



(19)中華民國智慧財產局

(12)發明說明書公告本 (11)證書號數：TW I506556 B

(45)公告日：中華民國 104 (2015) 年 11 月 01 日

(21)申請案號：101102912

(22)申請日：中華民國 101 (2012) 年 01 月 30 日

(51)Int. Cl. : G06F9/45 (2006.01)

(30)優先權：2011/01/25 美國 61/436,013

(71)申請人：美光科技公司 (美國) MICRON TECHNOLOGY, INC. (US)
美國

(72)發明人：葛蘭汀尼 保羅 GLENDENNING, PAUL (AU)；許郡君 XU, JUNJUAN (CN)

(74)代理人：陳長文

(56)參考文獻：

TW 200301420A

TW 200412724A

US 5537580A

US 6748588B1

審查人員：李惟任

申請專利範圍項數：29 項 圖式數：16 共 89 頁

(54)名稱

用於編譯正規表達式之方法及裝置

METHOD AND APPARATUS FOR COMPILING REGULAR EXPRESSIONS

(57)摘要

描述用於一編譯程式之裝置、系統及方法。一種此類編譯程式將原始程式碼轉換成包含狀態及該等狀態之間的轉變的一自動機，其中該自動機中之該等狀態包括對應於一特殊目的硬體元件之一特殊目的狀態。該編譯程式將該自動機轉換成一接線對照表，且對該接線對照表進行置放及路徑選擇以提供用於組態一目標器件之機器碼。

Apparatus, systems, and methods for a compiler are described. One such compiler converts source code into an automaton comprising states and transitions between the states, wherein the states in the automaton include a special purpose state that corresponds to a special purpose hardware element. The compiler converts the automaton into a netlist, and places and routes the netlist to provide machine code for configuring a target device.

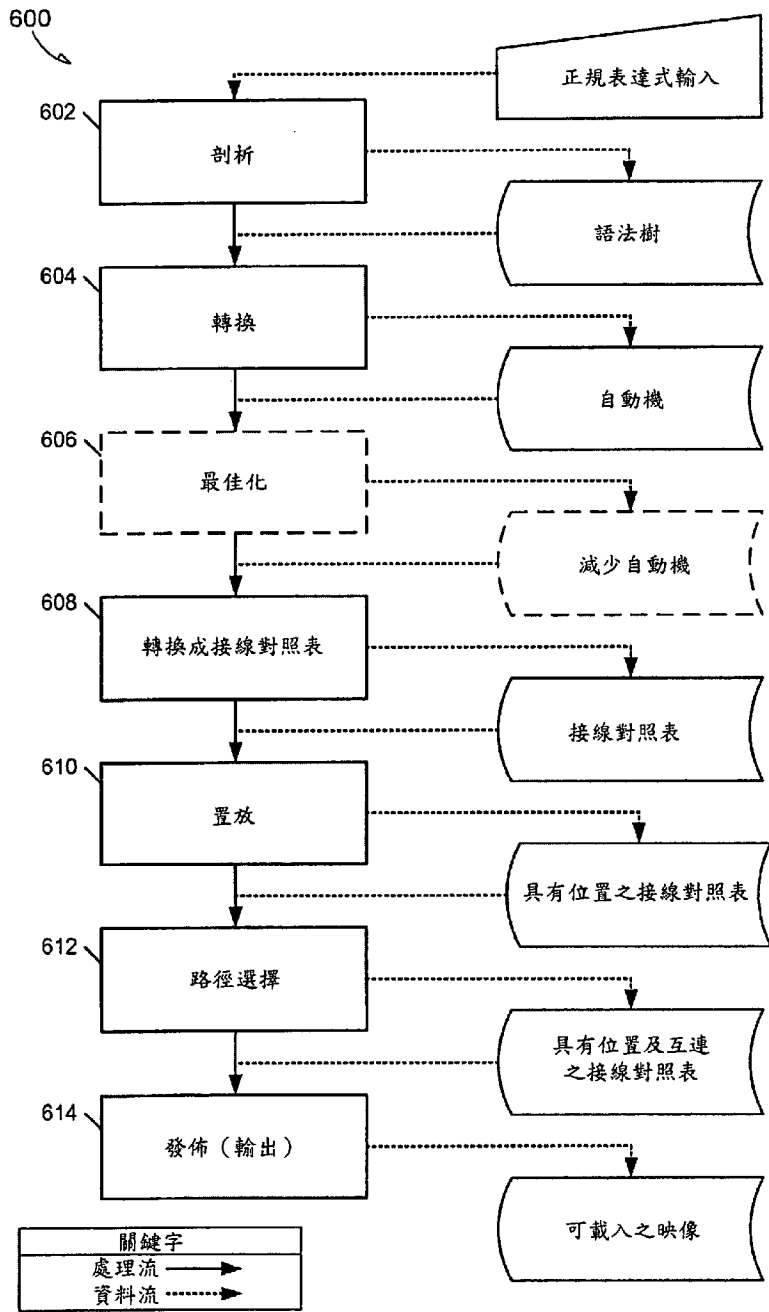


圖6

發明專利說明書

公告本

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※申請案號：101102912

※申請日：101. 1. 30 ※IPC 分類：G06F 9/45 (2006.01)

一、發明名稱：^{101. 1. 30}(中文/英文)

用於編譯正規表達式之方法及裝置

METHOD AND APPARATUS FOR COMPILING REGULAR EXPRESSIONS

二、中文發明摘要：

描述用於一編譯程式之裝置、系統及方法。一種此類編譯程式將原始程式碼轉換成包含狀態及該等狀態之間的轉變的一自動機，其中該自動機中之該等狀態包括對應於一特殊目的硬體元件之一特殊目的狀態。該編譯程式將該自動機轉換成一接線對照表，且對該接線對照表進行置放及路徑選擇以提供用於組態一目標器件之機器碼。

三、英文發明摘要：

Apparatus, systems, and methods for a compiler are described. One such compiler converts source code into an automaton comprising states and transitions between the states, wherein the states in the automaton include a special purpose state that corresponds to a special purpose hardware element. The compiler converts the automaton into a netlist, and places and routes the netlist to provide machine code for configuring a target device.

四、指定代表圖：

(一)本案指定代表圖為：第(6)圖。

(二)本代表圖之元件符號簡單說明：

(無元件符號說明)

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

(無)

六、發明說明：

本專利申請案依據35 U.S.C.章節119(e)主張2011年1月25日申請之題為「METHOD AND APPARATUS FOR COMPILING REGULAR EXPRESSIONS」之美國臨時專利申請案第61/436,013號的優先權之利益，該美國臨時專利申請案之全文以引用方式併入本文中。

【先前技術】

有限狀態機(FSM)(亦被稱作有限狀態自動機、自動機或簡稱為狀態機)為狀態、狀態之間的轉變及動作的表示。有限狀態機可用以設計數位邏輯、電腦程式或平行機之映像。有限狀態機為由有限數目個狀態、彼等狀態之間的轉變及輸出組成的行為模型。有限狀態機可表示為一圖，其中該圖之頂點對應於該有限狀態機之狀態，且該圖之邊對應於歸因於至該有限狀態機之一或多個輸入而發生的狀態之間的轉變。有限狀態機亦可具有機率轉變、模糊狀態或其他異事。有限狀態機具有一有限內部記憶體、一輸入特徵及一選用之輸出特徵。具有一輸出之有限狀態機可被稱作有限狀態轉換器。

有限狀態機之應用包括電子設計自動化、通信協定設計、生物學及人工智慧研究，以及用以描述自然語言之文法的語言學。

【實施方式】

以下描述及圖式充分地說明使熟習此項技術者能夠實踐的特定實施例。其他實施例可併有結構、邏輯、電、過程

及其他改變。一些實施例之部分及特徵可包括於其他實施例中或可替代其他實施例之部分及特徵。申請專利範圍中所陳述之實施例涵蓋彼等技術方案之所有可用等效物。

本文件尤其描述用於將原始程式碼(例如，正規表達式)轉換成機器碼(例如，用於組態(例如，程式化)平行機之映像)的編譯程式。由編譯程式產生之映像(輸出檔案)可程式化該平行機以執行某些功能。在某些實例中，該平行機可包括一有限狀態機(FSM)引擎、一場可程式化閘陣列(FPGA)及其變體。

圖1說明目標器件(例如，平行機100)之實例。平行機100可接收輸入資料並基於該輸入資料提供一輸出。平行機100可包括用於接收輸入資料之資料輸入埠110及用於將輸出提供至另一器件的輸出埠114。資料輸入埠110提供用於將輸入至該平行機100之資料的介面。

平行機100包括複數個可程式化元件，包括通用元件102及特殊目的元件112。通用元件102可包括一或多個輸入104及一或多個輸出106。可將通用元件102程式化至複數個狀態中之一者中。通用元件102之狀態判定該等通用元件102將基於給定輸入提供哪一(些)輸出。亦即，通用元件102之狀態判定可程式化元件將如何對給定輸入起反應(例如，回應)。輸入至資料輸入埠110之資料可提供至該複數個通用元件102以使該等通用元件102對其採取行動。如下文詳細論述，通用元件102之實例可包括(例如)狀態機元件(SME)、計數器，及/或可組態邏輯區塊，以及其他可程式

化元件。在一實例中，SME可經程式化(例如，設定)以當在資料輸入埠110處接收到一給定輸入時提供某一輸出(例如，高或「1」信號)。當在資料輸入埠110處接收到不同於該給定輸入之一輸入時，該SME可提供一不同輸出(例如，低或「0」信號)。在一實例中，一可組態邏輯區塊可經設定以基於在資料輸入埠110處接收到之輸入來執行布林邏輯函數(例如，「及」(AND)、「或」(OR)、「或非」(NOR)等)。稍後在本文中論述計數器之實例。特殊目的元件112可包括記憶體(例如，RAM)、邏輯閘、計數器、查找表、場可程式化閘陣列(FPGA)及其他硬體元件。特殊目的元件112可與通用元件102互動並執行特殊目的功能。

平行機100亦可包括用於將程式(例如，映像)載入至平行機100上之程式設計介面111。該映像可程式化(例如，設定)通用元件102之狀態。亦即，該映像可組態通用元件102以按某一方式來對給定輸入起反應。舉例而言，一通用元件102可經設定，以當在資料輸入埠110處接收到字元「a」時輸出一高信號。在一些實例中，平行機100可使用一時脈信號來控制通用元件102之操作的時序。在一些實施例中，在資料輸入埠110處接收到之資料可包括隨時間過去或一齊接收到之一組固定之資料或隨時間過去而接收到之資料串流。資料可自耦接至平行機100之任何來源接收或藉由耦接至平行機100之任何來源產生，該任何來源諸如資料庫、感測器、網路等。

平行機100亦包括用於將平行機100之不同元件(例如，

通用元件102、資料輸入埠110、輸出埠114、程式設計介面111及特殊目的元件112)選擇性地耦接在一起的複數個可程式化開關108。因此，平行機100包含在該等元件間形成之可程式化矩陣。在一實例中，一可程式化開關108可將兩個或兩個以上元件選擇性地耦接至彼此，使得一通用元件102之輸入104、資料輸入埠110、程式設計介面111或特殊目的元件112可經由一或多個可程式化開關108而耦接至一通用元件102之輸出106、輸出埠114、程式設計介面111或特殊目的元件112。因此，該等元件之間的信號之投送可藉由設定可程式化開關108來控制。儘管圖1說明一給定元件與一可程式化開關108之間的某一數目之導體(例如，電線)，但應理解，在其他實例中，可使用不同數目之導體。而且，儘管圖1說明每一通用元件102個別地耦接至一可程式化開關108，但在其他實例中，多個通用元件102可作為一群組(例如，區塊202，如圖2中所說明)而耦接至一可程式化開關108。在一實例中，資料輸入埠110、資料輸出埠114及/或程式設計介面111可實施為暫存器，以使得至暫存器之寫入將資料提供至各別元件或自各別元件提供資料。

在一實例中，一單一平行機100實施於一實體器件上，然而，在其他實例中，兩個或兩個以上平行機100可實施於單一實體器件(例如，實體晶片)上。在一實例中，多個平行機100中之每一者可包括一不同之資料輸入埠110、一不同之輸出埠114、一不同之程式設計介面111，及一組不

同之通用元件102。此外，每一組通用元件102可對其對應輸入資料埠110處之資料起反應(例如，輸出高或低信號)。舉例而言，對應於第一平行機100之第一組通用元件102可對對應於第一平行機100之第一資料輸入埠110處之資料起反應。對應於第二平行機100之第二組通用元件102可對對應於第二平行機100之第二資料輸入埠110起反應。因此，每一平行機100包括一組通用元件102，其中通用元件102之不同集合可對不同輸入資料起反應。類似地，每一平行機100及每一組對應之通用元件102可提供一不同輸出。在一些實例中，來自第一平行機100之輸出埠114可耦接至第二平行機100之輸入埠110，使得用於第二平行機100之輸入資料可包括來自第一平行機100之輸出資料。

在一實例中，用於載入至平行機100上之映像包含用於在平行機100內設定通用元件102之狀態、程式化可程式化開關108及組態特殊目的元件112的複數個資訊位元。在一實例中，該映像可載入至平行機100上以程式化該平行機100以基於某些輸入而提供所要輸出。輸出埠114可基於通用元件102對在輸入埠110處接收到之資料所起之反應來提供來自平行機100之輸出。來自輸出埠114之輸出可包括指示一給定型樣之匹配的單一位元、包含指示與複數個型樣之匹配及不匹配之複數個位元的字，及對應於在給定時刻所有或某些通用元件102之狀態的輸出向量。

平行機100之實例用途包括型樣識別(例如，語音識別、影像識別等)、信號處理、成像、電腦視覺、密碼學及其

他。在某些實例中，平行機100可包含一有限狀態機(FSM)引擎、一場可程式化閘陣列(FPGA)及其變體。此外，平行機100可為較大器件中之一組件，該較大器件諸如電腦、尋呼機、蜂巢式電話、個人行事曆、攜帶型音訊播放器、網路器件(例如，路由器、防火牆、交換機或其任何組合)、控制電路、攝影機等。

圖2至圖5說明實施為有限狀態機(FSM)引擎200之另一平行機。在一實例中，FSM引擎200包含有限狀態機之硬體實施方案。因此，FSM引擎200實施對應於FSM中之複數個狀態的複數個可選擇性地耦接之硬體元件(例如，可程式化元件)。類似於FSM中之狀態，一硬體元件可分析一輸入串流且基於該輸入串流來啟動一下游硬體元件。

FSM引擎200包括複數個可程式化元件，包括通用元件及特殊目的元件。通用元件可經程式化以實施許多不同功能。此等通用元件包括SME 204、205(展示於圖5中)，該等SME 204、205階層式組織成列206(展示於圖3及圖4中)及區塊202(展示於圖2及圖3中)。為了在經階層式組織之SME 204、205之間投送信號，使用可程式化開關之階層，該階層包括區塊間開關203(展示於圖2及圖3中)、區塊內開關208(展示於圖3及圖4中)及列內開關212(展示於圖4中)。SME 204、205可對應於藉由FSM引擎200實施的FSM之狀態。如下文所描述，可藉由使用可程式化開關將SME 204、205耦接在一起。因此，可藉由程式化SME 204、205以對應於狀態之功能及藉由將SME 204、205選擇性地耦接

在一起以對應於FSM中之狀態之間的轉變來在FSM引擎200上實施FSM。

圖2說明實例FSM引擎200之全圖。FSM引擎200包括複數個區塊202，該等區塊可與可程式化區塊間開關203選擇性地耦接在一起。另外，區塊202可選擇性地耦接至用於接收信號(例如，資料)及將資料提供至區塊202的輸入區塊209(例如，資料輸入埠)。區塊202亦可選擇性地耦接至用於將來自區塊202之信號提供至外部器件(例如，另一FSM引擎200)的輸出區塊213(例如，輸出埠)。FSM引擎200亦可包括一程式設計介面211以將一程式(例如，映像)載入至FSM引擎200上。該映像可程式化(例如，設定)SME 204、205之狀態。亦即，該映像可組態SME 204、205以按某一方式對輸入區塊209處之給定輸入起反應。舉例而言，一SME 204可經設定，以當在輸入區塊209處接收到字元「a」時輸出一高信號。

在一實例中，輸入區塊209、輸出區塊213及/或程式設計介面211可實施為暫存器，以使得至暫存器之寫入將資料提供至各別元件或自各別元件提供資料。因此，來自儲存於對應於程式設計介面211之暫存器中的映像之位元可載入於SME 204、205上。儘管圖2說明區塊202、輸入區塊209、輸出區塊213及區塊間開關203之間的某一數目之導體(例如，電線、跡線)，但應理解，在其他實例中，可使用更少或更多之導體。

圖3說明區塊202之實例。區塊202可包括複數個列206，

該等列可與可程式化區塊內開關 208 選擇性地耦接在一起。另外，列 206 可藉由區塊間開關 203 而選擇性地耦接至另一區塊 202 內之另一列 206。在一實例中，包括緩衝器 201 以控制去往/來自區塊間開關 203 之信號的時序。列 206 包括複數個 SME 204、205，該複數個 SME 204、205 組織成本文中被稱作成對群組(group of two；GOT)210 的元件對。在一實例中，區塊 202 包含十六(16)個列 206。

圖 4 說明列 206 之實例。GOT 210 可藉由可程式化列內開關 212 而選擇性地耦接至該列 206 內之其他 GOT 210 及任何其他元件 224。GOT 210 亦可藉由區塊內開關 208 而耦接至其他列 206 中之其他 GOT 210，或藉由區塊間開關 203 而耦接至其他區塊 202 中之其他 GOT 210。在一實例中，GOT 210 具有第一輸入 214 及第二輸入 216 以及一輸出 218。第一輸入 214 耦接至 GOT 210 之第一 SME 204，且第二輸入 216 耦接至 GOT 210 之第二 SME 204。

