202414203

(19)中華民國智慧財產局

(12)發明說明書公開本 (11)公開編號：TW 202414203 A

(43)公開日： 中華民國 113 (2024) 年 04 月 01 日

(21)申請案號：112132669 (22)申請日： 中華民國 112 (2023) 年 08 月 30 日

(51)Int. Cl.： *G06F9/46 (2006.01)* *G06F9/30 (2018.01)*

G06F13/24 (2006.01)

(30)優先權：2022/09/28 英國 2214155.0

(71)申請人：英商ＡＲＭ股份有限公司 (英國) ARM LIMITED (GB)

英國

(72)發明人：威廉斯　麥克約翰 WILLIAMS, MICHAEL JOHN (GB)

(74)代理人：李世章；彭國洋

申請實體審查：無　申請專利範圍項數：19 項　圖式數：12　　共 116 頁

(54)名稱

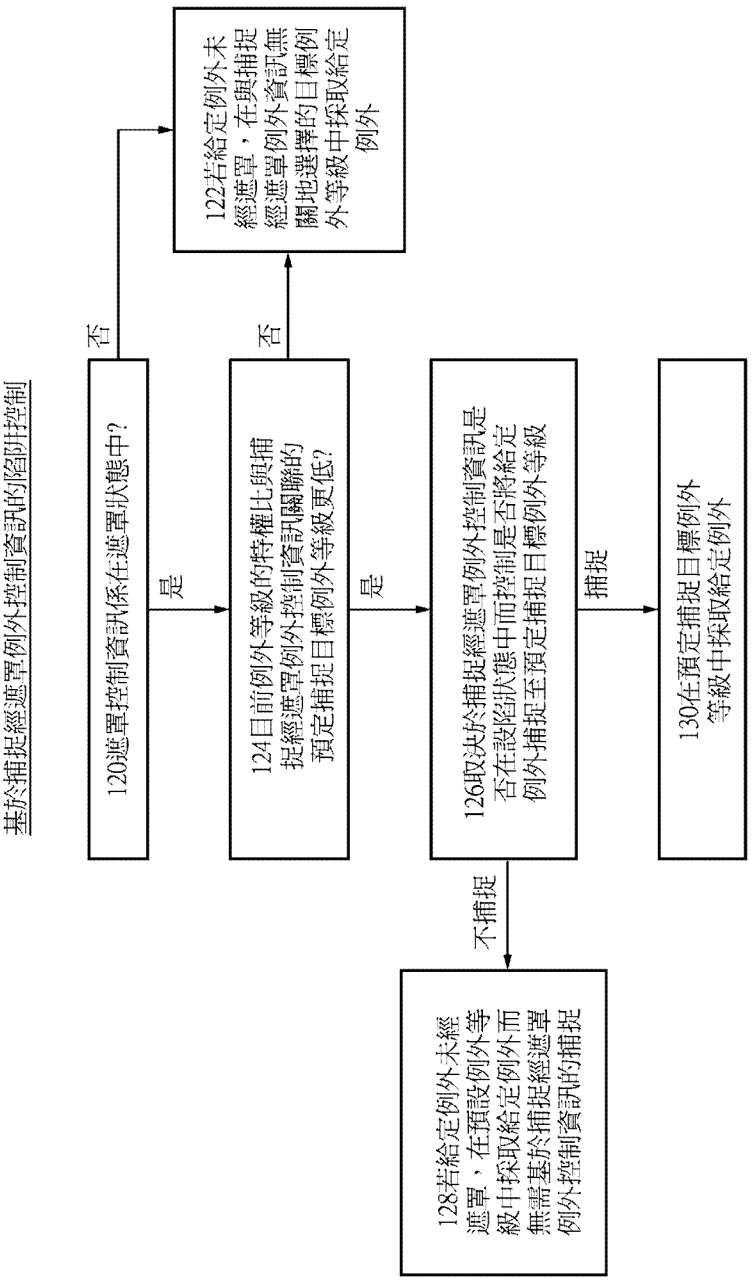
例外控制

(57)摘要

例外控制電路系統 40 取決於儲存在至少一個暫存器 14 中的控制資訊而控制由處理電路系統 4 採取例外，該控制資訊包括可設定成一遮罩狀態或未遮罩狀態的遮罩控制資訊；及可設定成一未設陷狀態或設陷狀態的捕捉經遮罩例外控制資訊。回應於一可遮罩例外類別的一給定例外，在該遮罩控制資訊係在該遮罩狀態中且一目前例外等級的特權低於一預定捕捉目標例外等級的至少一個情境中，該例外控制電路系統取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級。當該遮罩控制資訊係在該未遮罩狀態中時，用於處理該給定例外的一目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

Exception control circuitry 40 controls taking of exceptions by processing circuitry 4, depending on control information stored in at least one register 14, the control information including masking control information settable to a masked state or unmasked state; and trap-masked-exception control information settable to an untrapped state or trapped state. In response to a given exception of a maskable class of exceptions, in at least one scenario when the masking control information is in the masked state and a current exception level is less privileged than a predetermined trap target exception level, the exception control circuitry controls whether to trap the given exception to the predetermined trap target exception level depending on whether the trap-masked-exception control information is in the trapped state. When the masking control information is in the unmasked state, a target exception level for handling the given exception is selected independent of the trap-masked-exception control information.

指定代表圖：



【圖5】

符號簡單說明：

120:步驟

122:步驟

124:步驟

126:步驟

128:步驟

130:步驟

202414203

【發明摘要】

【中文發明名稱】 例外控制

【英文發明名稱】 EXCEPTION CONTROL

【中文】

例外控制電路系統40取決於儲存在至少一個暫存器14中的控制資訊而控制由處理電路系統4採取例外，該控制資訊包括可設定成一遮罩狀態或未遮罩狀態的遮罩控制資訊；及可設定成一未設陷狀態或設陷狀態的捕捉經遮罩例外控制資訊。回應於一可遮罩例外類別的一給定例外，在該遮罩控制資訊係在該遮罩狀態中且一目前例外等級的特權低於一預定捕捉目標例外等級的至少一個情境中，該例外控制電路系統取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級。當該遮罩控制資訊係在該未遮罩狀態中時，用於處理該給定例外的一目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【英文】

Exception control circuitry 40 controls taking of exceptions by processing circuitry 4, depending on control information stored in at least one register 14, the control information including masking control information settable to a masked state or unmasked state; and trap-masked-exception control information settable to an untrapped state or trapped state. In response to a given exception of a maskable class of exceptions, in at least one scenario when the masking control information is in the masked state and a current exception level is less privileged than a predetermined trap

target exception level, the exception control circuitry controls whether to trap the given exception to the predetermined trap target exception level depending on whether the trap-masked-exception control information is in the trapped state. When the masking control information is in the unmasked state, a target exception level for handling the given exception is selected independent of the trap-masked-exception control information.

【指定代表圖】圖5

【代表圖之符號簡單說明】

120:步驟

122:步驟

124:步驟

126:步驟

128:步驟

130:步驟

【特徵化學式】無

【發明說明書】

【中文發明名稱】　例外控制

【英文發明名稱】　EXCEPTION CONTROL

【技術領域】

【0001】　本技術係關於資料處理領域。更具體而言，其係關於例外控制。

【先前技術】

【0002】　在執行資料處理時，處理電路系統可遭遇例外，例外係需要中斷程式執行的正常流程的事件，使得處理可切換至例外處理器以處理導致該例外的事件。例如，例外可由於若干個原因而出現，諸如未定義指令的執行、導因於未定義虛擬至實體位址轉譯映射或記憶體存取權限檢查失敗的記憶體存取而出現的位址錯誤、或外部事件（諸如記憶體系統錯誤或偵測到由外部裝置或週邊設備傳訊的中斷事件）的發生。

【發明內容】

【0003】　至少一些實例提供一種設備，其包含：處理電路系統，其在由該處理電路系統支援的複數個例外等級的一者中執行資料處理；及例外控制電路系統，其取決於儲存在至少一個暫存器中的控制資訊控制由該處理電路系統採取例外，該控制資訊包括至少：遮罩控制資訊，其可設定成一遮罩狀態及一未遮罩狀態的一者；及捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷

狀態及一設陷狀態的一者；其中：回應於基於將該遮罩控制資訊設定成該遮罩狀態而可遮罩之一可遮罩例外類別的一給定例外，該例外控制電路系統經組態以：判定該遮罩控制資訊是否在該遮罩狀態中；在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【0004】　至少一些實例提供用於上述設備之製造的電腦可讀碼。該電腦可讀碼可儲存在一儲存媒體上。該儲存媒體可係暫時性或非暫時性儲存媒體。

【0005】　至少一些實例提供一種方法，其包含：在由該處理電路系統支援的複數個例外等級的一者中執行資料處理；及取決於儲存在至少一個暫存器中的控制資訊而控制由該處理電路系統採取例外，該控制資訊包括至少：遮罩控制資訊，其可設定成一遮罩狀態及一未遮罩狀態的一者；及捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷狀態及一設陷狀態的一者；其中：回應於基於將該遮罩控制資訊設定成該遮罩狀態而可遮罩之一可遮罩例外類別的一給定例外，該方法包含：判定該遮罩控制資訊是否在該遮罩狀態中；在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及在該遮罩控制資訊係在該未遮罩狀

態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【0006】　至少一些實例提供一種包含指令的電腦程式，當該等指令由一主機資料處理設備執行時，控制該主機資料處理設備提供用於執行目標碼的一指令執行環境，該電腦程式包含：處理程式邏輯，其用以模擬該目標碼的指令在複數個例外等級的一者中的執行；及例外控制程式邏輯，其用以取決於儲存在該主機資料處理設備之代表由該目標碼使用的一目標指令集架構的至少一個暫存器的至少一個儲存位置中的控制資訊而模擬採取例外，該控制資訊包括至少：遮罩控制資訊，其可由軟體設定成一遮罩狀態及一未遮罩狀態的一者；及捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷狀態及一設陷狀態的一者；其中：回應於基於將該遮罩控制資訊設定成該遮罩狀態的軟體而可遮罩之一可遮罩例外類別的一給定例外，該例外控制程式邏輯經組態以：判定該遮罩控制資訊是否在該遮罩狀態中；在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【0007】　該電腦程式可儲存在一儲存媒體上。該儲存媒體可係暫時性或非暫時性儲存媒體。

【圖式簡單說明】

【0008】　本技術的進一步態樣、特徵、及優點將由於結合附圖閱讀的以下實例描述而顯而易見，在該等附圖中：

〔圖1〕示意地繪示資料處理設備的一實例；

〔圖2〕繪示處理電路系統可於其中執行資料處理的若干個例外等級；

〔圖3〕繪示儲存在暫存器中的控制資訊；

〔圖4〕係說明判定可遮罩例外類別的給定例外是否應遮罩的流程圖；

〔圖5〕係說明取決於遮罩控制資訊是否在遮罩狀態中及捕捉經遮罩例外控制資訊是否在設陷狀態中而判定是否將給定例外捕捉至預定捕捉目標例外等級的流程圖；

〔圖6〕係說明控制對捕捉經遮罩例外控制資訊之更新的流程圖；

〔圖7〕係說明基於捕捉經遮罩例外控制資訊控制是否將非可遮罩例外類別的例外捕捉至預定捕捉目標例外等級的流程圖；

〔圖8A〕至〔圖8O〕繪示判定可遮罩例外類別的給定例外是否應遮罩，且若未遮罩，哪個例外等級係採取給定例外之目標例外等級的特定實例；

〔圖9A〕至〔圖9I〕繪示判定用於非可遮罩例外類別的例外的目標例外等級的特定實例；

〔圖10A〕及〔圖10B〕係說明取決於給定例外發生時的目前例外等級及控制資訊是否遮罩可遮罩類別的給定例外，且若未遮罩，在哪個例外等級中採取該給定例外的表；

〔圖11〕係說明取決於給定例外發生時的目前例外等級及控制資訊選擇用於非可遮罩類別的目標例外等級的表；及

〔圖12〕繪示模擬實例。

【實施方式】

【0009】　一種設備具有處理電路系統，該處理電路系統用於在若干個例外等級的一者中執行資料處理，及例外控制電路系統，其用以取決於儲存在至少一個暫存器中的控制資訊控制由該處理電路系統採取例外。該控制資訊影響是否遮罩或採取例外，且若係採取，影響哪個例外等級係在其中採取該例外的目標例外等級。

【0010】　該控制資訊包括可設定成遮罩狀態及未遮罩狀態的一者的遮罩控制資訊。該遮罩控制資訊可回應於某些事件而自動地由處理電路系統或例外控制電路系統的硬體設定成遮罩狀態（例如，遮罩控制資訊可回應於採取例外而設定成遮罩狀態），及/或可由在給定例外等級執行的軟體設定，以控制是否遮罩可遮罩例外類別，以防止其等在給定例外等級採取。若在採取例外上係以硬體設定，軟體可控制何時將遮罩控制資訊清除成未遮罩狀態的時序。例如，可在指令集架構中支援指令，以允許軟體觸發依需將遮罩控制資訊設定成遮罩或未遮罩狀態。若藉由軟體設定成遮罩狀態，當軟體期望將採取例外停用一段時間時，其可由於任何任意原因而選擇將遮罩控制資訊設定成遮罩狀態。無論遮罩控制資訊係以硬體或軟體設定，遮罩例外的一個原因可係保留與目前由軟體執行的處理關聯的狀態資訊，若採取例外，該狀態資訊可能被覆寫。

【0011】　然而，無論處理電路系統（採硬體或在軟體的控制下）在以較低特權例外等級操作時是否已將遮罩控制資訊設定成遮罩狀態，在處理電路系統上執行的其他軟體（例如，監督軟體（諸如，超管理器或安全性監測器））

可將可遮罩例外類別視為係應迅速處理的重要例外。例如，若可遮罩例外類別表示偵測到錯誤，可能存在導因於基於遮罩控制資訊的遮罩延遲例外的處理而可能使錯誤傳播的風險，使得問題變得更糟（例如，儲存於記憶體中的錯誤值可能由其他操作處理而將錯誤傳播至記憶體中的其他位置）的顧慮。一種用於確保可遮罩例外類別迅速地被處理的方法可係提供一控制，其中在較高特權例外等級的軟體可設定以指示可遮罩例外類別應在該較高特權例外等級採取，以置換基於遮罩控制資訊的遮罩，即使該遮罩控制資訊係在遮罩狀態中。然而，此方法的問題係當此控制經設定時，可遮罩例外類別的處理始終從用於處理該例外的正常預設等級取走，由於在開發在不同例外等級執行的軟體的不同軟體開發者之間可能需要更多合作，其可增加軟體開發的複雜性。

【0012】　在以下討論的實例中，該控制資訊包括可由軟體設定成未設陷狀態及設陷狀態的一者的捕捉經遮罩例外控制資訊。回應於基於將該遮罩控制資訊設定成該遮罩狀態而可遮罩之一可遮罩例外類別的一給定例外，該例外控制電路系統經組態以：判定該遮罩控制資訊是否在該遮罩狀態中；在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【0013】　因此，捕捉經遮罩例外控制資訊提供可用以組態例外處理電路系統的架構控制選項，使得在遮罩控制資訊已設定成遮罩狀態的情形中，可將

例外捕捉至預定捕捉目標例外等級，以降低處理例外上的延遲。然而，基於捕捉經遮罩例外控制資訊的此捕捉的前提係遮罩控制資訊設定成遮罩狀態。在遮罩控制資訊係在未遮罩狀態中且給定例外不被遮罩的情形中，可在與捕捉經遮罩例外控制資訊無關地選擇的目標例外等級中採取例外。實際上，由於遮罩控制資訊一般對相對小部分的執行時間設定成遮罩狀態，此可大幅降低必須將可遮罩例外類別捕捉至比會正常採取該例外之預設等級更高特權之例外等級之時機的數目。與指示應將可遮罩類別的所有例外皆捕捉至更高特權例外等級（不論遮罩控制資訊的狀態）的控制選項相比，此允許更大部分的例外在較低特權例外等級處理，促成較低特權軟體可在大部分情形中處理可遮罩例外類別的「核心優先(kernel-first)」例外處理（而非更高特權碼一般處理大部分例外的「韌體優先(firmware-first)」例外處理）。與韌體優先例外處理相比，核心優先例外處理常係較佳的，因為其可簡化軟體開發。因此，由於支援支援捕捉經遮罩例外控制資訊之指令集架構的處理器對於開發用於其等之軟體的軟體開發者更簡單，其等相較於不支援捕捉經遮罩例外控制資訊的處理器係經改善的。

【0014】　回應於請求對在比與該捕捉經遮罩例外控制資訊關聯的該預定捕捉目標例外等級更低特權的一例外等級執行的該捕捉經遮罩例外控制資訊的一更新的一指令，該處理電路系統可拒絕或捕捉對該捕捉經遮罩例外控制資訊的所請求更新。因此，捕捉經遮罩例外控制資訊可限於由在該例外將捕捉至其的較高特權例外等級執行的軟體設定（或可由甚至比該例外等級更高特權的例外等級設定）。因此，可將捕捉經遮罩例外控制資訊視為係使更高特權碼能夠置換基於在遮罩狀態中的遮罩控制資訊而遮罩可遮罩例外類別的偏好（與在較低特權例外等級的操作關聯）的架構控制選項。

【0015】 在一些實例中，可僅存在捕捉經遮罩例外控制資訊的單一項，其控制是否將相依於遮罩控制資訊的陷阱設立至特定例外等級。

【0016】 然而，提供捕捉經遮罩例外控制資訊之各與複數個預定捕捉目標例外等級的各別一者關聯的多個項可係有用的。該例外控制電路系統可在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且該目前例外等級的特權比該複數個預定捕捉目標例外等級的一給定一者更低的至少一個情境中取決於捕捉經遮罩例外控制資訊之與該複數個預定捕捉目標例外等級的該給定一者關聯的一對應項是否在該設陷狀態中而控制是否將該給定例外捕捉至該複數個預定捕捉目標例外等級的該給定一者。藉由提供可由在不同例外等級的較高特權軟體設定的捕捉經遮罩例外控制資訊的多個項，此給予多件更高特權的軟體（例如，超管理器或安全性監測器碼）在遮罩控制資訊經設定成遮罩狀態的情形中在是否應捕捉而非遮罩例外上的單獨控制。此可將影響處理可遮罩例外類別之方式的更多彈性提供給不同的軟體提供者。

【0017】 在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生、捕捉經遮罩例外控制資訊的至少二個項係在該設陷狀態中、且捕捉經遮罩例外控制資訊的該至少二個項之各者滿足捕捉至該預定捕捉目標例外等級的一或多個條件的至少一個情境中：該例外控制電路系統經組態以將該給定例外捕捉至與捕捉經遮罩例外控制資訊的一經選擇項關聯的該預定捕捉目標例外等級；及捕捉經遮罩例外控制資訊的該經選擇項包含捕捉經遮罩例外控制資訊的該至少二個項之在該目前例外等級與該關聯預定捕捉目標例外等級之間在特權上有一最小差的一者。因此，若捕捉經遮罩例外控制資訊的多於一個項已設定成設陷狀態（例如，藉由在不同例外等級的軟體），若例外係基於捕捉經遮罩例外控制

資訊捕捉，將其取至與捕捉經遮罩例外控制資訊之已設定成設陷狀態的該等項關聯的關聯預定捕捉目標例外等級的最低特權。可將此視為係反直覺的，因為其與若存在由在不同例外等級之軟體設定的衝突控制，通常會假設在更高特權例外等級設定的控制應取得優先權的其他例外處理控制不同。然而，由於在可用於採取例外之最低特權例外等級中採取例外可簡化軟體開發，即使更高特權例外等級亦已將捕捉經遮罩例外控制資訊設定成設陷狀態，允許捕捉經遮罩例外控制資訊之與較低特權例外等級關聯的項取得優先權可係有用的。

【0018】　在該捕捉經遮罩例外控制資訊係在該未設陷狀態中的一情形中，該例外控制電路系統可至少取決於該遮罩控制資訊是否在該遮罩狀態中而判定是否遮罩該給定例外。是否遮罩給定例外亦可取決於其他因子。例如，在陷阱基於捕捉經遮罩例外控制資訊設立的情形中，即使遮罩控制資訊係在遮罩狀態中，仍不遮罩例外。

【0019】　再者，亦可存在遮罩例外的其他原因，其可係例如基於例外發生時的目前特權等級。例如，該例外控制電路系統可取決於該目前例外等級的特權是否比若該例外等級係在該捕捉經遮罩例外控制資訊係在該未設陷狀態中時採取，將採取該給定例外的一例外等級更高而判定是否遮罩該給定例外。此由於可假設若目前在比初始將處理例外的例外等級更高特權的例外等級中（不考慮基於捕捉經遮罩例外控制資訊的捕捉），則可將目前處理視為比例外的處理更重要，且因此不論遮罩控制資訊是否在遮罩狀態中，可隱含地將例外視為經遮罩而係有用的。

【0020】　再者，即使遮罩控制資訊係在遮罩狀態中，不一定始終意謂著在基於捕捉經遮罩例外控制資訊的陷阱不存在時例外被遮罩。例如，如下文更

詳細地解釋的，提供允許軟體表示至少在一些情境中應將可遮罩例外類別視為彷彿其係不可遮罩的，使得即使遮罩控制資訊係在遮罩狀態中，給定例外可不被遮罩的架構控制選項亦係可行的。

【0021】　因此，將理解雖然通常在捕捉經遮罩例外控制資訊係在未設陷狀態中的情形中，例外控制電路系統經組態以至少取決於遮罩控制資訊是否在遮罩狀態中而判定是否遮罩給定例外，在一些實施方案中，是否實際遮罩例外的判定亦可考慮數個其他參數。

【0022】　在一些實例中，該控制資訊包含可設定成一非可遮罩狀態及一可遮罩狀態的一者以指示該可遮罩類別的例外是否可遮罩的非可遮罩例外控制資訊；及在該非可遮罩例外控制資訊係在該非可遮罩狀態中的至少一個情境中，即使在該遮罩控制資訊經設定成該遮罩狀態時，該例外控制電路系統經組態以判定該給定例外將不被遮罩。此控制對於允許更高特權軟體表示應將可遮罩例外類別視為非可遮罩的以確保更迅速地處理其等的偏好而不論遮罩控制資訊是否已設定成遮罩狀態可係有用的。

【0023】　在該給定例外在該目前例外等級係一最低特權例外等級時發生、該非可遮罩例外控制資訊係在該非可遮罩狀態中、及該給定例外將取至經啟用以在該給定例外未遮罩且該捕捉經遮罩例外控制資訊係在該未設陷狀態中的一情形中用於採取該給定例外的一次一最低特權例外等級的一情境中，即使在該捕捉經遮罩例外控制資訊係在該設陷狀態中時，該例外控制電路系統經組態以控制該給定例外在該次一最低特權例外等級中採取而不捕捉至該預定捕捉目標例外等級。因此，在目前例外等級係最低特權等級的情形中，在非可遮罩狀態中的非可遮罩例外控制資訊亦置換基於捕捉經遮罩例外控制資訊的任何捕

捉。應注意到經啟用以用於採取給定例外的次一最低特權例外等級可基於其他控制資訊變化。例如，指定系統係在客作業系統體系中（其中超管理器管理客作業系統，但不執行作業系統功能本身）或主機作業系統體系中（其中超管理器亦管理應用程式且執行作業系統功能）的控制資訊可控制用於客作業系統體系中之客作業系統的例外等級是否在主機作業系統體系中停用，所以經啓用的次一最低特權例外等級可係客作業系統體系中的核心級例外等級（客作業系統在其執行）且可係主機作業系統體系中的超管理器級例外等級（超管理器/主機作業系統在其中執行）。

【0024】　當目前例外等級非最低特權例外等級時，在非可遮罩狀態中的非可遮罩例外控制資訊可置換遮罩本身，但不置換基於在設陷狀態中的捕捉經遮罩例外控制資訊及在遮罩狀態中的遮罩控制資訊經判定係需要的任何陷阱。此係由於將遮罩控制資訊設定成遮罩狀態表示對不採取例外以保留在採取例外時可能被覆寫的某個狀態的偏好（假定處理的目前狀態在通常預設將例外取至其的預設例外等級），且因此即使該例外實際上並未經遮罩，藉由將陷阱觸發至預定捕捉目標例外等級以在較高等級採取例外可仍係較佳的。

【0025】　在一些實例中，該控制資訊包含非可遮罩例外控制資訊之與不同例外等級關聯的複數個項；且該例外控制電路系統取決於非可遮罩例外控制資訊之該複數個項之基於該目前例外等級選擇的一者判定是否遮罩該給定例外。此可為在不同例外等級執行之軟體的不同提供者提供彈性以表示在是否允許遮罩可遮罩例外類別上的獨立偏好。

【0026】　因此，將理解除了遮罩控制資訊及捕捉經遮罩例外控制資訊外，是否將陷阱設立至預定捕捉目標例外等級的準確判定可取決於若干個因

子，諸如（在一些實施方案中）目前例外等級及（在至少一個例外等級中）非可遮罩例外控制資訊的目前狀態。

【0027】　在一些實施方案中，亦可存在置換捕捉經遮罩例外控制資訊的其他（可選）控制，使得即使上文描述之用於設立陷阱的條件獲得滿足（遮罩控制資訊係在遮罩狀態中、目前例外等級的特權比與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低、及捕捉經遮罩例外控制資訊係在設陷狀態中），至預定捕捉目標例外等級的陷阱可能仍不設立（例如，由於另一控制已指定將可遮罩例外類別捕捉至不同的例外等級）。

【0028】　例如，該控制資訊可包含捕捉控制資訊之可由軟體設定成一設陷狀態及一未設陷狀態的一者的至少一個其他項；該例外控制電路系統可取決於捕捉控制資訊的該其他項是否在該設陷狀態中而控制是否將至少該可遮罩類別的例外捕捉至一特定例外等級；及捕捉控制資訊的該至少一個其他項可優先於該捕捉經遮罩例外控制資訊，使得在捕捉控制資訊的該其他項係在該設陷狀態中的一情形中，該例外控制電路系統經組態以基於捕捉控制資訊的該其他項將該例外捕捉至該特定例外等級而不論該捕捉經遮罩例外控制資訊是否在該設陷狀態中。例如，控制資訊的此其他項可類似於上文所描述之用於比較的控制，其意謂著可將可遮罩例外類別捕捉至某個較高特權例外等級而不論遮罩控制資訊是否在遮罩狀態中。捕捉經遮罩例外控制資訊的使用可因此在捕捉控制資訊的該至少一個其他項係在未設陷狀態中的情形中有關，但在捕捉控制資訊的該至少一個其他項係在設陷狀態中時無關。

【0029】　在一些情形中，捕捉控制資訊的其他項的至少一者可特定用於可遮罩例外類別，其中另一類別的例外不受捕捉控制資訊的該項影響。

【0030】　再者，捕捉控制資訊的其他項的至少一者係影響可遮罩例外類別及至少一個其他類別之例外兩者的處理的更一般性控制係可能的。

【0031】　因此，取決於支援哪些額外架構控制設定及定義該等架構控制設定的資訊已如何設定，存在大數目的可以其使用捕捉經遮罩例外控制資訊的方式。一般而言，在給定例外在遮罩控制資訊係在遮罩狀態中時發生且目前例外等級的特權低於與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級的至少一個情境中，提供取決於捕捉經遮罩例外控制資訊是否在設陷狀態中而控制至預定捕捉目標例外等級的陷阱，但在遮罩控制資訊係在未遮罩狀態中的情形中不會影響目標例外等級在可遮罩例外將不被遮罩的情形中的選擇的捕捉經遮罩例外控制資訊係有用的。然而，在可發生例外的所有情境中考慮捕捉經遮罩例外控制資訊係非必要的。

【0032】　可遮罩例外類別可係可可選地遮除但有時可能需要更迅速地處理的任何類別的例外。

【0033】　然而，上文討論的該等技術在該可遮罩例外類別包含指示與正確服務的一偏離的非同步系統錯誤例外的一情形中可係特別有用的。錯誤例外指示與正確服務的偏離，例如，設備本身（或外部設備）不遵從其之正確功能規格。此可與硬體系統本身正正確地運作，但軟體組態資訊已錯誤地設定的其他故障原因（諸如記憶體管理故障或未定義指令錯誤）不同（例如，已提供未定義指令、頁表映射未指定用於記憶體中存取的位址、或記憶體存取權限已遭違反（例如，寫入至記憶體的唯讀區域））。在許多情形中，錯誤例外可在記憶體系統存取之後發生，但其等亦可由於其他原因（例如，處理電路系統之暫存器中的故障）發生。非同步例外係不能歸因於所執行之指令串流中的特定點

