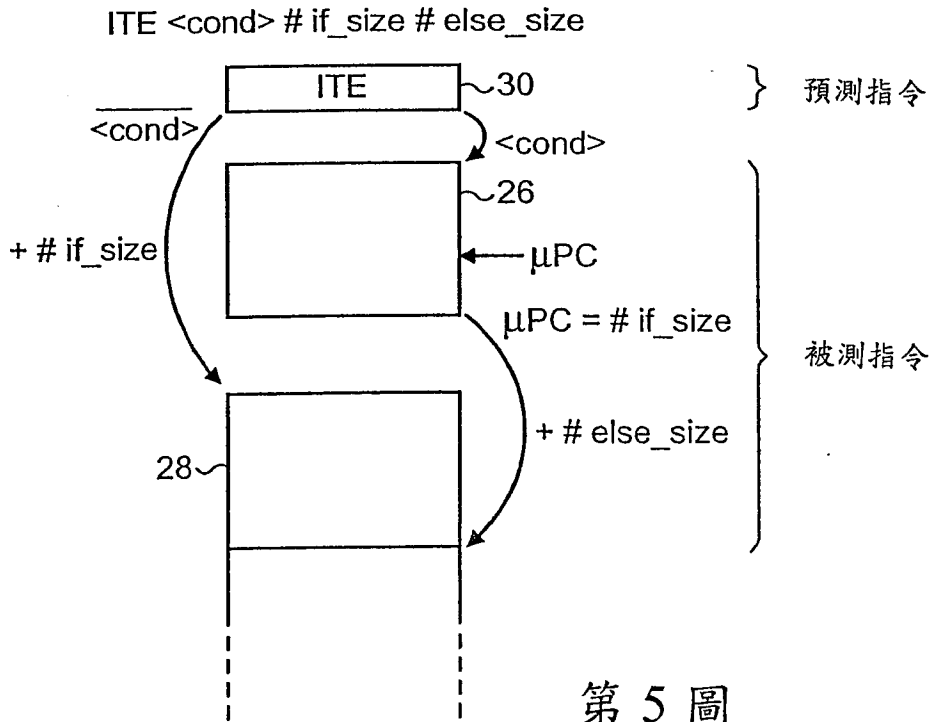
第 2 圖

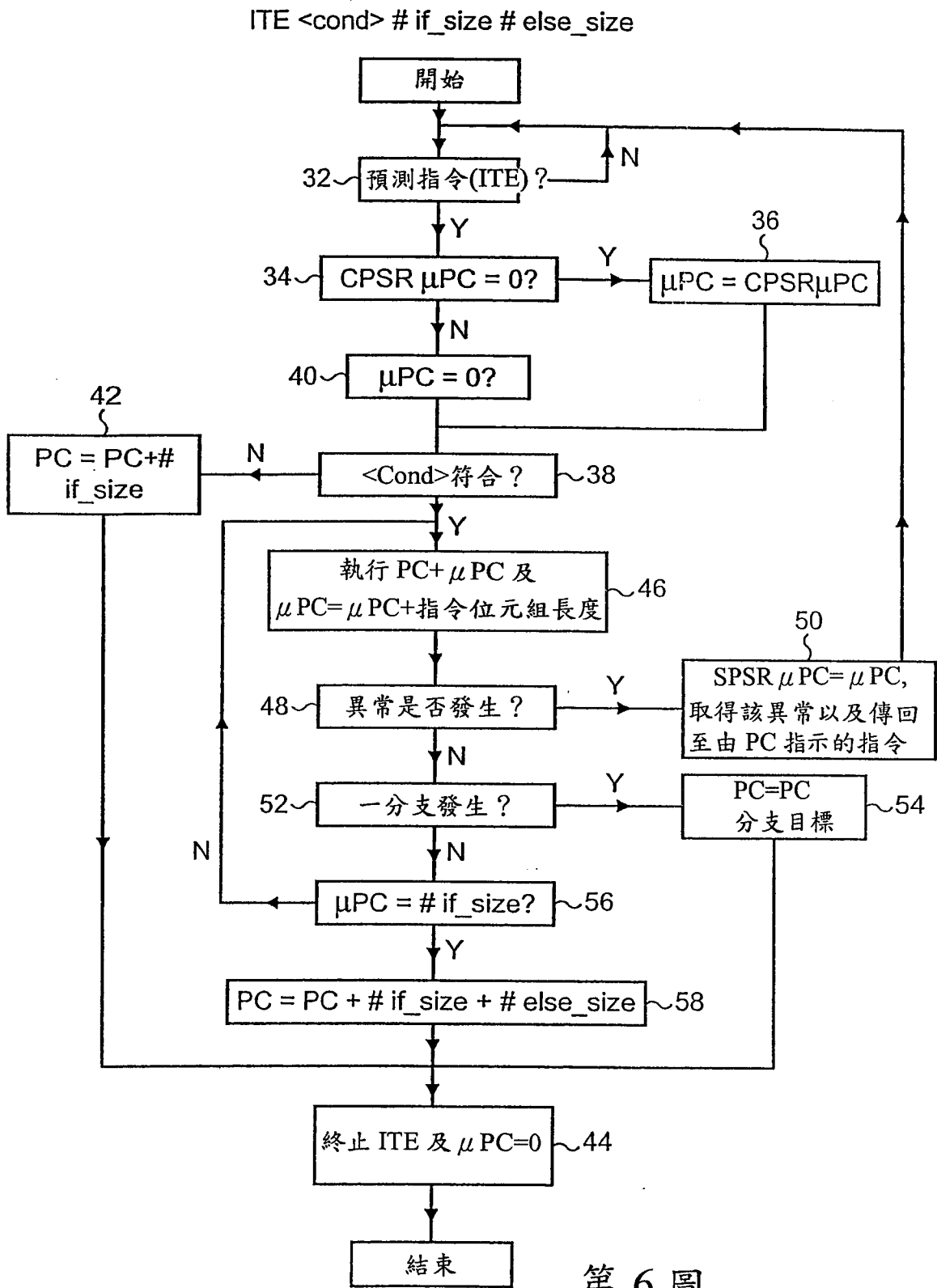


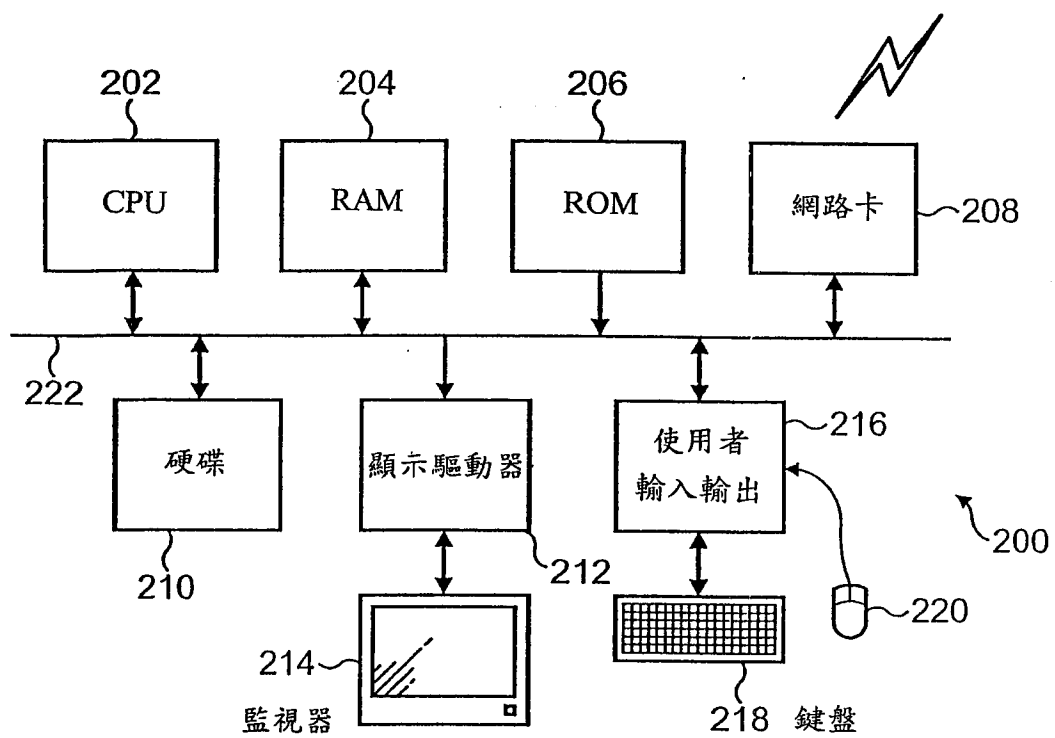
第 3 圖



第 4 圖







第 7 圖

## 柒、指定代表圖：

(一)、本案指定代表圖為：第 6 圖。

(二)、本代表圖之元件代表符號簡單說明：

- |   |   |
|---|---|
| 32 預測指令(ITE)?                                   | 48 異常是否發生?  |
| 34 系統確認是否在目前程式狀態暫存器內儲存的環境計數值不是零                 | 50 SPSR $\mu$ PC = $\mu$ PC, 取得該異常以及傳回至由 PC 指示的指令 |
| 36 在目前程式狀態暫存器內, 接受和儲存環境值                        | 52 一支發生?  |
| 38 <Cond>符合?                                    | 54 改變分支目標的程式計數值                                   |
| 40 環境值為零  | 56 測試以了解是否預測指令計數值已經觸及 If 區塊的大小                    |
| 42 提昇程式計數值至 Else 區塊 28 中的第一指令                   | 58 把程式計數值移至 Else 區塊的末端之上                          |
| 44 終止 ITE 及 $\mu$ PC=0                          | 48 異常是否發生?  |
| 46 執行 PC+ $\mu$ PC 及 $\mu$ PC= $\mu$ PC+指令位元組長度 |   |

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無