在一實例中，列 206 包括第一複數個列互連導體 220 及第二複數個列互連導體 222。在一實例中，GOT 210 之輸入 214、216 可耦接至一或多個列互連導體 220、222，且輸出 218 可耦接至一個列互連導體 220、222。在一實例中，第一複數個列互連導體 220 可耦接至列 206 內之每一 GOT 210 之每一 SME 204。第二複數個列互連導體 222 可耦接至列 206 內之每一 GOT 210 之一 SME 204，但不可耦接至 GOT 210 之另一 SME 204。在一實例中，該第二複數個列互連導體 222 之前一半可耦接至一列 206 內之 SME 204 之前一半(來

自每一 GOT 210之一 SME 204)，且該第二複數個列互連導體222之後一半可耦接至一系列206內之 SME 204的後一半(來自每一 GOT 210之另一 SME 204)。該第二複數個列互連導體222與 SME 204、205之間的有限連接性在本文中被稱作「對等性」。

在一實例中，列206亦可包括一特殊目的元件224，諸如計數器、可程式化布林邏輯元件、場可程式化閘陣列(FPGA)、特殊應用積體電路(ASIC)、可程式化處理器(例如，微處理器)及其他元件。另外，在一實例中，該特殊目的元件224在不同列206中為不同的。舉例而言，一區塊202中之列206中之四個可包括布林邏輯作為特殊目的元件224，且一區塊202中之其他八個列206可包括計數器作為特殊目的元件224。

在一實例中，特殊目的元件224包括一計數器(本文中亦被稱作計數器224)。在一實例中，計數器224包含一12位元可程式化遞減計數器。該12位元可程式化計數器224具有一計數輸入、一重設輸入，及零計數輸出。該計數輸入在被確證時使計數器224之值遞減一。該重設輸入在被確證時使計數器224載入來自相關聯之暫存器的初始值。對於該12位元計數器224，可載入高達12位元之數目作為初始值。當計數器224之值遞減至零(0)時，確證該零計數輸出。計數器224亦具有至少兩種模式，脈衝及保持。當將計數器224設定為脈衝模式時，在計數器224遞減至零時的第一時脈循環期間確證零計數輸出，且在接下來之時脈循

環處，即使確證了計數輸入，仍不再確證零計數輸出。此狀態繼續，直至藉由重設輸入被確證而重設計數器224為止。當將計數器224設定為保持模式時，在計數器224遞減至零時之第一時脈循環期間確證零計數輸出，且在確證了計數輸入時保持為經確證的，直至藉由重設輸入被確證而重設計數器224為止。

圖5說明GOT 210之實例。GOT 210包括第一SME 204及第二SME 205，其具有輸入214、216且具有其耦接至OR閘230及3比1多工器242之輸出226、228。該3比1多工器242可經設定以將GOT 210之輸出218耦接至第一SME 204、第二SME 205或OR閘230。可使用OR閘230將輸出226、228耦接在一起以形成GOT 210之共同輸出218。在一實例中，如上文所論述，第一SME 204及第二SME 205展現對等性，其中第一SME 204之輸入214可耦接至列互連導體222中之一些，且第二SME 205之輸入216可耦接至其他列互連導體222。在一實例中，可藉由設定開關240中之任一者或兩者來使GOT 210內之兩個SME 204、205級聯及/或迴圈回至其自身。可藉由將SME 204、205之輸出226、228耦接至另一SME 204、205之輸入214、216而使SME 204、205級聯。可藉由將輸出226、228耦接至其自身之輸入214、216而使SME 204、205迴圈回至其自身。因此，第一SME 204之輸出226可不耦接至第一SME 204之輸入214及第二SME 205之輸入216中之任一者，或可耦接至該等輸入中之一者或兩者。

在一實例中，一狀態機元件204、205包含與偵測線234並聯耦接之複數個記憶體胞232、諸如動態隨機存取記憶體(DRAM)中所常用之彼等記憶體胞。一種此類記憶體胞232包含可設定為一資料狀態之記憶體胞，該資料狀態諸如對應於高或低值(例如，1或0)之狀態。記憶體胞232之輸出耦接至偵測線234，且記憶體胞232之輸入基於資料串流線236上之資料而接收信號。在一實例中，解碼資料串流線236上之輸入以選擇該等記憶體胞232中之一者。選定之記憶體胞232將其所儲存之資料狀態作為輸出提供至偵測線234上。舉例而言，在資料輸入埠209處接收到之資料可提供至一解碼器(未圖示)，且該解碼器可選擇該等資料串流線236中之一者。在一實例中，該解碼器可將一ACSII字元轉換成256位元之1。

因此，當一記憶體胞232設定至高值且該資料串流線236上之資料對應於記憶體胞232時，該記憶體胞232輸出一高信號至偵測線234。當資料串流線236上之資料對應於記憶體胞232且記憶體胞232設定至低值時，記憶體胞232輸出一低信號至偵測線234。在偵測線234上的來自記憶體胞232之輸出藉由一偵測電路238來感測。在一實例中，一輸入線214、216上之信號將各別偵測電路238設定至一作用中或非作用中狀態。當設定至非作用中狀態時，偵測電路238在各別輸出226、228上輸出低信號，而不管各別偵測線234上之信號。當設定至作用中狀態時，當偵測到來自各別SME 204、205之記憶體胞234中之一者的高信號時，

偵測電路238在各別輸出線226、228上輸出高信號。當在作用中狀態時，當來自各別SME 204、205之所有記憶體胞234的信號為低時，偵測電路238在各別輸出線226、228上輸出低信號。

在一實例中，SME 204、205包括256個記憶體胞232，且每一記憶體胞232耦接至一不同之資料串流線236。因此，SME 204、205可經程式化以當資料串流線236中之選定的一或多者上面具有高信號時輸出一高信號。舉例而言，SME 204可使第一記憶體胞232(例如，位元0)設定為高且使所有其他記憶體胞232(例如，位元1-255)設定為低。當各別偵測電路238處於作用中狀態時，當對應於位元0之資料串流線236上面具有高信號時，SME 204在輸出226上輸出一高信號。在其他實例中，SME 204可經設定，以當藉由將適當之記憶體胞232設定至高值而使多個資料串流線236中之一者上面具有高信號時輸出一高信號。

在一實例中，可藉由自相關聯之暫存器讀取位元來將記憶體胞232設定至高值或低值。因此，可藉由將由編譯程式產生之映像儲存至暫存器中且將暫存器中之位元載入至相關聯之記憶體胞232中來程式化SME 204。在一實例中，由該編譯程式產生之映像包括高及低(例如，1及0)位元之二進位映像。該映像可藉由將SME 204、205級聯而程式化FSM引擎200以操作為FSM。舉例而言，可藉由將偵測電路238設定至作用中狀態而將第一SME 204設定至作用中狀態。第一SME 204可經設定，以當對應於位元0之資料串流

線236上面具有高信號時輸出一高信號。第二SME 205最初可設定至非作用中狀態，但可經設定以在作用中時當對應於位元1之資料串流線236上面具有高信號時輸出一高信號。可藉由設定第一SME 204之輸出226以耦接至第二SME 205之輸入216來將第一SME 204與第二SME 205級聯。因此，當在對應於位元0之資料串流線236上感測到高信號時，第一SME 204在輸出226上輸出一高信號且將第二SME 205之偵測電路238設定至作用中狀態。當在對應於位元1之資料串流線236上感測到高信號時，第二SME 205在輸出228上輸出一高信號以啟動另一SME 205或用於來自FSM引擎200之輸出。

圖6說明供編譯程式用以將原始程式碼轉換成經組態以程式化平行機之映像的方法600之實例。方法600包括：將該原始程式碼剖析成一語法樹(區塊602)；將該語法樹轉換成一自動機(區塊604)；最佳化該自動機(區塊606)；將該自動機轉換成一接線對照表(區塊608)；將該接線對照表置放於硬體上(區塊610)；對該接線對照表進行路徑選擇(區塊612)；及發佈所得映像(區塊614)。

在一實例中，該編譯程式包括一應用程式設計介面(API)，該API允許軟體開發者產生用於在FSM引擎600上實施FSM之映像。該編譯程式提供用以將該原始程式碼中之一組輸入之正規表達式轉換成經組態以程式化FSM引擎600之映像的方法。該編譯程式可藉由用於具有馮諾依曼架構之電腦之指令來實施。此等指令可使該電腦上之處理

器實施該編譯程式之功能。舉例而言，該等指令在藉由該處理器執行時可使該處理器對可由該處理器存取之原始程式碼執行如區塊602、604、606、608、610、612及614中所描述之動作。具有馮諾依曼架構之實例電腦展示於圖16中且在下文描述。

在一實例中，原始程式碼描述用於識別一群符號內之符號的型樣之搜尋字串。為了描述該等搜尋字串，該原始程式碼可包括複數個正規表達式(regex)。正規表達式可為用於描述符號搜尋型樣之字串。正規表達式廣泛用在各種電腦領域中，諸如程式設計語言、文字編輯器、網路安全及其他領域。在一實例中，由編譯程式支援之正規表達式包括用於無結構資料之搜尋的搜尋準則。無結構資料可包括以下資料，其為自由形式的且無索引施加於資料內之字。字可包括資料內之任何位元組組合，可列印及不可列印的。在一實例中，編譯程式可支援用於實施正規表達式之多種不同的原始程式碼語言，包括Perl(例如，Perl相容正規表達式(PCRE))、PHP、Java及.NET語言。

返回參看圖6，在區塊602處，編譯程式可剖析原始程式碼以形成有關係連接之運算子的配置，其中不同類型之運算子對應於藉由原始程式碼實施之不同功能(例如，藉由原始程式碼中之正規表達式實施的不同功能)。剖析原始程式碼可產生該原始程式碼之一般表示。在一實例中，該一般表示包含該原始程式碼中的呈被稱為語法樹之樹形圖形式的正規表達式之經編碼表示。本文中所描述之實例將

該配置稱為語法樹(亦被稱為「抽象語法樹」)，然而，在其他實例中，可使用具體語法樹或其他配置。

由於如上文所提及，編譯程式可支援多種語言之原始程式碼，因此，不管是何種語言，剖析將該原始程式碼轉換成非語言特定表示，例如語法樹。因此，藉由編譯程式進行之其他處理(區塊604、606、608、610)可自一共同輸入結構來起作用，而不管原始程式碼之語言。

如上所述，該語法樹包括有關係連接之複數個運算子。語法樹可包括多種不同類型之運算子。亦即，不同運算子可對應於藉由原始程式碼中之正規表達式實施的不同功能。

在區塊604處，將該語法樹轉換(例如，變換)成自動機。在一實例中，該自動機包含一FSM之軟體模型，且可因此分為確定性或非確定性。確定性自動機在給定時間具有單一執行路徑，而非確定性自動機具有多個同時之執行路徑。一自動機包括可藉由節點表示之複數個狀態。為了將一語法樹轉換成自動機，將語法樹中之運算子及運算子之間的關係轉換成狀態(藉由節點表示)，其具有自動機中之狀態之間的轉變(例如，藉由有向邊表示)。在一實例中，可部分基於FSM引擎200之硬體來轉換該自動機。

在一實例中，用於自動機之輸入符號包括字母、數字0-9及其他可列印字元之符號。在一實例中，該等輸入符號藉由位元組值0至255(包括在內)來表示。在一實例中，一自動機可表示為有向圖，其中該圖之節點對應於狀態集

合。在一實例中，被一自動機接受(例如，匹配)之資料為在按順序地輸入至該自動機中時將達到最終狀態的所有可能字元資料之集合。被該自動機接受之資料中的每一符號沿自開始狀態至一或多個最終狀態之路徑而行。

在一實例中，該自動機包含通用狀態以及特殊目的狀態。該等通用狀態及特殊目的狀態對應於藉由一目標器件支援之通用元件及特殊目的元件，編譯程式為該目標器件產生機器碼。不同類型之目標器件可支援不同類型之通用元件以及一或多種不同類型之特殊目的元件。通用元件通常可用以實施廣泛範圍之功能，而特殊目的元件通常可用以實施更窄範圍之功能。然而，在一實例中，一特殊目的元件可在其窄範圍之功能內達成(例如)較大效率。因此，一特殊目的元件可用以(例如)減少在目標器件中實施某些功能所需的機器循環或機器資源。在一些實例中，該目標器件僅支援特殊目的元件，其中支援多種不同類型之特殊目的元件。

在編譯程式產生用於FSM引擎200之機器碼的實例中，通用狀態可對應於SME 204、205，且通用狀態因此在本文中被稱作「SME狀態」。此外，當編譯程式產生用於FSM引擎600之機器碼時，特殊目的狀態可對應於計數器224且因此在本文中被稱作「計數器狀態」。在一實例中，自動機中之SME狀態1:1映射至FSM引擎200中之SME(例如，SME 204、205)，不映射至SME的自動機之開始狀態除外。計數器224可能或可能不1:1映射至計數器狀態。

在一實例中，在該自動機中可使用輸入符號範圍外之特殊轉變符號。此等特殊轉變符號可用以(例如)使得能夠使用特殊目的元件224。此外，特殊轉變符號可用以提供在不同於輸入符號之其他者上發生的轉變。舉例而言，一特殊轉變符號可指示當啟用第二狀態及第三狀態時將啟用(例如，轉變至)第一狀態。因此，當啟動第二狀態及第三狀態時啟動第一狀態，且至該第一狀態之轉變並非直接取決於輸入符號。顯著地，指示在啟用第二狀態及第三狀態時將啟用第一狀態的特殊轉變符號可用以將(例如)藉由布林邏輯執行之布林AND函數表示為特殊目的元件224。在一實例中，可使用一特殊轉變符號來指示一計數器狀態已達到零，且因此轉變至一下游狀態。

在一實例中，自該語法樹產生之自動機為均齊自動機(homogeneous automaton)。一均齊自動機為對一般自動機定義之限制。該限制要求進入一狀態之所有轉變必須發生在同一(些)輸入符號上。均齊自動機滿足以下條件：對於任兩個狀態， q_1 及 q_2 ，若 $r \in \delta(q_1) \cap \delta(q_2)$ ，則表示 $S_1 = \{a | a \in \Sigma, r \in \delta(q_1, a)\}$ ， $S_2 = \{a | a \in \Sigma, r \in \delta(q_2, a)\}$ 。 S_1 為允許 q_1 轉變至 r 之符號的集合；且 S_2 為允許 q_2 轉變至 r 之符號的集合。在此， $S_1 = S_2$ ，亦即，若狀態 q_1 及狀態 q_2 均轉變至狀態 r ，則均齊限制為轉變必須發生在同一(些)符號上。

關於語法樹轉換成自動機之額外細節，參見下文參看圖7所作之論述。

在區塊606處，在建構了自動機之後，最佳化該自動機

以尤其減少其複雜性及大小。可藉由組合冗餘狀態來最佳化該自動機。

在區塊608處，將該自動機轉換成一接線對照表。將該自動機轉換成一接線對照表將該自動機之狀態映射至FSM引擎200之一硬體元件(例如，SME 204、205、GOT 210、特殊目的元件224)的例項，且判定該等例項之間的連接。在一實例中，該接線對照表包含複數個例項，每一例項對應於(例如，表示)FSM引擎200之一硬體元件。每一例項可具有用於連接至另一例項之一或多個連接點(本文中亦被稱作「埠」)。該接線對照表亦包含該等例項之埠之間的複數個連接，該複數個連接對應於(例如，表示)用以耦接對應於該等例項之硬體元件的導體。在一實例中，該接線對照表包含對應於不同類型之硬體元件的不同類型之例項。舉例而言，該接線對照表可包括對應於通用硬體元件之通用例項及對應於特殊目的硬體元件之特殊目的例項。作為一實例，通用狀態可轉換成通用例項，且特殊目的狀態可轉換成特殊目的例項。在一實例中，該等通用例項可包括用於一SME 204、205之一SME例項及用於包含一群SME之硬體元件之一SME群組例項。在一實例中，該SME群組例項包括對應於GOT 210之GOT例項；然而，在其他實例中，該SME群組例項可對應於包含一群三個或三個以上SME之硬體元件。特殊目的例項可包括用於計數器224之計數器例項及用於邏輯元件224之邏輯例項。由於一GOT 210包括兩個SME 204、205，因此一GOT例項含有兩

個 SME 例項。

為產生接線對照表，將自動機中之狀態轉換成接線對照表中之例項，除了開始狀態不具有對應例項之外。將 SME 狀態轉換成 GOT 例項，且將計數器狀態轉換成計數器例項。另外，針對自對應於第一例項之狀態至對應於第二例項之狀態的轉變產生自第一例項至第二例項之對應連接。由於 FSM 引擎 200 中之 SME 204、205 分組至被稱作 GOT 210 之對中，因此編譯程式可將 SME 狀態分組至 GOT 例項中之對中。歸因於 GOT 210 之實體設計，並非所有 SME 例項可一起成對以形成 GOT 210。因此，編譯程式判定在 GOT 210 中可將哪些 SME 狀態一起映射，且接著基於該判定將 SME 狀態成對至 GOT 例項中。下文參看下文之圖 15A 及圖 15B 來論述關於將自動機轉換成接線對照表的實例方法的額外細節。

在區塊 610 處，一旦已產生接線對照表，便置放該接線對照表以針對該接線對照表之每一硬體元件例項選擇該目標器件之一特定硬體元件(例如，SME 204、205、其他元件 224)。根據本發明之一實施例，置放基於針對硬體元件之一般輸入及輸出約束來選擇每一特定硬體元件。

置放可為困難之問題且通常藉由使用試探法來解決。其可使用諸如以下各者之方法來執行：力導向技術、分割技術、模擬退火，或上述技術之組合。

在一實例中，可使用兩種方法來解決大組合最佳化問題；此等方法為模擬退火及多級超圖分割。此等方法之間

的權衡為精確度對速度。模擬退火可產生極高品質之置放，但極耗費中央處理單元(CPU)時間。相比之下，超圖分割可快多個數量級，但傾向於產生不及最佳之置放。在一實例中，模擬退火可用以確保滿足目標硬體器件之需要的高品質置放。在另一實例中，超圖分割可用作第一級，其後跟著模擬退火操作以細化由超圖分割級產生之置放。在某一實例中，使用模擬退火與多級超圖分割之組合以便利利用每一次試探法之強度。