的例外。例如，對於非同步例外，不像可歸因於指令串流中的特定點的同步例外，可能無法清楚地判定哪些指令係不受錯誤影響的較老指令及可能受錯誤影響的較年輕指令。雖然非同步系統錯誤例外可具有若干個原因，非同步系統錯誤的共同實例可係由髒資料從快取線至外部記憶體的寫回所觸發的資料中止。

【0034】 此類錯誤可期望相對迅速地處理，由於延遲非同步系統錯誤例外的處理可有導致更大問題的錯誤傳播的風險。然而，支援軟體或硬體指定錯誤例外應使用遮罩控制資訊遮罩以防止重要的狀態資訊在採取例外時被覆寫的能力亦可係所欲的。捕捉經遮罩例外控制資訊的提供允許在非同步系統錯誤例外將不被遮罩的大部分情形中例外可使用核心優先處理的折衷，但在遮罩控制資訊已設定成遮罩狀態的情形中存在允許韌體優先處理例外的安全網。

【0035】 例外控制電路系統亦可支援不能夠基於將遮罩控制資訊設定成遮罩狀態而被遮罩的非可遮罩例外類別。可能預期由於此類別的例外無法基於遮罩控制資訊遮罩，上文討論之此遮罩導致延遲處理例外的問題不會發生，且因此不需要觸發基於捕捉經遮罩例外控制資訊的任何陷阱。然而，發明人認知到即使非可遮罩例外類別實際上將不會被遮罩，將遮罩控制資訊設定成遮罩狀態可傳訊將不採取例外以保留在採取例外時可能以其他方式被覆寫的重要處理器狀態的偏好。因此，即使非可遮罩例外類別係不可遮罩的，使用捕捉經遮罩例外控制資訊以在遮罩控制資訊係在遮罩狀態中的情形中控制是否設立至預定捕捉目標例外等級的陷阱可仍係有用的。因此，在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生、一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的該預定捕捉目標例外等級、及該目前例外等級係能夠係用於處理例外的一目標例外等級的一例外等級的至少一個情境中，該例外控制電路系

統可取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該非可遮罩例外捕捉至該預定捕捉目標例外等級。此有助於降低導因於當例外在遮罩控制資訊係在遮罩狀態中時發生時的覆寫而損失重要狀態的可能性。

【0036】 在目前例外等級係不能夠係用於處理例外之目標例外等級的例外等級（例如，最低特權例外等級）的情形中，則不需要基於捕捉經遮罩例外控制資訊觸發非可遮罩例外的捕捉。

【0037】 非可遮罩例外類別可例如包含指示與正確服務之偏離的同步外部中止例外。再次，與正確服務的偏離可由上文對非同步系統錯誤例外描述的實例事件的任一者引起，但與非同步例外相比，同步例外係能夠歸因於所執行之指令串流中的特定點的例外。例如，同步例外可係在指定指令執行之後發佈的需求記憶體存取之後在記憶體系統中偵測到的錯誤（諸如，失敗的錯誤校正碼檢查）（而非對快取記憶體寫回偵測到之非常可能被視為非同步錯誤的錯誤），因此能夠歸因於指令串流中的特定點。

【0038】 該設備可包含複數組一或多個例外症候群暫存器，各組一或多個例外症候群暫存器儲存在該等例外等級的一各別者中採取之例外的例外症候群資訊。雖然軟體可由於任何任意軟體特定原因而選擇將遮罩控制資訊設定成遮罩狀態，將遮罩控制資訊設定成遮罩狀態的常見原因可係正在處理先前例外，且若將後續例外取至該例外等級，與先前例外有關之在用於目前例外等級之相關症候群暫存器中的資訊經覆寫係非所欲的。因此，將陷阱設立至預定捕捉目標例外等級可幫助保護與目前例外等級關聯之該組一或多個症候群暫存器中的資訊，而仍允許迅速地採取例外。因此，上文描述的技術在存在用於特定

例外等級之允許與在不同例外等級處理之例外關聯之例外資訊的同時暫存器儲存的分組例外症候群暫存器組的架構中可係特別有用的。

【0039】 上文討論的技術可實施在具有提供如上文討論之處理電路系統及例外控制電路系統之硬體電路特徵的硬體處理器中。硬體設備可具有以硬體提供以儲存如上文討論之控制資訊的暫存器。

【0040】 然而，在其他實例中，該技術可實施在用於控制主機資料處理設備以提供用於執行目標碼的指令執行環境的程式內。提供該模擬的電腦程式可儲存在一儲存媒體上。該儲存媒體可係一暫時性儲存媒體或一非暫時性儲存媒體。

【0041】 此類模擬電腦程式對允許根據目標指令集架構開發的目標程式在本身不支援該目標指令集架構的主機資料處理設備上執行可係有用的。此可對於若干個原因係有用的，諸如允許傳統碼在不支援該碼為其開發之該架構的較新裝置上執行，或在支援待於未來釋放之新指令集架構的任何硬體裝置尚未可用之前，用於測試開發中之用於該新指令集架構的目標程式。儘管硬體實施裝置可具有提供上文討論之特徵的硬體電路，軟體模擬可提供對應的軟體特徵，諸如仿真對應硬體之功能性的程式邏輯及資料結構，使得目標程式可在模擬程式上執行以給出猶如目標程式係由原生地支援目標指令集架構的對應硬體裝置執行的相同結果。

【0042】 因此，電腦程式可具有以與上文討論之處理電路系統及例外控制電路系統類似的方式運作的處理程式邏輯及例外控制程式邏輯。該例外控制程式邏輯可基於儲存在主機資料處理設備之代表由目標碼使用之目標指令集架構的至少一個暫存器的至少一個儲存位置中的控制資訊而控制例外處理。可遮

罩類別的經模擬例外可以上文討論的方式處理，其中在遮罩控制資訊係在遮罩狀態中的情形中，至預定捕捉目標例外等級的陷阱係基於捕捉經遮罩例外控制資訊受控制。因此，在模擬程式上執行的目標碼（其可包括在不同經模擬例外等級執行的各別件軟體）可見到猶如其在支援與稍早討論相同的架構控制選項的硬體設備上執行的相同效應。

【0043】 圖1示意地繪示資料處理設備2的一實例。設備2具有提供用於回應於指令而執行資料處理之處理電路系統的處理管線4。處理管線4具有用於從指令快取記憶體8提取待執行之指令的提取級6。解碼級10解碼經提取指令以產生用於控制管線4之後續級以執行由指令表示之操作的控制信號。發佈級12接收經解碼指令且在等待任何所需運算元待於暫存器14中可用時將其等排在佇列中。當所需運算元經判定成可用時，則將指令傳遞至執行級16，該執行級包括用於回應於指令而執行處理操作的若干個執行單元18、20、22、24。處理操作可使用讀自暫存器14的運算元，並產生可由處理管線4的寫回級26寫回至暫存器14的處理結果。其他實例可具有不同配置的管線級（例如，可包括用於將由指令指定的架構暫存器映射至以硬體提供的實體暫存器的暫存器重命名級）

【0044】 在此實例中，執行級16包括作為執行單元之用於在整數運算元上執行算術或邏輯操作的算術/邏輯單元(arithmetic/logic unit, ALU) 18；用於執行浮點操作的浮點單元20，用於該等浮點操作的輸入運算元及輸出運算元的至少一者係以浮點表示表示；用於評估用於觸發非循序程式流程之分支指令之結果的分支單元22；及用於控制對記憶體之存取的載入/儲存單元24。載入/儲存單元24可執行控制載入/儲存單元將資料從記憶體系統載入至暫存器14的載入指令，或執行控制載入/儲存單元24將資料從暫存器14儲存至記憶體系統的儲存指

令。將理解顯示於圖1之實例中的該組特定執行單元18至24僅係一個可能實施方案且其他實施方案可具有不同組的執行單元。例如，可省略圖1所示之執行單元類型的一或多者，或可重複此等類型的一者使得可提供數個相同類型的執行單元。亦可提供未顯示於圖1中之專用於處理其他類別的處理操作的其他類型的執行單元。

【0045】 對於由載入/儲存單元24執行的載入/儲存操作，由載入/儲存指令指定的虛擬位址可基於定義在從記憶體系統內的頁表結構獲得的頁表資料中的位址映射而由記憶體管理單元28轉譯成實體位址。轉譯後備緩衝區(translation lookaside buffer, TLB) 30可由記憶體管理單元28提供以從頁表結構快取頁表項以用於更快的存取。

【0046】 在此實例中，記憶體系統包括1級指令快取記憶體8、1級資料快取記憶體32、用於指令或資料其中一者的共用2級快取記憶體34、及主記憶體36。然而，將理解此僅係可能的快取記憶體階層的一個實例，且其他實例可具有更大數目的快取記憶體級，或可在指令快取與資料快取之間具有不同關係。

【0047】 提供用於控制處理設備2之例外處理的例外控制電路系統40。指示若干種例外條件的一者已於何時發生的例外信號42可由例外控制電路系統接收。例外可包括若干種類型的例外。例如，解碼級10可在遭遇對應於未由設備2支援的未定義指令編碼的經提取指令時產生未定義指令例外。再者，若載入/儲存指令指定不具有定義在頁表結構中的位址映射或確實具有經定義的位址映射但用於其的對應位址映射指定未為發佈該載入/儲存指令之軟體程序所滿足的存取權限的目標位址，MMU 28可產生位址錯誤例外。再者，可接收指示外部中斷是否已發生的例外信號42，例如導因於接收到來自外部裝置或週邊裝置的

訊息，諸如指示使用者已按壓處理設備2包括於其中之裝置上的按鈕或指示與外側網路通訊的網路控制器已接收到訊息的訊息。若偵測到記憶體系統錯誤（諸如用以偵測經儲存資料中之由事件（諸如導因於粒子撞擊、游離輻射、或電子組件老化效應的隨機位元翻轉）導致之錯誤的同位碼或錯誤校正碼的檢查失敗、導因於記憶體系統匯流排上的不正確控制設定的匯流排錯誤、或意謂著所請求的記憶體系統操作不能正確地執行的其他情況），記憶體系統亦可產生例外信號。將理解可定義許多不同類型的例外且上述實例並非窮舉的。

【0048】　當例外發生時，例外控制電路系統可使用暫存器14中的暫存器狀態以判定是否採取例外，且若應採取該例外，處理電路系統在採取例外時應在什麼操作狀態中。若採取例外，則例外控制電路系統40中斷目前正由管線執行的處理且控制管線切換成執行例外處理器，該例外處理器可提供用於回應由例外類型指示的所發生事件的軟體指令。

【0049】　圖2係繪示當執行指令時處理電路系統4可於其中操作的不同執行狀態（亦稱為例外等級）的圖。在此實例中，存在四個例外等級EL0、EL1、EL2、EL3，其中例外等級EL0係最低特權例外等級且例外等級EL3係最高特權例外等級。一般而言，當在較高特權例外等級中執行時，處理電路系統可具有對不可為較低較少特權例外等級存取的一些記憶體位置或暫存器14的存取。

【0050】　在此實例中，例外等級EL0係用於執行由在例外等級EL1執行的對應作業系統或虛擬機器管理的應用程式。在多個虛擬機器共存在相同實體平台上時，則可提供在EL2操作的超管理器以管理各別的虛擬機器。雖然圖2顯示

超管理器管理虛擬機器且虛擬機器管理應用程式的實例，超管理器直接管理在EL0的應用程式亦係可行的（在此情形中，例外等級EL1可停用）。

【0051】 雖然並非必要的，一些實施方案可為處理電路系統實施操作之分開的硬體分割的安全域及非安全域。資料處理系統2可具有實施在處理器及記憶體系統內的硬體特徵，以確保與在安全域中操作之軟體程序關聯的資料及碼與由在非安全域中操作之程序的存取隔離。例如，可使用諸如由Arm® Limited of Cambridge, UK提供的TrustZone®架構的硬體架構。替代地，可使用其他硬體強制安全分割架構。安全應用程式（受信任服務）可在安全域中的例外等級EL0中操作，且安全（受信任）作業系統或虛擬機器可在安全域中的例外等級EL1中操作。在一些實施方案中，不支援在安全狀態中的EL2且超管理器僅可在非安全EL2中執行。在其他實施方案中，如藉由圖2中之星號指示的，可支援安全超管理器在安全EL2中執行。在一些實例中，可提供在例外等級EL3中執行之用於管理非安全域與安全域之間的轉變的安全監測器程式。其他實施方案可以硬體監督安全域之間的轉變，使得可不需要安全監測器程式。

【0052】 圖3顯示暫存器14的子集。暫存器包括例外症候群暫存器，該等例外症候群暫存器經分組以提供分開的一或多個例外症候群暫存器組，以儲存在各別例外等級EL1、EL2、及EL3採取之例外的例外症候群資訊（最低特權例外等級EL0不能夠採取例外等級，所以不需要例外症候群暫存器）。例如，與經採取例外關聯的症候群資訊可包括與例外的原因（例如，故障類型）有關的資訊、與例外關聯之位址有關的資訊、或對於使例外處理器能決定如何處理例外有用的任何其他資訊。

【0053】 圖3亦顯示例外控制電路系統40可用以控制例外之處理的控制資訊的子集。將理解圖3未顯示提供在處理器中的所有暫存器，且未顯示提供在該等暫存器中的所有控制狀態。再者，將理解橫跨不同暫存器之控制資訊的特定佈局可變化，且相同資訊可以不同形式表示。例如，圖3所示之在不同暫存器中的資訊可儲存在相同暫存器中，或在圖3中顯示在相同暫存器中的資訊可儲存在不同的暫存器中。因此，所示的特定暫存器布局係非必要的，且可使用其他暫存器布局提供控制選項的相同功能性。記號XXX_ELx.YYY係指提供在暫存器XXX內的欄位YYY，且指示可寫入暫存器的最低特權例外等級係例外等級ELx。若嘗試在比指示在暫存器後綴中的例外等級ELx更低特權的例外等級寫入至暫存器，該等寫入受拒絕或捕捉至更高的例外等級。

【0054】 在此特定實例中，控制資訊包括：

- PSTATE.EL：目前例外等級的指示；
- PSTATE.A：遮罩控制資訊，其可設定成遮罩狀態（在此實例中，PSTATE.A = 0b1）及未遮罩狀態（在此實例中，PSTATE.A = 0b0）的一者。當在遮罩狀態中時，此表示對於標定之目前例外等級的非同步例外（包括非同步記憶體系統錯誤例外–SError例外）被遮罩以防止其等被採取，直到PSTATE.A經清除成未遮罩狀態的偏好，雖然此偏好可由如下所示的其他控制狀態選項所置換。例如，遮罩非同步系統錯誤例外對保護儲存在例外症候群暫存器中的資訊（與先前例外有關）係有用的，否則若另一例外在與先前例外相同的例外等級中採取，該資訊將被覆寫。雖然未顯示於圖3中，（多個）PSTATE暫存器亦可包括用於遮罩非同步系統錯誤例外以外的其他類別的例外的其他

遮罩欄位。然而，下文討論的實例將非同步系統錯誤(SError)例外引用為可遮罩例外類別的特定實例。非同步系統錯誤係在偵測到不能夠歸因於經執行指令串流中的特定點的錯誤（偏離系統硬體的正確運作）時傳訊的例外。

- SCR_EL3.EA：進一步陷阱控制的實例，當其設定成0b1時，導致某些類別的例外（例如，同步外部中止及非同步系統錯誤(SError)例外）路由至EL3，而非在SCR_EL3.EA = 0b0時以其他方式處理其等的例外等級。
- HCR_EL2.TEA：進一步陷阱控制的另一實例，當設定成0b1時，其導致在EL0、EL1、或EL2發生的外部中止例外路由至EL2（若基於SCR_EL3.EA而未路由至EL3），而非在HCR_EL2.TEA = 0b0時以其他方式處理其等的例外等級，除非HCR_EL2.TGE或另一控制需要路由至EL2或EL3。
- HCR_EL2.TGE：進一步陷阱控制的另一實例，其指示是否「將一般例外捕捉」至EL2–若設定成0b0，在EL0的執行上沒有影響，而當設定成0b1時，通常將從EL0路由至EL1的例外將替代地路由至EL2。此有效地使EL1不能用於採取例外，且軟體不能到達EL1。若HCR_EL2.E2H亦係0b1，則EL1完全停用且EL0在主機作業系統中運行。若HCR_EL2.E2H係0b0，則不將例外採取至EL1，但EL1控制仍可具有效果。
- HCR_EL2.AMO：進一步陷阱控制的另一實例，其指示是否應將實體系統錯誤(SError)例外路由至EL2。當HCR_EL2.AMO = 0b1時，將

SError例外取至EL2，除非基於SCR_EL3.EA而將其等路由至EL3。當HCR_EL2.AMO係0b0時，此係非必需的（除非HCR_EL2.TGE或另一控制需要路由至EL2）。

- HCR_EL2.E2H：指示用於運行主機作業系統（類型2超管理器）的組態是否在EL2受支援，其可影響在較低等級（諸如EL0）的例外處理（若支援類型2超管理器，否則會在EL1處理的一些例外可替代地在EL2處理）。當HCR_EL2.E2H = 0b1時，啟用在EL2支援主機作業系統的設施。當HCR_EL2.E2H = 0b0時，停用在EL2支援主機作業系統的設施。
- SCR_EL3.NMEA、SCTLR2_EL2.NMEA、SCTLR2_EL1.NMEA：非可遮罩例外控制資訊的各別項分別與例外等級EL3、EL2、及EL1關聯，其等可藉由在該等例外等級（或更高特權例外等級）執行的軟體設定以控制SError例外是否可使用PSTATE.A遮罩以防止在此等例外等級採取其等。此等可各設定成非可遮罩狀態（例如，0b1）及可遮罩狀態（例如0b0）的一者。當在非可遮罩狀態中時，Serror例外可在對應例外等級採取而不論PSTATE.A的值。當在可遮罩狀態中時，Serror例外可基於PSTATE.A經設定成遮罩狀態（例如，0b1）而經遮罩以免於在對應例外等級採取。下文闡述此等控制如何影響Serror例外之處理的更多細節。
- SCR_EL3.TMEA、HCRX_EL2.TMEA：捕捉經遮罩例外控制資訊的各別項分別與EL3及EL2關聯，其等可設定在設陷狀態（例如，0b1）中以在PSTATE.A係在遮罩狀態(0b1)中且目前例外等級的特權低於對應

例外等級EL3或EL2的情形中控制將非同步Serror例外或同步外部中止例外捕捉至對應的例外等級EL3或EL2，或可設定成不發生捕捉的未設陷狀態（例如0b0）。TMEA控制由多個一般類型的進一步捕捉控制（SCR_EL3.EA、HCR_EL2.TEA、HCR_EL2.TGE、及HCR_EL2.AMO）的一者所導致的至EL2或EL3的任何陷阱所置換，但不像此等進一步陷阱控制，TMEA控制特定用於當遮罩控制資訊係在遮罩狀態(PSTATE.A = 0b1)中的情形，且當遮罩控制資訊係在未遮罩狀態(PSTATE.A = 0b0)中的情形中，在目標例外等級的選擇上不具有影響。

【0055】 因此，PSTATE.A旗標可用以保護症候群暫存器免於非同步例外。當採取例外（亦即，CPU填充症候群暫存器）時，旗標係由處理電路系統的硬體設定（自動地，無需軟體干預），且亦可由執行指令集架構之指示PSTATE.A旗標將設定或清除之指令的軟體設定/清除。在正常使用中，旗標大部分時間係清除的。當旗標經設定時，非同步例外經遮罩，亦即，不被採取。由於無法迅速地採取錯誤例外可導致錯誤傳播，此係非同步錯誤例外的問題。額外地，同步例外係從不被遮罩的非可遮罩例外，意謂著（非預期的）同步錯誤例外可覆寫症候群暫存器，其亦係錯誤回復的問題。

【0056】 此問題的一個可能解決方案係將錯誤例外導向（路由）至較高例外等級，各例外等級具有其自身的症候群暫存器組。此稱為韌體優先錯誤處理。此方法的問題係其始終將錯誤處理從在EL1的作業系統取走，且需要複雜的軟體構造以將錯誤告知作業系統，並允許其回復。當在EL2或EL3操作的韌體像往常一樣對系統管理者不透明時，其亦使識別錯誤的根本原因變得更難。

【0057】 此處描述的實例引入新的例外路由控制TMEA（捕捉經遮罩例外控制資訊），其係在較高例外等級（諸如EL2或EL3）的軟體的控制下。當PSTATE.A係清除時，像平常一樣將錯誤例外取至會正常地處理其等的預設例外等級而不路由至更高例外等級（假設無基於其他控制SCR_EL3.EA、HCR_EL2.TEA、HCR_EL2.TGE、及HCR_EL2.AMO的一者捕捉至EL2或EL3）。當PSTATE.A經設定且TMEA（在較高例外等級）經設定時，將錯誤例外（同步及非同步二者）取至設定TMEA旗標的較高例外等級。對於Arm架構中的EL1，存在二個更高例外等級EL2及EL3，且如上文所示，二者皆具有可設定的TMEA旗標。在SCR_EL3.TMEA、HCRX_EL2.TMEA二者皆設定成設陷狀態(0b1)且PSTATE.A經設定的情形中，將例外取至此等較高例外等級的最接近者（例如，當在EL0或EL1時，至EL2）。

【0058】 以此方式，PSTATE.A可保護症候群暫存器免於非預期錯誤例外，但由較高例外等級設定的TMEA控制提供「安全網」以確保在PSTATE.A經設定成0b1以指示遮罩的罕見情形中，錯誤由於仍可迅速地將例外取至較高例外等級而不會傳播。然而，在大多數情形中，PSTATE.A將設定成0b0，且因此錯誤例外可由在EL1的作業系統直接處理。

【0059】 額外地，作業系統或更高例外等級亦可設定NMEA控制位元（正交於TMEA），以指示非同步例外應係非可遮罩的而與PSTATE.A無關。提供NMEA控制以用於所有例外處理例外等級EL1、EL2、EL3。以下實例顯示此控制如何與其他控制相互作用。

【0060】 圖4係說明是否應將可遮罩例外類別（例如，非同步系統錯誤例外）的給定例外視為經遮罩以防止其等免於被採取的判定的流程圖。在步驟

100，例外控制電路系統40判定目前例外等級的特權是否比在捕捉經遮罩例外控制資訊（例如，EL2及EL3 TMEA控制）係在未設陷狀態中的情形中（若經採取）可遮罩類別的給定例外將採取的例外等級更高。再者，在步驟102，例外控制電路系統40判定遮罩控制資訊(PSTATE.A)是否在遮罩狀態中。

【0061】 若目前例外等級的特權比將處理例外（若經採取）的預設例外等級更高（忽略基於TMEA的任何捕捉）或遮罩控制資訊(PSTATE.A)係在遮罩狀態中，則例外可能被遮罩，但其可基於非可遮罩例外控制資訊（如稍早提及的NMEA控制）而被置換。在步驟104，例外控制電路系統40判定基於目前例外等級選擇的非可遮罩例外控制資訊（適當的NMEA控制）是否設定成非可遮罩狀態（例如，0b1）。若非可遮罩例外控制資訊不在非可遮罩狀態中（亦即，其在可遮罩狀態中，例如，0b0），則不置換在步驟100或102之例外應被遮罩的判定，且在步驟106，儘管經遮罩，例外控制電路系統判定是否存在捕捉例外的任何原因（例如，基於TMEA控制，如相關於圖5更詳細地討論的）。若不存在捕捉例外的原因，則在步驟107，判定給定例外經遮罩，使得該例外不被採取。

【0062】 若非可遮罩例外控制資訊係在非可遮罩狀態（例如，0b1）中，或在步驟106，判定存在捕捉例外的原因，則在步驟108，儘管步驟100及102的一者的結果係是，判定給定例外未經遮罩，使得例外仍被採取（由於NMEA控制已置換遮罩狀態的初始判定，或存在基於TMEA控制或另一控制而設下的陷阱）。

【0063】 一般而言，在步驟104使用的適用NMEA控制係基於目前例外等級選擇，但在一些實例中，其他控制資訊亦可係有關的。例如，如以下圖8E中

更詳細地顯示的，適用NMEA控制，若目前例外等級係EL3，係SCR_EL3.NMEA；若目前例外等級係EL2或控制資訊指示處理電路系統目前在主機作業系統體系中操作，係SCTLR2_EL2.NMEA；及若目前例外等級係EL1或EL0且處理電路系統目前不在主機作業系統體系中操作（見圖8B之如何基於HCR_EL2的E2H及TGE欄位及目前例外等級判定處理電路系統是否在主機作業系統體系中操作的實例），係SCTLR2_EL1.NMEA。

【0064】 若目前例外等級的特權不高於在TMEA控制係在未設陷狀態中的情形中（若經採取）給定例外將採取的預設例外等級（步驟100的否），且遮罩控制資訊不在遮罩狀態(PSTATE.A = 0b0)中，則在步驟108，給定例外未被遮罩且判定採取該例外。

【0065】 圖5係繪示基於捕捉經遮罩例外控制資訊（TMEA控制）判定是否捕捉可遮罩例外類別的給定例外的步驟的流程圖。在步驟120，例外控制電路系統判定遮罩控制資訊是否在遮罩狀態中（例如，PSTATE.A = 0b1）。若否，則在步驟122，若根據圖4未判定遮罩給定例外，則給定例外在與捕捉經遮罩例外控制資訊無關地選擇的目標例外等級中採取。

【0066】 在步驟124，例外控制電路系統判定目前例外等級的特權是否比與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低（例如，若使用TMEA控制的EL2或EL3版本其中一者，在EL0或EL1中；或若使用TMEA控制的EL3版本，在EL2中）。若目前例外等級的特權不低於與給定TMEA控制關聯的例外等級EL2或EL3，則TMEA控制係不相關的。若不存在適用於判定例外在目前例外等級中處理的相關TMEA控制，則在步驟122，若未經遮罩，給定例外在與捕捉經遮罩例外控制資訊無關地選擇的目標例外等級中採取。

【0067】 若遮罩控制資訊係在遮罩狀態中（步驟120的是），且目前例外等級的特權低於與捕捉經遮罩例外控制資訊的至少一個項關聯的預定捕捉目標例外等級（EL2或EL3），則在步驟126，例外控制電路系統40取決於捕捉經遮罩例外控制資訊是否在設陷狀態中而控制是否將給定例外捕捉至預定捕捉目標例外等級。若判定不需要捕捉，則在步驟128，若給定例外將不被遮罩，給定例外在預設例外等級採取而無需基於捕捉經遮罩例外控制資訊捕捉。此預設例外等級可基於其他控制狀態變化（例如，基於稍早提及的EA、TEA、AMO、E2H、TGE控制–見下文之此等控制在不存在基於TMEA之陷阱的情形中可如何影響目標例外等級的更詳細實例）。若基於捕捉經遮罩例外控制資訊判定需要陷阱，則在步驟130，將給定例外捕捉至與捕捉經遮罩例外控制資訊之在設陷狀態中的項關聯的預定捕捉目標例外等級。若存在捕捉經遮罩例外控制資訊之在設陷狀態中的多於一個的相關項，將陷阱設立至與捕捉經遮罩例外控制資訊之在設陷狀態(TMEA = 0b1)中之該等項之在目前例外等級與關聯預定捕捉目標例外等級之間在特權上有最小差的該一者（而與目前例外等級不同）關聯的例外等級。例如，若目前例外等級係EL1，且EL2及EL3 TMEA控制二者皆設定成1，將陷阱設立至EL2，而若目前例外等級係EL2，則將陷阱設立至EL3，即使EL2及EL3二者的TMEA控制皆經設定。

【0068】 在步驟126，是否基於TMEA控制捕捉的控制可取決於實施哪些其他控制而變化。一般而言，若不存在在設陷狀態中的相關捕捉經遮罩例外控制資訊（TMEA控制），陷阱不設立。然而，在存在捕捉經遮罩例外控制資訊之在設陷狀態中的相關項的情形中，陷阱是否發生亦可取決於其他參數。例如，在非可遮罩例外控制資訊（NMEA控制）受支援時，在一些情境中，經設

定成非可遮罩狀態的NMEA控制除了置換基於PSTATE.A的任何遮罩外亦可基於TMEA控制置換陷阱。例如，在給定例外在目前例外等級係最低特權例外等級(EL0)時發生、非可遮罩例外控制資訊係在非可遮罩狀態中（NMEA控制=0b1）、及給定例外將取至經啟用以在給定例外未經遮罩且捕捉經遮罩例外控制資訊係在未設陷狀態中的情形中用於採取給定例外的次一最低特權例外等級（例如，次一最低特權例外等級在客作業系統體系中可係EL1或在主機作業系統體系中可係EL2）的情境中，即使在捕捉經遮罩例外控制資訊(TMEA)係在設陷狀態中時，例外控制電路系統可判定給定例外應在次一最低特權例外等級中採取而不捕捉至預定捕捉目標例外等級。再者，儘管相關TMEA控制經設定成設陷狀態，步驟130的捕捉可不發生的另一情境可係當稍早提及的EA、AMO、TGE控制已指定應將例外路由至EL3或EL2時（不論遮罩資訊PSTATE的值或TMEA控制的值）。因此，在一些實例中，是否基於TMEA捕捉的決定可比只檢查TMEA控制本身的狀態更複雜。特定決策流程可取決於哪些其他控制選項實施在特定實施例中。