在區塊612處，對經置放之接線對照表進行路徑選擇以判定可程式化開關(例如，區塊間開關203、區塊內開關208及列內開關212)之設定以便將選定硬體元件耦接在一起以達成接線對照表所描述之連接。在一實例中，該等可程式化開關之設定藉由判定FSM引擎200之將用以連接選定之硬體元件的特定導體及該等可程式化開關之設定來判定。路徑選擇可能要考量比區塊610處之置放多的在硬體元件之間的連接的特定限制。因此，路徑選擇可調整如由全域置放判定的硬體元件中之一些的位置，以便在給出對FSM引擎200之導體的實際限制的情況下進行適當連接。

一旦對接線對照表進行置放及路徑選擇，便可將所置放及經路徑選擇之接線對照表轉換成用於程式化FSM引擎200之複數個位元。該複數個位元在本文中被稱作映像。

在一些實例中，在區塊608處將自動機轉換成接線對照表之前，將該自動機分裂成多個較小之自動機，且在區塊608處將每一較小之自動機個別地轉換成接線對照表。由

於在區塊610處之置放的複雜性隨著例項數目增加而增加，因此將自動機分裂成複數個較小之自動機及將較小之自動機轉換成個別接線對照表可提供用於在區塊610及區塊612處進行置放及路徑選擇的較小之接線對照表。因此，置放較小之接線對照表可幫助減少判定可接受之組態所需之時間。在一實例中，使用圖論將自動機分裂成多個較小之自動機。可接著將每一較小之自動機個別地轉換成一接線對照表(區塊608)，且置放(區塊610)於一經指派區域內。因此，可劃分可用區域並將其指派至不同接線對照表，其中每一接線對照表個別地置放於其經指派之部分中。如上所述，可用區域之一些部分可保持為未經指派的，且因此可用於置放其他接線對照表。在一實例中，由較小之自動機形成的每一接線對照表可具有平行地判定之組態以便減少總處理時間。

在區塊614處，藉由編譯程式發佈一映像。該映像包含用於程式化FSM引擎200之特定硬體元件及/或可程式化開關的複數個位元。在該映像包含複數個位元(例如，0及1)之實施例中，該映像可被稱作二進位映像。該等位元可載入至FSM引擎200上以程式化SME 204、205、特殊目的元件224及可程式化開關之狀態，以使得經程式化之FSM引擎200實施具有藉由原始程式碼描述之功能性的FSM。置放(區塊610)及路徑選擇(區塊612)可將在FSM引擎200中之特定位置處的特定硬體元件映射至自動機中之特定狀態。因此，該映像中之位元可程式化該等特定硬體元件及/或

可程式化開關以實施所要功能。在一實例中，可藉由將機器碼儲存至電腦可讀媒體來發佈該映像。在另一實例中，可藉由在顯示器件上顯示該映像來發佈映像。在又一實例中，可藉由將該映像發送至另一器件(諸如，用於將該映像載入至FSM引擎200上之程式設計器件)來發佈映像。在再一實例中，可藉由將該映像載入至平行機(例如，FSM引擎200)上來發佈映像。

在一實例中，可藉由將來自映像之位元值直接載入至SME 204、205及其他硬體元件224或藉由將映像載入至一或多個暫存器中且接著將來自暫存器之位元值寫入至SME 204、205及其他硬體元件224來將映像載入至FSM引擎200上。在一實例中，FSM引擎200之硬體元件(例如，SME 204、205、其他元件224、可程式化開關203、208、212)經記憶體映射，以使得一電腦(例如，耦接至該電腦或與該電腦整合之程式設計器件)可藉由將映像寫入至一或多個記憶體位址而將映像載入至FSM引擎200上。

圖7說明可包括於供編譯程式用以將語法樹轉換成自動機的實例方法604中之各種額外操作。方法604可包括用於插入符號(710)、處理該語法樹以在目標器件內操作(720)、對語法樹元素分類(730)及替換非確定性運算子(740)的操作。圖7內所描繪之方法604說明可在一或多個語法樹轉換成一或多個自動機期間執行的各種操作。圖7內所描繪之操作的次序僅為說明性的；可按各種次序來執行該等操作。另外，在某些實例中，可使用不同操作組合。

在一實例中，方法700將特殊轉變符號插入至自動機中，同時將語法樹轉換成自動機。在一種此類實例中，特殊轉變符號對應於不1:1映射至自動機狀態之運算子。如上文所提及，可為布林運算、計數器及資料結束函數以及其他者保留特殊轉變符號。

在720處，方法700可包括用以鑒於目標硬體器件之約束來調整語法樹的操作。在一實例中，該目標硬體器件(例如，FSM引擎200)之約束可將限制強加於自動機之結構上。在此等約束強加限制之情形中，編譯程式可包括在轉換級處用以調整在自動機內產生之狀態及/或轉變以便遵照硬體約束的操作。

在730處，方法700可包括用以使用一組屬性對每一語法樹分類的操作。在一實例中，可使用諸如Glushkov之方法的標準技術中之一者來對操作分類。

在740處，方法700可包括用等效之確定性運算子替換語法樹之非確定性運算子的操作。在一實例中，可使用計數器結合標準可程式化元件來實施某些類型之非確定性運算子，諸如迴圈。在一實例中，若該非確定性運算子不適合於用特殊目的硬體元件(諸如，計數器)來實施，則可展開非確定性運算子。展開運算子可藉由串行化對應於非確定性運算子之狀態的所有可能組合來達成。

某些正規表達式在轉換成自動機時可導致大數目之狀態。大數目之狀態可使用大數目之通用元件102來實施。為了減少狀態之數目且因此減少所使用之通用元件102之

數目，可使用特殊目的硬體元件112來實施某些正規表達式。舉例而言，在轉換成標準通用元件102時可能需要大量狀態之一個正規表達式為一量化表達式。一量化表達式對應於重複一或多個表達式許多次之迴圈結構。可展開一量化表達式並藉由大量通用狀態連續地實施。然而，在一實例中，可使用特殊目的硬體元件(例如，其他元件112)(諸如，計數器)來利用量化表達式中之重複表達式以減少用以實施該量化表達式之狀態的數目。

量化在此項技術中為眾所周知的，且用以描述重複型樣。作為一實例，「 $A(B) \{n1, n2\}C$ 」為一般正規表達式，其中A、B及C為子表達式，且「 $(B)\{n1, n2\}$ 」包含一量化。如本文中所描述，使用大寫字母來表示正規表達式或正規表達式之一部分(例如，子表達式)。可將雙引號添加到正規表達式或子表達式周圍以避免混淆。因此，描述一表達式之大寫字母可對應於用於多個輸入符號之搜尋字串。舉例而言，表達式「A」可對應於輸入字串「abbc」。

此外，應理解，術語表達式及子表達式在本文中僅用於關係描述(例如，子表達式為表達式之一部分)，且術語表達式及子表達式不應限於任何特定長度、語法或字元數目。詳言之，原始程式碼可包括大量字元(包括元字元及搜尋字元)，其中整組字元或其任何個別部分可被視為一「表達式」。舉例而言，以下各者中之每一者可被視為一表達式：「 $a(bb|d?)\{5, 20\}c$ 」、「 $(b)\{0, 10\}$ 」、「 $(b|d)$ 」及「b」。

將一量化以正規表達式表達為「 $(B)\{n1, n2\}$ 」，其中 B 為子表達式，且 $n1$ 及 $n2$ 為指定允許前述子表達式出現多少次之整數。 B 在本文中被稱作為重複子表達式，因為 B 為重複了由 $n1$ 及 $n2$ 指定之次數的子表達式。為了匹配量化「 $(B)\{n1, n2\}$ 」，必須匹配重複子表達式 B $n1$ 至 $n2$ 數目次。舉例而言，正規表達式「 $(B)\{5, 7\}$ 」將要求子表達式 B 匹配5、6或7次。在正規表達式「 $A(B)\{n1, n2\}C$ 」中，子表達式 A 在本文中被稱作為驅動表達式，因為子表達式 A 在匹配時轉變至一量化。另外，為了繼續重複及遞增針對該量化之計數，必須連續地匹配該量化之重複子表達式。亦即，當一重複子表達式在該量化之給定迴圈期間不匹配時，該量化結束。在一實例中，符號「？」亦對應於量化，其中在「？」之前的符號可識別一次或零次。

當目標器件為FSM引擎200時，方法800可識別某些量化並將該等量化映射至FSM引擎200上之計數器224。藉由計數器224實施某些量化可導致優於藉由狀態機元件204、205實施量化的效率。因此，可簡化針對FSM引擎200之自動機及所得映像。舉例而言，實施量化之語法樹之部分可能需要大量SME 204、205來實施。然而，在一實例中，此等量化中之一些可使用計數器224來實施，計數器224具有比SME 204、205所需要之狀態少的狀態。

在區塊802處，編譯程式識別語法樹之對應於可能藉由FSM引擎200中之計數器224實施的量化之部分。若該語法樹之部分不對應於一量化，則方法800前進至區塊803，在

區塊803處，將該部分轉換成用於藉由SME 204、205實施之通用狀態。若該語法樹之部分確實對應於一量化，則進一步分析該量化以判定是否可藉由計數器224來實施所識別之部分。

在作出是否可能藉由計數器224來實施一量化之判定之前，若寫成 $\mathcal{L}(B)$ 的型樣「B」之語言(亦即，「B」匹配之所有字串)包括空字串，則「 $B\{n1, n2\}$ 」之量化重寫為「 $B'\{0, n2\}$ 」，其中 B' 為 B 之非空字串版本， $\mathcal{L}(B')=\mathcal{L}(B)-\Phi$ 。舉例而言，「 $(bc|)\{10, 20\}$ 」可重寫為「 $(bc)\{0, 20\}$ 」，因為此等正規表達式接受正好相同之資料。接著，對於一給定之量化 $B\{n1, n2\}$ ，根據以下條件，該量化可能藉由一計數器來實施(方法前進至區塊804)或者藉由SME來實施且無計數器(方法前進至區塊808)：

1)當 $(n1=0, n2=-1)$ 時，將以SME 204、205來展開該量化且無計數器224(區塊808)。在此，不需要計數器224。

2)當 $(n1=1, n2=-1)$ 時，將以SME 204、205來展開該量化且無計數器224(區塊808)。在此，不需要計數器224。

3)當 $(n1 > 1, n2=-1)$ 時，該量化將分裂成兩個正規表達式 $B\{n1-1\}$ 及 B^+ ，因為 $B\{n, -1\}$ 等於 $B\{n1-1\}B^+$ 。可接著可能藉由計數器來實施量化 $B\{n1-1\}$ (區塊804)，而將以SME 204、205來實施 B^+ 且無計數器224(區塊808)。對於 B^+ ，不需要計數器224。

4)當 $(n1=0, n2 > 0)$ 時，將該量化修改成 $(B\{1, n2\})?$ ，因為 $(B\{1, n2\})?$ 等於 $B\{0, n2\}$ 。可接著可能藉由計數器224來

實施不可空 $B\{1, n2\}$ (區塊 804)。

5) 當 $(n1 > 0, n2 > 0)$ 時，可能藉由計數器 224 將該量化實施為 $B\{n1, n2\}$ (區塊 804)。

總言之，在不修改之情況下可能藉由計數器 224 實施 (區塊 804) 之量化可被寫成 $B\{n1, n2\}$ ，其中 B 為不可空的， $n1 > 0$ 、 $n2 > 0$ 且 $n1 \leq n2$ 。

在區塊 804 處，一旦編譯程式已識別了可能藉由計數器 224 實施之量化，該編譯程式便判定語法樹之對應於經識別部分的一部分是否為確定性的。當該經識別部分為確定性時，可藉由一或多個計數器 224 來實施該經識別部分，且方法 800 前進至區塊 806 及 807，在區塊 806 及 807 處，將該經識別部分轉換成一或多個計數器狀態以及一或多個 SME 狀態。當該經識別部分為非確定性時，不使用計數器 224 來實施該經識別部分，且方法 800 前進至區塊 808，在區塊 808 處，如下文參看圖 13 所描述，使用一或多個 SME 狀態來展開該經識別部分。

大體上，區塊 806 及區塊 808、810 對應於用以將量化轉換成自動機的兩種方式。在區塊 806 處，使用一或多個計數器狀態來轉換該量化，可能結合一或多個 SME 狀態以將該量化實施為一迴圈。在區塊 808、810 處，藉由「展開」該量化來轉換該量化，其包括使用 SME 狀態且無計數器狀態。展開包含藉由非量化語法來重寫該量化。舉例而言，正規表達式「 $(b|c)\{1,2\}$ 」可展開為「 $(b|c)(b|c)?$ 」。展開之優點包括 (1) 所得自動機為有向非循環圖 (DAG) 且可易於分

析及實施，及(2)所得自動機可藉由通用元件(尤其係狀態機元件)而非特殊目的元件來實施。然而，用以實施展開量化之通用狀態且因此狀態機元件之數目與 $n1$ 及 $n2$ 成線性關係。因此，當 $n1$ 或 $n2$ 為大數目時，狀態之數目可為大的。詳言之，現實資源為有限的，因此，在一些實例中，此展開技術僅用於有限類別之量化。在下文關於區塊808、810及圖13A至圖13C來提供關於展開量化之更多細節。

然而，當目標器件具有經設計以實施計數功能之特殊目的元件(諸如，計數器224)時，在某些情況下可避免展開。此方法之優點為在自動機中需要重複表達式之較少複本，且複本之數目獨立於 $n1$ 及 $n2$ 。因此，可節省顯著資源。舉例而言，可使用一或多個計數器224來藉由用重複表達式及該一或多個計數器224產生一迴圈來實施量化。每當重複表達式匹配，計數器224便可遞增(或遞減)。可接著重新啟動該(等)重複表達式以搜尋另一匹配。當計數器224已遞增(或遞減)了等於該量化所規定之次數時，計數器224可在該量化之後啟動該(等)狀態。因此，可藉由較少SME 204、205來實施該量化，因為再用了用以實施重複表達式之SME。然而，歸因於整個自動機(例如，對應於整個語法樹)之平行性(亦即，可同時處於作用中之多個狀態)，在一些實例中，計數器224可僅用於對應於整個自動機之確定性部分的量化。

圖9說明轉換成自動機900之正規表達式的實例，該自動

機900使用特殊目的計數器狀態902來實施量化。自動機900對應於正規表達式「 $A(B)\{n1, n1\}C$ 」，其中該量化之兩個計數值(例如， $n1$ 、 $n2$)為相等的。由於該等計數值中之兩者為相等的，因此使用單一計數器224來實施該量化。如圖9中所展示，自動機900可表示為有向圖，其中該圖之節點對應於一組狀態。

正規表達式「 $A(B)\{n1, n1\}C$ 」轉換成若干SME狀態904、906、910、908及計數器狀態902。SME狀態904、906、908、910對應於子表達式「A」、「B」及「C」。SME狀態904、906、910、908可藉由SME 204、205來實施，而計數器狀態902可藉由計數器224來實施。當在FSM引擎200上實施自動機910時，對應於計數器狀態902之計數器224最初載入有值 $n1$ 且經設定以在計數器224中之值達到零時確證零計數輸出。當 $n1$ 等於 $n2$ 時，計數器224可經設定至停止0及脈衝輸出模式，其意謂一旦計數器224之值達到零，計數器224便確證其輸出，且計數器224將維持於零且不發出任何信號直至重設計數器224為止。

自動機900在狀態904處開始且在匹配子表達式「A」後即轉變至狀態906。當處於狀態906處，每當子表達式「B」匹配，便啟動計數器狀態902之IN埠，且計數器狀態902遞減一。另外，每當子表達式「B」匹配，狀態906便啟動其自身以及啟動狀態910。當計數器狀態902達到零時，啟動輸出，且自動機900將接著搜尋子表達式「C」。在接下來之循環中，將出現兩種情形：第一種情形在

「 $\sim B$ 」匹配時出現。當「 $\sim B$ 」匹配時，重設計數器狀態902且將其值往回設定至 $n1$ 。因此，下次子表達式「A」匹配，過程便自狀態904開始。在第二種情形中，狀態906之自迴圈仍在作用中，且在子表達式「B」匹配時繼續觸發計數器902之IN埠。由於計數器狀態902係在脈衝模式下組態，因此儘管狀態906之自迴圈仍處於作用中，但計數器狀態902將不會再次啟動其輸出。

子表達式「B」之否定版本在本文中亦被稱作「 $\sim B$ 」。在一實例中，子表達式「B」之否定版本用以啟動計數器狀態902之重設埠。此係因為，由於「B」係量化「 $(B)\{n1, n1\}$ 」之重複表達式，因此當在輸入處接收到不同於B之任一者(例如，「B」之否定版本)(一旦已啟動了狀態906)，則該量化結束且因此重設計數器。因此，一旦啟動了狀態910，便重設計數器狀態902，且當子表達式「B」之否定版本匹配時，該量化不匹配。在一實例中，使用標準自動機理論來否定重複表達式。

儘管當 $n1$ 等於 $n2$ 時，說明並描述單一計數器狀態224實施一量化，但應認識到，多個計數器224可級聯以考量比藉由單一計數器224支援之數目大的數目。

圖10說明轉換成自動機1000之正規表達式的另一實例，該自動機1000使用複數個特殊目的計數器狀態1002、1004來實施具有量化之正規表達式。自動機1000對應於正規表達式「 $A(B)\{n1, n2\}C$ 」，其中 $n1$ 小於 $n2$ 。使用兩個計數器狀態1002、1004，因為在量化「 $(B)\{n1, n2\}$ 」中 $n1$ 小於

n2。計數器狀態1002、1004經組態至停止0及保持模式，其意謂，當計數器狀態1002、1004達到零時，計數器狀態1002、1004啟動其輸出，且在重設計數器狀態1002、1004之前，計數器狀態1002、1004維持於零且每當啟動IN埠便保持啟動其輸出。在此實例中，自計數器狀態1002至計數器狀態1004之潛時佔用兩個循環。

計數器狀態1002最初設定至n1，且計數器狀態1004最初設定至n2。當子表達式「A」匹配時，該自動機自狀態1006轉變至狀態1008。一旦啟動了狀態1008，每當子表達式「B」匹配，便啟動計數器狀態1002及計數器狀態1004之IN埠。因此，計數器狀態1002及計數器狀態1004遞減一。當計數器狀態1002達到零時，啟動其輸出，且自動機1000接著搜尋子表達式「C」之匹配且啟動狀態1010。一旦子表達式「B」已匹配了n1次，計數器狀態1004之值便為n2-n1。稍後，每當子表達式「B」匹配，便啟動計數器狀態1002之IN埠，且計數器狀態1002之值維持於零，且仍啟動其輸出。同時，計數器狀態1004繼續遞減。當子表達式「B」匹配了n2次時，計數器狀態1004亦達到零，且啟動其輸出，此驅動計數器狀態1002之重設埠。由於計數器狀態1004至計數器狀態1002之潛時為兩個循環，因此計數器狀態1002繼續啟動其輸出至狀態1010。在下一個循環中，自計數器狀態1004之輸出重設計數器狀態1002，且不自計數器狀態1002確證輸出。在接下來之循環中，將出現兩種情形。在第一種情形中，「 $\sim B$ 」匹配。計數器狀態

1002及計數器狀態1004均藉由狀態1012重設，且其值分別設定為 $n1$ 及 $n2$ 。因此，下次狀態1006為作用中且下次子表達式「A」匹配時，啟動狀態1008，且計數器狀態1002、1004再次遞減。在第二種情形中，狀態1008之自迴圈維持於啟動，且啟動計數器狀態1002、1004 IN埠。由於計數器狀態1004繼續啟動其輸出，因此計數器狀態1002繼續重設，且只要狀態1008之自迴圈處於作用中便不啟動其輸出。

另外，在狀態1008處於作用中時子表達式「B」之匹配啟動了狀態1012。一旦狀態1012啟動且「 $\sim B$ 」匹配，則重設計數器狀態1002、1004，且不匹配該量化。使用子表達式「B」之否定版本，因為「B」為量化「 $(B) \{n1, n2\}$ 」之重複表達式。因此，處於狀態1008之表達式「B」可重複地匹配 $n1$ 至 $n2$ 次。儘管說明並描述了單一計數器分別實施下(例如， $n1$)及上(例如， $n2$)臨限值，但應認識到，如熟習此項技術者所知，多個計數器可級聯以考量比藉由單一計數器支援之數目大的數目。

在使用計數器狀態轉換量化之前，在區塊804處，編譯程式判定對應於該量化之自動機是否為確定性的。在一實例中，當表達式滿足下文所論述之無首碼、無重入條件時，該自動機為確定性的。亦即，為了使量化映射至計數器224，該量化應滿足如下文所論述之無首碼及無重入條件。

參考圖10之自動機1000，無重入條件要求自狀態1006至

狀態 1008 之邊不可啟動，而計數器狀態 1002 處於作用中（例如，當計數器狀態 1002 正在計數時）。亦即，判定在已處理該量化時是否可匹配該量化之驅動表達式。匹配驅動表達式意謂，緊跟在該量化之前的狀態將轉變至對應於該量化之狀態。因此，將「重入」該量化，而計數器狀態仍處理一重複表達式。由於在 FSM 引擎 200 之此實例中，計數器 224 在任何給定時間時可僅實施單一迴圈，因此在迴圈已經處理時轉變至一量化可使計數器 224 在給定迴圈期間不準確地計數。

圖 11A 及圖 11B 說明自動機 1100 及 1114，可用以進一步解釋無重入條件。圖 11A 說明對應於語法樹中之一量化的實例自動機 1100，其中編譯程式可進行分析以判定對應於該量化之自動機是否為確定性的。

自動機 1100 對應於正規表達式「 $abb?(b|c)\{1,2\}$ 」且包括開始狀態 1102 及最終狀態 1112、1104。該等最終狀態在圖 11A 中識別為雙圓。最初啟動開始狀態 1102 並在輸入符號「a」之後即刻使其轉變至狀態 1106。在輸入符號「b」時，狀態 1106 轉變至狀態 1108 及狀態 1110。在輸入符號「b」時狀態 1108 轉變至狀態 1110，且在輸入符號「b」或「c」時，狀態 1110 轉變至狀態 1112。在輸入符號「b」或「c」時，自動機 1100 自狀態 1112 轉變至狀態 1104。

自動機 1100 包含正規表達式「 $abb?(b|c)\{1,2\}$ 」之自動機，將檢查該正規表達式是否符合無重入條件。自動機 1114 包含自自動機 1100 之正規表達式「 $abb?(b|c)\{1,2\}$ 」導

出之正規表達式 $SS(\text{「abb?」}, \text{「(b|c)\{2\}」})$ 的自動機。 $SS(M, N)$ 被定義為自 M 、 N 導出之正規表達式。導出步驟包括：1) 串接 M 及 N ，結果表示為「 MN 」。2) 建構「 MN 」之自動機，表示為 $A(MN)$ 。3) 如下地修改 $A(MN)$ ：a) 使 $A(MN)$ 之開始狀態驅動所有其他狀態，及 b) 使對應於「 N 」之所有狀態為最終狀態。最後，4) 將經修改之自動機的正規表達式表示為 $SS(M, N)$ 。 $SS(M, N)$ 之被接受資料由自「 MN 」之任何狀態開始且在 N 之任何狀態處結束的子字串組成。

無重入條件可定義如下。給出具有量化「 $AB\{n_1, n_2\}C$ 」之正規表達式，無重入條件需要 $\mathcal{L}(SS(A, B\{n_1, n_2\})) \cap \mathcal{L}(A) = \emptyset$ 。換言之，一旦子表達式「 A 」匹配且計數器狀態 1002 開始計數，則為了滿足無重入條件，將不再次啟動自狀態 1006 至狀態 1008 之邊，直至進行「 $B\{n_1, n_2\}$ 」(匹配或失敗)為止。舉例而言，「 abb 」 $\in \mathcal{L}(\text{「abb?」}) \cap \mathcal{L}(SS(\text{「abb?」}, \text{「(b|c)\{2\}」}))$ ，且因此藉由計數器 224 將不會正確地實施「 $abb?(b|c)\{1, 2\}$ 」。

現在參看圖 12，將參考自動機 1200 來解釋無首碼條件。無首碼條件規定 $\mathcal{L}(B)$ 之任何字串不應為 $\mathcal{L}(B)$ 之另一字串的首碼，此將保證 B 不會使計數器計數多於一次。換言之，當一量化之第一重複子表達式為該量化之第二重複子表達式之首碼時，該量化不實施為(且因此轉換為)計數器 224。正式之陳述為：對於所有 $l_i, l_j \in \mathcal{L}(B)$ ， $l_i \neq l_j$ ，吾人要求 $\{l_i.\}\cap\{l_j.\} = \emptyset$ 。

舉例而言，正規表達式「 $a(b|bc)\{3\}$ 」不滿足無首碼條

件。因此，將不使用計數器狀態來轉換正規表達式「 $a(b|bc)\{3\}$ 」，且因此不藉由計數器224來實施。實情為，不藉由任何計數器狀態來將正規表達式「 $a(b|bc)\{3\}$ 」轉換成通用狀態。

若藉由計數器224來實施正規表達式「 $a(b|bc)\{3\}$ 」，則將錯誤地匹配輸入「abbc」。舉例而言，自動機1200為使用計數器狀態1212對正規表達式「 $a(b|bc)\{3\}$ 」之假設轉換的結果。如下文所描述，此轉換導致計數器狀態1212之不正確執行。最初啟動狀態1202，且在輸入「a」處，狀態1202啟動狀態1204。在狀態1204啟動之情況下，在輸入「b」處，狀態1204啟動狀態1206、1208且再啟動自身，即狀態1204。而且，在輸入「b」處，狀態1204啟動計數器1212之IN埠，其中計數器狀態1212之初始值處於3且接著減少至2。在狀態1204、1206及1208啟動之情況下，在另一輸入「b」處藉由狀態1204再次啟動計數器狀態1212之IN埠，且計數器狀態1212中之值減少至1。此時，啟動狀態1204、1206及1208。接著，輸入值「c」使計數器狀態1212之IN埠藉由狀態1208啟動以將計數器1212中之值減少至0。在計數器1212中之值處於零之情況下，啟動輸出，且啟動狀態1214，此指示一匹配。然而，此匹配為一錯誤肯定，因為在序列「abbc」不滿足正規表達式「 $a(b|bc)\{3\}$ 」時，輸入「abbc」已造成一匹配。因此，正規表達式「 $a(b|bc)\{3\}$ 」不滿足無首碼條件，且不應使用計數器狀態來轉換且不應藉由計數器224來實施。

若在區塊 804 處，該量化滿足無首碼條件及無重入條件，則在區塊 806 處使用特殊目的計數器狀態來轉換該量化。可如上文關於圖 10 及圖 11 所描述般轉換該量化。然而，若該量化不滿足無首碼或無重入條件，則在區塊 808、810 處藉由展開該量化且將該量化轉換成通用狀態且無計數器狀態 224 來轉換該量化。因此，藉由 SME 204、205 且無計數器 224 來實施該量化。

返回參看區塊 808，展開可與單一數目個迴圈匹配的量化以形成具有複數個連續地連接之重複子表達式的自動機。具有單一數目個迴圈之量化對應於 $n1$ 等於 $n2$ 之量化。舉例而言，量化「 $B\{n1\}$ 」可展開為「 $BB...B$ 」，具有 B 之 $n1$ 個複本。

在區塊 810 處，當 $n1$ 不等於 $n2$ 時，且當 $n1$ 等於 1 且 $n2$ 大於 1 時，展開可與多個數目個迴圈匹配的量化。當 $n1$ 大於 1 時，將該量化分裂成可與 $n1-1$ 數目個迴圈匹配的第一量化及可與一個至 $n2-n1+1$ 數目個迴圈匹配的第二量化。舉例而言，量化 $B\{n1, n2\}$ (其中 $n1 > 1$ 、 $n2 > 1$ 且 $n1 < n2$) 可與多個數目個迴圈匹配，詳言之，自 $n1$ 至 $n2$ 數目個迴圈。此量化 $B\{n1, n2\}$ 可分裂成以下量化： $B\{n1-1\}B\{1, n2-n1+1\}$ 。第一量化為可與數目等於 $n1-1$ 之迴圈匹配的重複子表達式 B 。將此第一量化與一第二量化串接，該第二量化具有可與數目為一個至 $n2-n1+1$ 個之迴圈匹配的重複子表達式。如 1302 處所述，展開第一量化 $B\{n1-1\}$ 。

可基於所得自動機之內分支度(in-degree)及/或外分支度

(out-degree)來展開第二量化 $B\{1, n2-nl+1\}$ 。展開一量化可產生具有大內分支度或大外分支度之狀態。在一實例中，內分支度對應於至該自動機之狀態的轉變之數目，且外分支度對應於自該自動機之狀態發生的轉變之數目。因此，可展開該第二量化以控制當將第二量化轉換成自動機時進(內分支度)或出(外分支度)狀態之轉變。舉例而言，可展開該量化以限制每一展開狀態之內分支度使之低於一臨限值數目。舉例而言，限制內分支度可經執行以考量目標器件中之元件的態樣及/或限制。此外，在展開期間限制內分支度可減少編譯程式之後續處理。

在一實例中，當展開量化 $B\{1, n2-nl+1\}$ 時，產生自動機，作為內分支度與外分支度之間的權衡。因此，減少內分支度可增加外分支度，且減少外分支度可增加內分支度。在一實例中，為了展開量化 $B\{1, n2-nl+1\}$ 之迴圈結構，進行至或自展開狀態之許多轉變以使該自動機接受B串接了 k 次之任何字串，其中 $1 \leq k \leq n2-nl+1$ 。控制是否進行至或自展開狀態之轉變可用以控制該自動機之內分支度/外分支度。

儘管將方法800描述為對應於單一量化，但可針對一語法樹內之複數個量化而重複方法800，且可接著將所得之單獨自動機連接至較大自動機中。

圖13A說明自動機1300之一實例，其中展開表達式 $AB\{1, 4\}$ 以最小化內分支度。自經展開以最小化內分支度之量化產生的自動機在本文中亦被稱作為散射型樣。表達

式 $AB\{1, 4\}$ 之散射型樣展開版本直接對應於自動機 1300 所對應於之表達式 $A((((B?)B)?B)?B)$ 。自動機 1300 包括用於量化 $B\{1, 4\}$ 之驅動狀態 1302 及複數個展開狀態 1304-1308，包括該量化之第一狀態 1304 及該量化之最後一個狀態 1308。在一實例中，表達式 A 及 B 中之每一者可對應於未展示之較小自動機的多個狀態。為了最小化自動機 1300 之內分支度，將該量化之轉變指派為自第一狀態 1304 至其他展開狀態 1305-1308 之外轉變。因此，第一狀態 1304 具有大外分支度 (4 個外轉變)，且所有量化狀態 1304-1308 具有小內分支度 (1 或 2 個內轉變)。

圖 13B 說明自動機 1310 之一實例，其中展開表達式 $AB\{1, 4\}$ 以最小化外分支度。自經展開以最小化外分支度之量化產生的自動機在本文中亦被稱作為合併型樣。表達式 $AB\{1,4\}$ 之合併型樣展開版本直接對應於展開表達式 $AB(B(B(B)?))?)$ 。自動機 1308 包括與自動機 1300 相同之狀態 1302、1304-1308，但在狀態 1302、1304-1308 之間具有不同轉變。為了最小化自動機 1310 之外分支度，將該量化之轉變指派為至該量化之最後一個狀態 1308 之內轉變。內轉變係來自展開狀態 1304-1307 中之每一者。因此，所有量化狀態 1304-1308 具有小外分支度 (1 或 2 個外轉變)，但該量化之最後一個狀態 1308 具有大內分支度 (4 個內轉變)。

在一實例中，展開具有一量化之一表達式以限制外分支度或內分支度中之一者使之小於一臨限值。在一實例中，為了展開表達式 $AB\{1, n1\}$ 以將內分支度限制於一臨限值，

可將數目高達該臨限值的該量化 $B\{1, nl\}$ 之轉變指派為至量化 $B\{1, nl\}$ 之最後一個狀態的內轉變，且可將其他轉變指派為針對量化 $B\{1, nl\}$ 之第一狀態的外轉變。相反，為了展開表達式 $AB\{1, nl\}$ 以將外分支度限制於一臨限值，可將數目高達該臨限值的該量化 $B\{1, nl\}$ 之轉變指派為至該量化之第一狀態的外轉變，且可將其他轉變指派為針對該量化 $B\{1, nl\}$ 之最後一個狀態的外轉變。

圖 13C 說明自動機 1312 之另一實例，其中展開表達式 $AB\{1,4\}$ 以限制任何狀態之內轉變使之小於或等於 3。自動機 1312 包括與自動機 1300 及 1308 相同之狀態 1302、1304-1308，但在狀態 1302、1304-1308 之間具有不同轉變。在一實例中，為了限制自動機 1312 之內分支度使之小於或等於三(3)個內轉變，最初將該量化之轉變指派為至該量化 $B\{1, 4\}$ 之最後一個狀態 1308 的內轉變，直至達到三個之極限為止，且將其他轉變指派為自量化狀態 1304-1308 之外轉變。因此，最後一個狀態 1308 以及該量化之其他量化狀態 1304-1307 具有等於或低於 3 之極限的內分支度，且第一狀態 1304 具有為 3 之外分支度。

在其他實例中，可將表達式之內分支度及外分支度設定為彼此成某一比率(例如，1 比 1、2 比 1)。在又一些實例中，可將表達式之內分支度及外分支度設定為彼此成某一比率，直至內轉變或外轉變達到一臨限值為止，且接著可使用另一比率，或可將所有轉變分別指派為內轉變或外轉變。

接線對照表為連接性例項，其中例項對應於硬體元件且有向邊為連接硬體元件之網。若一狀態驅動另一狀態，則此兩個狀態所指派至之SME之間將存在電連接。多數實體器件在硬體組件之間的連接之數目上具有一些限制。為了獲得自一自動機至實體器件之可行映射，必須轉變該自動機，以使得所有狀態之內分支度滿足硬體連接限制。

如上所述，編譯程式基於FSM引擎200之限制(若有)來判定哪些SME可一起分組。因此，對於GOT 210，編譯程式基於GOT 210中之SME 204、205之輸出限制來判定哪些SME可一起成對。

在一種此類實施例中，FSM引擎200對於所有硬體組件具有有限之連接。為了獲得自一自動機至FSM引擎200之可行映射，必須轉變該自動機，以使得所有狀態之內分支度滿足連接限制。

因此，在一實例中，可基於目標器件之硬體約束來限制一自動機之內分支度。在一實例中，兩級方案可用於FSM引擎200以限制SME 204、205之進入。首先，一個級可利用由GOT 210提供之OR閘230，其將藉由將來自SME狀態之輸出配對成單一輸出來使內分支度減少達至多50%。另一組件為布林，其可經組態以提供複合邏輯函數。在此實例中，吾人僅將其視為簡單OR組件。在以下演算法中，吾人以謹慎之方式來估計GOT 210之使用。在該演算法中，當估計內分支度大於FSM引擎200之約束時，一些狀態分裂成多個狀態，或在必要時插入布林OR以減少內分

支度。

狀態分裂之基本理念係將一狀態分裂成多個狀態且將預分裂狀態之進入分配給該等分裂狀態，使得每一分裂狀態之內分支度滿足約束。當自動機為有向非循環圖(DAG)時，該自動機之簡單廣度優先遍歷便可解決該問題。然而，當迴圈(例如，量化)存在時，分裂可使狀態之數目按指數增加或可造成任何可行解決方案均不可能的情形。布林邏輯可幫助減少增加之狀態且因此減輕此情形。在一實施例中，使用布林OR或其等效物來處置迴圈狀況。

分裂狀態之一實例展示於圖 14A 及圖 14B 中。在圖 9A 中所示之實例中，八個狀態 1430 饋入一個狀態 1432，狀態 1432 又饋入兩個狀態 1434。如圖 9B 中所示，如上文所詳述之狀態分裂導致增添兩個新的狀態 1436(C2 及 C3)。然而，現在，三個狀態饋入兩個狀態 1434。