【0069】　然而，一般而言，提供TMEA控制以使使用者能定義在PSTATE.A旗標經設定成遮罩狀態的至少一個情境中設立至更高特權例外等級的陷阱係有用的，以允許滿足二個競爭需求（例外迅速地處理及保護與較低特權例外等級關聯的症候群資訊），而不一定在PSTATE.A旗標係在未遮罩狀態中的情形中需要將每一個例外取至較高等級。

【0070】　圖6係說明控制是否允許更新捕捉經遮罩例外控制資訊（TMEA控制資訊）的流程圖。在步驟150，執行請求更新捕捉經遮罩例外控制資訊的指令。在步驟152，處理電路系統4判定目前例外等級的特權是否比與經請求更

新之捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低（對於HCRX_EL2.TMEA係EL2，對於SCR_EL3.TMEA係EL3）。若如此，則在步驟154，更新捕捉經遮罩例外控制資訊的嘗試被徹底拒絕或捕捉至更高特權例外等級，使得在較高特權例外等級的軟體可判定是否允許該經請求更新。若目前例外等級係預定捕捉目標例外等級或係比預定捕捉目標例外等級更高特權的例外等級，則在步驟156，捕捉經遮罩例外控制資訊的經請求更新在目前例外等級被允許（除非存在經實施而將由於其他原因導致捕捉對捕捉經遮罩例外控制資訊的此一寫入的任何其他控制）。例如，在EL3的軟體可設定控制以捕捉對HCRX_EL2的存取，導致對EL2 TMEA控制的寫入從EL2捕捉至EL3。

【0071】 圖7係說明處理非可遮罩例外（例如，同步外部中止例外）之類別的流程圖。在步驟160，偵測到此類別的非可遮罩例外。在步驟162，例外控制電路系統判定暫存器14中的控制資訊是否指示目前例外等級的特權低於與捕捉經遮罩例外控制資訊之給定項關聯的預定捕捉目標例外等級，且目前例外等級能夠係用於處理例外之目標例外等級的情境（例如，EL0可能不能夠係目標例外等級）。若否，則在步驟164，不需要基於TMEA控制的陷阱，且非可遮罩類別的例外在基於目前例外等級及/或基於（TMEA控制以外的）其他控制資訊選擇的目標例外等級採取。

【0072】 若在步驟162，判定目前例外等級的特權低於與捕捉經遮罩例外控制資訊之給定項關聯的預定捕捉目標例外等級，且目前例外等級能夠係用於處理例外的目標例外等級（且可選地，在一些實例中，至少一個其他條件獲得滿足，例如，取決於可能要求與由TMEA所指示的陷阱有不同的行為的

SCR_EL3.EA或HCR_EL2.TEA控制），則基於TMEA控制（捕捉經遮罩例外控制資訊）但取決於其他參數的陷阱係可行的。

【0073】 在步驟166，例外控制電路系統判定遮罩控制資訊是否在遮罩狀態中（例如，PSTATE.A = 0b1）。若否，則在步驟168，不需要基於TMEA的陷阱，且非可遮罩例外在基於目前例外等級及/或其他控制資訊選擇的目標例外等級採取。

【0074】 在步驟170，例外控制電路系統判定捕捉經遮罩例外控制資訊是否在設陷狀態中（例如，相關TMEA控制經設定成0b1）。

【0075】 若遮罩控制資訊係在遮罩狀態中（步驟166的是）且捕捉經遮罩例外控制資訊係在設陷狀態中（步驟170的是），則將陷阱設立至與經設定成設陷狀態的TMEA控制關聯的預定捕捉目標例外等級（再次，若二個TMEA控制經設定且目前例外等級係EL1，則將陷阱設立至EL2，在特權上與目前例外等級最接近的等級）。若遮罩控制資訊係在未遮罩狀態中(PSTATE = 0b0)或捕捉經遮罩例外控制資訊係在未設陷狀態中（所有TMEA控制皆係0b0），基於TMEA的陷阱未設立（步驟168），雖然此不一定由於其他原因而排除至更高等級的陷阱，諸如基於EA、TEA控制。

【0076】 因此，即使PSTATE.A旗標未導致非可遮罩例外的遮罩，然而可考慮將其用於基於TMEA控制而控制是否將例外捕捉至更高特權等級。此由於在PSTATE.A旗標經設定的情形中，存在保留症候群資訊的偏好而係有用的，所以捕捉至較高例外等級有助於滿足此需求，即使實際上已經不存在任何遮罩。然而，使此捕捉取決於PSTATE.A有助於增加核心優先例外處理的使用，且降低韌體需要處理例外之時機的數目，簡化軟體開發。

【0077】 將理解在流程圖之結果取決於若干個條件之檢查的部分中，該等條件可以任何順序檢查或在一些情形中至少部分地平行地檢查，所以顯示在流程圖中的步驟的特定次序僅係此可如何完成的一個實例。

【0078】 現在描述稍早描述之控制資訊的各種項可如何用以控制例外處理的更詳細實例。

【0079】 圖8A至圖8O說明指示在記憶體系統存取之後偵測到錯誤之非同步系統錯誤(SError)例外的處理，其中該錯誤無法歸因於指令串流的特定點，所以不知道哪些指令可能受該錯誤影響。此係上文描述之可遮罩例外類別的實例。作為整體的此等圖式說明用於基於例外時的目前例外等級及圖3所示的控制資訊判定例外是否應遮罩的程序，且若不遮罩，哪個例外等級應採取該例外。

【0080】 圖8A說明指示SError例外是否將從較低例外等級路由以在EL3採取的值route_to_el3的判定。若目前例外等級非EL3（步驟1000的否）且SCR_EL3.EA = 0b1(步驟1002的是)，route_to_el3係真（步驟1006），指示例外將路由至EL3。若目前例外等級係EL3（步驟1000的是）或SCR_EL3.EA = 0b0（步驟1002的否），則route_to_el3係偽（步驟1004）。

【0081】 圖8B說明指示處理目前是否在主機作業系統體系（類型2超管理器）中的值IsInHost()的判定。IsInHost()若在客作業系統體系中係偽且若在主機作業系統體系中係真。若目前例外等級既非EL2（步驟1008的否）也非EL0（步驟1014的否），則IsInHost()係偽（步驟1018）。若目前例外等級係EL2（步驟1008的是），但HCR_EL2.E2H係0b0（步驟1010的否）或若目前例外等級係EL0（步驟1014的是）且HCR_EL2.E2H及HCR_EL2.TGE的一或二者係0b0

（步驟1016的否），IsInHost()亦係偽（步驟1018）。若目前例外等級係EL2（步驟1008的是）且HCR_EL2.E2H = 0b1（步驟1010的是）或目前例外等級係EL0（步驟1014的是）且HCR_EL2.E2H及HCR_EL2.TGE二者皆設定成0b1（步驟1016的是），IsInHost()係真（步驟1012）。

【0082】 圖8C說明指示SError例外是否將從較低例外等級EL0或EL1路由以在EL2採取的值route_to_el2的判定。若route_to_el3係真（步驟1020的否），則route_to_el2係偽（步驟1032）。若route_to_el3係偽（步驟1020的是）且目前例外等級係EL2或EL3（步驟1022及1030二者皆否），則route_to_el2係偽（步驟1032）。

【0083】 若route_to_el3係偽（步驟1020的是）且目前例外等級係EL1（步驟1022的是），則：

- 在步驟1026，若HCR_EL2.AMO係0b1（步驟1024的是），route_to_el2係真且
- 在步驟1028，若HCR_EL2.AMO係0b0（步驟1024的否），route_to_el2係偽。

【0084】 若route_to_el3係偽（步驟1020的是），目前例外等級係EL0（步驟1022的否，步驟1030的是），且如圖8B之程序中判定的IsInHost()係真（步驟1034的否）或HCR_EL2.E2H及HCR_EL2.TGE二者皆設定成0b0（步驟1038的是），route_to_el2係偽（步驟1036）。

【0085】 若route_to_el3係偽（步驟1020的是），目前例外等級係EL0（步驟1022的否，步驟1030的是），IsInHost()在圖8B的程序中經判定係偽（步

驟1034的是），且HCR_EL2.E2H及HCR_EL2.TGE的一或二者經設定成0b1（步驟1038的否），route_to_el2係真（步驟1040）。

【0086】 圖8D說明表示SError例外是否應遮罩之初始判定（如在圖8E及圖8F中於下文提及的，基於非可遮罩例外控制資訊（NMEA控制）置換此初始判定係可行的）的值soft_masked的判定。soft_masked的判定取決於目前例外等級（步驟1042）。

【0087】 若目前例外等級係EL3，則soft_masked係真（步驟1050），若SCR_EL3.EA = 0b0（步驟1044的是）或PSTATE.A = 0b1（步驟1046的是），且若SCR_EL3.EA = 0b1（步驟1044的否）及PSTATE.A = 0b0（步驟1046的否）二者，則soft_masked係偽（步驟1048）。

【0088】 若目前例外等級係EL2，則：

- 若如圖8A中判定的route_to_el3係真（步驟1052的否），soft_masked係偽（步驟1054）。
- 若route_to_el3係偽（步驟1052的是），則：
    - 若HCR_EL2.TGE及HCR_EL2.AMO二者皆係0（步驟1056的是）或若PSTATE.A = 1（步驟1058的是），soft_masked係真（步驟1060）。
    - 若HCR_EL2.TGE及HCR_EL2.AMO的一或二者係0b1（步驟1056的否）且PSTATE.A = 0（步驟1058的否），soft_masked係偽（步驟1062）。

【0089】 若目前例外等級係EL0或EL1，則：

- 若如圖8A中判定的route_to_el3係真（步驟1064的否）或如圖8C中判定的route_to_el2係真（步驟1066的否），soft_masked係偽（步驟1070）。
- 若route_to_el3及route_to_el2皆係偽，則soft_masked，若PSTATE.A = 1（步驟1068的是），係真（步驟1072），且若PSTATE.A = 0（步驟1068的否），係偽（步驟1070）。

【0090】 應注意，因為步驟1044、1056通過，若EA、TGE、AMO控制尚未設定以將SError例外捕捉至較高等級的一者，步驟1044、1056的效果係在目前例外等級（在步驟1050的情形中係EL3或在步驟1060的情形中係EL2）的特權比指定用於採取例外等級的例外等級更高（忽略TMEA捕捉控制）的情形中可將SError例外視為在步驟1050或1060經遮罩。

【0091】 圖8E說明選擇使用非可遮罩例外控制資訊（NMEA控制）之各別項的何者以用以判定在圖8D中判定的遮罩狀態是否應置換。此取決於目前例外等級及如在圖8B中判定的IsInHost()狀態。若目前例外等級係EL3（步驟1076的是）則所使用的NMEA控制係SCR_EL3.NMEA（步驟1078）。若目前例外等級係EL2（步驟1080的是）或IsInHost()係真（步驟1084的是），則所使用的NMEA控制係SCTLR2_EL2.NMEA（步驟1082）。若目前例外等級係EL1或EL0（步驟1080的否）且IsInHost()係偽（步驟1084的否），則所選擇的NMEA控制係SCTLR2_EL1.NMEA（步驟1086）。

【0092】 圖8F說明應用經選擇NMEA控制以判定圖8D的初始遮罩判定是否應置換以防止遮罩SError例外。若在圖8D的程序中已將soft_masked判定為偽（步驟1098的否），則不論經選擇NMEA控制的值，其將保持偽（步驟

1100）。若soft_masked係真（步驟1098的是）且經選擇NMEA控制係在非可遮罩狀態0b1中（步驟1102的否），則soft_masked變成偽（步驟1100），使得先前遮罩狀態已基於由NMEA控制提供的非可遮罩指示而置換。若soft_masked係真（步驟1098的是）且經選擇NMEA控制係在可遮罩狀態0b0中（步驟1102的是），則soft_masked保持真（步驟1104），且因此SError例外將被遮罩以防止其被採取（又–該例外保持待決且因此若PSTATE.A旗標於稍後清除成未遮罩狀態0b0，仍可被採取）。

【0093】 圖8G說明指示是否應基於TMEA控制的EL2版本HCRX_EL2.TMEA設立至EL2之陷阱的信號route_soft_masked_to_el2的判定。此係當PSTATE.A = 1時設立的陷阱以確保SError例外可在EL2更迅速地處理，同時仍保留在該EL1症候群暫存器組中的例外症候群資訊。

【0094】 在步驟1122，若在圖8A中判定的route_to_el3係真（步驟1108的否）或外部偵錯器已停用例外免於在低於EL3的例外等級採取（步驟1110的是），參數route_soft_masked_to_el2經判定成偽（亦即，基於TMEA的捕捉未發生）。假設此等條件皆未滿足（步驟1108的是及步驟1110的否），則：

- 若HCRX_EL2.TMEA係在未設陷狀態中（0b0–步驟1112的否），則route_soft_masked_to_el2係偽（步驟1122），因為在此情形中，EL2軟體尚未設定捕捉經遮罩例外控制資訊以指示相依於PSTATE.A之至EL2的陷阱係所欲的。
- 若HCRX_EL2.TMEA係在設陷狀態中（0b1–步驟1112的是），則導因於TMEA而至EL2的陷阱係可能的，但其發生與否亦取決於其他參數：

- 若目前例外等級係EL1（步驟1114的是）且PSTATE.A係在遮罩狀態中（0b1–步驟1116的是），則route_soft_masked_to_el2係真（步驟1118）。
- 若目前例外等級係EL1（步驟1114的是）但PSTATE.A係在未遮罩狀態中（0b0–步驟1116的否），則route_soft_masked_to_el2係偽（步驟1120）。
- 若目前例外等級係EL0（步驟1124的是），遮罩狀態（在任何NMEA置換經評估之後）係真（步驟1126的是）且在圖8B中判定的IsInHost()係偽（步驟1128的是），則route_soft_masked_to_el2係真（步驟1130）。應注意在步驟1126，用語「soft_masked」係指在圖8F的程序中基於NMEA判定的最終遮罩狀態，而非在圖8D中判定的初始遮罩狀態（在任何NMEA置換之前）。
- 若目前例外等級係EL0（步驟1124的是），遮罩狀態如在圖8F的處理中在NMEA置換之後（非在圖8D中判定的初始遮罩狀態）經判定係偽（步驟1126的否）或在圖8B中判定的IsInHost()係真（步驟1128的否），則route_soft_masked_to_el2係偽（步驟1122）。
- 若目前例外等級係EL2或EL3（步驟1124的否），則route_soft_masked_to_el2係偽（步驟1122）。

【0095】 圖8H說明指示至EL3的陷阱是否應基於TMEA控制的EL3版本SCR_EL3.TMEA設立的信號route_soft_masked_to_el3的判定。再次，若採取，

此捕捉意圖允許SError例外在EL3更迅速地處理，同時假定PSTATE.A = 1，仍促成在EL1或EL2組的症候群暫存器中的例外症候群資訊的保留。

【0096】 若在EL3的例外處理由外部偵錯器所停用（步驟1136的是），則route_soft_masked_to_el3係偽（步驟1134）。

【0097】 若SCR_EL3.TMEA係在未設陷狀態中（0b0–步驟1138的否），則route_soft_masked_to_el3係偽（步驟1134），因為在此情形中，EL3軟體未表示出當PSTATE.A = 1時避免在較低等級遮罩SError例外的任何偏好。

【0098】 若例外未由外部偵錯器在EL3停用（步驟1136的否）且SCR_EL3.TMEA係在設陷狀態中（0b1–步驟1138的是），則導因於TMEA而至EL3的陷阱係可能的，但其發生與否亦取決於其他參數：

- 若route_to_el2或route_soft_masked_to_el2其中一者係真（步驟1140或步驟1142的否），則：
    - 若目前例外等級的特權低於EL3（步驟1150的否）且在比EL3更低特權的等級採取例外目前由外部偵錯器所停用（步驟1152的是），route_soft_masked_to_el3係真（步驟1154）；及
    - 若目前例外等級係EL3（步驟1150的是）或在比EL3更低特權的等級採取例外目前未由外部偵錯器所停用（步驟1152的否），route_soft_masked_to_el3係偽（步驟1156）。

因此，在外部偵錯器停用例外免於在EL0至EL2採取不存在時，如圖8G所判定的至EL2的TMEA陷阱將優先於至EL3的TMEA陷阱–導因於步驟1142，若SCR_EL3.TMEA及HCRX_EL2.TMEA兩者皆設定成設陷

狀態，陷阱將設立至EL2（當HCRX_EL2.TMEA相關時，SError可發生之最接近EL1/EL0等級的例外等級（在特權上的最小差））。

- 若route_to_el2及route_soft_masked_to_el2二者皆係偽（步驟1140及1142的是），若以下之其中一者係真，則route_soft_masked_to_el3係真（步驟1146）：
    - 目前例外等級係EL2或EL1且PSTATE.A係在遮罩狀態中（0b1-步驟1144的是），或
    - 目前例外等級係EL2或EL0且遮罩狀態係真（步驟1148的是）。再次，步驟1148中的soft_masked係指在圖8F的程序中在NMEA置換之後的遮罩狀態（非在圖8D中判定的初始遮罩狀態）。
- 若route_to_el2及route_soft_masked_to_el2二者皆係偽（步驟1140及1142的是），在步驟1144、1148測試的條件皆未滿足，目前例外等級的特權低於EL3（步驟1150的否），且在比EL3更低特權的等級採取例外目前由外部偵錯器停用（步驟1152的是），route_soft_masked_to_el3亦可係真（步驟1154）。

【0099】 注意到圖8G的步驟1116與1126之間及圖8H的步驟1144與1148之間的不同。對於目前例外等級係EL1時發生的例外，TMEA陷阱取決於係1的PSTATE.A而不論在圖8F中判定的實際遮罩狀態是否已使用NMEA控制置換。此係由於在EL1中，將PSTATE.A設定成遮罩狀態(0b1)指示在EL1之處理的目前狀態使得存在一些重要的症候群資訊要保護在EL1症候群暫存器中，因此即使已使用NMEA控制置換遮罩，執行至EL2或EL3的TMEA陷阱以保護在EL1症候

群暫存器中的症候群資訊仍係所欲的。相比之下，對於例外等級EL0，TMEA陷阱取決於在圖8F中判定的實際遮罩狀態（在使用NMEA控制置換之後）。因此，設定至1的NMEA控制將防止TMEA陷阱從EL0設立至EL2或EL3，且替代地非遮罩例外將取至其預設例外等級（例如，EL1），除非另一控制取得優先權（例如，EA、AMO、TGE控制）。EL0的此行為由於在EL0例外等級中（其不能夠係用於處理例外的目標例外等級）不存在要保護的例外症候群資訊而係有用的，由於可假設從擷取EL1症候群資訊至今已存在傳回至EL0的先前例外，不再需要任何EL1症候群資訊，因此即使相關TMEA控制經設定，不需要至EL2或EL3的陷阱。因此，即使相關TMEA控制經設定成設陷狀態(0b1)，在NMEA控制已防止遮罩發生的情形中，不需要至EL2或EL3的陷阱以保護EL1的症候群資訊。

【0100】　圖8I顯示指示是否存在任何原因以將SError例外從否則將採取該例外的預設例外等級（例如，在客作業系統環境中係EL1，或在主機作業系統環境中係EL2）路由離開的「routed_away」指示的判定。若如在分別顯示於圖8C、圖8G、圖8A、圖8H中的程序中判定的route_to_el2、route_soft_masked_to_el2、route_to_el3、或route_soft_masked_to_el3的任一者係真（步驟1158、1162、1164、1166的是），routed_away指示係真（步驟1160）。若route_to_el2、route_soft_masked_to_el2、route_to_el3、及route_soft_masked_to_el3皆係偽（步驟1166的否），則routed_away係偽（步驟1168）。

【0101】　圖8J說明指示是否實際遮罩例外（防止被採取）的最終遮罩狀態的判定。若如在圖8F中在任何NMEA置換之後判定的soft_masked係真（步驟

1170的是）且如在圖8I中判定的routed_away係偽（步驟1174的是），則masked係真（步驟1176），其指示將防止該例外免於被採取。若soft_masked（在任何NMEA置換之後）係偽（步驟1170的否）或routed_away係真（步驟1174的否），則由於例外將被採取，masked係偽（步驟1172）。

【0102】 圖8K說明例外是否應在EL3採取的判定。若目前例外等級係EL3（步驟1178的是）且例外未經遮罩（步驟1180的是），則take_in_el3係真（步驟1184），否則take_in_el3係偽（步驟1182）。

【0103】 圖8L說明例外是否應在EL2採取(take_in_el2_0)的判定。若目前例外等級係EL2（步驟1186的是）或如圖8B中判定的IsInHost()係真（步驟1188的是），如在圖8I中判定的routed_away係偽（步驟1192的是）且如圖8J中例外經判定為未遮罩（步驟1194的是），則take_in_el2_0係真（步驟1196）。否則，take_in_el2_0係偽（步驟1190）。

【0104】 圖8M說明例外是否應在EL1採取(take_in_el1_0)的判定。若目前例外等級係EL1（步驟1198的是）或目前例外等級係EL0且IsInHost()係偽（步驟1200的是），routed_away係偽（步驟1204的是）且例外經判定為未遮罩（步驟1206的是），則take_in_el1_0係真（步驟1208）。否則，take_in_el1_0係偽（步驟1202）。

【0105】 圖8N說明用於採取例外之目標例外等級的選擇。若例外根據圖8J判定成經遮罩（步驟1210的否），則由於例外將不被採取，目標例外等級係未知。

【0106】 若「masked」狀態係偽（步驟1210的是）且take_in_el3（圖8K）、route_to_el3（圖8A）、及route_soft_masked_to_el3（圖8H）的任一者係真（步驟1214的是），則目標例外等級係EL3（步驟1216）。

【0107】 若masked係偽（步驟1210的是），例外將不在EL3採取（步驟1214的否），且take_in_el2_0（圖8L）、route_to_el2（圖8C）、及route_soft_masked_to_el2（圖8G）的任一者係真（步驟1218的是），則目標例外等級係EL2（步驟1220）。

【0108】 若masked係偽（步驟1210的是），例外將不在EL3或EL2採取（步驟1214及1218的否），則目標例外等級係EL1（步驟1224）。應注意，在一些情形中，在判定目標例外等級係EL1之前，可存在「take_in_el1_0」是否係真的額外檢查（根據圖8M判定）。然而，實際上，如下文之虛擬碼中提到的，在步驟1210、1214、1218的結果皆係否且「take_in_el1_0」係偽時的狀態應係不可到達的，所以「take_in_el1_0」是否係真的檢查可係冗餘的。

【0109】 圖8O說明基於外部偵錯器在圖8N判定的在目標例外等級停用例外而遮罩例外的進一步能力。若外部偵錯器已在目標例外等級停用例外（步驟1228的是），masked變成真且目標例外等級變成未知（步驟1232）。否則，維持如分別根據圖8J及圖8N所判定的masked狀態及目標例外等級。

【0110】 圖9A至圖9I說明指示在記憶體系統存取之後偵測到外部記憶體系統錯誤（除了MMU故障外）的同步外部中止例外的處理，其中該錯誤能夠歸因於所執行之指令串流的特定點。此係上文描述之非可遮罩例外類別的實例。作為整體的此等圖式說明用於基於例外時的目前例外等級及圖3所示的控制資訊判定應採取例外的哪個例外等級的程序。此類型的例外不能被遮罩。對

於圖9A至圖9I中對「IsInHost()」的參考，此以與圖8B所示之用於非同步SError例外相同的方式判定，所以此不於此處再次重複。

【0111】 圖9A說明指示同步外部中止例外是否將從較低例外等級路由以在EL3採取的值route_to_el3的判定。若目前例外等級非EL3（步驟1234的否）且SCR_EL3.EA = 0b1（步驟1236的是），則route_to_el3係真（步驟1240），指示例外將路由至EL3。若目前例外等級係EL3（步驟1234的是）或SCR_EL3.EA = 0b0（步驟1236的否），則route_to_el3係偽（步驟1238）。

【0112】 圖9B說明指示同步外部中止是否將從較低例外等級路由以在EL2採取的值route_to_el2的判定。對「tea_bit」的參考係指HCR_EL2.TEA控制的值（步驟1254）。若如在圖9A中判定的route_to_el3係真（步驟1256的否），則route_to_el2係偽（步驟1258）。

【0113】 若route_to_el3係偽（步驟1256的是）且目前例外等級係EL1（步驟1260的是），則：

- 若tea_bit係1或故障係待於EL2處理的特定類型（步驟1262的是），則route_to_el2係真（步驟1264）。例如，待於EL2處理的特定類型的故障可包括與對正通過對記憶體的存取仿真之由在EL2執行的軟體管理的系統暫存器的存取有關的故障，或可係由在二級位址轉譯體系中的MMU偵測到的第二級位址轉譯故障，其中位址轉譯係基於由作業系統定義的第一級頁表控制以將虛擬位址映射至中間位址，且基於由超管理器定義的第二級頁表控制以將中間位址映射至實體位址。
- 若tea_bit係0且該故障非待於EL2處理的類型（步驟1262的否），則route_to_el2係偽（步驟1266）。

【0114】 若route_to_el3係偽（步驟1256的是）且目前例外等級係EL0（步驟1268的是），則：

- 若IsInHost()（根據圖8B判定）係偽（步驟1270的是）且tea_bit係1、HCR_EL2.TGE控制係1、或故障係待於EL2處理的類型之一者（步驟1274的是），route_to_el2係真（步驟1276）。
- 若IsInHost()係真（步驟1270的否）或所有下列條件皆獲得滿足，route_to_el2係偽（步驟1272）：tea_bit = 0b0、HCR_EL2.TGE = 0b0且故障非待於EL2處理的類型（步驟1274的否）。

【0115】 若目前例外等級係EL3或EL2，則route_to_el2係偽（步驟1278，在步驟1268的否之後）。

【0116】 圖9C說明指示是否應基於TMEA控制的EL2版本HCRX_EL2.TMEA設立至EL2之陷阱的信號route_soft_masked_to_el2的判定。此係當PSTATE.A = 1時設立的陷阱以確保同步外部中止例外可在EL2更迅速地處理，同時促成該組EL1症候群暫存器中的例外症候群資訊的保留。

【0117】 若施加下列的任何一或多者，參數route_soft_masked_to_el2經判定成係偽（步驟1282）：

- route_to_el3（如根據圖9A判定）係真（步驟1280的否）；
- 目前例外等級係EL1以外的任何例外等級（步驟1284的否）；
- PSTATE.A = 0（步驟1286的否）；
- HCRX_EL2.TMEA = 0（步驟1288的否）。

【0118】 若所有以下條件皆獲得滿足，參數route_soft_masked_to_el2經判定成係真（步驟1282）：

- route_to_el3係偽（步驟1280的是）；
- 目前例外等級係EL1（步驟1284的是）；
- PSTATE.A = 1（步驟1286的是）；
- HCRX_EL2.TMEA = 1（步驟1288的是）。

【0119】 因此，HCRX_EL2.TMEA經設定成1（設陷狀態）導致若同步外部中止例外在PSTATE.A設定成遮罩狀態時在EL1發生，將該等同步外部中止例外捕捉至EL2。若PSTATE.A係0（未遮罩狀態），此陷阱未設立。

【0120】 圖9D說明指示至EL3的陷阱是否應基於TMEA控制的EL3版本SCR_EL3.TMEA設立的信號route_soft_masked_to_el3的判定。

【0121】 若施加下列的任何一或多者，參數route_soft_masked_to_el3經判定成係偽（步驟1294）：

- route_to_el2（如根據圖9B判定）係真（步驟1292的否）；
- route_soft_masked_to_el2（如根據圖9C判定）係真（步驟1296的否）；
- 目前例外等級係EL1或EL2以外的任何例外等級（步驟1298的否）。
- PSTATE.A = 0（步驟1300的否）。
- SCR_EL3.TMEA = 0（步驟1302的否）。

【0122】 若所有以下條件皆獲得滿足，參數route_soft_masked_to_el3經判定成係真（步驟1304）：

- route_to_el2係偽（步驟1292的是）；
- route_soft_masked_to_el2係偽（步驟1296的否）；
- 目前例外等級係EL1或EL2（步驟1298的是）；