返回參看圖 5，一 GOT 210 具有對 SME 204、205 之輸出限制。詳言之，GOT 210 具有由兩個 SME 204、205 共用之單一輸出 218。因此，一 GOT 210 中之每一 SME 204、205 不可獨立地驅動輸出 218。此輸出限制限制了在一 GOT 例項中哪些 SME 狀態可一起配對。顯著地，驅動(例如，轉變至、啟動)外部 SME 狀態(例如，對應於 GOT 例項外之 SME 的 SME 狀態)之不同集合的兩個 SME 狀態在一 GOT 例項中不可一起成對。然而，此限制不限制該兩個 SME 狀態是否驅動彼此或自迴圈，因為一 GOT 210 可在內部藉由開關 240 提供此功能性。儘管將 FSM 引擎 200 描述為具有對應於

SME 204、205之某一實體設計，但在其他實例中，SME 204、205可具有其他實體設計。舉例而言，SME 204、205可一起分組到SME 204、205之三個或三個以上集合中。另外，在一些實例中，可存在對至SME 204、205之輸入214、216的限制，對來自SME 204、205之輸出226、228可具有或不具有限制。

然而，在任一種情況下，編譯程式基於FSM引擎200之實體設計來判定哪些SME狀態可一起分組。因此，對於GOT例項，編譯程式基於GOT 210中之SME 204、205之輸出限制來判定哪些SME狀態可一起成對。在一實例中，存在五種情形，其中兩個SME狀態可基於GOT 210之實體設計而一起成對以形成GOT 210。

在第一SME狀態及第二SME狀態可在GOT 210中一起成對時的第一種情形出現在第一SME狀態或第二SME狀態皆非最終狀態時，及出現在第一SME狀態及第二SME狀態中之一者不驅動不同於第一SME狀態或第二SME狀態之任何狀態時。作為一實例，當第一狀態轉變至第二狀態時，認為該第一狀態驅動該第二狀態。當此第一種情形出現時，至多第一SME狀態及第二SME狀態中之一者驅動一(些)外部狀態。因此，第一SME狀態及第二SME狀態可一起成對，而不受GOT 210之輸出限制影響。然而，歸因於GOT 210在內部將SME 204、205耦接至彼此的能力，允許第一SME狀態及第二SME狀態驅動彼此及自迴圈以驅動自身。在自動機項中，當 $q1$ 或 $q2$ 皆非最終狀態，且 $\delta(q1) = \{q1, q2\}$

為空時，或當 $\delta(q2) - \{q1, q2\}$ 為空時，第一 SME 狀態(對應於狀態 $q1$)及第二 SME 狀態(對應於狀態 $q2$)可一起成對。

在第一 SME 狀態及第二 SME 狀態可在 GOT 210 中一起成對時的第二種情形出現在第一 SME 狀態或第二 SME 狀態皆非自動機中之最終狀態時，及出現在第一 SME 狀態及第二 SME 狀態驅動相同之外部狀態時。如本文中所使用，外部狀態對應於 GOT 例項外部之狀態，例如，不管 GOT 例項中之第一 SME 狀態及第二 SME 狀態是否驅動彼此或自迴圈。再次，GOT 210 之輸出限制不影響第一 SME 狀態及第二 SME 狀態，因為第一 SME 狀態及第二 SME 狀態驅動相同之外部狀態。而且，歸因於 GOT 210 在內部將 SME 204、205 耦接至彼此的能力，對驅動相同狀態之限制不包括第一狀態及第二狀態是否驅動彼此或自迴圈。使用自動機項，當 $q1$ 或 $q2$ 皆非最終狀態，且 $\delta(q1) - \{q1, q2\} = \delta(q2) - \{q1, q2\}$ 時，第一 SME 狀態(對應於狀態 $q1$)及第二 SME 狀態(對應於狀態 $q2$)可一起成對。

第一 SME 狀態及第二 SME 狀態可在 GOT 210 中一起成對的第三及第四種情形出現在第一 SME 狀態及第二 SME 狀態中之一者為最終狀態且第一 SME 狀態及第二 SME 狀態中之另一者並不驅動任一外部狀態時。亦即，當 $q1$ 為最終狀態，且 $\delta(q2) - \{q1, q2\}$ 為空時，或當 $q2$ 對應於最終狀態且 $\delta(q1) - \{q1, q2\}$ 為空時，第一 SME 狀態(對應於狀態 $q1$)及第二 SME 狀態(對應於狀態 $q2$)可一起成對。由於最終狀態輸出與正規表達式匹配之指示，因此對應於最終狀態之 SME

狀態應能夠獨立使用 GOT 210 之輸出 218 以便指示該匹配。因此，不允許 GOT 210 中之另一 SME 狀態使用輸出 218。

在第一 SME 狀態及第二 SME 狀態可在 GOT 210 中一起成對時的第五種情形出現在第一 SME 狀態及第二 SME 狀態均對應於一自動機中之最終狀態且第一 SME 狀態及第二 SME 狀態均驅動相同之外部狀態時。使用自動機項，當 $q1$ 及 $q2$ 均為最終狀態，且 $\delta(q1) - \{q1, q2\} = \delta(q2) - \{q1, q2\}$ 時，第一狀態（對應於狀態 $q1$ ）及第二 SME 狀態（對應於狀態 $q2$ ）可一起成對。

一旦編譯程式判定一或多個 SME 狀態是否可一起成對，編譯程式便將該等 SME 狀態成對為 GOT 例項。在一實例中，編譯程式按 SME 狀態被判定為能夠成對以形成 GOT 例項的次序來將 SME 狀態成對為 GOT 例項。亦即，一旦兩個特定 SME 狀態被判定為能夠一起成對，此兩個 SME 狀態便可成對為一 GOT 例項。一旦兩個 SME 狀態已成對以形成 GOT 例項，此等成對之 SME 狀態不可用於與其他 SME 狀態成對。此過程可繼續，直至不再有任何 SME 狀態留待成對為止。

在一實例中，編譯程式使用圖論來判定哪些 SME 一起成對為 GOT 例項。由於僅某些 SME 可一起成對，因此一些 SME 成對可導致其他 SME 必須在其自身之 GOT 例項中實施，以致 GOT 例項中之其他 SME 位置未用且因此被浪費。可使用圖論來藉由減少接線對照表之 GOT 例項中之未用 SME 例項的數目來最佳化 GOT 210 中之 SME 利用（例如，減

少未用 SME 之數目)。為了使用圖論，編譯程式首先根據上文所論述之 FSM 引擎 200 之實體設計來判定 SME 狀態之間的所有可能成對。編譯程式接著產生一圖，其中該圖之頂點對應於 SME 狀態，且該圖之邊對應於 SME 狀態之可能成對。亦即，若判定兩個 SME 狀態能夠在一 GOT 例項中一起成對，則用一條邊連接兩個對應之頂點。因此，該圖含有 SME 狀態之所有可能成對。

編譯程式可接著找出該圖之匹配頂點以識別哪些 SME 狀態在一 GOT 210 中一起成對。亦即，編譯程式識別邊(且因此識別頂點對)，使得該圖之匹配頂點之間的兩條邊不會共用一個共同頂點。在一實例中，編譯程式可找出該圖之最大匹配。在另一實例中，編譯程式可找出該圖之最大匹配。最大匹配為含有最大之可能數目的邊的匹配。可存在許多最大匹配。可以多項式時間來解決找出一個圖之最大匹配的問題。

一旦已識別了所有匹配頂點(例如，作為最大匹配)，則將對應於匹配頂點之每一對 SME 狀態映射至一 GOT 例項。將對應於未匹配頂點之 SME 狀態映射至其自身之 GOT 例項。亦即，將對應於未匹配頂點之 SME 狀態映射至 GOT 例項中之 SME 位置中之一者中，且 GOT 例項中之其他 SME 位置未用。因此，給定接線對照表 N 及其匹配頂點 M 之對應集合，所使用之 N 的 GOT 例項之數目等於 $|Q|-1-|M|$ ，其中 Q 為自動機之狀態集合，且「-1」係因為在此實例中自動機之開始狀態並不對應於 SME 狀態。

在一實例中，接線對照表N係使用最少數目之GOT例項由G之最大匹配M來建構。此可藉由以下事例證明：若存在使用更少數目之GOT例項的另一接線對照表N'，則將該對應匹配表示為M'。由於N'之GOT例項之數目等於 $|Q|-1-|M'|$ ，因此吾人得到 $|M| < |M'|$ 。此與M為最大匹配之事實衝突。因此，接線對照表N使用最少數目之GOT例項。

一旦SME狀態成對為GOT例項，便根據自動機中之狀態之間的轉變來連接GOT例項、計數器例項及邏輯例項。由於每一GOT 210具有單一輸出，因此接線對照表中之每一GOT例項具有單一輸出埠以供連接至其他例項。因此，若第一GOT例項中之任一SME狀態驅動第二GOT例項中之一SME狀態，則第一GOT例項之輸出埠耦接至第二GOT例項之輸入。

圖15A及圖15B說明自均齊自動機產生之實例接線對照表1500、1502。接線對照表1500、1502包括SME例項1506、1508、1510、1512及1514。接線對照表1500為非最佳接線對照表之實例。接線對照表1500使用四個GOT例項1516，同時留下三個SME例項1518未用。然而，接線對照表1502為使用圖論識別最大匹配而產生之最佳接線對照表的實例。接線對照表1502使用三個GOT例項1516且具有單一未用SME例項1518。在接線對照表1502中，可藉由GOT例項內部之連接(例如，經由開關240)將例項1510連接至例項1512。

本文中所描述之方法實例可至少部分為機器或電腦實施

的。一些實例可包括編碼有指令之電腦可讀媒體或機器可讀媒體，該等指令可操作以組態一電子器件以執行如上述實例中所描述之方法。此等方法之一實施方案可包括程式碼，諸如微碼、組合語言程式碼、高級語言程式碼或其類似者。此類程式碼可包括用於執行各種方法之電腦可讀指令。程式碼可形成電腦程式產品之部分。另外，程式碼在執行期間或在其他時間時可有形地儲存於一或多個揮發性或非揮發性電腦可讀媒體上。此等電腦可讀媒體可包括(但不限於)硬碟、可抽取式磁碟、可抽取式光碟(例如，壓縮光碟及數位影音光碟)、磁帶盒、記憶體卡或記憶體棒、隨機存取記憶體(RAM)、唯讀記憶體(ROM)及其類似者。

圖16大體上說明具有馮諾依曼架構之電腦1600的實例。在閱讀及理解了本發明之內容之後，一般熟習此項技術者將理解，在一基於電腦之系統中可自電腦可讀媒體載入軟體程式以使之執行以軟體程式定義之功能的方式。一般熟習此項技術者將進一步理解，可使用各種程式設計語言來產生經設計以實施並執行本文所揭示之方法的一或多個軟體程式。可使用物件導向語言(諸如，Java、C++或一或多種其他語言)以物件導向格式來結構化該等程式。或者，可使用程序語言(諸如，組合語言、C等)以程序導向格式來結構化該等程式。軟體組件可使用一般熟習此項技術者所熟知的許多方案中之任一者來通信，諸如應用程式設計介面或過程間通信技術，包括遠端程序呼叫或其他。各個

實施例之教示不限於任何特定程式設計語言或環境。

因此，可實現其他實施例。舉例而言，一製造物件(諸如，電腦、記憶體系統、磁碟或光碟、某其他儲存器件，或任何類型之電子器件或系統)可包括一或多個處理器1602，該一或多個處理器1602耦接至上面儲存有指令1624(例如，電腦程式指令)之一電腦可讀媒體1622，諸如一記憶體(例如，可抽取式儲存媒體，以及包括電、光或電磁導體之任何記憶體)，該等指令在藉由該一或多個處理器1602執行時導致執行相對於上述方法所描述之動作中之任一者。

電腦1600可採取具有處理器1602之電腦系統的形式，該處理器1602直接及/或使用匯流排1608來耦接至許多組件。此等組件可包括主記憶體1604、靜態或非揮發性記憶體1606及大容量儲存器件1616。耦接至處理器1602之其他組件可包括輸出器件1610(諸如，視訊顯示器)、輸入器件1612(諸如，鍵盤)及游標控制器件1614(例如，滑鼠)。用以將處理器1602及其他組件耦接至網路1626之網路介面器件1620亦可耦接至匯流排1608。指令1624可利用許多熟知之傳送協定(例如，HTTP)中之任一者經由網路介面器件1620跨越網路1626來進一步傳輸或接收。耦接至匯流排1608之此等元件中之任一者可不存在，單獨地存在，或以複數形式存在，此取決於待實現之特定實施例。

在一實例中，處理器1602、記憶體1604、1606或儲存器件1616中之一或多者可各自包括指令1624，該等指令1624

在執行時可使電腦1600執行本文所描述之方法中之任何一或多者。在替代實施例中，電腦1600操作為單獨器件或可連接(例如，網路連接)至其他器件。在網路環境中，電腦1600可以伺服器-用戶端網路環境中之伺服器或用戶端器件之能力來操作，或操作為對等(或分散式)網路環境中之對等器件。電腦1600可包括個人電腦(PC)、平板PC、機上盒(STB)、個人數位助理(PDA)、蜂巢式電話、web器具、網路路由器、交換機或橋接器，或能夠執行一組指令(順序的或其他)的任何器件，該組指令指定將由該器件採取之行動。另外，雖然僅說明了單一電腦1600，但術語「電腦」亦應被視為包括個別地或共同地執行一組(或多組)指令以執行本文中所論述之方法中之任何一或多者的器件之任何集合。

電腦1600亦可包括用於使用一或多個通信協定(例如，通用串列匯流排(USB)、IEEE 1394等)與周邊器件通信的輸出控制器1628。輸出控制器1628可(例如)將一映像提供至通信地耦接至電腦1600之程式設計器件1630。程式設計器件1630可經組態以程式化一平行機(例如，平行機100、FSM引擎200)。在其他實例中，程式設計器件1630可與電腦1600整合且耦接至匯流排1608或可經由網路介面器件1620或另一器件與電腦1600通信。

雖然電腦可讀媒體1624展示為單一媒體，但術語「電腦可讀媒體」應被視為包括儲存該一或多組指令1624的單一媒體或多個媒體(例如，集中式或分散式資料庫，或相關

聯之快取記憶體及伺服器，及或各種儲存媒體，諸如處理器1602暫存器、記憶體1604、1606及儲存器件1616)。術語「電腦可讀媒體」亦應被視為包括以下任何媒體，其能夠儲存、編碼或載運供電腦執行之一組指令且使該電腦執行本發明之方法中之任何一或多者，或能夠儲存、編碼或載運供此組指令利用或與此組指令相關聯的資料結構。術語「電腦可讀媒體」因此應被視為包括(但不限於)有形媒體，諸如固態記憶體、光媒體及磁性媒體。

提供摘要以遵照需要摘要的37 C.F.R.章節1.72(b)，此將允許讀者確定技術揭示內容之性質及要旨。在理解到摘要將不用以限制或解釋申請專利範圍之範疇或涵義的情況下提交摘要。以下申請專利範圍藉此併入詳細描述中，其中每一請求項依賴於其自身而作為一單獨實施例。

實例實施例

實例1包括一種電腦，其包括上面儲存有指令之一記憶體。該電腦亦包括一通信地耦接至該記憶體之處理器，其中該等指令在由該處理器執行時使該處理器將原始程式碼轉換成一自動機，該自動機包含狀態及該等狀態之間的轉變，其中該自動機中之該等狀態包括對應於一特殊目的硬體元件之一特殊目的狀態。該等指令亦使該處理器將該自動機轉換成一接線對照表並置放及路徑選擇該接線對照表以提供用於組態一目標器件之機器碼。

實例2包括一種電腦實施方法，其包括使用一或多個處理器將原始程式碼剖析成一語法樹。該方法亦包括使用該

一或多個處理器將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中自動機結構係藉由一目標硬體器件指定。該方法亦包括使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括複數個例項，每一例項對應於一目標器件之一硬體元件。該方法亦包括使用該一或多個處理器來置放該等例項中之每一者，其中置放包括將該接線對照表中之每一例項指派至該目標器件之一硬體元件。該方法亦包括根據該接線對照表使用該一或多個處理器來對該等硬體元件之間的連接進行路徑選擇，及基於置放及路徑選擇來產生用以程式化該目標器件之程式設計資料。

實例3包括一種可程式化器件，其包括複數個可程式化元件，該複數個可程式化元件包括一或多個輸入及一或多個輸出。該可程式化器件亦包括用於將該複數個可程式化元件之一部分與外部電路建立介面的一輸入區塊及一輸出區塊。該可程式化器件亦包括將該複數個可程式化元件與該輸入區塊及該輸出區塊通信地連接之複數個可程式化開關，其中一或多個可程式化開關之設定選擇性地控制該複數個可程式化元件及該複數個可程式化開關中之任兩者或兩者以上之間的信號投送。該可程式化器件亦包括複數個暫存器，該複數個暫存器經組態以儲存能夠組態該複數個可程式化元件及該複數個可程式化開關的可程式化資料，其中程式設計資料係藉由使用一或多個處理器將原始程式

碼剖析成一語法樹而產生；使用該一或多個處理器將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中自動機結構係藉由一目標硬體器件指定；使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括複數個例項，每一例項對應於一目標器件之一硬體元件；使用該一或多個處理器來置放該等例項中之每一者，其中置放包括將該接線對照表中之每一例項指派至該目標器件之一硬體元件；根據該接線對照表使用該一或多個處理器來對該等硬體元件之間的連接進行路徑選擇；及產生用以基於置放及路徑選擇來程式化該目標器件之程式設計資料。

實例4包括一種電腦可讀媒體，該電腦可讀媒體包括指令，該等指令在由一或多個處理器實施時執行以下操作：將原始程式碼剖析成一語法樹；將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中自動機結構係藉由一目標硬體器件指定；將該自動機轉換成一接線對照表，其中該接線對照表包括與該目標器件相關聯之複數個硬體元件，且其中該接線對照表定義該等硬體元件之間的連接；置放該等硬體元件中之每一者，其中置放包括將該接線對照表中之每一硬體元件指派至該目標器件內之一位置；根據該接線對照表來對該等硬體元件之間的連接進行路徑選擇；及產生用以程式化該目標器件以反映置放及路徑選擇的程

式設計資料。

實例5包括一種電腦實施方法，其包括：使用一或多個處理器將原始程式碼剖析成一語法樹；使用該一或多個處理器將該語法樹轉換成一自動機，其中該轉換包括基於一目標器件來限制自動機結構，其中該目標硬體器件包括成對為成對群組之狀態機元件。該方法亦包括使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括與該目標器件相關聯之複數個硬體元件，且其中該接線對照表定義該等硬體元件之間的連接；使用該一或多個處理器來置放該等硬體元件中之每一者，其中置放包括將該接線對照表中之每一硬體元件指派至該目標器件內之一位置；使用該一或多個處理器來根據該接線對照表對該等硬體元件之間的連接進行路徑選擇；及產生用以程式化該目標器件以反映置放及路徑選擇的複數個位元。

在實例6中，實例1至5中任一者之標的可視情況地包括，其中轉換原始程式碼包括在該量化滿足將被映射至一特殊目的硬體元件之條件時將量化轉換成包括該特殊目的硬體狀態之複數個狀態。

在實例7中，實例1至6中任一者之標的可視情況地包括，其中轉換原始程式碼包括在該量化不滿足將被映射至一特殊目的硬體元件之該條件時將量化展開成複數個通用狀態。

在實例8中，實例1至7中任一者之標的可視情況地包括，其中展開包括展開該量化以控制該自動機之一內分支

度。

在實例9中，實例1至8中任一者之標的可視情況地包括最佳化該自動機，其中最佳化包括在該自動機之一特定狀態之一估計內分支度大於該目標器件之約束時將該特定狀態分裂成多個狀態。

在實例10中，實例1至9中任一者之標的可視情況地包括，其中分裂該特定狀態包含將該特定狀態之驅動狀態分配給該多個狀態，使得該多個狀態中之每一者的一內分支度滿足該約束。

在實例11中，實例1至10中任一者之標的可視情況地包括，其中將該自動機轉換成該接線對照表包括將該等狀態映射至該接線對照表的例項，其中映射包括將該特殊目的狀態映射至對應於一特殊目的元件之一特殊目的例項。

在實例12中，實例1至11中任一者之標的可視情況地包括，其中將該自動機轉換成一接線對照表包括基於該目標器件之一實體設計來將狀態一起分組。

在實例13中，實例1至12中任一者之標的可視情況地包括，其中該等例項包括對應於一狀態機元件(SME)硬體元件之一SME例項及對應於包含一群SME之一硬體元件的一SME群組例項，且其中分組包括將狀態分組至一SME群組例項中。

在實例14中，實例1至13中任一者之標的可視情況地包括，其中將該語法樹轉換成一自動機包括將呈該原始程式碼形式之一量化轉換成複數個狀態，該複數個狀態包括對

應於該目標器件之一計數器元件之一特殊目的狀態。

在實例15中，實例1至14中任一者之標的可視情況地包括，其中對應於SME之複數個通用狀態基於一GOT硬體元件之輸出限制而一起分組以形成一GOT例項。

在實例16中，實例1至15中任一者之標的可視情況地包括限制該自動機之該內分支度，其中限制該內分支度包括限制至該自動機之一狀態的轉變之數目。

在實例17中，實例1至16中任一者之標的可視情況地包括，其中限制該內分支度包括將一特定狀態分裂成多個狀態且將該特定狀態之驅動狀態分配給該多個狀態，以使得該多個狀態中之每一者之一內分支度滿足一約束。

在實例18中，實例1至17中任一者之標的可視情況地包括，其中限制該內分支度包括將一量化展開成複數個展開狀態且限制該等展開狀態中之任一者的內轉變之數目。

在實例19中，實例1至18中任一者之標的可視情況地包括，其中轉換包括判定呈原始程式碼形式之一量化是否滿足將被映射至該目標器件之一計數器的條件；當該量化滿足該條件時，將該量化轉換成包括一計數器狀態之複數個狀態；及當該量化不滿足該條件時，藉由展開該量化而將該量化轉換成複數個SME狀態。

在實例20中，實例1至19中任一者之標的可視情況地包括，其中判定該量化是否滿足該條件包括判定在處理該量化時是否可匹配該量化之一驅動表達式。

在實例21中，實例1至20中任一者之標的可視情況地包

括，其中判定該量化是否滿足該條件包括判定該量化之一重複表達式是否為該量化之另一重複表達式之一首碼。

在實例22中，實例1至21中任一者之標的可視情況地包括，其中將該量化轉換成包括一計數器狀態之複數個狀態包括將該複數個狀態實施為一迴圈，該迴圈包含該量化之一重複表達式及該計數器狀態，其中該計數器狀態經組態以對該重複表達式匹配之次數計數，且其中在該重複表達式匹配了藉由該量化指定之次數時該計數器狀態啟動一下游狀態。

在實例23中，實例1至22中任一者之標的可視情況地包括，其中展開包括展開該量化以基於該目標器件之一內分支度約束來控制該自動機之一內分支度。

在實例24中，實例1至23中任一者之標的可視情況地包括，其中轉換包括基於共用一共同輸出之成對群組來限制自動機結構。

在實例25中，實例1至24中任一者之標的可視情況地包括，其中限制包括基於該目標器件之一計數器元件來限制自動機結構。

在實例26中，實例1至25中任一者之標的可視情況地包括發佈該複數個位元。

在實例27中，實例1至26中任一者之標的可視情況地包括最佳化該自動機以減少該複數個狀態。

實例28包括藉由使用技術方案1至27中任一者之標的產生的一映像來程式化的一平行機。

【圖式簡單說明】

圖 1 說明根據本發明之各種實施例的平行機之實例。

圖 2 說明根據本發明之各種實施例的實施為有限狀態機引擎的圖 1 之平行機的實例。

圖 3 說明根據本發明之各種實施例的圖 2 之有限狀態機引擎的區塊之實例。

圖 4 說明根據本發明之各種實施例的圖 3 之區塊之列的實例。

圖 5 說明根據本發明之各種實施例的圖 4 之列之成對群組的實例。

圖 6 說明根據本發明之各種實施例的供編譯程式用以將原始程式碼轉換成經組態以程式化圖 1 之平行機的映像之方法的實例。

圖 7 說明根據本發明之各種實施例的可包括於用以將語法樹轉換成自動機之實例方法中的各種操作。

圖 8 說明根據本發明之各種實施例的用於將語法樹轉換成自動機的方法。

圖 9 說明根據本發明之各種實施例的具有特殊目的計數器狀態的實例自動機。

圖 10 說明根據本發明之各種實施例的具有特殊目的計數器狀態的另一實例自動機。

圖 11A 及圖 11B 說明根據本發明之各種實施例的實例自動機。

圖 12 說明根據本發明之各種實施例的具有不滿足無首碼

條件之量化的正規表達式之實例自動機。

圖 13A 至圖 13C 說明根據本發明之各種實施例的用於展開量化之實例自動機。

圖 14A 及圖 14B 說明根據本發明之各種實施例的作為限制自動機之內分支度的部分的狀態分裂。

圖 15A 及圖 15B 說明根據本發明之各種實施例的實例接線對照表。

圖 16 說明根據本發明之各種實施例的具有馮諾依曼(Von Nuemann)架構的電腦之實例。

【主要元件符號說明】

100	平行機
102	通用元件
104	輸入
106	輸出
108	可程式化開關
110	資料輸入埠
111	程式設計介面
112	特殊目的元件
114	輸出埠
200	有限狀態機(FSM)引擎
201	緩衝器
202	區塊
203	區塊間開關
204	狀態機元件

205	狀態機元件
206	列
208	區塊內開關
209	輸入區塊
210	成對群組 (GOT)
211	程式設計介面
212	列內開關
213	輸出區塊
214	第一輸入
216	第二輸入
218	輸出
220	第一複數個列互連導體
222	第二複數個列互連導體
224	特殊目的元件
226	輸出
228	輸出
230	OR閘
232	記憶體胞
234	偵測線
236	資料串流線
238	偵測電路
900	自動機
902	特殊目的計數器狀態
904	SME狀態

906	SME 狀態
908	SME 狀態
910	SME 狀態
1000	自動機
1002	特殊目的計數器狀態
1004	特殊目的計數器狀態
1006	狀態
1008	狀態
1010	狀態
1100	自動機
1102	開始狀態
1104	最終狀態
1106	狀態
1108	狀態
1110	狀態
1112	最終狀態
1114	自動機
1200	自動機
1202	狀態
1204	狀態
1206	狀態
1208	狀態
1212	計數器狀態
1214	狀態

1300	自 動 機
1302	驅 動 狀 態
1304	展 開 狀 態
1305	展 開 狀 態
1306	展 開 狀 態
1307	展 開 狀 態
1308	展 開 狀 態
1310	自 動 機
1312	自 動 機
1430	狀 態
1432	狀 態
1434	狀 態
1436	狀 態
1500	接 線 對 照 表
1502	接 線 對 照 表
1506	SME例 項
1508	SME例 項
1510	SME例 項
1512	SME例 項
1514	SME例 項
1516	GOT例 項
1518	SME例 項
1600	電 腦
1602	處 理 器

1604	主記憶體
1606	靜態或非揮發性記憶體
1608	匯流排
1610	輸出器件
1612	輸入器件
1614	游標控制器件
1616	大容量儲存器件
1620	網路介面器件
1622	電腦可讀媒體
1624	指令
1626	網路
1628	輸出控制器
1630	程式設計器件

七、申請專利範圍：

1. 一種電腦，其包含：

一記憶體，其包括上面所儲存之指令；及

一處理器，其通信地耦接至該記憶體，其中該等指令在由該處理器執行時使該處理器：

將原始程式碼轉換成包含狀態及該等狀態之間的轉變的一自動機，其中該自動機中之該等狀態包括對應於一特殊目的硬體元件之一特殊目的狀態；

將該自動機轉換成一接線對照表；及

對該接線對照表進行置放及路徑選擇以提供用於組態一目標器件之機器碼。

2. 如請求項1之電腦，其中該原始程式碼包括一量化；且

其中使該處理器轉換原始程式碼包括：當該量化滿足將被映射至一特殊目的硬體元件之一條件時，將該量化轉換成包括該特殊目的硬體狀態之複數個狀態。

3. 如請求項2之電腦，其中使該處理器轉換原始程式碼包括：當該量化不滿足將被映射至一特殊目的硬體元件之該條件時，將該量化展開成複數個通用狀態。

4. 如請求項3之電腦，其中展開包括：展開該量化以控制該自動機之一內分支度。

5. 如請求項1之電腦，其進一步包含使該處理器執行最佳化該自動機的指令，其中最佳化包括在該自動機之一特定狀態之一估計內分支度大於該目標器件之約束時將該特定狀態分裂成多個狀態。

6. 如請求項5之電腦，其中分裂該特定狀態包含：將該特定狀態之驅動狀態分配給該多個狀態，使得該多個狀態中之每一者的一內分支度滿足該約束。
7. 如請求項1之電腦，其中使該處理器將該自動機轉換成該接線對照表包括將該等狀態映射至該接線對照表之例項，其中映射包括將該特殊目的狀態映射至對應於一特殊目的元件之一特殊目的例項。
8. 如請求項7之電腦，其中使該處理器將該自動機轉換成一接線對照表包括：基於該目標器件之一實體設計來將狀態一起分組。
9. 如請求項1之電腦，其中該等例項包括對應於一狀態機元件(SME)硬體元件之一SME例項及對應於包含一群SME之一硬體元件的一SME群組例項，且其中分組包括將狀態分組至一SME群組例項中。
10. 一種電腦實施方法，其包含：
 - 使用一或多個處理器將原始程式碼剖析成一語法樹；
 - 使用該一或多個處理器將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中該自動機的一結構係藉由一目標硬體器件指定；
 - 使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括複數個例項，每一例項對應於一目標硬體器件之一硬體元件；
 - 使用該一或多個處理器來置放該等例項中之每一者，

其中置放包括將該接線對照表中之每一例項指派至該目標硬體器件之一硬體元件；

使用該一或多個處理器來根據該接線對照表對該等硬體元件之間的連接進行路徑選擇；及

基於置放及路徑選擇產生用以程式化該目標硬體器件之程式設計資料。

11. 如請求項10之電腦實施方法，其中將該語法樹轉換成一個自動機包括：將呈該原始程式碼形式之一量化轉換成複數個狀態，該複數個狀態包括對應於該目標硬體器件之一計數器元件之一特殊目的狀態。

12. 如請求項10之電腦實施方法，其中對應於SME之複數個通用狀態基於一成對群組(GOT)硬體元件之一輸出限制而一起分組以形成一成對群組例項。

13. 如請求項10之電腦實施方法，其包括：

限制該自動機之內分支度，其中限制該內分支度包括限制至該自動機之一狀態的轉變之一數目。

14. 如請求項13之電腦實施方法，其中限制該內分支度包括：將一特定狀態分裂成多個狀態且將該特定狀態之驅動狀態分配給該多個狀態，以使得該多個狀態中之每一者之一內分支度滿足一約束。

15. 如請求項13之電腦實施方法，其中限制該內分支度包括：將一量化展開成複數個展開狀態及限制該等展開狀態中之任一者的內轉變之數目。

16. 一種可程式化器件，其包含：

複數個可程式化元件，其等包括一或多個輸入及一或多個輸出；

一輸入區塊及一輸出區塊，其用於將該複數個可程式化元件之一部分與外部電路建立介面；

複數個可程式化開關，其等將該複數個可程式化元件與該輸入區塊及該輸出區塊通信地連接，其中一或多個可程式化開關之設定選擇性地控制該複數個可程式化元件及該複數個可程式化開關中之任兩者或兩者以上之間的信號投送；及

複數個暫存器，其等經組態以儲存能夠組態該複數個可程式化元件及該複數個可程式化開關之程式設計資料，其中該程式設計資料係藉由以下操作產生：

使用一或多個處理器將原始程式碼剖析成一語法樹；

使用該一或多個處理器將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中該自動機的一結構係藉由一目標器件指定；

使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括複數個例項，每一例項對應於該目標器件之一硬體元件；

使用該一或多個處理器來置放該等例項中之每一者，其中置放包括將該接線對照表中之每一例項指派至該目標器件之一硬體元件；

使用該一或多個處理器來根據該接線對照表對該等硬

體元件之間的連接進行路徑選擇；及

基於置放及路徑選擇產生用以程式化該目標器件之程式設計資料。

17. 一種電腦可讀媒體，該電腦可讀媒體非暫態傳遞信號 (transitory propagating signal)，且該電腦可讀媒體包括若干指令，該等指令在由一或多個處理器實施時執行以下操作：

將原始程式碼剖析成一語法樹；

將該語法樹轉換成一自動機，其中該自動機定義具有複數個狀態及該複數個狀態之間的轉變的一行為模型，且其中該自動機的一結構係藉由一目標器件指定；

將該自動機轉換成一接線對照表，其中該接線對照表包括與該目標器件相關聯之複數個硬體元件，且其中該接線對照表定義該等硬體元件之間的連接；

置放該等硬體元件中之每一者，其中置放包括將該接線對照表中之每一硬體元件指派至該目標器件內之一位置；

根據該接線對照表對該等硬體元件之間的該等連接進行路徑選擇；及

產生用以程式化該目標器件以反映置放及路徑選擇之程式設計資料。

18. 如請求項17之電腦可讀媒體，其中轉換包括判定呈該原始程式碼形式之一量化是否滿足將被映射至該目標器件之一計數器的一條件；

當該量化滿足該條件時，將該量化轉換成包括一計數器狀態之複數個狀態；及

當該量化不滿足該條件時，藉由展開該量化將該量化轉換成複數個SME狀態。

19. 如請求項18之電腦可讀媒體，其中判定該量化是否滿足該條件包括：判定在處理該量化時是否可匹配該量化之一驅動表達式。
20. 如請求項19之電腦可讀媒體，其中判定該量化是否滿足該條件包括：判定該量化之一重複表達式是否為該量化之另一重複表達式的一首碼。
21. 如請求項20之電腦可讀媒體，其中將該量化轉換成包括一計數器狀態之複數個狀態包括將該複數個狀態實施為一迴圈，該迴圈包含該量化之一重複表達式及該計數器狀態，其中該計數器狀態經組態以對該重複表達式匹配之次數計數，且其中在該重複表達式匹配了藉由該量化指定之次數時該計數器狀態啟動一下游狀態。
22. 如請求項20之電腦可讀媒體，其中展開包括展開該量化以基於該目標器件之一內分支度約束來控制該自動機之一內分支度。
23. 一種可程式化器件，其係使用藉由請求項17之方法產生之程式設計資料來程式化。
24. 一種電腦實施方法，其包含：
 - 使用一或多個處理器將原始程式碼剖析成一語法樹；
 - 使用該一或多個處理器將該語法樹轉換成一自動機，