- PSTATE.A = 1（步驟1300的是）；
- SCR_EL3.TMEA = 1（步驟1302的是）。

【0123】 因此，SCR_EL3.TMEA經設定成1（設陷狀態）導致若同步外部中止例外基於HCRX_EL2.TMEA或HCR_EL2.TEA將不捕捉至EL2且在PSTATE.A設定成1（遮罩狀態）時在EL1或EL2發生，將該等同步外部中止例外捕捉至EL3。若PSTATE.A係0（未遮罩狀態），此陷阱未設立。

【0124】 圖9E顯示指示是否存在任何原因以將同步外部中止例外從否則將採取該例外的預設例外等級（例如，EL1或EL2）路由離開的「routed_away」指示的判定。若如在分別顯示於圖9B、圖9C、圖9A、圖9D中的程序中判定的route_to_el2、route_soft_masked_to_el2、route_to_el3、或route_soft_masked_to_el3的任一者係真（步驟1306、1310、1312、1314的是），routed_away指示係真（步驟1308）。若route_to_el2、route_soft_masked_to_el2、route_to_el3、及route_soft_masked_to_el3皆係偽（步驟1306、1310、1312、1314的否），則routed_away係偽（步驟1316）。

【0125】 圖9F說明例外是否應在EL3採取(take_in_el3)的判定。若目前例外等級係EL3（步驟1318的是），則take_in_el3係真（步驟1322），否則take_in_el3係偽（步驟1320）。

【0126】 圖9G說明例外是否應在EL2採取(take_in_el2_0)的判定。若目前例外等級係EL2（步驟1324的是）或如圖8B中判定的IsInHost()係真（步驟1326的是），且如在圖9E中判定的routed_away係偽（步驟1330的是），則take_in_el2_0係真（步驟1332）。否則，take_in_el2_0係偽（步驟1328）。

【0127】 圖9H說明例外是否應在EL1採取(take_in_el1_0)的判定。若目前例外等級係EL1（步驟1334的是）或目前例外等級係EL0且IsInHost()係偽（步驟1336的是）且routed_away係偽（步驟1340的是），則take_in_el1_0係真（步驟1342）。否則，take_in_el1_0係偽（步驟1338）。

【0128】 圖9I說明用於採取例外之目標例外等級的選擇。若take_in_el3（圖9F）、route_to_el3（圖9A）、及route_soft_masked_to_el3（圖9D）的任一者係真（步驟1344的是），則目標例外等級係EL3（步驟1346）。

【0129】 若例外將不在EL3採取（步驟1344的否）且take_in_el2_0（圖9G）、route_to_el2（圖9B）、及route_soft_masked_to_el2（圖9C）的任一者係真（步驟1348的是），則目標例外等級係EL2（步驟1350）。

【0130】 若例外將不在EL3或EL2中採取（步驟1348的否），則目標例外等級係EL1（步驟1354）。應注意，在一些情形中，在判定目標例外等級係EL1之前，可存在「take_in_el1_0」是否係真的額外檢查（根據圖9H判定）。然而，實際上，如下文之虛擬碼中提到的，在步驟1344及1348的結果皆係否且「take_in_el1_0」係偽時的狀態應係不可到達的，所以「take_in_el1_0」是否係真的檢查可係冗餘的。

【0131】 對於顯示於上方的流程圖，將理解相同邏輯功能可藉由以不同順序執行步驟或平行地執行一些步驟而達成。再者，並非所有步驟對給定實施例皆係必要的。在一些實施方案中，可能不支援某些類型的例外路由控制，如以下虛擬碼中所示，其中存在對於某些可選特徵是否受支援的一些額外測試（例如，見「???Enabled()」或「Have???()」形式的變數測試，其等係指示某些可選特徵是否經啟用及經實施的變數）。若此等可選特徵的一者不受支援，顯

示於虛擬碼中的AND或布林關係顯示將發生什麼事。在大多數情形中，若例外路由控制不受支援，將對應控制值視為猶如其係0b0。

【0132】　用於實施圖8A至圖8O之功能的實例虛擬碼顯示如下：

```
// Returns a tuple of whether SError exception can be taken and, if
// so, the target Exception level
(boolean, bits(2)) AArch64.PhysicalSErrorTarget()
boolean route_to_el3;
boolean route_to_el2;
// The exception is explicitly routed to EL3
if PSTATE.EL != EL3 then
    route_to_el3 = (HaveEL(EL3) && SCR_EL3.EA == '1');
else
    route_to_el3 = FALSE;
// The exception is explicitly routed to EL2
if !route_to_el3 && EL2Enabled() && PSTATE.EL == EL1 then
    route_to_el2 = (HCR_EL2.AMO == '1');
elsif !route_to_el3 && EL2Enabled() && PSTATE.EL == EL0 then
    route_to_el2 = (!IsInHost() && HCR_EL2.<TGE,AMO> != '00');
else
    route_to_el2 = FALSE;
// The exception is "masked" by software or hardware
boolean soft_masked;
case PSTATE.EL of
```

```
        when EL3
            soft_masked = (SCR_EL3.EA == '0' || PSTATE.A == '1');
        when EL2
            soft_masked = (!route_to_el3 &&
                           (HCR_EL2.<TGE,AMO> == '00' ||
                            PSTATE.A == '1'));
        when EL1, EL0
            soft_masked = (!route_to_el3 && !route_to_el2
                                         && PSTATE.A == '1');
    // The exception is "disabled" by external debug in the current Security state
    boolean intdis = ExternalDebugInterruptDisabled(EL1);
    // When FEAT_DoubleFault or FEAT_DoubleFault2 is implemented,
    // the mask might be overridden
    if HaveDoubleFault2Ext() then
        bit nmea_bit;
        if PSTATE.EL == EL3 then
            nmea_bit = SCR_EL3.NMEA;
        elsif PSTATE.EL == EL2 || IsInHost() then
            nmea_bit = if IsSCTLR2EL2Enabled() then SCTLR2_EL2.NMEA
                       else '0';
        else
            nmea_bit = if IsSCTLR2EL1Enabled() then SCTLR1_EL2.NMEA
                       else '0';
        soft_masked = soft_masked && (nmea_bit == '0');
```

```
elsif HaveDoubleFaultExt() && PSTATE.EL == EL3 then
        bit nmea_bit = SCR_EL3.NMEA AND SCR_EL3.EA;
        soft_masked = soft_masked && (nmea_bit == '0');
boolean route_soft_masked_to_el3;
boolean route_soft_masked_to_el2;
if HaveDoubleFault2Ext() then
    // The soft_masked exception is routed to EL2
    route_soft_masked_to_el2 =
                    (EL2Enabled() && !route_to_el3 && !intdis &&
                    ((PSTATE.EL == EL1 && PSTATE.A == '1') ||
                    (PSTATE.EL == EL0 && soft_masked && !IsInHost()))
                     && (IsHCRXEL2Enabled() && HCRX_EL2.TMEA == '1'));
    // The soft_masked exception is routed to EL3
    route_soft_masked_to_el3 =
                (HaveEL(EL3) && !ExternalDebugInterruptDisabled(EL3) &&
                    ((!(route_to_el2 || route_soft_masked_to_el2) &&
                     ((PSTATE.EL IN {EL2, EL1} && PSTATE.A == '1') ||
                      (PSTATE.EL IN {EL2, EL0} && soft_masked))) ||
                      (PSTATE.EL != EL3 && intdis)) &&
                      SCR_EL3.TMEA == '1');
else
    route_soft_masked_to_el2 = FALSE;
    route_soft_masked_to_el3 = FALSE;
// The exception is routed away from the current exception regime
```

```
routed_away = (route_to_el2 || route_soft_masked_to_el2 ||
                    route_to_el3 || route_soft_masked_to_el3);
// The exception is actually "masked" - "disabled" will be factored in
// later
masked = soft_masked && !routed_away;
// The exception is taken at EL3
take_in_el3 = PSTATE.EL == EL3 && !masked;
// The exception is taken at EL2 or from the Host EL0 to EL2
take_in_el2_0 = ((PSTATE.EL == EL2 || IsInHost()) &&
                        !routed_away && !masked);
// The exception is taken at EL1 or from the non-Host EL0 to EL1
take_in_el1_0 = ((PSTATE.EL == EL1 ||
                    (PSTATE.EL == EL0 && !IsInHost())) &&
                   !routed_away && !masked);
bits(2) target_el;
if masked then
    target_el = bits(2) UNKNOWN;
elsif take_in_el3 || route_to_el3 || route_soft_masked_to_el3 then
    target_el = EL3;
elsif take_in_el2_0 || route_to_el2 || route_soft_masked_to_el2 then
    target_el = EL2;
elsif take_in_el1_0 then
    target_el = EL1;
else
```

```
        Unreachable();
// The exception is "disabled" by external debug for the actual target
if ExternalDebugInterruptDisabled(target_el) then
        masked = TRUE;
        target_el = bits(2) UNKNOWN;
return (masked, target_el);
```

Example pseudocode for the function IsInHost() is as follows:

```
// IsInHost()
// ==========
boolean IsInHost(bits(2) el)
        if !HaveVirtHostExt() || ELUsingAArch32(EL2) then
                return FALSE;
        case PSTATE.EL of
                when EL3
                        return FALSE;
                when EL2
                        return EL2Enabled() && HCR_EL2.E2H == '1';
                when EL1
                        return FALSE;
                when EL0
                        return EL2Enabled() && HCR_EL2.<E2H,TGE> == '11';
                otherwise
                        Unreachable();
```

【0133】 圖10A及圖10B顯示指示對於控制資訊之值的各組合，在例外等級EL0、EL1、EL2、EL3的任一者中發生的非同步系統錯誤(SError)例外是否將被採取或遮罩，且若採取，哪個例外等級會係採取例外的目標例外等級的彙總表。

【0134】 表之左側中的各行係指用於經標記例外等級之系統暫存器的一者中的位元，或係指PSTATE.A旗標（其中「x」指示對應控制係0b0或0b1並不重要），說明如下：

- EA EL3意指SCR_EL3.EA
- TMEA EL3意指SCR_EL3.TMEA
- NMEA EL3意指SCR_EL3.NMEA
- TGE EL2意指HCR_EL2.TGE
- E2H EL2意指HCR_EL2.E2H
- AMO EL2意指HCR_EL2.AMO
- NMEA EL2意指SCTLR2_EL2.NMEA
- TMEA EL2意指HCRX_EL2.TMEA
- NMEA EL1意指SCTLR2_EL1.NMEA
- A PSTATE意指PSTATE.A。

【0135】 該表之右側中的各行顯示若SError例外針對該組對應控制資訊值發生，當目前例外等級分別係EL3、EL2、EL1、EL0時的結果。右側行中的各單元表示以下結果的一者：

- ELx意謂著將錯誤例外取至ELx。
- B意謂著由於錯誤例外由PSTATE.A遮罩，其未被採取且保持待決。

- C意謂著由於錯誤例外經隱含地遮罩（導因於目前例外等級的特權比應採取該例外的例外等級更高），其未被採取且保持待決。
- 不適用(n/a)：意謂著該例外等級不可在此組態中存取。

【0136】 此等值係根據以上虛擬碼及顯示於圖8A至圖8O中的功能判定。

【0137】 TMEA及PSTATE.A以外的任何控制選項皆係可選的，且若省略，可以與猶如將其等設定成0b0的相同方式處理。

【0138】 圖10A及圖10B中值得注意的實例行包括：

- 行1：將在EL1及EL0的錯誤例外取至預設例外等級EL1，且由於係在比預設例外等級EL1更高特權的例外等級中，在EL2及EL3被隱含地遮罩。陷阱控制尚未設定。
- 行2：將PSTATE.A設定成1導致錯誤例外在EL1及EL0被遮罩。
- 行3：將NMEA EL1設定成1防止遮罩錯誤例外，即使PSTATE.A係1。
- 行4及5：將TMEA EL2設定成1，若PSTATE.A係1，導致錯誤例外從EL1或EL0捕捉至EL2，但若PSTATE.A係0，行為仍如同行1。
- 行6：若TMEA EL2及NMEA EL1二者皆係1且PSTATE.A係1，若錯誤例外在EL0發生，由NMEA = 1導致的非可遮罩狀態置換至EL2的陷阱，但若錯誤例外在EL1發生，則否，使得當TMEA EL2、NMEA EL1、及PSTATE.A皆係1，但沒有其他捕捉控制經設定時，在EL0發生的錯誤例外在EL1採取，但在EL1發生的錯誤例外係在EL2採取。
- 行7：將NMEA EL2設定成1意謂著即使錯誤例外在EL2發生，仍將該錯誤例外取至EL2，而非如行1至6中隱含地遮罩。

- 行13至19及63至68：當EA EL3、TGE EL2、及AMO EL2控制的任一者經設定時，由於在此等情形中存在錯誤例外至EL2或EL3的明確路由，TMEA控制不具有影響。
- 行41：若TMEA EL3及TMEA EL2二者皆經設定且PSTATE.A經設定，將來自EL0及EL1之基於TMEA的陷阱設立至EL2，亦即，在特權上最接近目前例外等級的TMEA關聯例外等級，且將來自EL2之基於TMEA的陷阱設立至EL3。行42顯示若NMEA EL1係1，則此如何置換來自EL0的基於TMEA的陷阱，使得例外替代地在EL1採取（類似於行6）。

【0139】　用於實施圖9A至圖9I之功能的實例虛擬碼顯示如下：

```
// AArch64.SyncExternalAbortTarget()
// ================================
// Returns the target Exception level for a Synchronous External Data
// or Instruction Abort
bits(2) AArch64.SyncExternalAbortTarget(FaultRecord fault)
    boolean route_to_el3;
    boolean route_to_el2;
    // The exception is explicitly routed to EL3
    if PSTATE.EL != EL3 then
        route_to_el3 = (HaveEL(EL3) && SCR_EL3.EA == '1');
    else
        route_to_el3 = FALSE;
    // The exception is explicitly routed to EL2
```

```
bit tea_bit = (if HaveRASExt() then HCR_EL2.TEA else '0');
if !route_to_el3 && EL2Enabled() && PSTATE.EL == EL1 then
    route_to_el2 = (tea_bit == '1' ||
                    fault.acctype == AccType_NV2REGISTER ||
                    IsSecondStage(fault));
elsif !route_to_el3 && EL2Enabled() && PSTATE.EL == EL0 then
    route_to_el2 = (!IsInHost() && (HCR_EL2.TGE == '1' ||
        tea_bit =='1' || IsSecondStage(fault)));
else
    route_to_el2 = FALSE;
boolean route_soft_masked_to_el3;
boolean route_soft_masked_to_el2;
if HaveDoubleFault2Ext() then
    // The soft masked exception is routed to EL2
    route_soft_masked_to_el2 = (EL2Enabled() && !route_to_el3 &&
                                (PSTATE.EL == EL1 && PSTATE.A == '1')
                && (IsHCRXEL2Enabled() && HCRX_EL2.TMEA == '1'));
    // The soft masked exception is routed to EL3
    route_soft_masked_to_el3 =
                                (HaveEL(EL3) &&
                !(route_to_el2 || route_soft_masked_to_el2) &&
                (PSTATE.EL IN {EL2, EL1} && PSTATE.A == '1') &&
                                SCR_EL3.TMEA == '1');
else
```

```
        route_soft_masked_to_el2 = FALSE;
        route_soft_masked_to_el3 = FALSE;
    // The exception is routed away from the current exception regime
    routed_away = (route_to_el2 || route_soft_masked_to_el2 ||
                   route_to_el3 || route_soft_masked_to_el3);
    // The exception is taken at EL3
    take_in_el3 = PSTATE.EL == EL3;
    // The exception is taken at EL2 or in the Host EL0
take_in_el2_0 = ((PSTATE.EL == EL2 || IsInHost()) && !routed_away);
    // The exception is taken at EL1 or in the non-Host EL0
take_in_el1_0 = ((PSTATE.EL == EL1 ||
                  (PSTATE.EL == EL0 && !IsInHost()))
                  && !routed_away);
    bits(2) target_el;
    if take_in_el3 || route_to_el3 || route_soft_masked_to_el3 then
        target_el = EL3;
    elsif take_in_el2_0 || route_to_el2 || route_soft_masked_to_el2
      then
        target_el = EL2;
    elsif take_in_el1_0 then
        target_el = EL1;
    else
        Unreachable();
    return target_el;
```

【0140】 圖11顯示指示對於控制資訊之值的各組合，哪個例外等級係用於在例外等級EL0、EL1、EL2、EL3的任一者中發生的（非可遮罩）同步外部中止例外的目標例外等級的彙總表。行TEA EL2係指HCR_EL2.TEA的值，且其他行具有與圖10A及圖10B相同的意義。應注意行2及3，其顯示當TMEA EL2係1時，當PSTATE.A係1時，將陷阱從EL1（但不從EL0）設立至EL2，但當PSTATE.A係0時，則將例外從EL1及EL0二者取至EL1。

【0141】 本文所描述之概念可體現於用於製造體現所描述之概念的設備的電腦可讀碼中。例如，電腦可讀碼可在半導體設計及製造程序之一或多個階段中使用，該半導體設計及製造程序包括電子設計自動化(electronic design automation, EDA)階段，以製造一積體電路，其包含實現概念之設備。上述電腦可讀碼可另外或替代地促成實現本文所述之概念之設備的定義、模型化、模擬、驗證及/或測試。

【0142】 例如，用於製造實現本文所述之概念的設備之電腦可讀碼可以定義代表該等概念之硬體描述語言(HDL)的碼實施。例如，碼可定義用於定義實現概念的設備之一或多個邏輯電路的暫存器轉移層(register-transfer-level, RTL)抽象概念。碼可以Verilog、SystemVerilog、Chisel、或VHDL（超高速積體電路硬體描述語言）以及諸如FIRRTL的中間表示定義實現設備的一或多個邏輯電路的HDL表示。電腦可讀碼可使用系統級模型化語言提供實現概念之定義，諸如系統C及系統Verilog或可藉由電腦解譯以促成概念的模擬、功能及/或正式驗證及測試之概念的其他行為表示。

【0143】 另外或替代地，電腦可讀碼可定義實現本文所描述之概念的積體電路組件的低階描述，諸如一或多個接線對照表或積體電路佈局定義，包括

諸如GDSII之表示。積體電路組件之一或多個接線對照表或其他電腦可讀表示可藉由施加一或多個邏輯合成程序至RTL表示以產生用於製造實現本發明之設備的定義來產生。替代地或額外地，一或多個邏輯合成程序可從電腦可讀碼產生一位元流，該位元流被載入至一場可程式化閘陣列(FPGA)中以組態FPGA以實現所描述之概念。FPGA可部署用於積體電路中之製造之前的驗證及測試概念的目的，或FPGA可直接部署於產品中。

【0144】　電腦可讀碼可包含用於製造設備之碼表示之混合，例如包括RTL表示、接線對照表表示、或用於半導體設計及製造程序以製造實現本發明之設備的另一電腦可讀定義之一或多者之混合。替代地或額外地，概念可定義在半導體設計及製造程序中使用以製造設備之電腦可讀定義與一旦製造由所定義設備執行的電腦可讀碼定義指令的組合。

【0145】　此類電腦可讀碼可設置於任何已知暫時性電腦可讀媒體（諸如，網路上之有線或無線傳輸碼）或非暫時性電腦可讀媒體（諸如，半導體、磁碟或光碟）中。使用電腦可讀碼製造的積體電路可包含組件，諸如中央處理單元、圖形處理單元、神經處理單元、數位信號處理器或單獨或共同實現概念的其他組件之一或多者。

【0146】　圖12繪示可使用的模擬器實施方案。雖然稍早所述之實施例以用於操作支援所關注技術的特定處理硬體之設備及方法來實施本發明，但亦可能根據本文所述之實施例提供一指令執行環境，其係透過使用電腦程式實施。此類電腦程式常稱為模擬器，因為其等提供硬體架構之基於軟體的實施方案。模擬器電腦程式的種類包括仿真器、虛擬機、模型、及二進制轉譯器（包括動態二進制轉譯器）。一般而言，模擬器實施方案可在可選地運行主機作業系統

320、支援模擬器程式310的主機處理器330上運行。在一些配置中，在硬體與所提供的指令執行環境及/或相同的主機處理器上提供的多個相異指令執行環境之間可有多層模擬。歷史上，已需要強大的處理器來提供模擬器實施方案，其以合理速度執行，但此種方法在某些情況下可係有正當理由的，諸如當因為相容性或再使用原因此需要執行另一處理器原生的程式碼時。例如，模擬器實施方案可提供具有不為主機處理器硬體所支援之額外功能性的指令執行環境，或提供一般與不同的硬體架構相關聯的指令執行環境。模擬的綜述係於「Some Efficient Architecture Simulation Techniques」中給出，Robert Bedichek, Winter 1990 USENIX Conference，頁數53至63。

【0147】 在先前已參照特定硬體架構或特徵來描述實施例之情況下，在一模擬實施例中，可藉由合適的軟體架構或特徵提供等效功能。例如，可在模擬實施例中將特定電路系統實施為電腦程式邏輯。類似地，記憶體硬體（諸如暫存器或快取）可在模擬實施例中實施為軟體資料結構。在先前描述的實施例中提及的硬體元件的一或多者存在於主機硬體（例如，主機處理器330）上的配置中，一些模擬實施例可（在適當處）利用主機硬體。

【0148】 模擬器程式310可儲存在電腦可讀儲存媒體（其可係非暫時性媒體）上，並提供程式介面（指令執行環境）給目標碼300（其可包括應用程式、作業系統、及超管理器），該程式介面與藉由模擬器程式310模型化之硬體架構的應用程式介面相同。因此，目標碼300的程式指令可使用模擬器程式310從指令執行環境內執行，使得實際上不具有上文討論之設備2之硬體特徵的主機電腦330可仿真此等特徵。因此，模擬器程式310模擬目標處理電路系統的存在。

【0149】 模擬器碼310包括在功能上對應於上文討論的指令解碼器10、處理電路系統4、及例外控制電路系統40的指令解碼程式邏輯312、處理程式邏輯314、及例外控制程式邏輯318。再者，暫存器仿真資料結構316可由模擬器碼310維持以仿真提供在由模擬器碼310模擬之目標處理器的指令集架構中的暫存器14。例如，暫存器仿真資料結構316可儲存在主機資料處理設備的記憶體中，且可包括儲存對應於暫存器14之資料的儲存位置，該等暫存器包括如圖3所示之用於儲存控制資訊的暫存器。例外控制程式邏輯318可以由上文描述之例外控制電路系統40對例外之回應的對應方式回應藉由模擬器碼310表示之標定目標處理器的經模擬例外，包括基於如上文討論的捕捉經遮罩例外控制資訊及其他控制資訊控制是否遮罩例外及控制捕捉（在此模擬中，例外路由控制資訊可維持為暫存器仿真資料結構316的部分）。因此，模擬器碼310可提供與在稍早實例中描述之處理設備2的實體硬體實施方案上達成的優點類似的優點。

【0150】 在本申請案中，用語「經組態以...(configured to...)」係用以意指一設備的一元件具有能夠實行該經定義作業的一組態。在此上下文中，「組態(configuration)」意指硬體或軟體之互連的配置或方式。例如，該設備可具有專用硬體，其提供經定義的作業，或者一處理器或其他處理裝置可經程式化以執行該功能。「經組態以(configured to)」並不意味著設備元件需要以任何方式改變以提供所定義的作業。

【0151】 在本申請案中，以片語「之至少一者(at least one of)」前綴的特徵列表意謂著此等特徵的任何一或多者可個別地或組合地提供。例如，「[A]、[B]、及[C]之至少一者」涵蓋下列選項的任一者：單獨A（不具有B或C）、單

獨B（不具有A或C）、單獨C（不具有A或B）、A及B的組合（不具有C）、A及C的組合（不具有B）、B及C的組合（不具有A）、或A、B、及C的組合。

【0152】 雖然本文已參照附圖詳細地描述本發明的說明性實施例，應瞭解本發明不限於該等精確實施例，且所屬技術領域中具有通常知識者可於其中實行各種變化與修改，而不脫離如隨附申請專利範圍所定義的本發明的範圍。

【符號說明】

【0153】

2:資料處理設備；設備；資料處理系統

4:處理管線；管線；處理電路系統

6:提取級

8:指令快取記憶體；1級指令快取記憶體

10:解碼級；指令解碼器

12:發佈級

14:暫存器

16:執行級

18:執行單元；算術/邏輯單元(ALU)

20:執行單元；浮點單元

22:執行單元；分支單元

24:執行單元；載入/儲存單元

26:寫回級

28:記憶體管理單元；MMU

30:轉譯後備緩衝區(TLB)

32:1級資料快取記憶體

34:共用2級快取記憶體

36:主記憶體

40:例外控制電路系統

42:例外信號

100:步驟

102:步驟

104:步驟

106:步驟

107:步驟

108:步驟

120:步驟

122:步驟

124:步驟

126:步驟

128:步驟

130:步驟

150:步驟

152:步驟

154:步驟

156:步驟

160:步驟

162:步驟

164:步驟

166:步驟

168:步驟

170:步驟

300:目標碼

310:模擬器程式；模擬器碼

312:指令解碼程式邏輯

314:處理程式邏輯

316:暫存器仿真資料結構

318:例外控制程式邏輯

320:主機作業系統

330:主機處理器；主機電腦

1000:步驟

1002:步驟

1004:步驟

1006:步驟

1008:步驟

1010:步驟

1012:步驟

1014:步驟

1016:步驟

1018:步驟

1020:步驟

1022:步驟

1024:步驟

1026:步驟

1028:步驟

1030:步驟

1032:步驟

1034:步驟

1036:步驟

1038:步驟

1040:步驟

1042:步驟

1044:步驟

1046:步驟

1048:步驟

1050:步驟

1052:步驟

1054:步驟

1056:步驟

1058:步驟

1060:步驟

1062:步驟

1064:步驟

1066:步驟

1068:步驟

1070:步驟

1072:步驟

1076:步驟

1078:步驟

1080:步驟

1082:步驟

1084:步驟

1086:步驟

1098:步驟

1100:步驟

1102:步驟

1104:步驟

1108:步驟

1110:步驟

1112:步驟

1114:步驟

1116:步驟

1118:步驟

1120:步驟

1122:步驟

1124:步驟

1126:步驟

1128:步驟

1130:步驟

1134:步驟

1136:步驟

1138:步驟

1140:步驟

1142:步驟

1144:步驟

1146:步驟

1148:步驟

1150:步驟

1152:步驟

1154:步驟

1156:步驟

1158:步驟

1160:步驟

1162:步驟

1164:步驟

1166:步驟

1168:步驟

1170:步驟

1172:步驟

1174:步驟

1176:步驟

1178:步驟

1180:步驟

1182:步驟

1184:步驟

1186:步驟

1188:步驟

1190:步驟

1192:步驟

1194:步驟

1196:步驟

1198:步驟

1200:步驟

1202:步驟

1204:步驟

1206:步驟

1208:步驟

1210:步驟

1214:步驟

1216:步驟

1218:步驟

1220:步驟

1224:步驟

1228:步驟

1232:步驟

1234:步驟

1236:步驟

1238:步驟

1240:步驟

1254:步驟

1256:步驟

1258:步驟

1260:步驟

1262:步驟

1264:步驟

1266:步驟

1268:步驟

1270:步驟

1272:步驟

1274:步驟

1276:步驟

1278:步驟

1280:步驟

1282:步驟

1284:步驟

1286:步驟

1288:步驟

1292:步驟

1294:步驟

1296:步驟

1298:步驟

1300:步驟

1302:步驟

1304:步驟

1306:步驟

1308:步驟

1310:步驟

1312:步驟

1314:步驟

1316:步驟

1318:步驟

1320:步驟

1322:步驟

1324:步驟

1326:步驟

1328:步驟

1330:步驟

1332:步驟

1334:步驟

1336:步驟

1338:步驟

1340:步驟

1342:步驟

1344:步驟

1346:步驟

1348:步驟

1350:步驟

1354:步驟

EL0:例外等級

EL1:例外等級

EL2:例外等級

EL3:例外等級

【發明申請專利範圍】

【請求項1】 一種設備，其包含：

處理電路系統，其在由該處理電路系統支援的複數個例外等級的一者中執行資料處理；及

例外控制電路系統，其取決於儲存在至少一個暫存器中的控制資訊控制由該處理電路系統採取例外，該控制資訊包括至少：

遮罩控制資訊，其可設定成一遮罩狀態及一未遮罩狀態的一者；及

捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷狀態及一設陷狀態的一者；

其中：

回應於基於該遮罩控制資訊係在該遮罩狀態中而可遮罩之一可遮罩例外類別的一給定例外，該例外控制電路系統經組態以：

判定該遮罩控制資訊是否在該遮罩狀態中；

在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及

在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【請求項2】 如請求項1之設備，其中：

該控制資訊包含捕捉經遮罩例外控制資訊之複數個項，其各與複數個預定捕捉目標例外等級的各別一者關聯，且

該例外控制電路系統經組態以在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生、且該目前例外等級的特權比該複數個預定捕捉目標例外等級的一給定一者更低的至少一個情境中，取決於捕捉經遮罩例外控制資訊之與該複數個預定捕捉目標例外等級的該給定一者關聯的一對應項是否在該設陷狀態中而控制是否將該給定例外捕捉至該複數個預定捕捉目標例外等級的該給定一者。

【請求項3】 如請求項2之設備，其中在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生、捕捉經遮罩例外控制資訊的至少二個項係在該設陷狀態中、且捕捉經遮罩例外控制資訊的該至少二個項之各者滿足捕捉至該預定捕捉目標例外等級的一或多個條件的至少一個情境中：

該例外控制電路系統經組態以將該給定例外捕捉至與捕捉經遮罩例外控制資訊的一經選擇項關聯的該預定捕捉目標例外等級；及

捕捉經遮罩例外控制資訊的該經選擇項包含捕捉經遮罩例外控制資訊的該至少二個項之在該目前例外等級與該關聯預定捕捉目標例外等級之間在特權上有一最小差的一者。

【請求項4】 如前述請求項中任一項之設備，其中在該捕捉經遮罩例外控制資訊係在該未設陷狀態中的一情形中，該例外控制電路系統經組態以至少取決於該遮罩控制資訊是否在該遮罩狀態中而判定是否遮罩該給定例外。

【請求項5】 如請求項4之設備，其中該例外控制電路系統亦經組態以取決於該目前例外等級的特權是否比若該例外等級係在該捕捉經遮罩例外控制資訊

係在該未設陷狀態中時採取，將採取該給定例外的一例外等級更高而判定是否遮罩該給定例外。

【請求項6】 如請求項4及5中任一項之設備，其中：

該控制資訊包含可設定成一非可遮罩狀態及一可遮罩狀態的一者以指示該可遮罩類別的例外是否可遮罩的非可遮罩例外控制資訊；及

在該非可遮罩例外控制資訊係在該非可遮罩狀態中的至少一個情境中，即使在該遮罩控制資訊經設定成該遮罩狀態時，該例外控制電路系統經組態以判定該給定例外將不被遮罩。

【請求項7】 如請求項6之設備，其中在該給定例外在該目前例外等級係一最低特權例外等級時發生、該非可遮罩例外控制資訊係在該非可遮罩狀態中、及該給定例外將採取至經啟用以在該給定例外未遮罩且該捕捉經遮罩例外控制資訊係在該未設陷狀態中的一情形中用於採取該給定例外的一次一最低特權例外等級的一情境中，該例外控制電路系統經組態以：

即使在該捕捉經遮罩例外控制資訊係在該設陷狀態中時，控制該給定例外在該次一最低特權例外等級中採取而不捕捉至該預定捕捉目標例外等級。

【請求項8】 如請求項7之設備，其中該控制資訊包含非可遮罩例外控制資訊之與不同例外等級關聯的複數個項；及

該例外控制電路系統經組態以取決於非可遮罩例外控制資訊之該複數個項之基於該目前例外等級選擇的一者判定是否遮罩該給定例外。

【請求項9】 如前述請求項中任一項之設備，其中該控制資訊包含捕捉控制資訊之可由軟體設定成一設陷狀態及一未設陷狀態的一者的至少一個其他項；

該例外控制電路系統經組態以取決於捕捉控制資訊的該其他項是否在該設陷狀態中，而控制是否將至少該可遮罩類別的例外捕捉至一特定例外等級；及

捕捉控制資訊的該至少一個其他項優先於該捕捉經遮罩例外控制資訊，使得在捕捉控制資訊的該其他項係在該設陷狀態中的一情形中，該例外控制電路系統經組態以基於捕捉控制資訊的該其他項將該例外捕捉至該特定例外等級，而不論該捕捉經遮罩例外控制資訊是否在該設陷狀態中。

【請求項10】如前述請求項中任一項之設備，其中該可遮罩例外類別包含指示與正確服務的一偏離的非同步錯誤例外。

【請求項11】如前述請求項中任一項之設備，其中回應於不能夠基於將該遮罩控制資訊設定成該遮罩狀態而被遮罩之一非可遮罩例外類別的一非可遮罩例外，該例外控制電路系統經組態以：

在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生、一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的該預定捕捉目標例外等級、及該目前例外等級係能夠係用於處理例外的一目標例外等級的一例外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中，而控制是否將該非可遮罩例外捕捉至該預定捕捉目標例外等級。

【請求項12】如請求項11之設備，其中該非可遮罩例外類別包含指示與正確服務的一偏離的同步外部中止例外。

【請求項13】如前述請求項中任一項之設備，其包含複數組一或多個例外症候群暫存器，各組一或多個例外症候群暫存器儲存在該等例外等級的一各別者中採取之例外的例外症候群資訊。

【請求項14】如前述請求項中任一項之設備，其中回應於請求對在比與該捕捉經遮罩例外控制資訊關聯的該預定捕捉目標例外等級更低特權的一例外等級執行的該捕捉經遮罩例外控制資訊的一更新的一指令，該處理電路系統經組態以拒絕或捕捉對該捕捉經遮罩例外控制資訊的所請求之該更新。

【請求項15】一種用於製造前述請求項中任一項之設備的電腦可讀碼。

【請求項16】一種儲存媒體，其儲存如請求項15的電腦可讀碼。

【請求項17】一種方法，其包含：

在由該處理電路系統支援的複數個例外等級的一者中執行資料處理；及

取決於儲存在至少一個暫存器中的控制資訊而控制由該處理電路系統採取例外，該控制資訊包括至少：

遮罩控制資訊，其可設定成一遮罩狀態及一未遮罩狀態的一者；及

捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷狀態及一設陷狀態的一者；

其中：

回應於基於將該遮罩控制資訊設定成該遮罩狀態而可遮罩之一可遮罩例外類別的一給定例外，該方法包含：

判定該遮罩控制資訊是否在該遮罩狀態中；

在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例

外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中，而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及

在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【請求項18】一種包含指令的電腦程式，當該等指令由一主機資料處理設備執行時，控制該主機資料處理設備提供用於執行目標碼的一指令執行環境，該電腦程式包含：

處理程式邏輯，其用以模擬該目標碼的指令在複數個例外等級的一者中的執行；及

例外控制程式邏輯，其用以取決於儲存在該主機資料處理設備之代表由該目標碼使用的一目標指令集架構的至少一個暫存器的至少一個儲存位置中的控制資訊而模擬採取例外，該控制資訊包括至少：

遮罩控制資訊，其可設定成一遮罩狀態及一未遮罩狀態的一者；及

捕捉經遮罩例外控制資訊，其可由軟體設定成一未設陷狀態及一設陷狀態的一者；

其中：

回應於基於將該遮罩控制資訊設定成該遮罩狀態而可遮罩之一可遮罩例外類別的一給定例外，該例外控制程式邏輯經組態以：

判定該遮罩控制資訊是否在該遮罩狀態中；

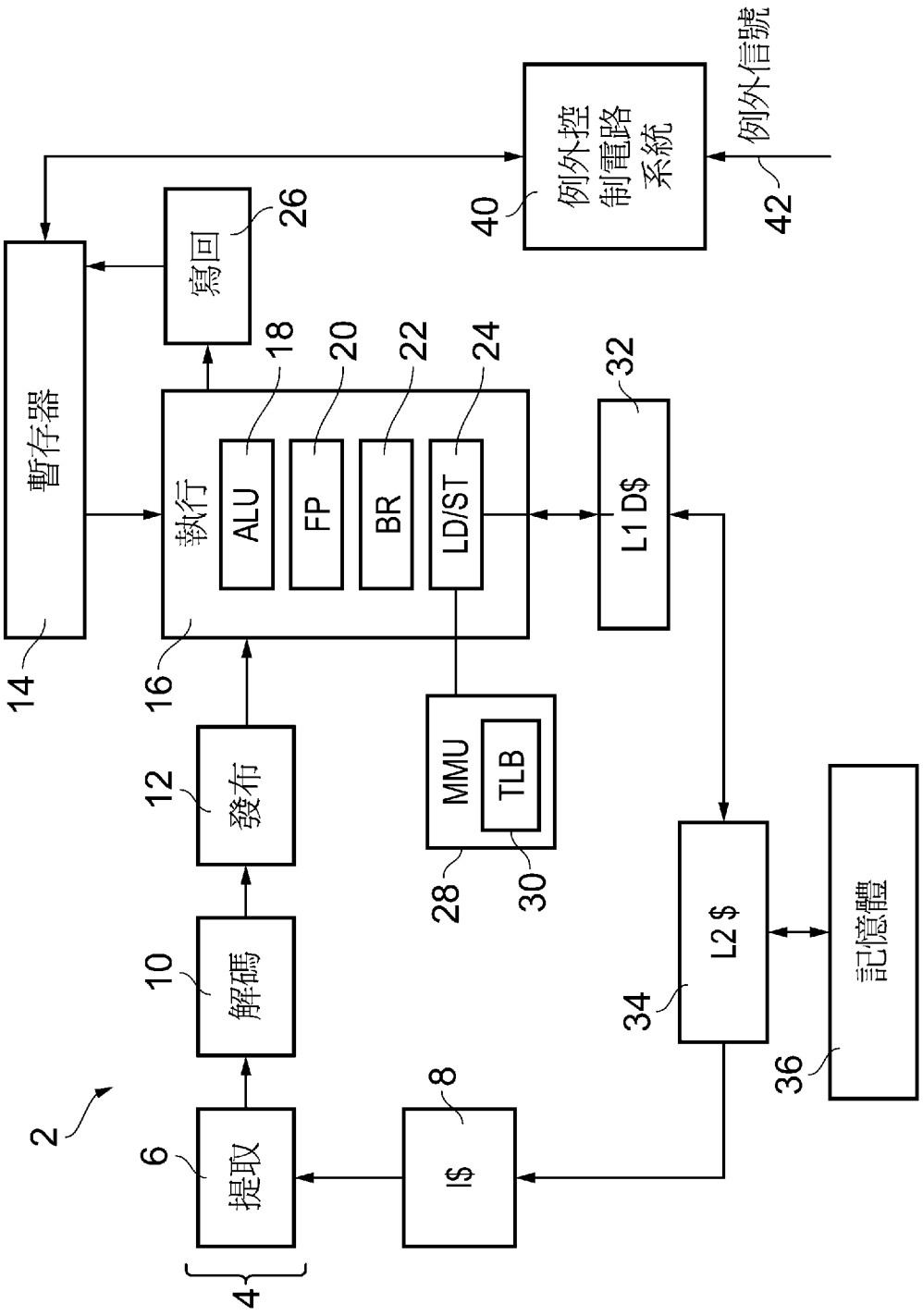
在該給定例外在該遮罩控制資訊係在該遮罩狀態中時發生且一目前例外等級的特權低於與該捕捉經遮罩例外控制資訊關聯的一預定捕捉目標例

外等級的至少一個情境中，取決於該捕捉經遮罩例外控制資訊是否在該設陷狀態中而控制是否將該給定例外捕捉至該預定捕捉目標例外等級；及

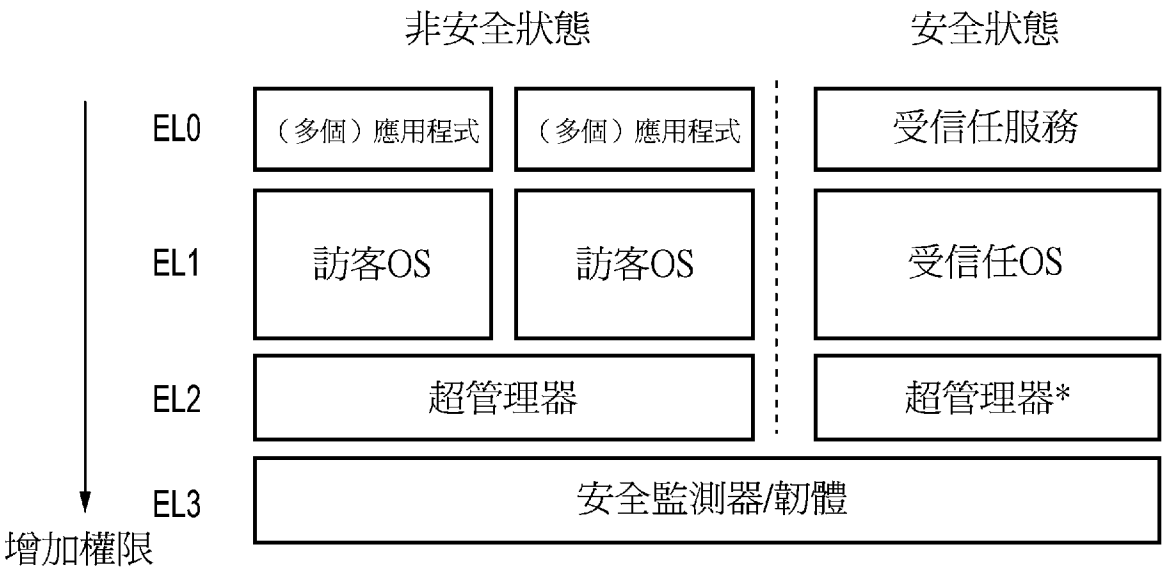
在該遮罩控制資訊係在該未遮罩狀態中且該給定例外經判定未經遮罩的一情形中，選擇用於處理該給定例外的一目標例外等級，該目標例外等級係與該捕捉經遮罩例外控制資訊無關地選擇。

【請求項19】一種儲存媒體，其儲存如請求項18的電腦程式。

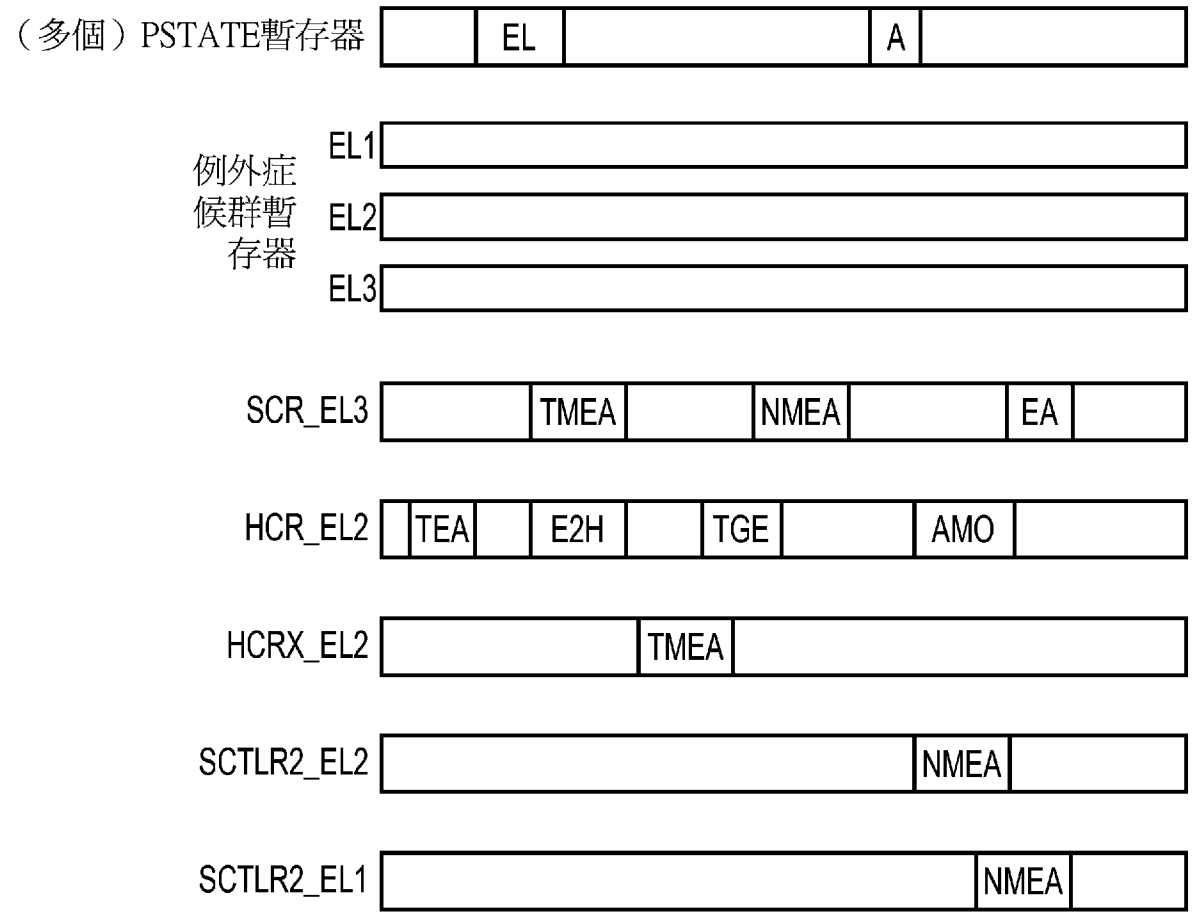
【發明圖式】


例外控制電路系統
例外信號
40
42
寫回
26
暫存器
14
執行
16
ALU
18
FP
20
BR
22
LD/ST
24
L1 D$
32
MMU
TLB
28
30
發布
12
解碼
10
提取
6
I$
8
4
2
L2 $
34
記憶體
36

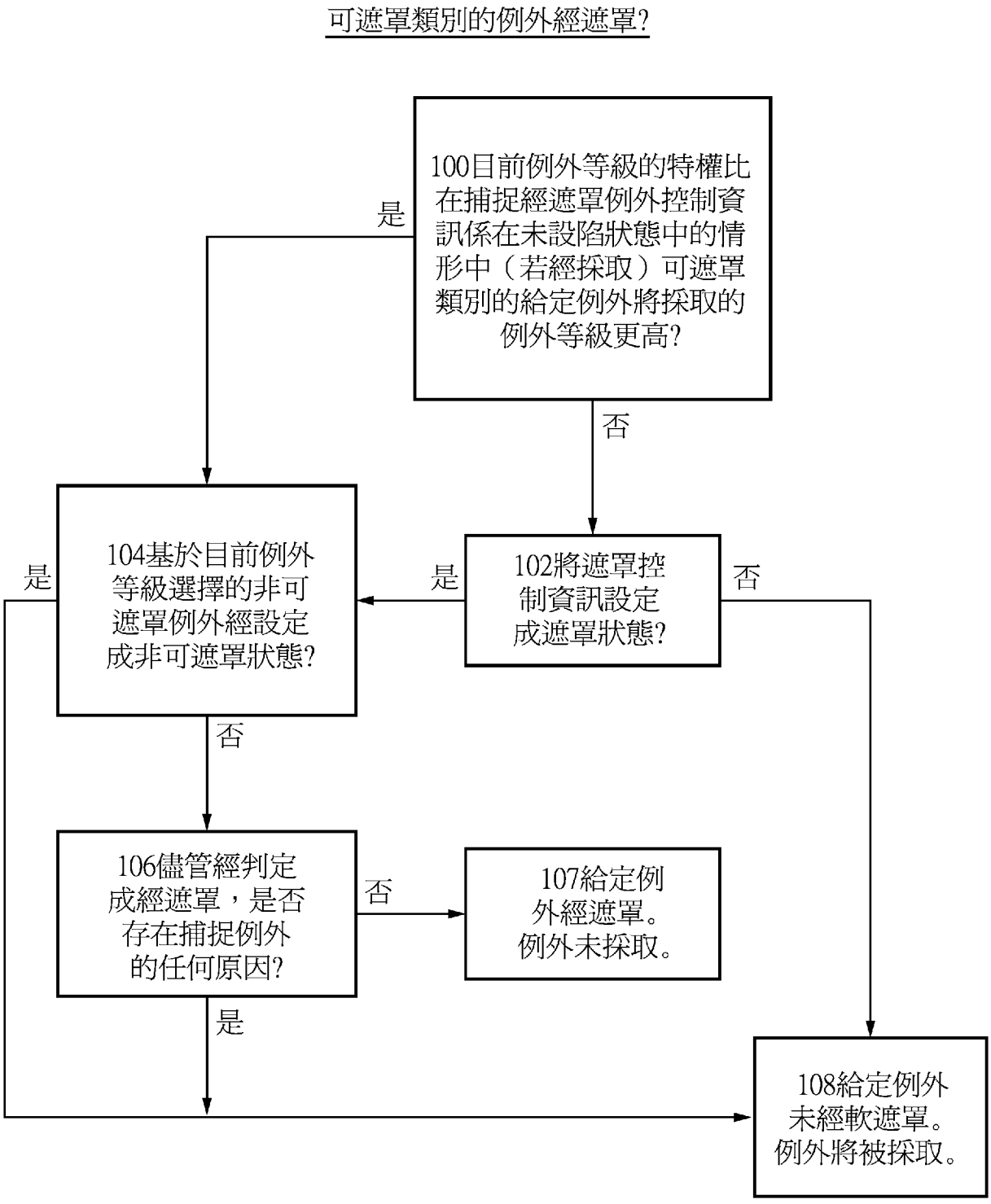
【圖1】


非安全狀態
安全狀態
EL0
（多個）應用程式
（多個）應用程式
受信任服務
EL1
訪客OS
訪客OS
受信任OS
EL2
超管理器
超管理器*
EL3
安全監測器/韌體
增加權限

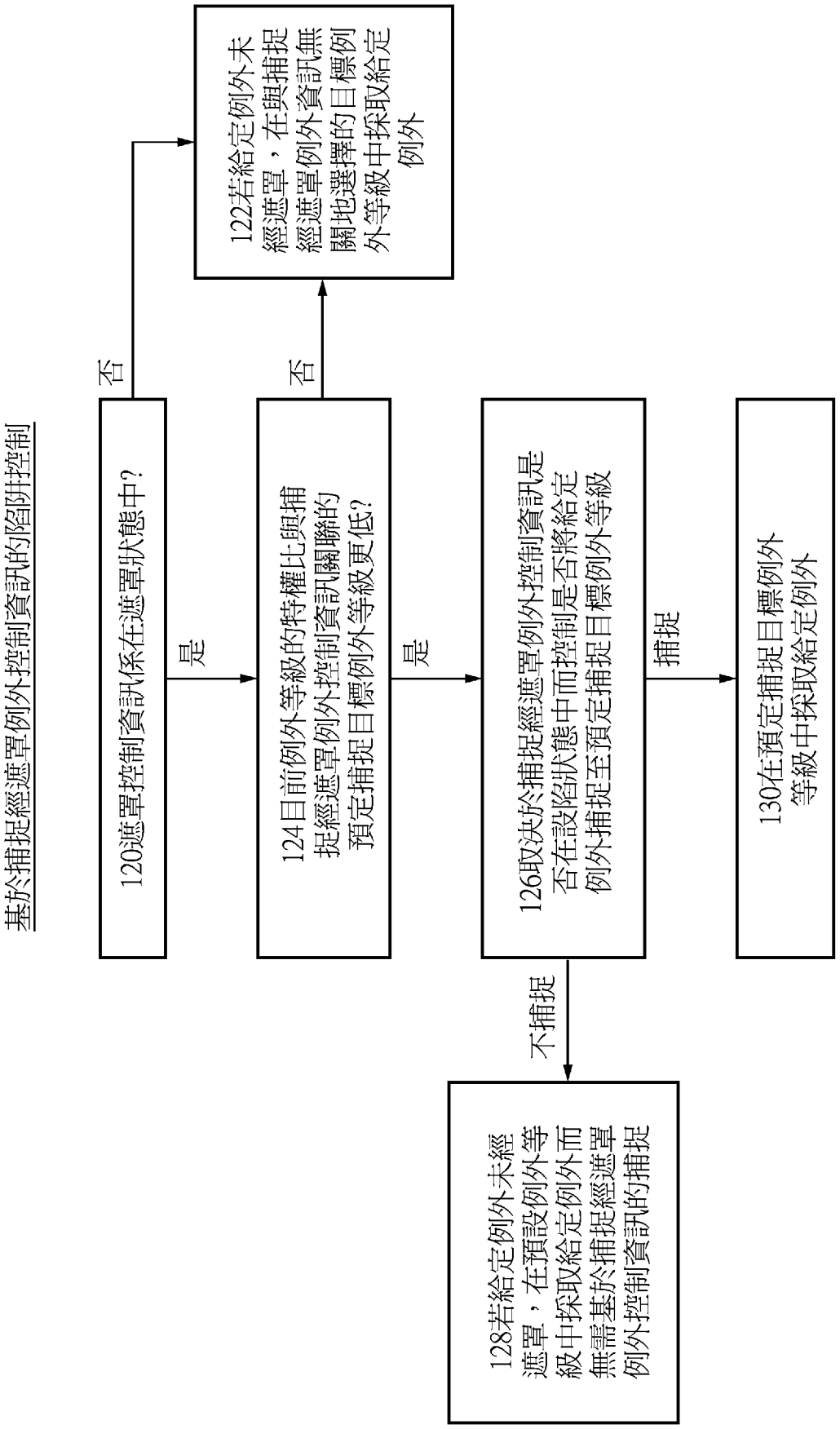
【圖2】


（多個）PSTATE暫存器
EL
A
例外症候群暫存器
EL1
EL2
EL3
SCR_EL3
TMEA
NMEA
EA
HCR_EL2
TEA
E2H
TGE
AMO
HCRX_EL2
TMEA
SCTLR2_EL2
NMEA
SCTLR2_EL1
NMEA

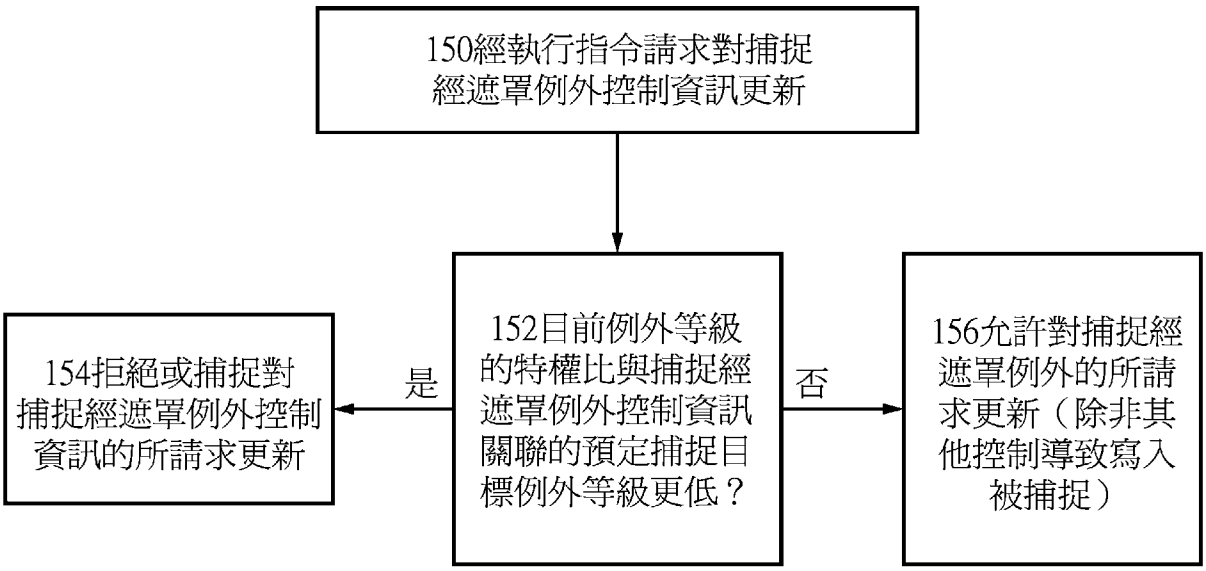
【圖3】


可遮罩類別的例外經遮罩?
100目前例外等級的特權比在捕捉經遮罩例外控制資訊係在未設陷狀態中的情形中（若經採取）可遮罩類別的給定例外將採取的例外等級更高?
是
否
102將遮罩控制資訊設定成遮罩狀態?
是
否
104基於目前例外等級選擇的非可遮罩例外經設定成非可遮罩狀態?
是
否
106儘管經判定成經遮罩，是否存在捕捉例外的任何原因?
否
107給定例外經遮罩。例外未採取。
是
108給定例外未經軟遮罩。例外將被採取。