其中該轉換包括：

基於一目標硬體器件來限制該自動機之一結構，其中該目標硬體器件包括成對為成對群組之狀態機元件；

使用該一或多個處理器將該自動機轉換成一接線對照表，其中該接線對照表包括與該目標硬體器件相關聯之複數個硬體元件，且其中該接線對照表定義該等硬體元件之間的連接；

使用該一或多個處理器來置放該等硬體元件中之每一者，其中置放包括將該接線對照表中之每一硬體元件指派至該目標硬體器件內之一位置；

使用該一或多個處理器來根據該接線對照表對該等硬體元件之間的該等連接進行路徑選擇；及

產生用以程式化該目標硬體器件以反映置放及路徑選擇的複數個位元。

25. 如請求項24之方法，其中將該語法樹轉換成一自動機包括：基於共用一共同輸出之該等成對群組來限制該自動機的該結構。

26. 如請求項24之方法，其中限制該自動機的該結構包括：基於該目標硬體器件之一計數器元件來限制該自動機的該結構。

27. 如請求項24之方法，其進一步包含：

發佈該複數個位元。

28. 如請求項24之方法，其進一步包含最佳化該自動機以減

少該複數個狀態。

29. 一種可程式化器件，其係使用藉由請求項24之方法產生之程式設計資料來程式化。