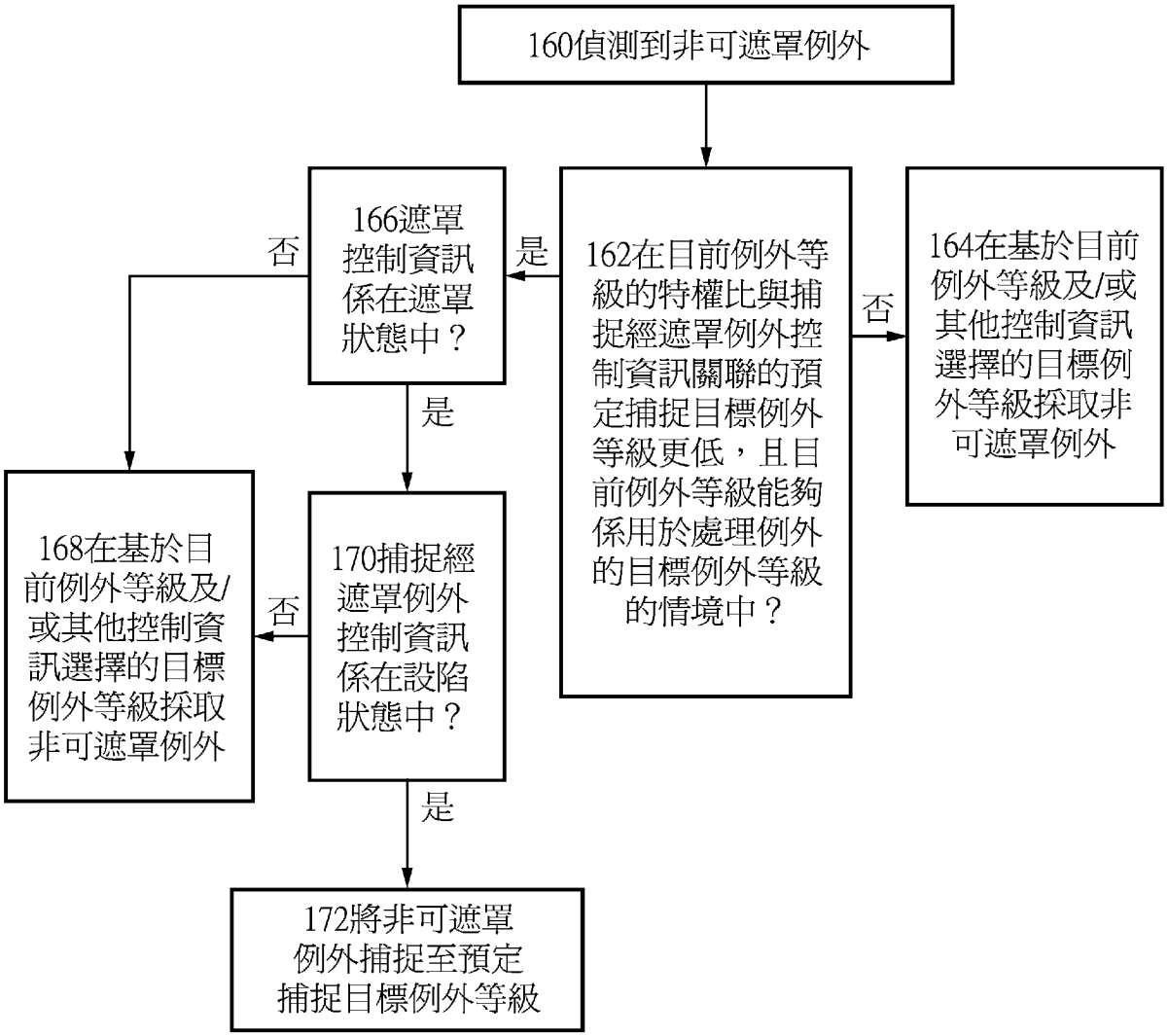
【圖4】


基於捕捉經遮罩例外控制資訊的陷阱控制
120遮罩控制資訊係在遮罩狀態中?
是
否
124目前例外等級的特權比與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低?
是
否
122若給定例外未經遮罩，在與捕捉經遮罩例外資訊無關地選擇的目標例外等級中採取給定例外
126取決於捕捉經遮罩例外控制資訊是否在設陷狀態中而控制是否將給定例外捕捉至預定捕捉目標例外等級
捕捉
不捕捉
130在預定捕捉目標例外等級中採取給定例外
128若給定例外未經遮罩，在預設例外等級中採取給定例外而無需基於捕捉經遮罩例外控制資訊的捕捉

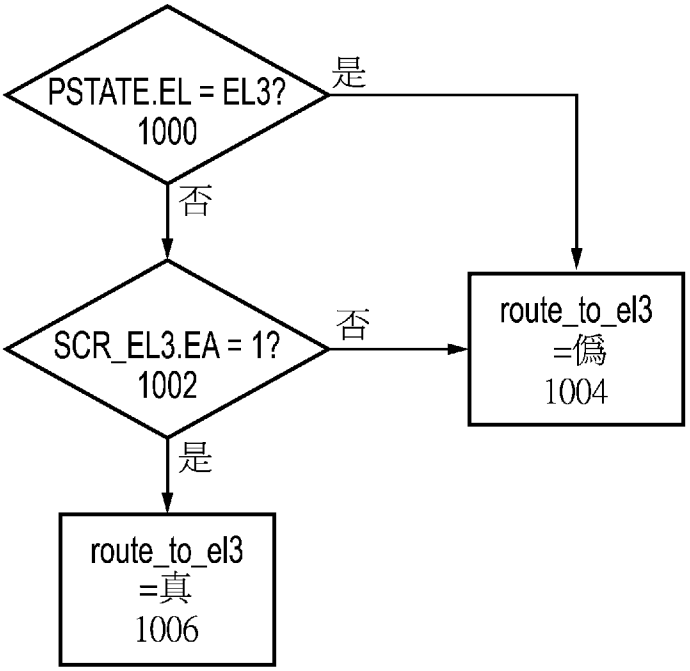
【圖5】


150經執行指令請求對捕捉經遮罩例外控制資訊更新
152目前例外等級的特權比與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低？
是
否
154拒絕或捕捉對捕捉經遮罩例外控制資訊的所請求更新
156允許對捕捉經遮罩例外的所請求更新（除非其他控制導致寫入被捕捉）

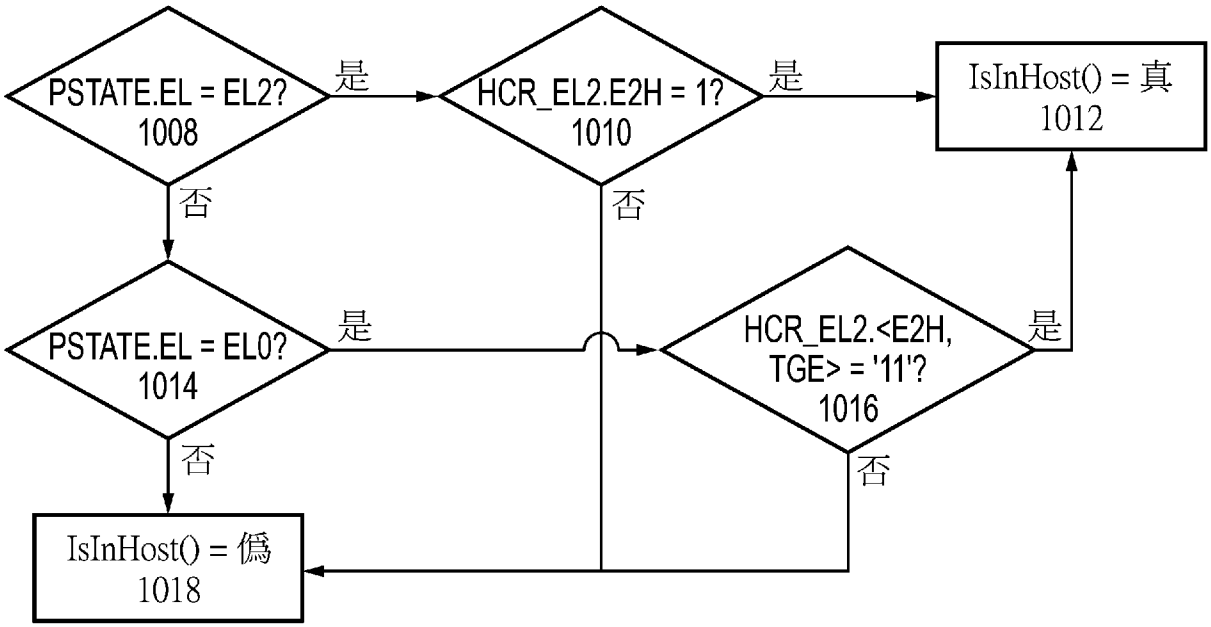
【圖6】


160偵測到非可遮罩例外
162在目前例外等級的特權比與捕捉經遮罩例外控制資訊關聯的預定捕捉目標例外等級更低，且目前例外等級能夠係用於處理例外的目標例外等級的情境中？
是
否
164在基於目前例外等級及/或其他控制資訊選擇的目標例外等級採取非可遮罩例外
166遮罩控制資訊係在遮罩狀態中？
否
是
168在基於目前例外等級及/或其他控制資訊選擇的目標例外等級採取非可遮罩例外
170捕捉經遮罩例外控制資訊係在設陷狀態中？
否
是
172將非可遮罩例外捕捉至預定捕捉目標例外等級

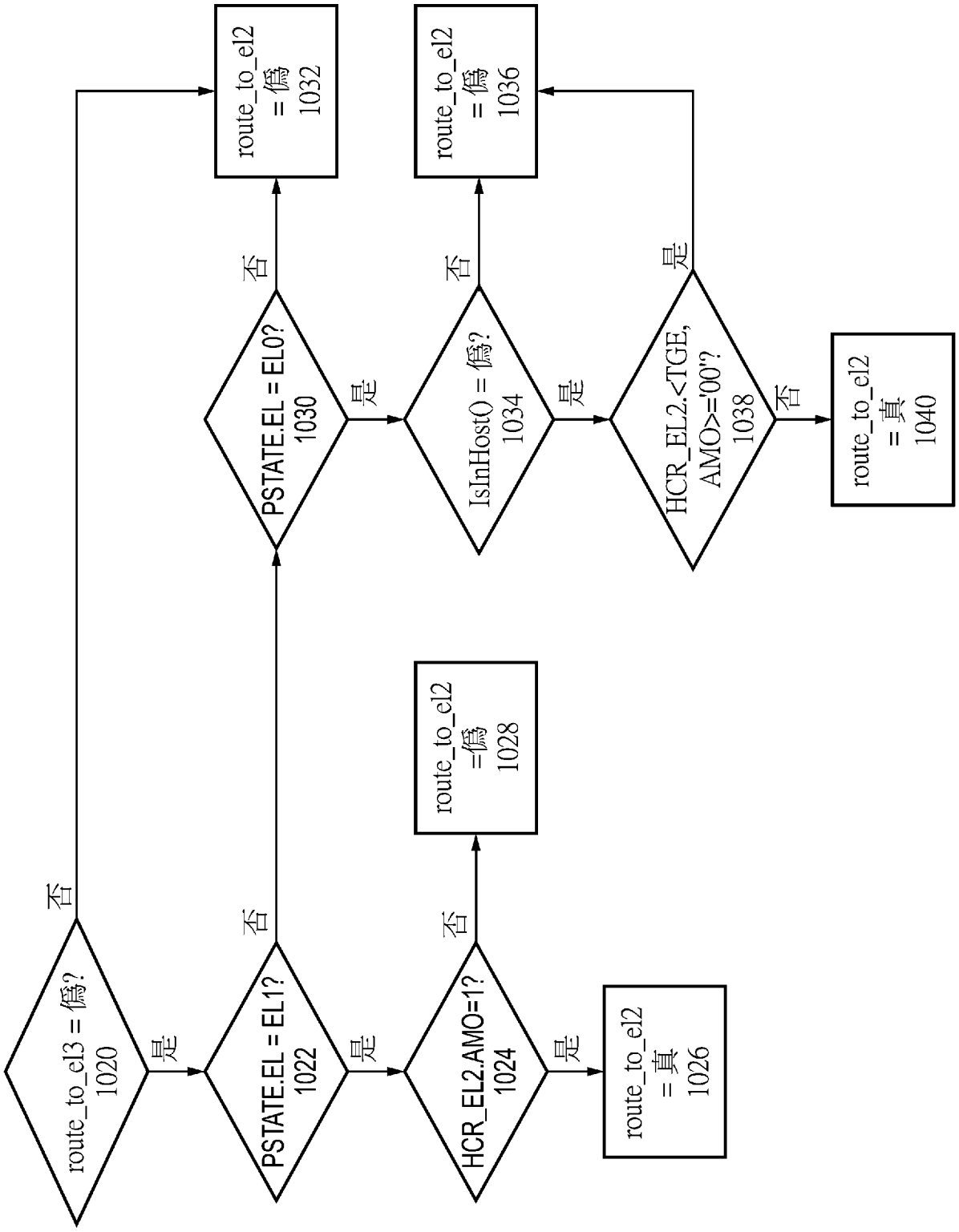
【圖7】


PSTATE.EL = EL3?
1000
是
否
SCR_EL3.EA = 1?
1002
否
是
route_to_el3
=僞
1004
route_to_el3
=真
1006

【圖8A】


PSTATE.EL = EL2?
1008
是
HCR_EL2.E2H = 1?
1010
是
IsInHost() = 真
1012
否
否
PSTATE.EL = EL0?
1014
是
HCR_EL2.<E2H,
TGE> = '11'?
1016
是
否
否
IsInHost() = 僞
1018

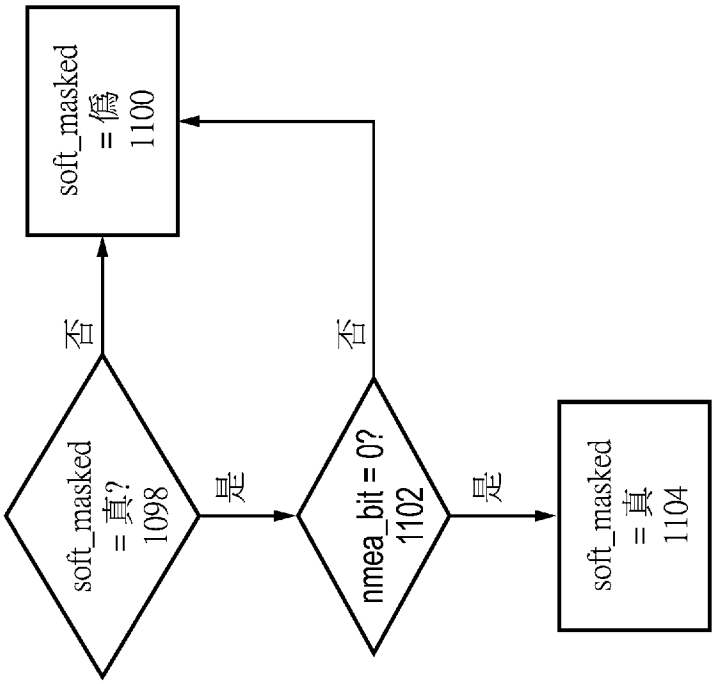
【圖8B】


route_to_el3 = 僞?
1020
否
是
PSTATE.EL = EL1?
1022
否
是
HCR_EL2.AMO=1?
1024
否
是
route_to_el2 = 真
1026
route_to_el2 = 僞
1028
PSTATE.EL = EL0?
1030
否
是
IsInHost() = 僞?
1034
否
是
HCR_EL2.<TGE, AMO>='00'?
1038
否
是
route_to_el2 = 真
1040
route_to_el2 = 僞
1032
route_to_el2 = 僞
1036

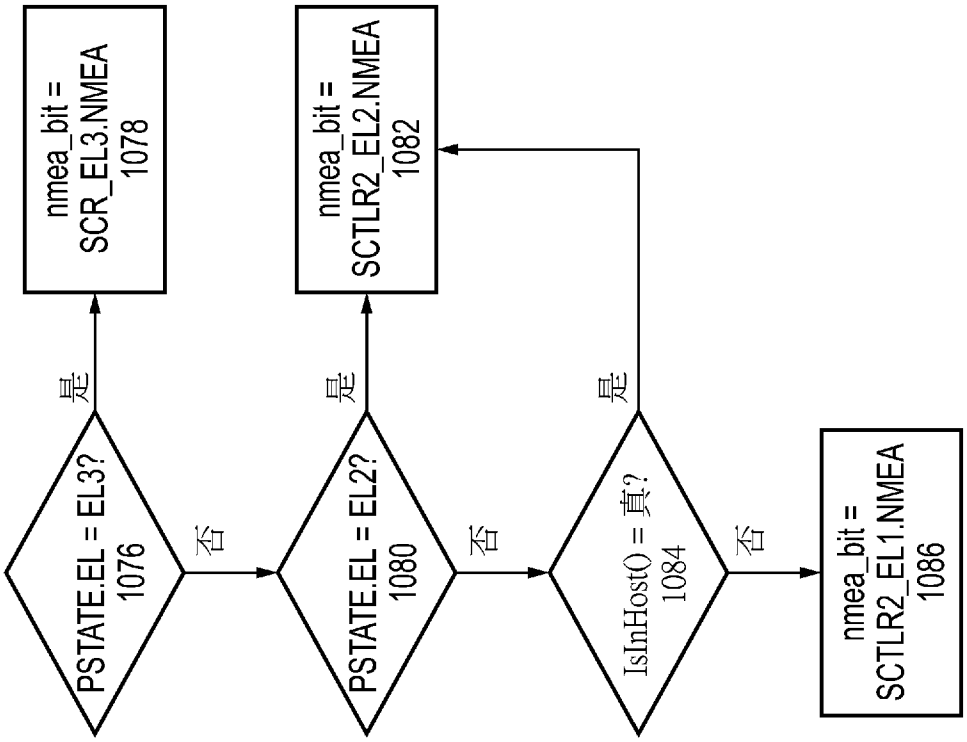
【圖8C】

soft_masked
soft_masked
soft_masked
soft_masked
soft_masked
soft_masked
soft_masked
soft_masked
soft_masked

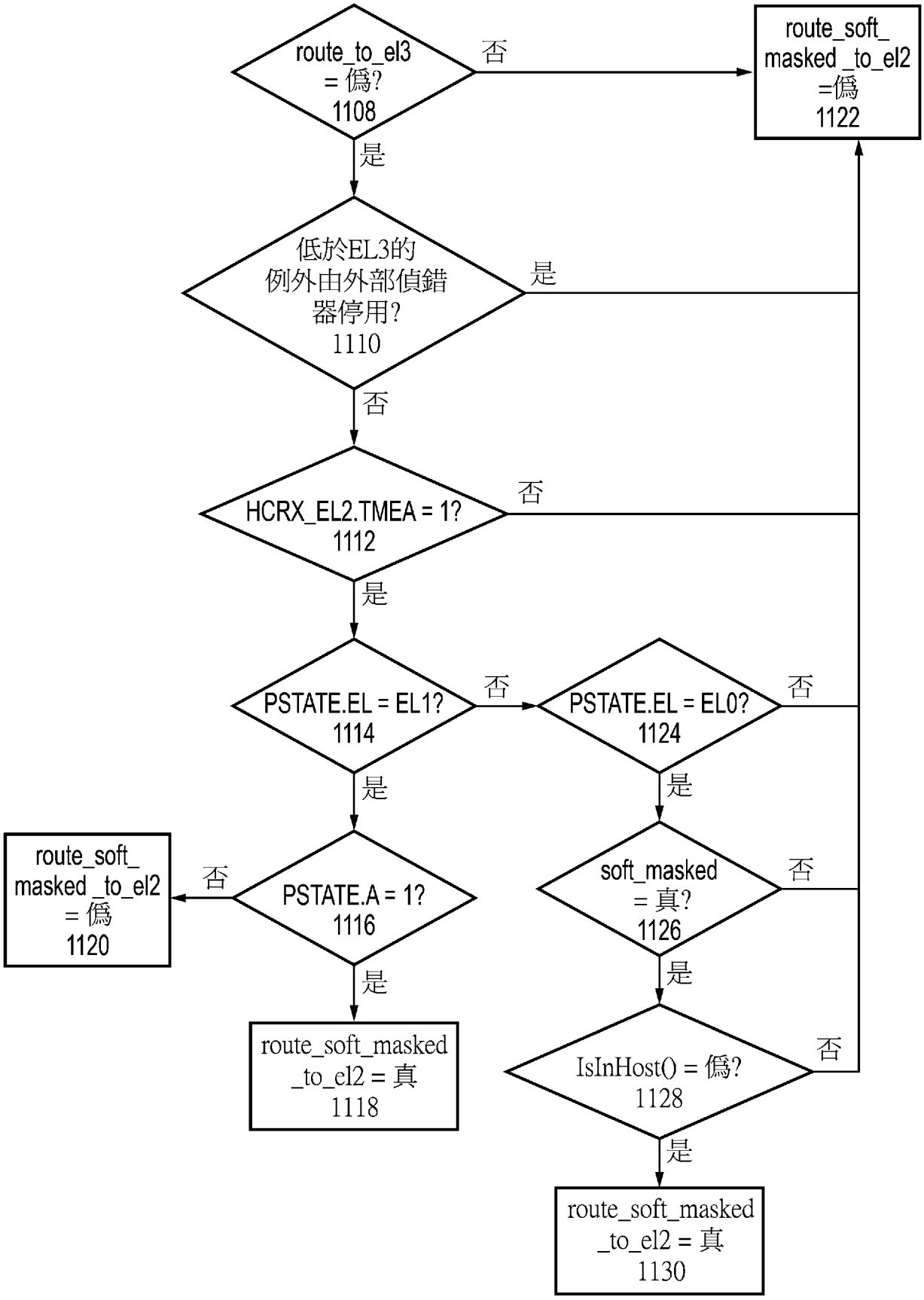
圖8D


soft_masked
= 偽
1100
否
否
soft_masked
= 真?
1098
是
nmea_bit = 0?
1102
是
soft_masked
= 真
1104

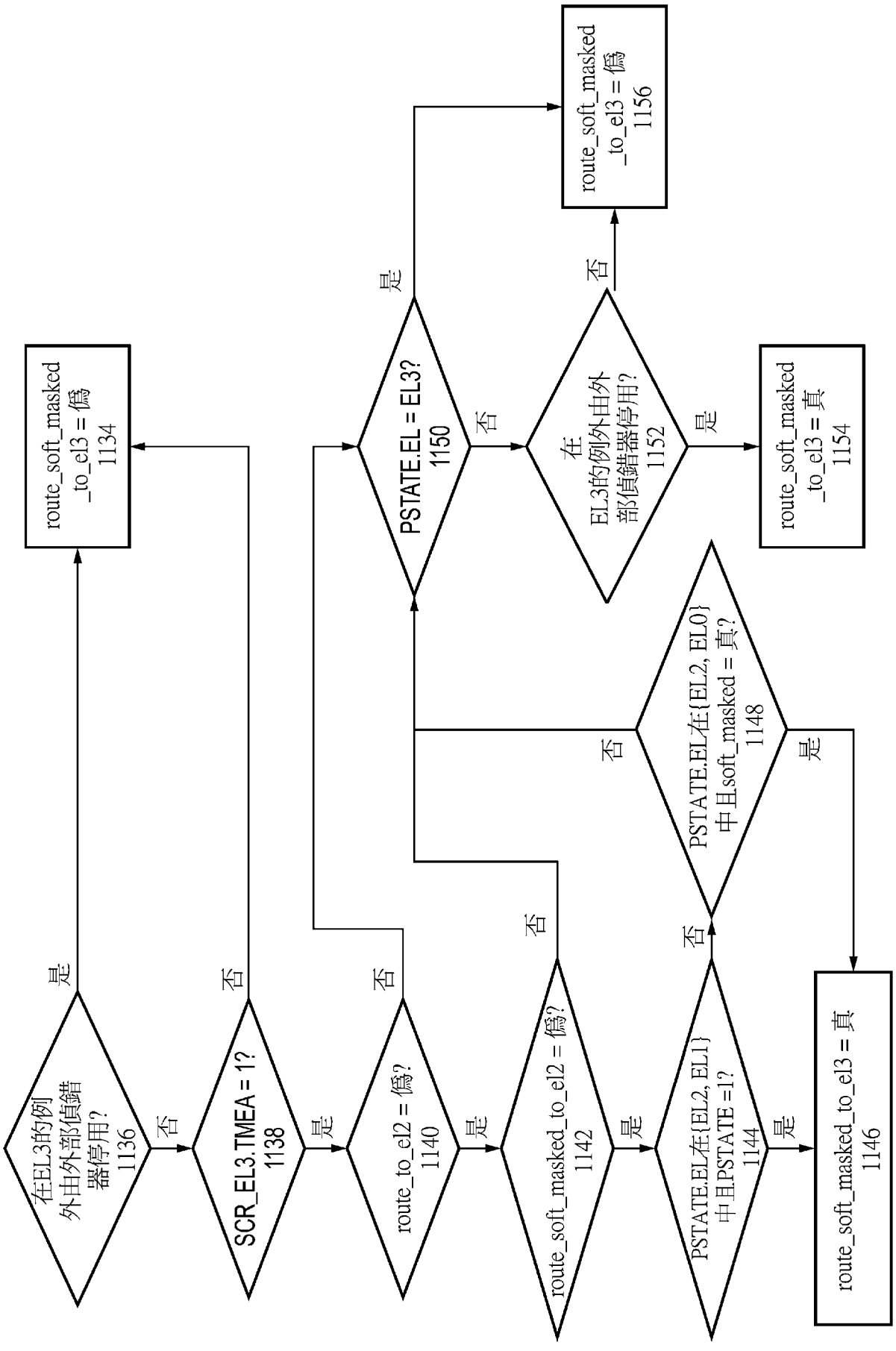
【圖8F】


PSTATE.EL = EL3?
1076
是
nmea_bit =
SCR_EL3.NMEA
1078
否
PSTATE.EL = EL2?
1080
是
nmea_bit =
SCTLR2_EL2.NMEA
1082
否
IsInHost() = 真?
1084
是
否
nmea_bit =
SCTLR2_EL1.NMEA
1086

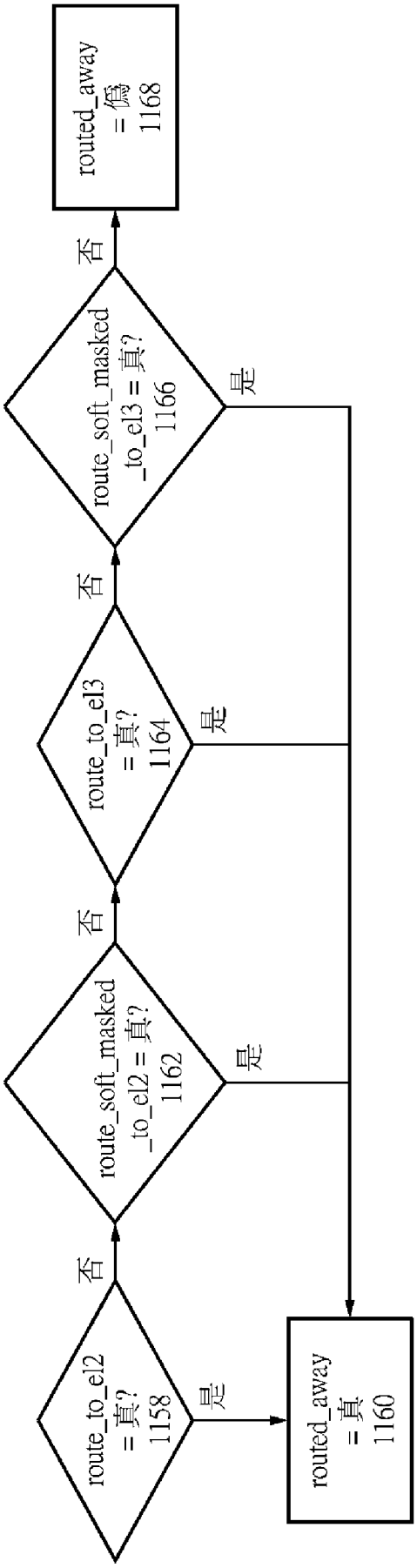
【圖8E】


route_to_el3
= 僞?
1108
否
route_soft_
masked _to_el2
=僞
1122
是
低於EL3的
例外由外部偵錯
器停用?
1110
是
否
HCRX_EL2.TMEA = 1?
1112
否
是
PSTATE.EL = EL1?
1114
否
PSTATE.EL = EL0?
1124
否
是
是
route_soft_
masked _to_el2
= 僞
1120
否
PSTATE.A = 1?
1116
soft_masked
= 真?
1126
否
是
是
route_soft_masked
_to_el2 = 真
1118
IsInHost() = 僞?
1128
否
是
route_soft_masked
_to_el2 = 真
1130

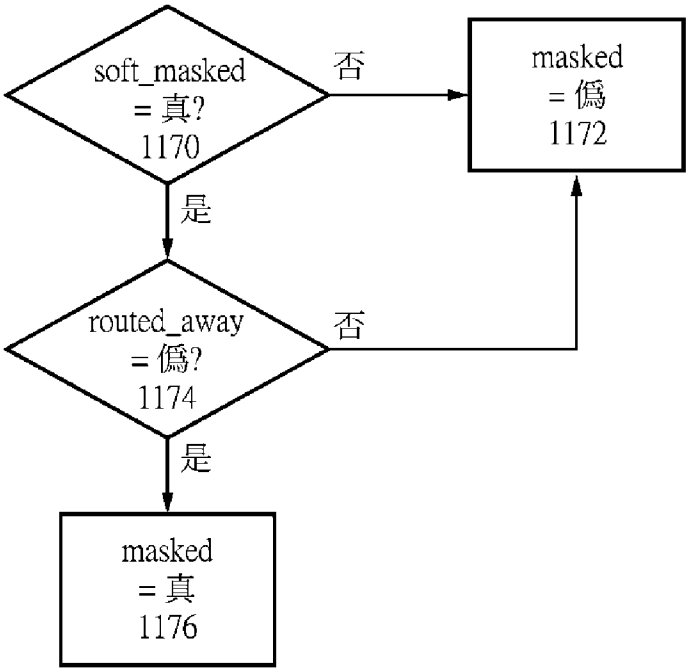
【圖8G】


在EL3的例外由外部偵錯器停用?
1136
是
否
SCR_EL3.TMEA = 1?
1138
否
是
route_to_el2 = 僞?
1140
否
是
route_soft_masked_to_el2 = 僞?
1142
否
是
PSTATE.EL在{EL2, EL1}中且PSTATE =1?
1144
否
是
PSTATE.EL在{EL2, EL0}中且soft_masked = 真?
1148
否
是
PSTATE.EL = EL3?
1150
否
是
在EL3的例外由外部偵錯器停用?
1152
否
是
route_soft_masked_to_el3 = 僞
1134
route_soft_masked_to_el3 = 真
1146
route_soft_masked_to_el3 = 僞
1156
route_soft_masked_to_el3 = 真
1154

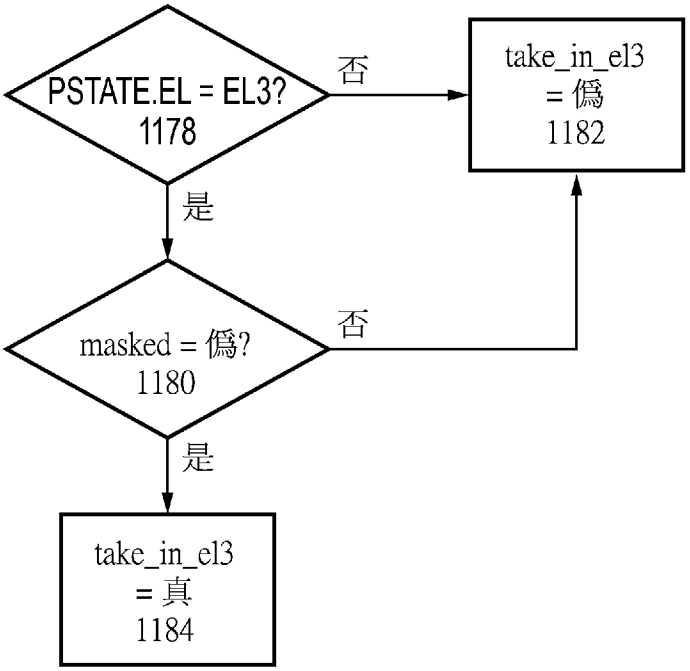
【圖8H】


route_to_el2 = 真? 1158
是
routed_away = 真 1160
否
route_soft_masked _to_el2 = 真? 1162
是
否
route_to_el3 = 真? 1164
是
否
route_soft_masked _to_el3 = 真? 1166
是
否
routed_away = 偽 1168

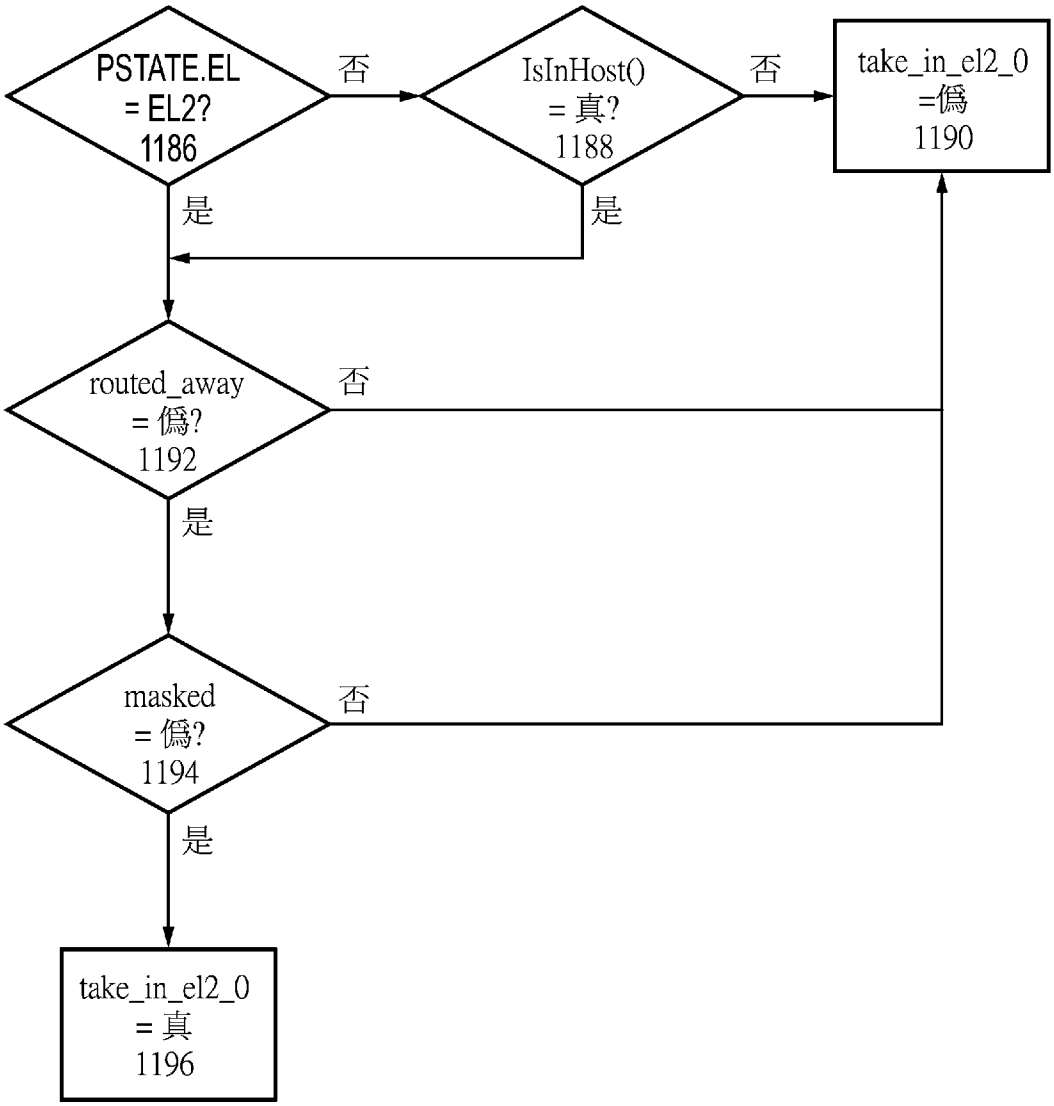
【圖8I】


soft_masked
= 真?
1170
否
masked
= 僞
1172
是
routed_away
= 僞?
1174
否
是
masked
= 真
1176

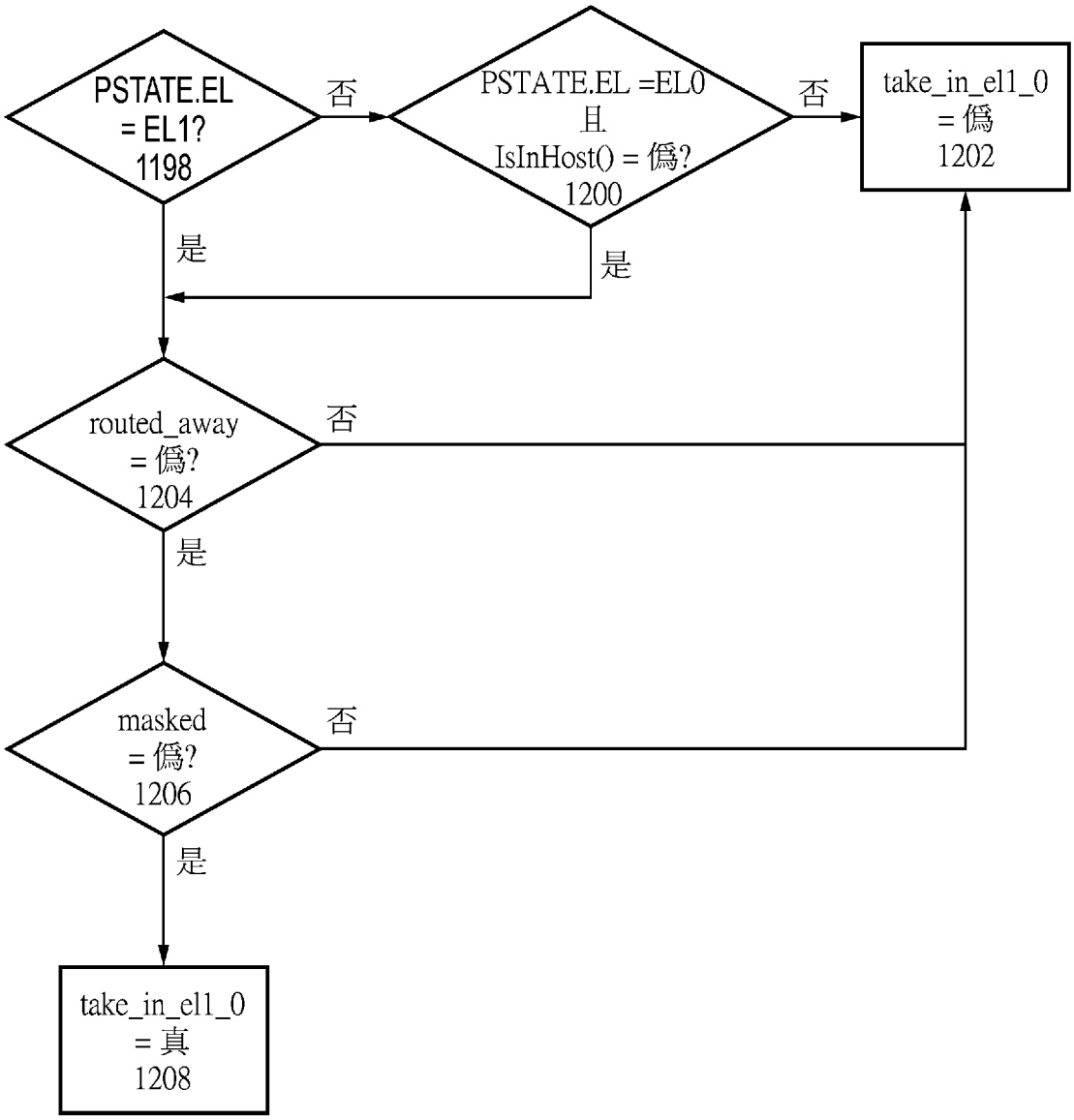
【圖8J】


PSTATE.EL = EL3?
1178
否
take_in_el3
= 僞
1182
是
masked = 僞?
1180
否
是
take_in_el3
= 真
1184

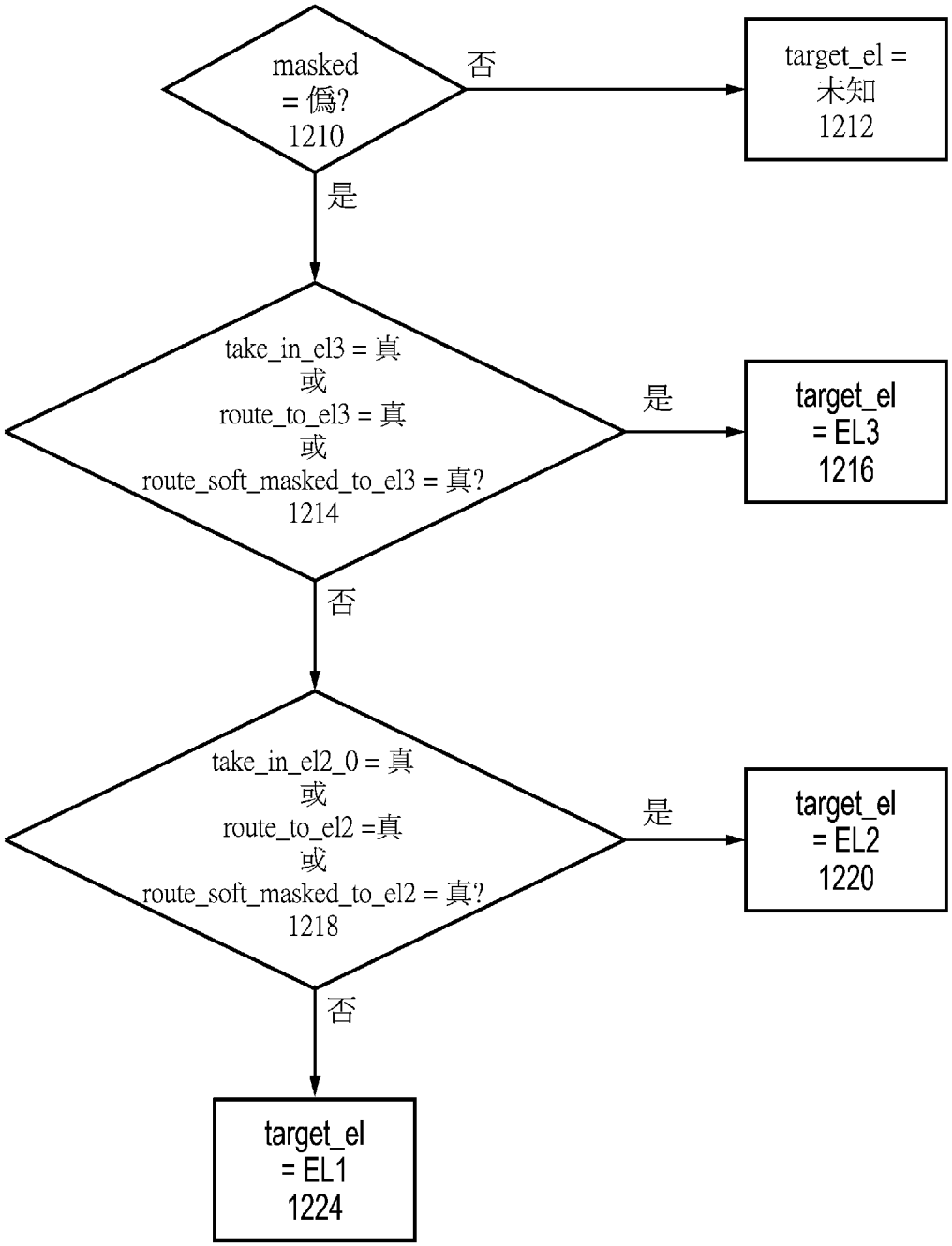
【圖8K】


PSTATE.EL
= EL2?
1186
否
IsInHost()
= 真?
1188
否
take_in_el2_0
=僞
1190
是
是
routed_away
= 僞?
1192
否
是
masked
= 僞?
1194
否
是
take_in_el2_0
= 真
1196

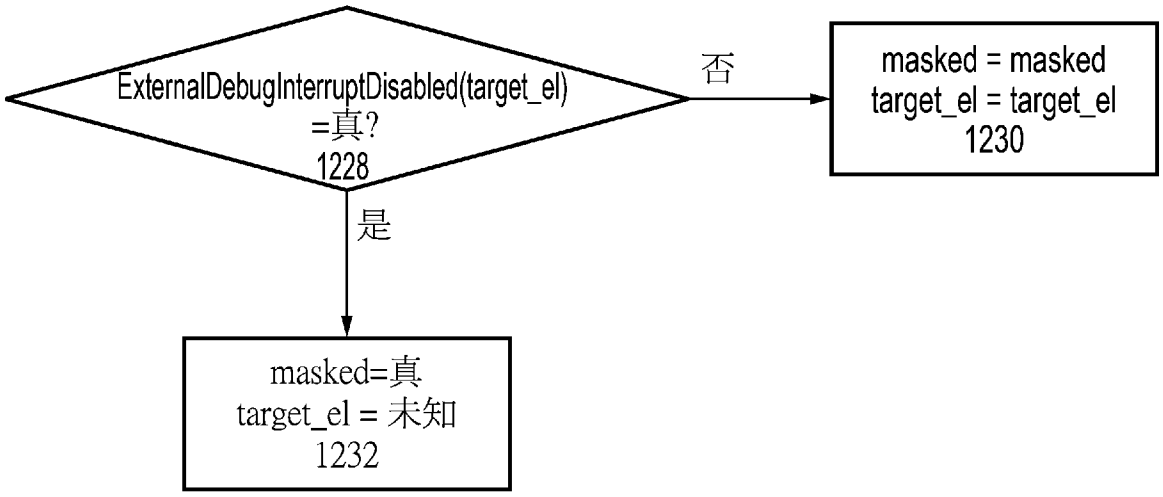
【圖8L】


PSTATE.EL
= EL1?
1198
否
PSTATE.EL =EL0
且
IsInHost() = 僞?
1200
否
take_in_el1_0
= 僞
1202
是
是
routed_away
= 僞?
1204
否
是
masked
= 僞?
1206
否
是
take_in_el1_0
= 真
1208

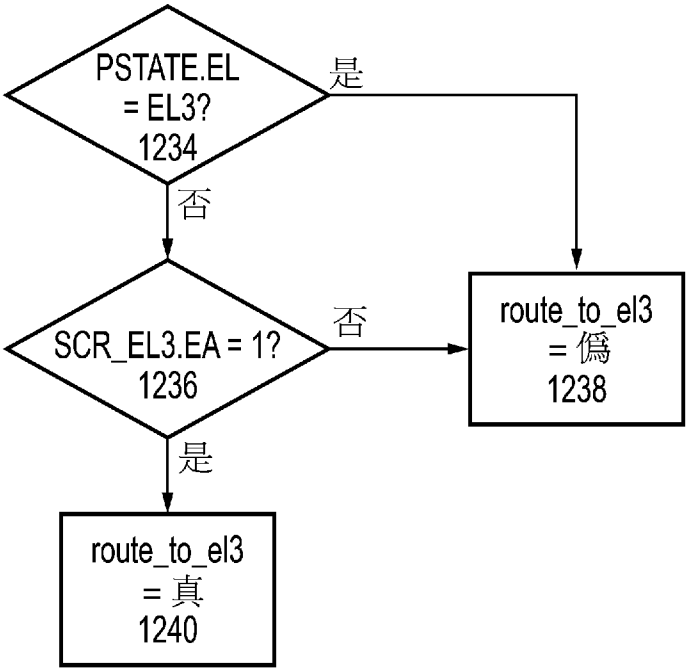
【圖8M】


masked
= 僞?
1210
否
target_el =
未知
1212
是
take_in_el3 = 真
或
route_to_el3 = 真
或
route_soft_masked_to_el3 = 真?
1214
是
target_el
= EL3
1216
否
take_in_el2_0 = 真
或
route_to_el2 =真
或
route_soft_masked_to_el2 = 真?
1218
是
target_el
= EL2
1220
否
target_el
= EL1
1224

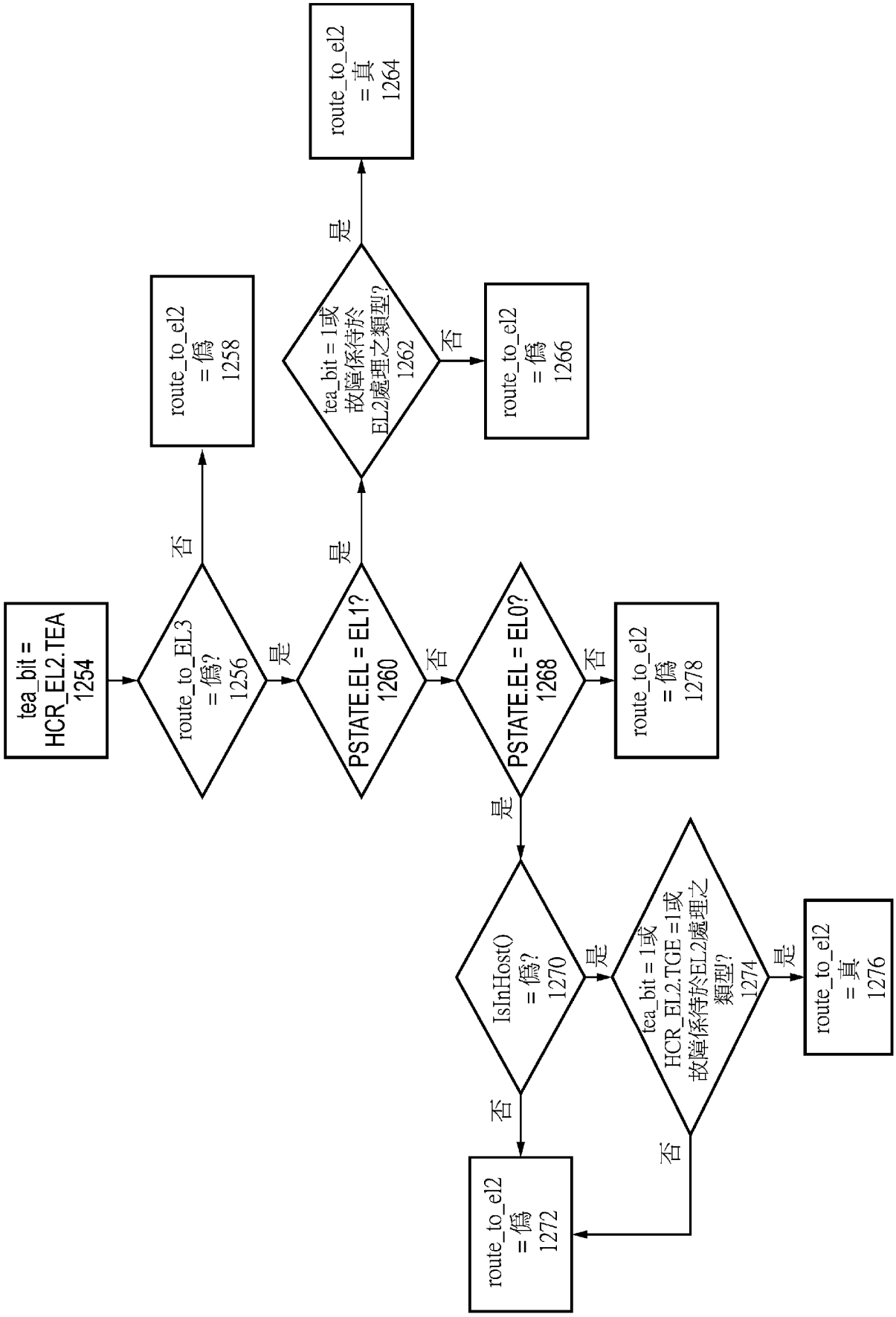
【圖8N】


ExternalDebugInterruptDisabled(target_el)
=真?
1228
否
masked = masked
target_el = target_el
1230
是
masked=真
target_el = 未知
1232

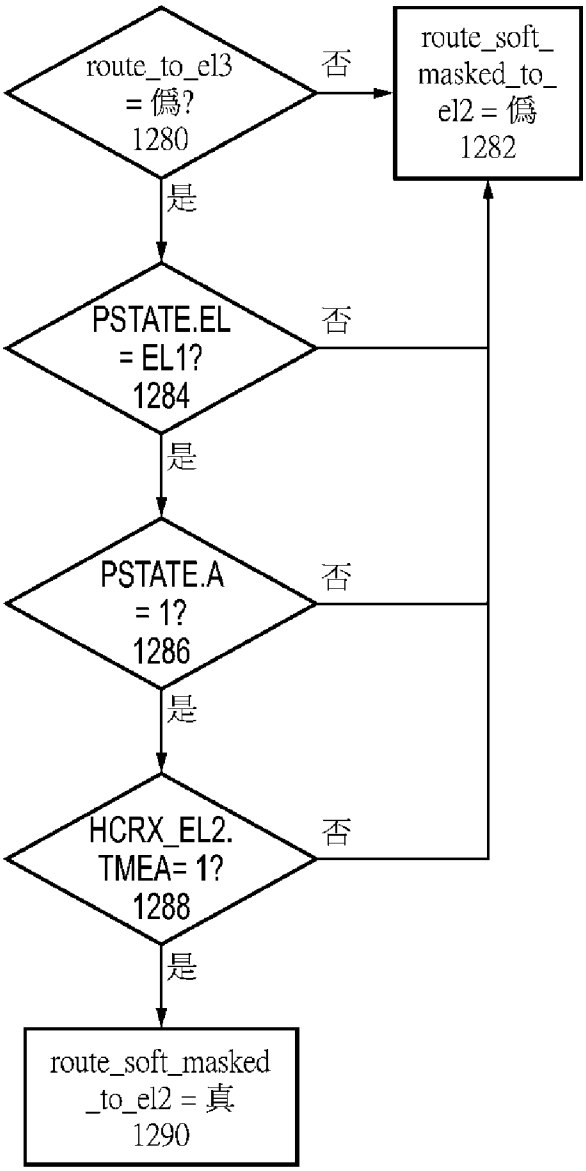
【圖8O】


PSTATE.EL
= EL3?
1234
是
否
SCR_EL3.EA = 1?
1236
否
route_to_el3
= 僞
1238
是
route_to_el3
= 真
1240

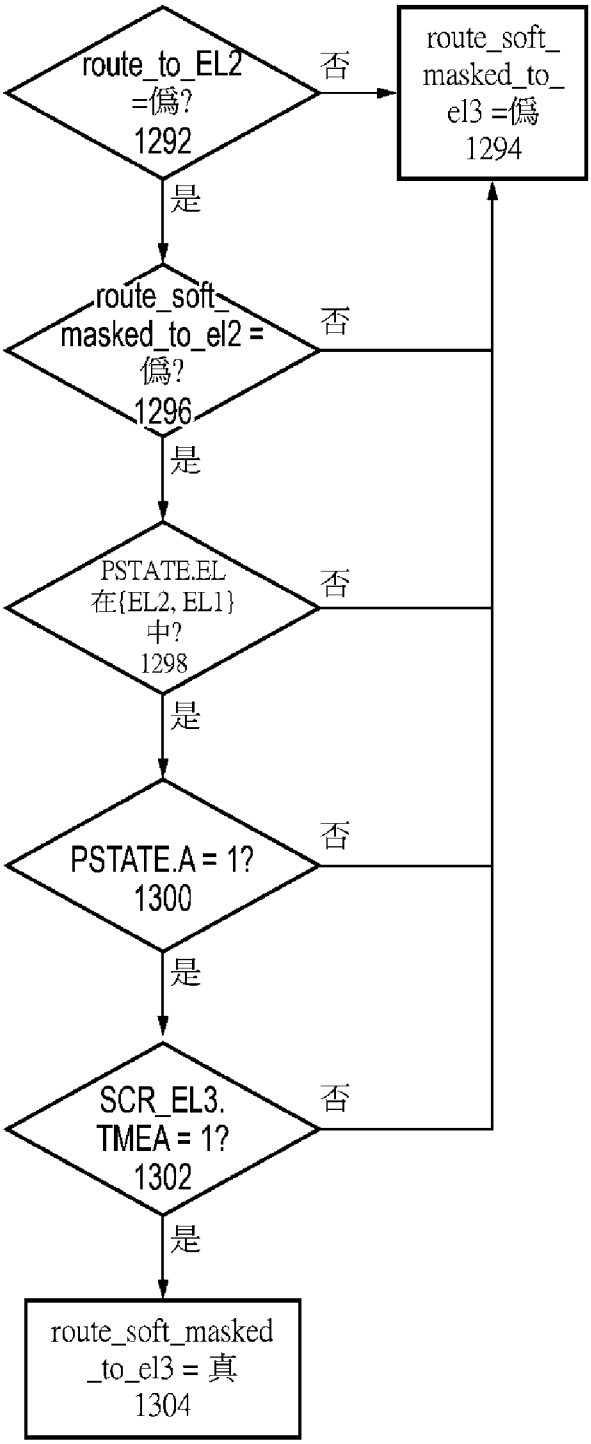
【圖9A】


tea_bit = HCR_EL2.TEA 1254
route_to_EL3 = 偽? 1256
否
是
route_to_el2 = 偽 1258
PSTATE.EL = EL1? 1260
是
否
tea_bit = 1或 故障係待於 EL2處理之類型? 1262
是
否
route_to_el2 = 真 1264
route_to_el2 = 偽 1266
PSTATE.EL = EL0? 1268
否
是
route_to_el2 = 偽 1278
IsInHost() = 偽? 1270
否
是
tea_bit = 1或 HCR_EL2.TGE =1或 故障係待於EL2處理之 類型? 1274
否
是
route_to_el2 = 偽 1272
route_to_el2 = 真 1276

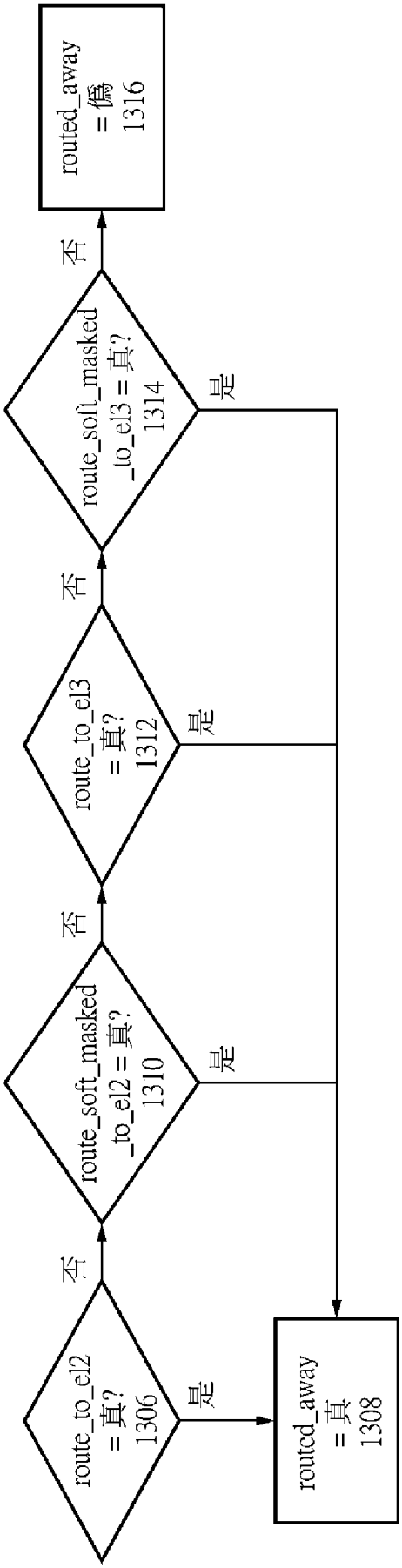
【圖9B】


route_to_el3
= 僞?
1280
否
route_soft_
masked_to_
el2 = 僞
1282
是
PSTATE.EL
= EL1?
1284
否
是
PSTATE.A
= 1?
1286
否
是
HCRX_EL2.
TMEA= 1?
1288
否
是
route_soft_masked
_to_el2 = 真
1290

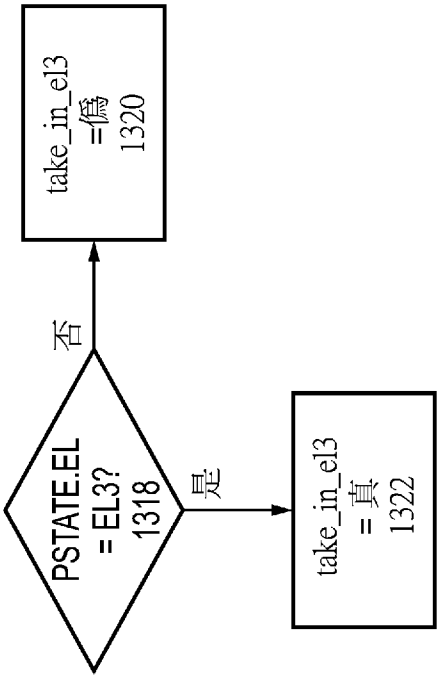
【圖9C】


route_to_EL2 =僞? 1292
否
route_soft_ masked_to_ el3 =僞 1294
是
route_soft_ masked_to_el2 = 僞? 1296
否
是
PSTATE.EL 在{EL2, EL1} 中? 1298
否
是
PSTATE.A = 1? 1300
否
是
SCR_EL3. TMEA = 1? 1302
否
是
route_soft_masked _to_el3 = 真 1304

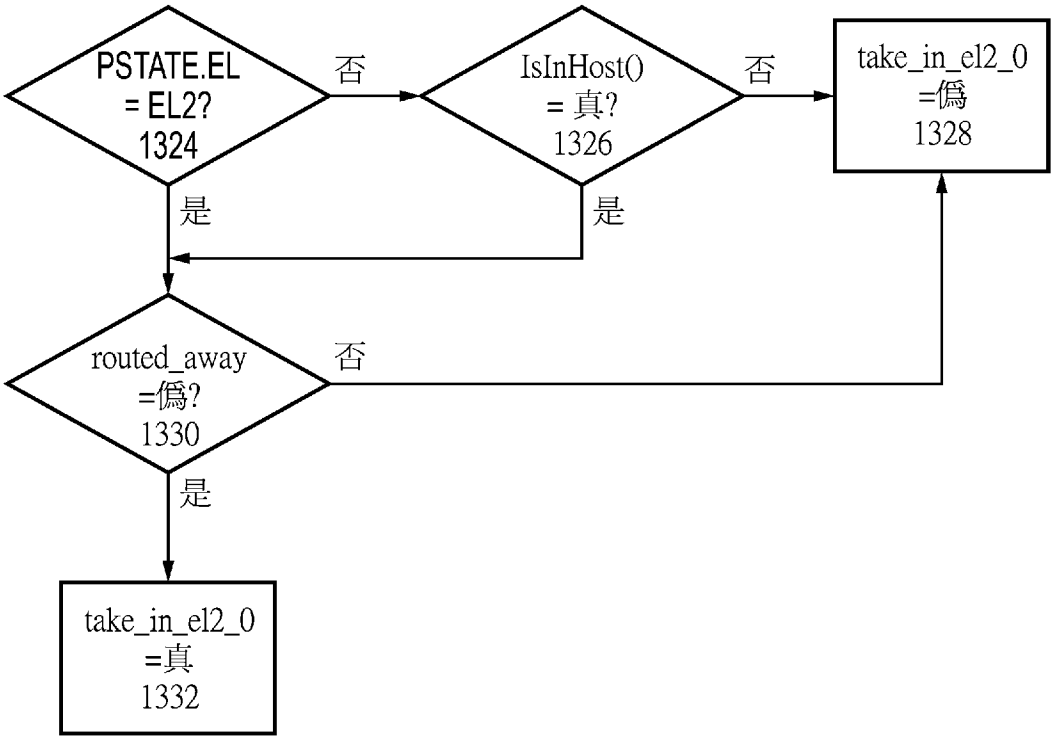
【圖9D】


routed_away
= 偽
1316
否
route_soft_masked
_to_el3 = 真?
1314
是
否
route_to_el3
= 真?
1312
是
否
route_soft_masked
_to_el2 = 真?
1310
是
否
route_to_el2
= 真?
1306
是
routed_away
= 真
1308

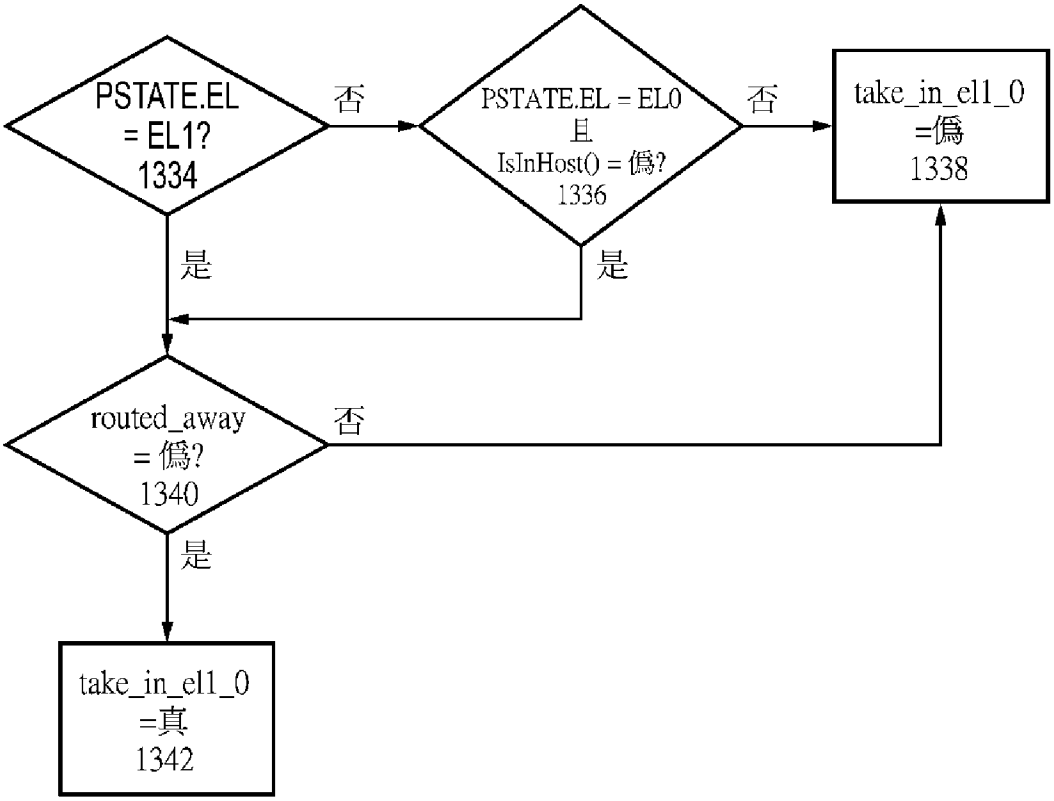
【圖9E】



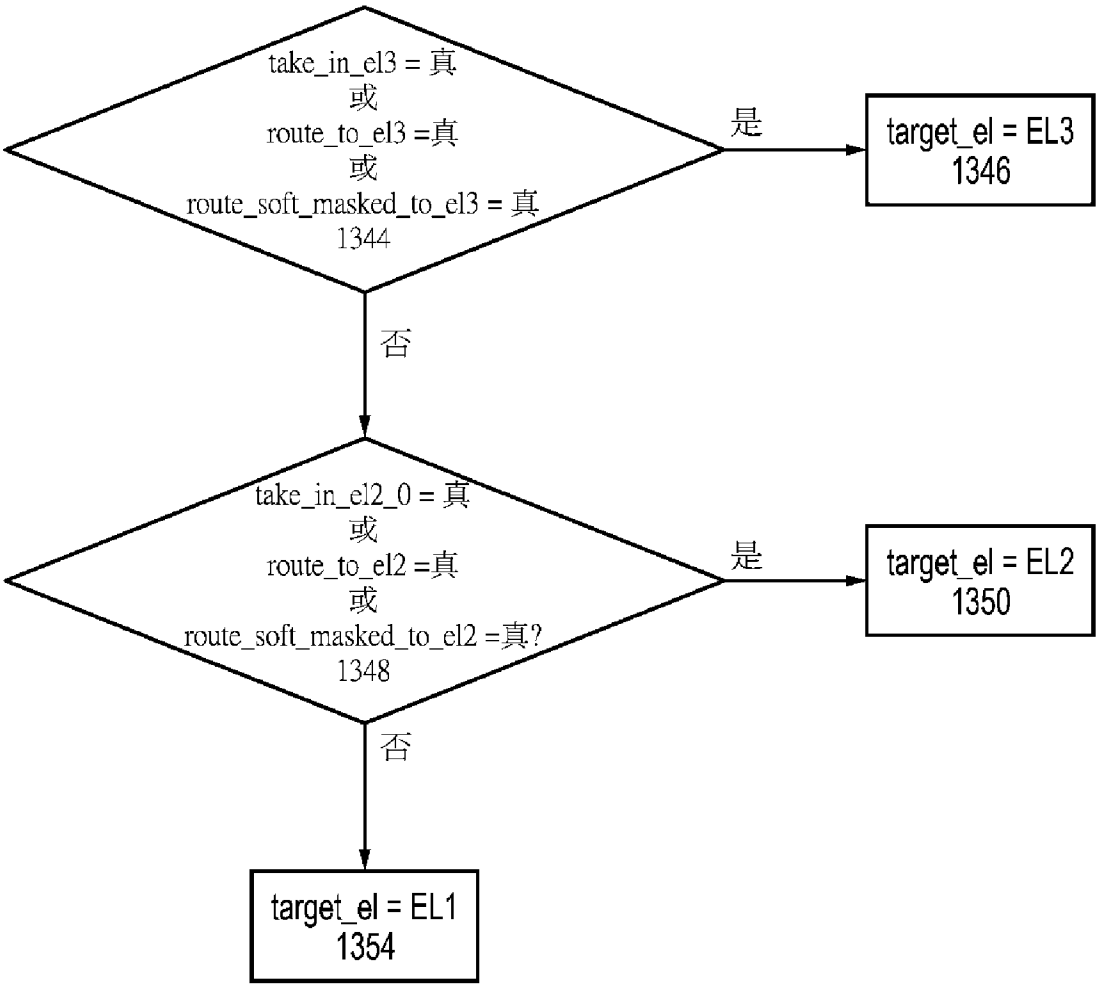
【圖9F】


PSTATE.EL
= EL2?
1324
否
IsInHost()
= 真?
1326
否
take_in_el2_0
=偽
1328
是
是
routed_away
=偽?
1330
否
是
take_in_el2_0
=真
1332

【圖9G】


PSTATE.EL
= EL1?
1334
否
PSTATE.EL = EL0
且
IsInHost() = 偽?
1336
否
take_in_el1_0
=偽
1338
是
是
routed_away
= 偽?
1340
否
是
take_in_el1_0
=真
1342

【圖9H】


take_in_el3 = 真
或
route_to_el3 =真
或
route_soft_masked_to_el3 = 真
1344
是
target_el = EL3
1346
否
take_in_el2_0 = 真
或
route_to_el2 =真
或
route_soft_masked_to_el2 =真?
1348
是
target_el = EL2
1350
否
target_el = EL1
1354

【圖9I】

	EA EL3	TMEA EL3	NMEA EL3	TGE EL2	E2H EL2	AMO EL2	NMEA EL2	TMEA EL2	NMEA EL1	A PSTATE	EL3	EL2	EL1	EL0
1	0b0	0b0	0b0	0b0	X	0b0	0b0	0b0	0b0	0b0	C	C	EL1	EL1
2	0b0	0b0	0b0	0b0	X	0b0	0b0	0b0	0b0	0b1	C	C	B	B
3	0b0	0b0	0b0	0b0	X	0b0	0b0	0b0	0b1	X	C	C	EL1	EL1
4	0b0	0b0	0b0	0b0	X	0b0	0b0	0b1	X	0b0	C	C	EL1	EL1
5	0b0	0b0	0b0	0b0	X	0b0	0b0	0b1	0b0	0b1	C	C	EL2	EL2
6	0b0	0b0	0b0	0b0	X	0b0	0b0	0b1	0b1	0b1	C	C	EL2	EL1
7	0b0	0b0	0b0	0b0	X	0b0	0b1	0b0	0b0	0b0	C	EL2	EL1	EL1
8	0b0	0b0	0b0	0b0	X	0b0	0b1	0b0	0b0	0b1	C	EL2	B	B
9	0b0	0b0	0b0	0b0	X	0b0	0b1	0b0	0b1	X	C	EL2	EL1	EL1
10	0b0	0b0	0b0	0b0	X	0b0	0b1	0b1	X	0b0	C	EL2	EL1	EL1
11	0b0	0b0	0b0	0b0	X	0b0	0b1	0b1	0b0	0b1	C	EL2	EL2	EL2
12	0b0	0b0	0b0	0b0	X	0b0	0b1	0b1	0b1	0b1	C	EL2	EL2	EL1
13	0b0	0b0	0b0	0b0	X	0b1	0b0	X	X	0b0	C	EL2	EL2	EL2
14	0b0	0b0	0b0	0b0	X	0b1	0b0	X	X	0b1	C	B	EL2	EL2
15	0b0	0b0	0b0	0b0	X	0b1	0b1	X	X	X	C	EL2	EL2	EL2
16	0b0	0b0	0b0	0b1	X	X	0b0	X	X	0b0	C	EL2	不適用	EL2
17	0b0	0b0	0b0	0b1	X	X	0b1	X	X	X	C	EL2	不適用	EL2
18	0b0	0b0	0b0	0b1	0b0	X	0b0	X	X	0b1	C	B	不適用	EL2
19	0b0	0b0	0b0	0b1	0b1	X	0b0	X	X	0b1	C	B	不適用	B
20	0b0	0b0	0b1	0b0	X	0b0	0b0	0b0	0b0	0b0	EL3	C	EL1	EL1
21	0b0	0b0	0b1	0b0	X	0b0	0b0	0b0	0b0	0b1	EL3	C	B	B
22	0b0	0b0	0b1	0b0	X	0b0	0b0	0b0	0b1	X	EL3	C	EL1	EL1
23	0b0	0b0	0b1	0b0	X	0b0	0b0	0b1	X	0b0	EL3	C	EL1	EL1
24	0b0	0b0	0b1	0b0	X	0b0	0b0	0b1	0b0	0b1	EL3	C	EL2	EL2
25	0b0	0b0	0b1	0b0	X	0b0	0b0	0b1	0b1	0b1	EL3	C	EL2	EL1
26	0b0	0b0	0b1	0b0	X	0b0	0b1	0b0	0b0	0b0	EL3	EL2	EL1	EL1
27	0b0	0b0	0b1	0b0	X	0b0	0b1	0b0	0b0	0b1	EL3	EL2	B	B
28	0b0	0b0	0b1	0b0	X	0b0	0b1	0b0	0b1	X	EL3	EL2	EL1	EL1
29	0b0	0b0	0b1	0b0	X	0b0	0b1	0b1	X	0b0	EL3	EL2	EL1	EL1
30	0b0	0b0	0b1	0b0	X	0b0	0b1	0b1	0b0	0b1	EL3	EL2	EL2	EL2
31	0b0	0b0	0b1	0b0	X	0b0	0b1	0b1	0b1	0b1	EL3	EL2	EL2	EL1
32	0b0	0b0	0b1	0b0	X	0b1	0b0	X	X	0b0	EL3	EL2	EL2	EL2
33	0b0	0b0	0b1	0b0	X	0b1	0b0	X	X	0b1	EL3	B	EL2	EL2
34	0b0	0b0	0b1	0b0	X	0b1	0b1	X	X	X	EL3	EL2	EL2	EL2
35	0b0	0b0	0b1	0b1	X	X	0b0	X	X	0b0	EL3	EL2	不適用	EL2
36	0b0	0b0	0b1	0b1	X	X	0b1	X	X	X	EL3	EL2	不適用	EL2

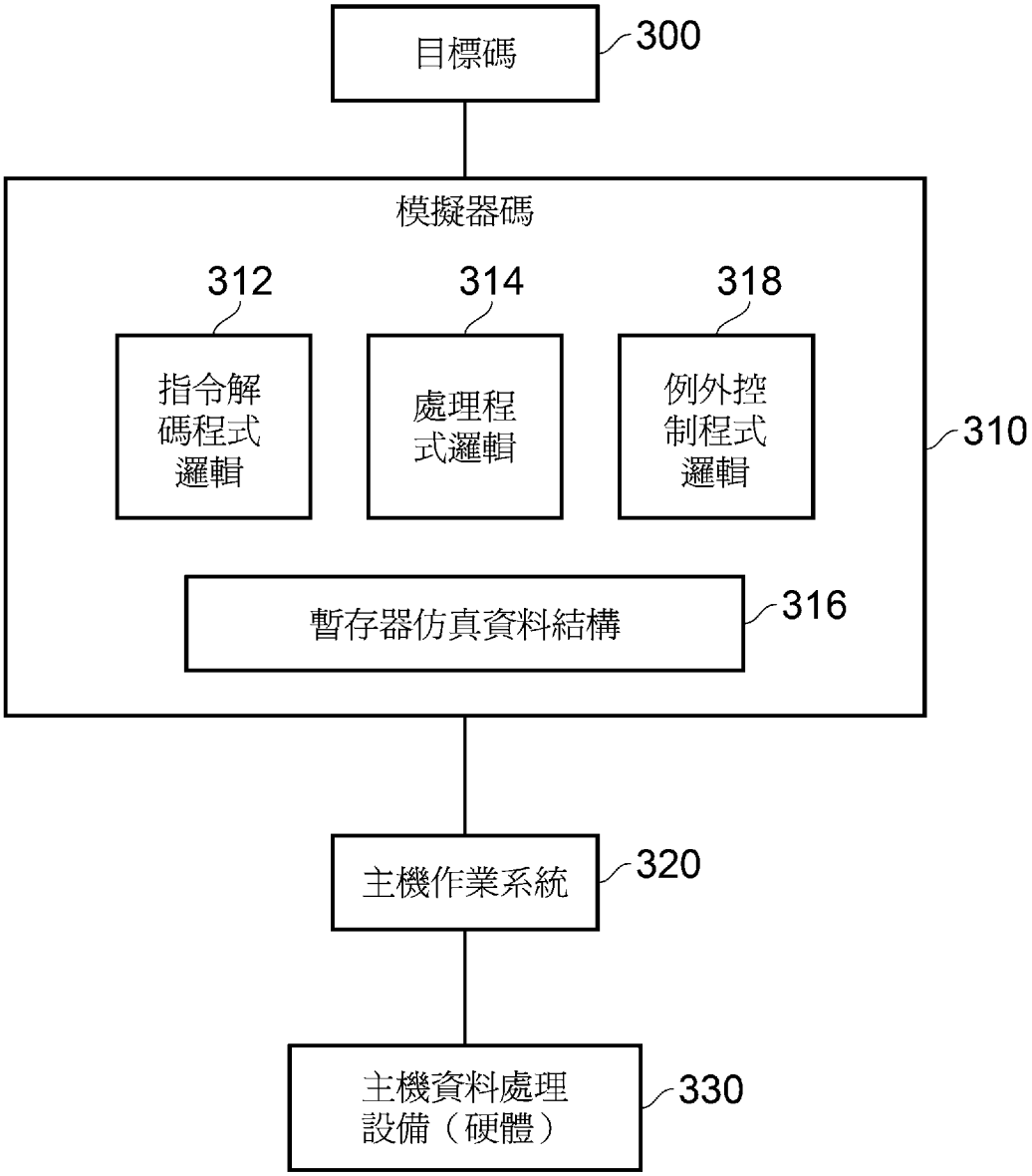
【圖10A】

	EA EL3	TMEA EL3	NMEA EL3	TGE EL2	E2H EL2	AMO EL2	NMEA EL2	TMEA EL2	NMEA EL1	A PSTATE	EL3	EL2	EL1	EL0
37	0b0	0b0	0b1	0b1	0b0	X	0b0	X	X	0b1	EL3	B	不適用	EL2
38	0b0	0b0	0b1	0b1	0b1	X	0b0	X	X	0b1	EL3	B	不適用	B
39	0b0	0b1	0b0	0b0	X	0b0	X	0b0	0b0	0b1	C	EL3	EL3	EL3
40	0b0	0b1	0b0	0b0	X	0b0	X	0b0	0b1	0b1	C	EL3	EL3	EL1
41	0b0	0b1	0b0	0b0	X	0b0	X	0b1	0b0	0b1	C	EL3	EL2	EL2
42	0b0	0b1	0b0	0b0	X	0b0	X	0b1	0b1	0b1	C	EL3	EL2	EL1
43	0b0	0b1	0b0	0b0	X	0b0	0b0	X	X	0b0	C	EL3	EL1	EL1
44	0b0	0b1	0b0	0b0	X	0b0	0b1	X	X	0b0	C	EL2	EL1	EL1
45	0b0	0b1	0b0	0b0	X	0b1	X	X	X	0b0	C	EL2	EL2	EL2
46	0b0	0b1	0b0	0b0	X	0b1	X	X	X	0b1	C	EL3	EL2	EL2
47	0b0	0b1	0b0	0b1	X	X	X	X	X	0b0	C	EL2	不適用	EL2
48	0b0	0b1	0b0	0b1	0b0	X	X	X	X	0b1	C	EL3	不適用	EL2
49	0b0	0b1	0b0	0b1	0b1	X	0b0	X	X	0b1	C	EL3	不適用	EL3
50	0b0	0b1	0b0	0b1	0b1	X	0b1	X	X	0b1	C	EL3	不適用	EL2
51	0b0	0b1	0b1	0b0	X	0b0	X	0b0	0b0	0b1	EL3	EL3	EL3	EL3
52	0b0	0b1	0b1	0b0	X	0b0	X	0b0	0b1	0b1	EL3	EL3	EL3	EL1
53	0b0	0b1	0b1	0b0	X	0b0	X	0b1	0b0	0b1	EL3	EL3	EL2	EL2
54	0b0	0b1	0b1	0b0	X	0b0	X	0b1	0b1	0b1	EL3	EL3	EL2	EL1
55	0b0	0b1	0b1	0b0	X	0b0	0b0	X	X	0b0	EL3	EL3	EL1	EL1
56	0b0	0b1	0b1	0b0	X	0b0	0b1	X	X	0b0	EL3	EL2	EL1	EL1
57	0b0	0b1	0b1	0b0	X	0b1	X	X	X	0b0	EL3	EL2	EL2	EL2
58	0b0	0b1	0b1	0b0	X	0b1	X	X	X	0b1	EL3	EL3	EL2	EL2
59	0b0	0b1	0b1	0b1	X	X	X	X	X	0b0	EL3	EL2	不適用	EL2
60	0b0	0b1	0b1	0b1	0b0	X	X	X	X	0b1	EL3	EL3	不適用	EL2
61	0b0	0b1	0b1	0b1	0b1	X	0b0	X	X	0b1	EL3	EL3	不適用	EL3
62	0b0	0b1	0b1	0b1	0b1	X	0b1	X	X	0b1	EL3	EL3	不適用	EL2
63	0b1	X	0b0	0b0	X	X	X	X	X	0b0	EL3	EL3	EL3	EL3
64	0b1	X	0b0	0b0	X	X	X	X	X	0b1	B	EL3	EL3	EL3
65	0b1	X	0b0	0b1	X	X	X	X	X	0b0	EL3	EL3	不適用	EL3
66	0b1	X	0b0	0b1	X	X	X	X	X	0b1	B	EL3	不適用	EL3
67	0b1	X	0b1	0b0	X	X	X	X	X	X	EL3	EL3	EL3	EL3
68	0b1	X	0b1	0b1	X	X	X	X	X	X	EL3	EL3	不適用	EL3

【圖10B】

	EA EL3	TMEA EL3	TGE EL2	TEA EL2	TMEA EL2	A PSTATE	EL3	EL2	EL1	EL0
1	0b0	0b0	0b0	0b0	0b0	X	EL3	EL2	EL1	EL1
2	0b0	0b0	0b0	0b0	0b1	0b0	EL3	EL2	EL1	EL1
3	0b0	0b0	0b0	0b0	0b1	0b1	EL3	EL2	EL2	EL1
4	0b0	0b0	0b0	0b1	X	X	EL3	EL2	EL2	EL2
5	0b0	0b0	0b1	X	X	X	EL3	EL2	不適用	EL2
6	0b0	0b1	0b0	0b0	X	0b0	EL3	EL2	EL1	EL1
7	0b0	0b1	0b0	0b0	0b0	0b1	EL3	EL3	EL3	EL1
8	0b0	0b1	0b0	0b0	0b1	0b1	EL3	EL3	EL2	EL1
9	0b0	0b1	0b0	0b1	X	0b0	EL3	EL2	EL2	EL2
10	0b0	0b1	0b0	0b1	X	0b1	EL3	EL3	EL2	EL2
11	0b0	0b1	0b1	X	X	0b0	EL3	EL2	不適用	EL2
12	0b0	0b1	0b1	X	X	0b1	EL3	EL3	不適用	EL2
13	0b1	X	0b0	X	X	X	EL3	EL3	EL3	EL3
14	0b1	X	0b1	X	X	X	EL3	EL3	不適用	EL3

【圖11】


目標碼
300
模擬器碼
312
314
318
指令解碼程式邏輯
處理程式邏輯
例外控制程式邏輯
310
暫存器仿真資料結構
316
主機作業系統
320
主機資料處理設備（硬體）
330

【圖12】