八、圖式：

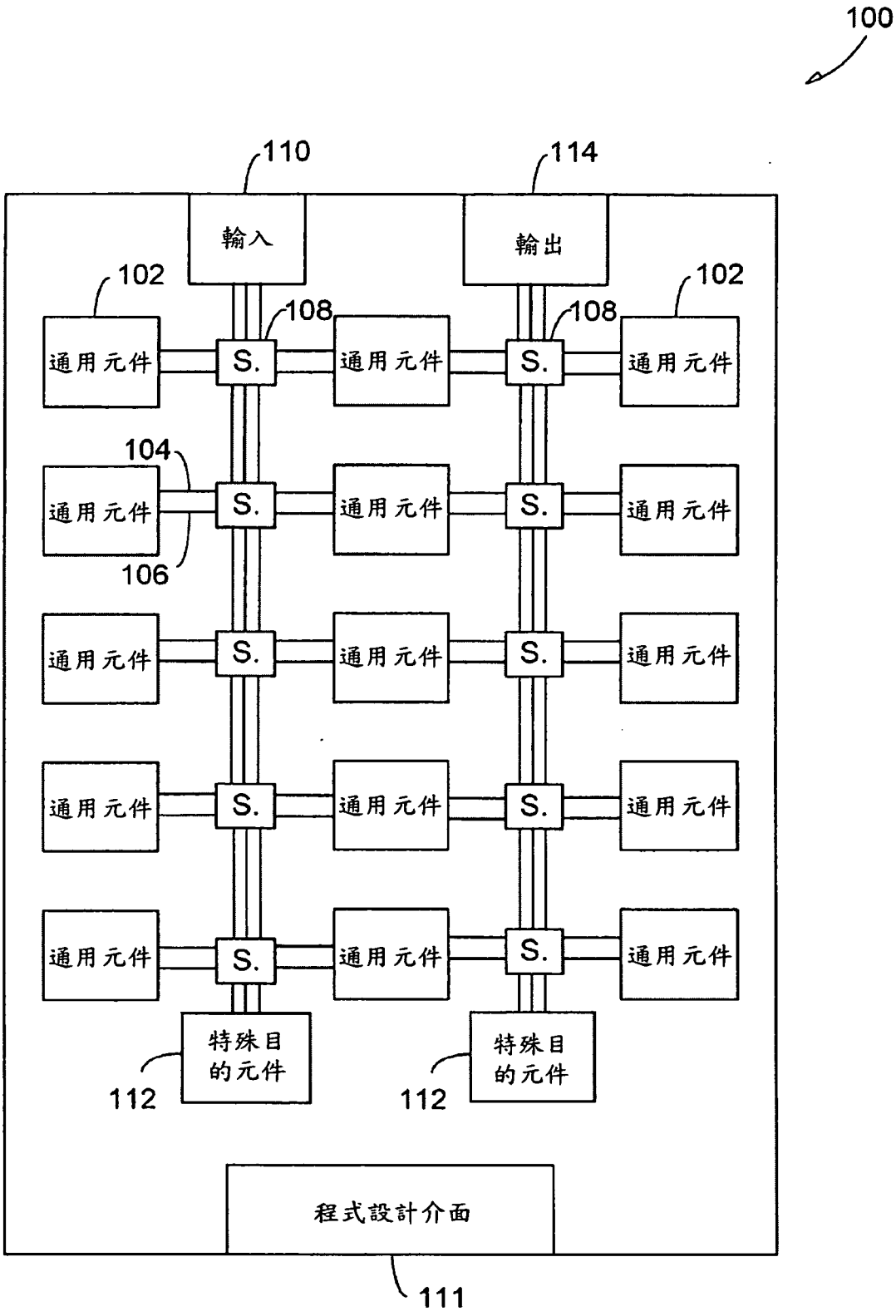


圖 1

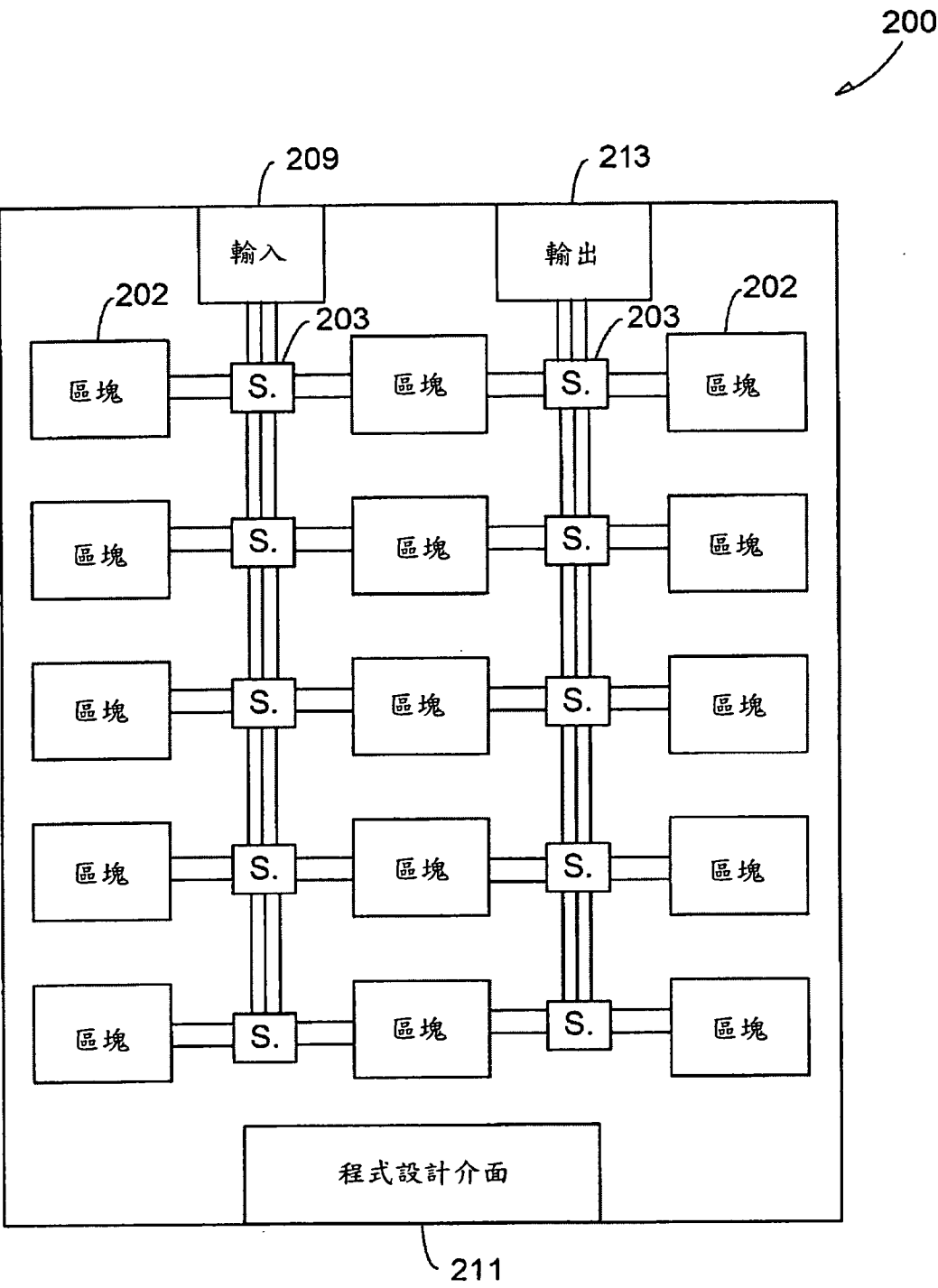


圖2

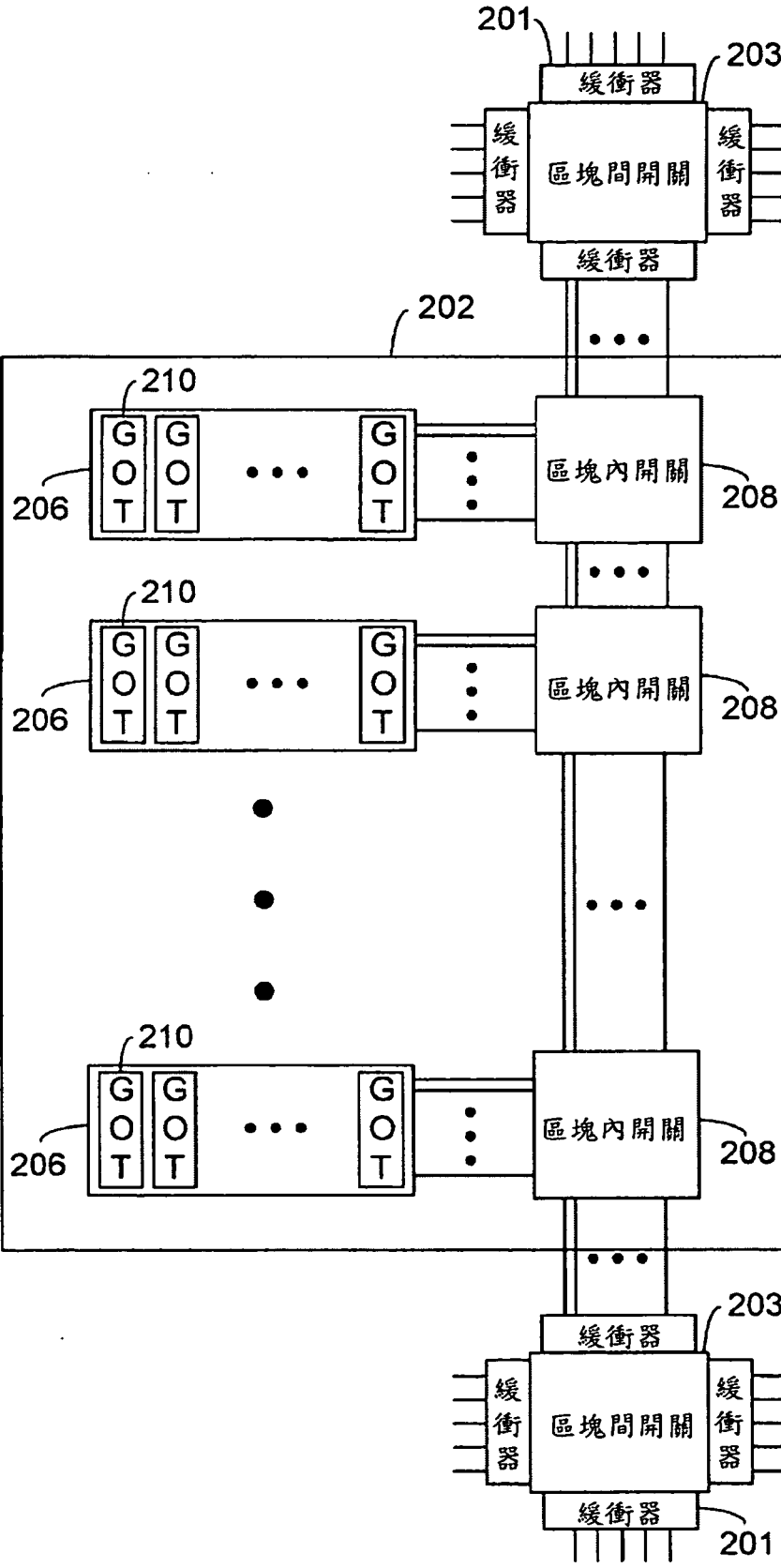


圖3

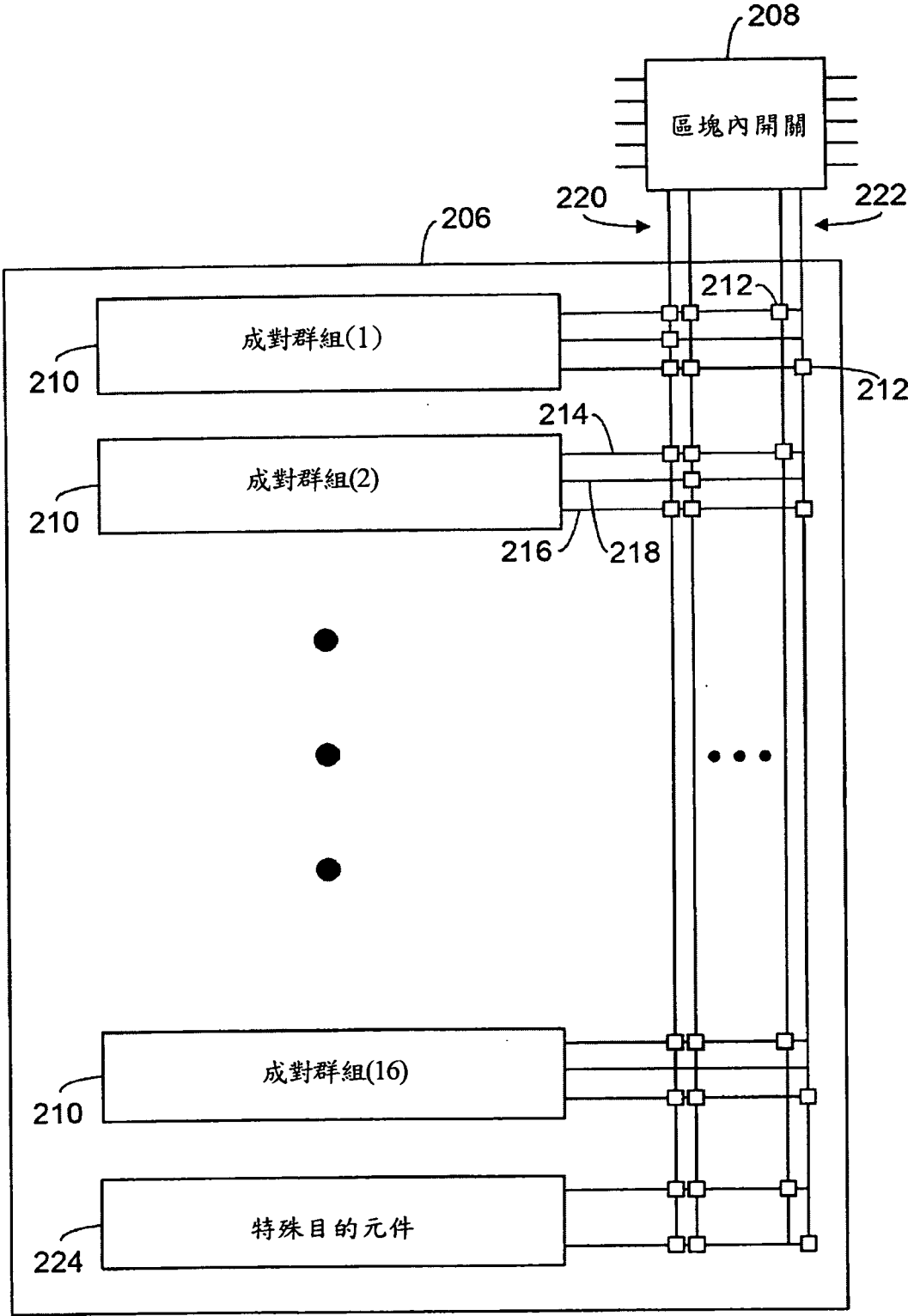


圖4

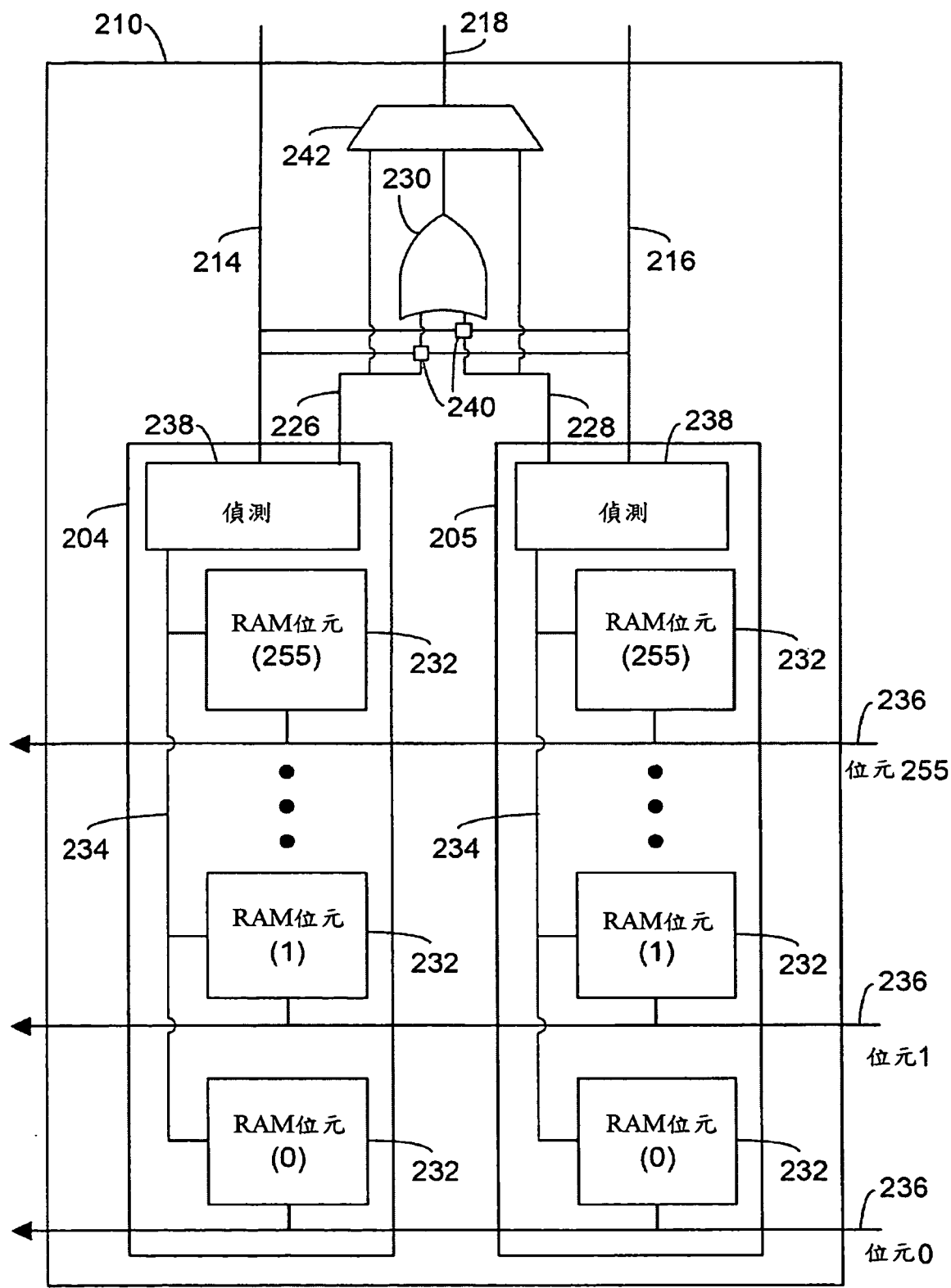


圖5

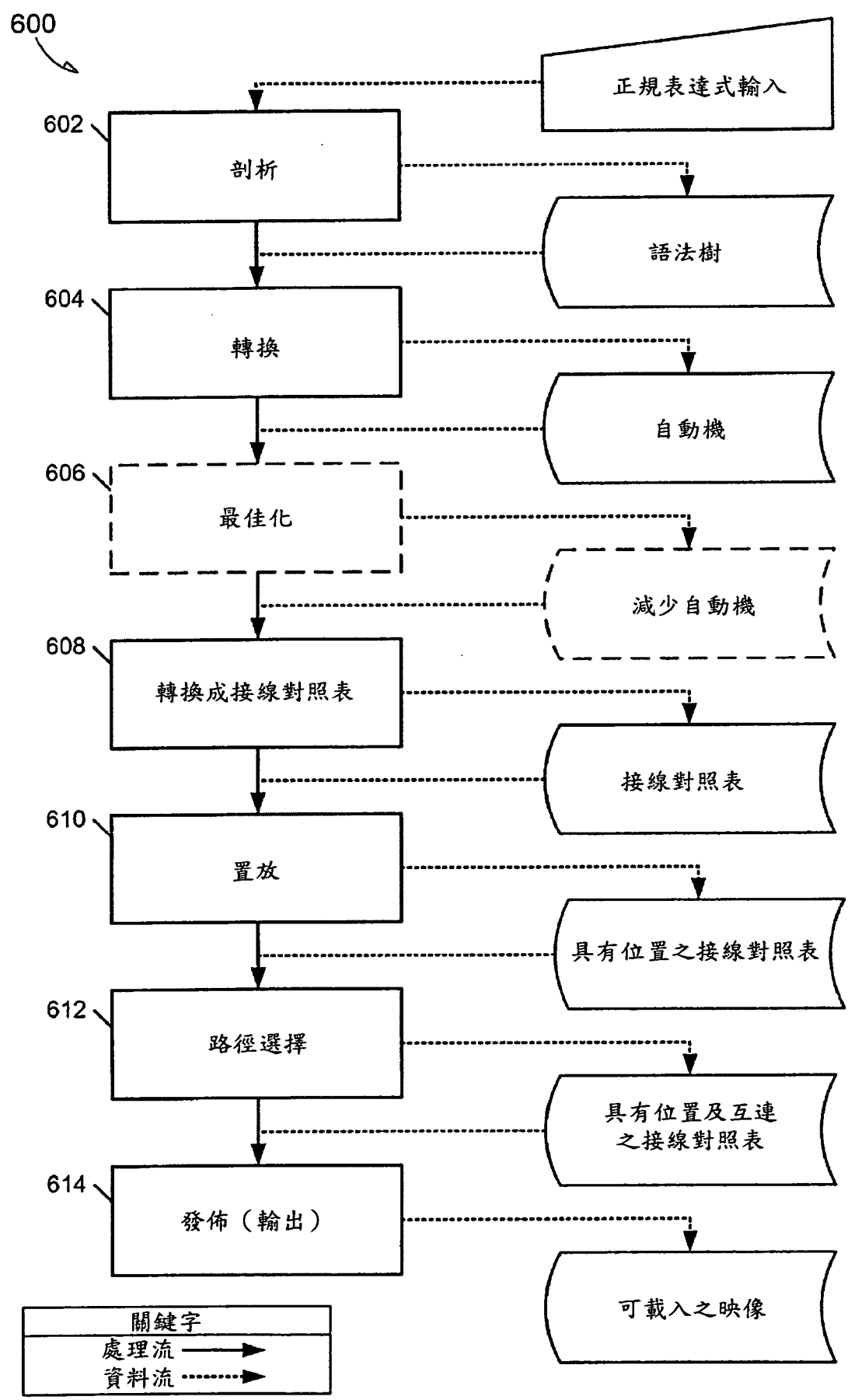


圖6

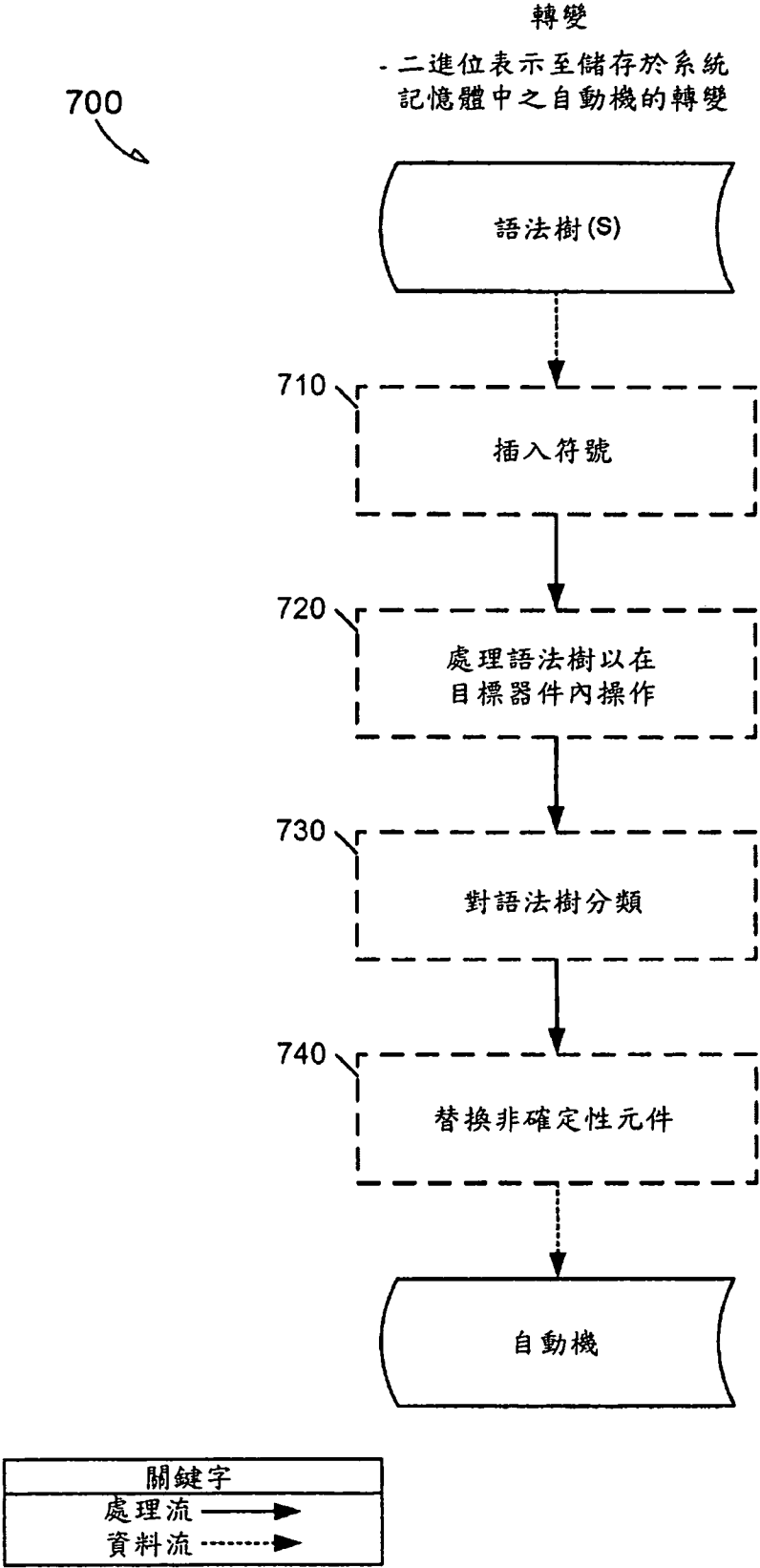


圖 7

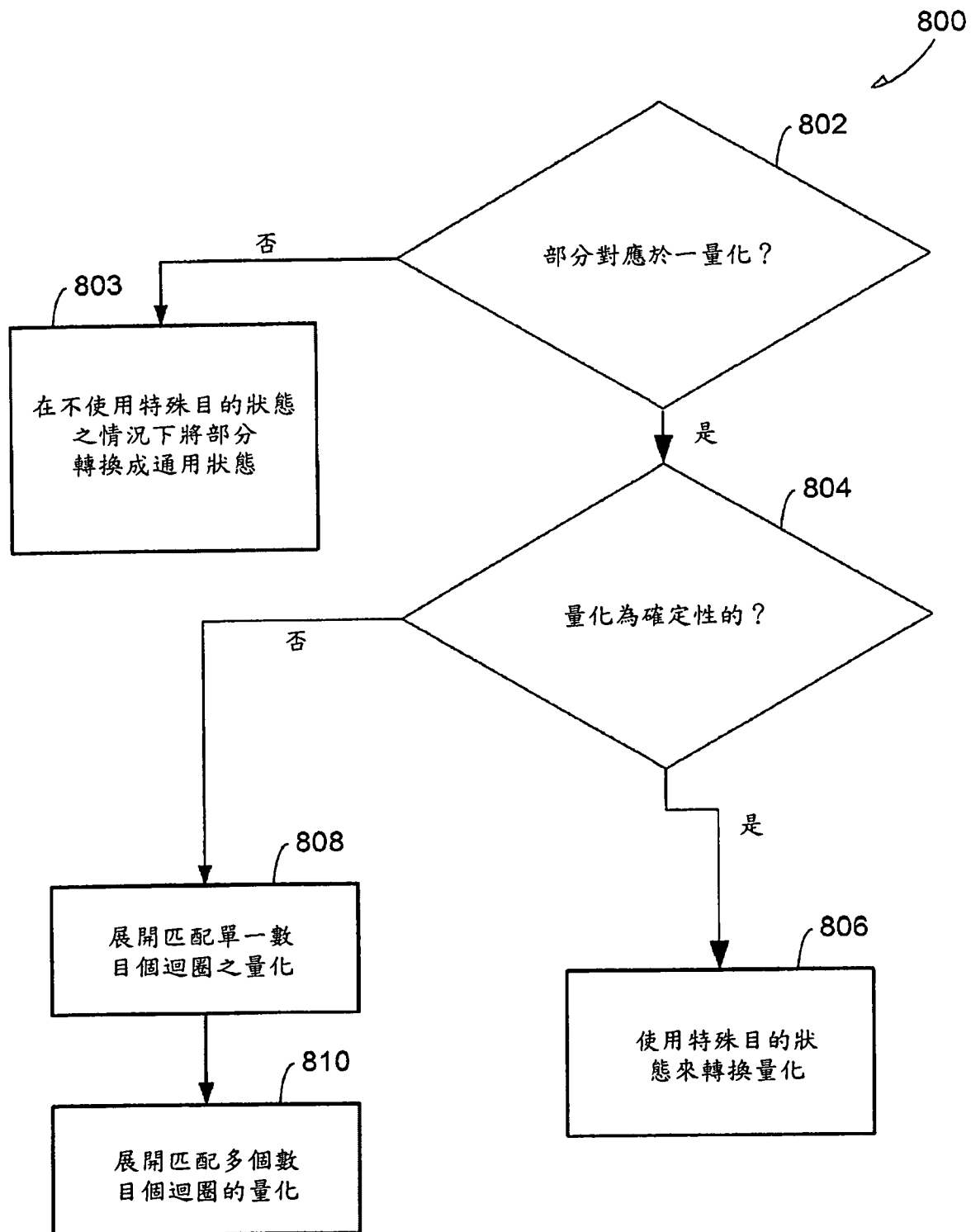


圖 8

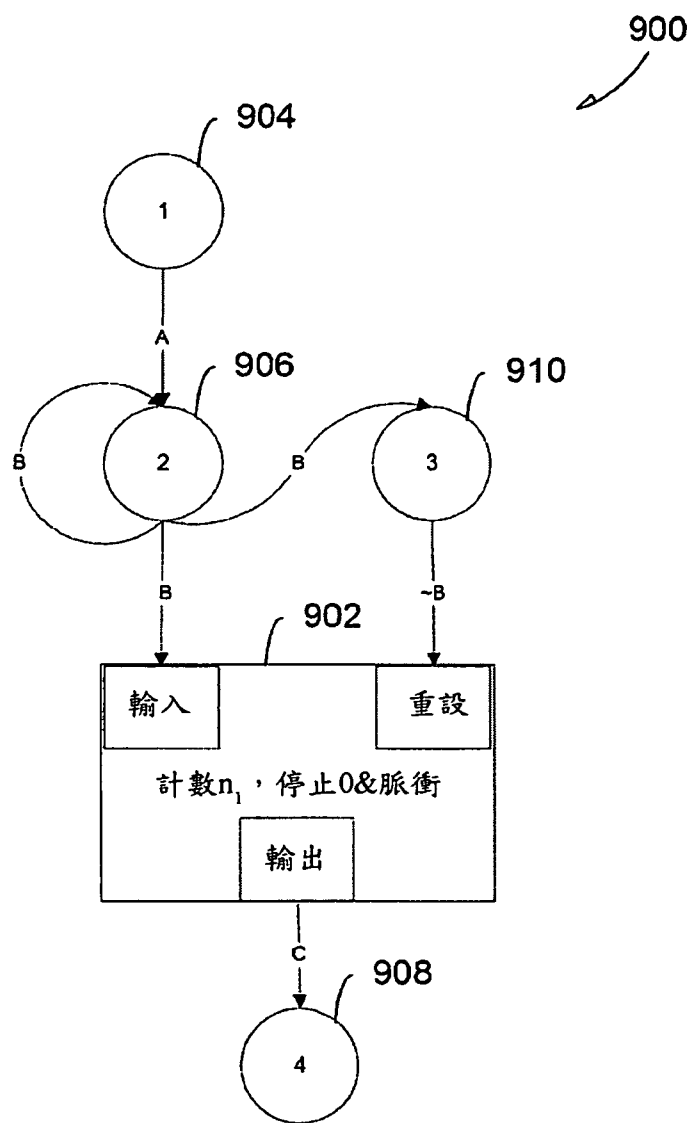


圖9

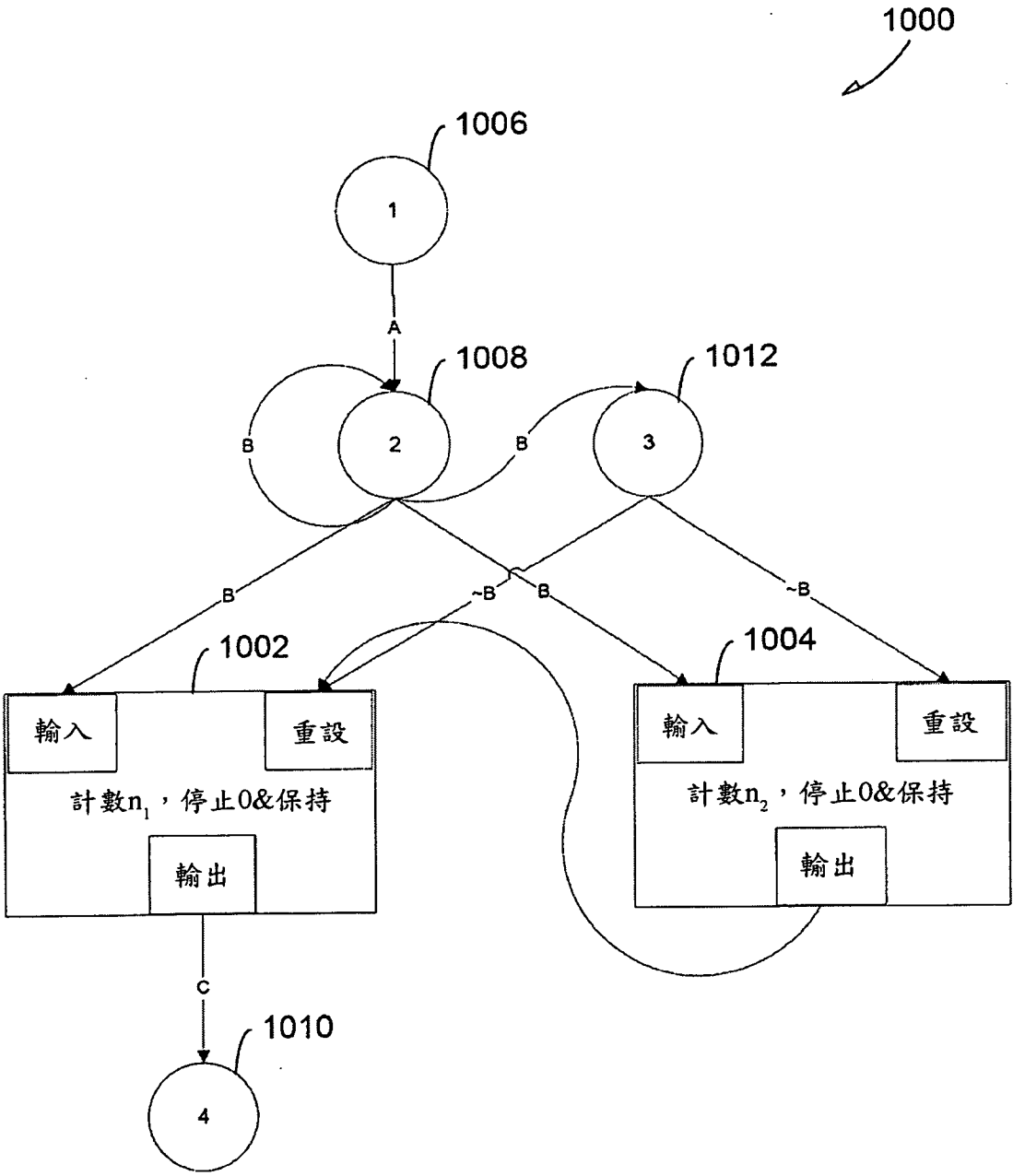
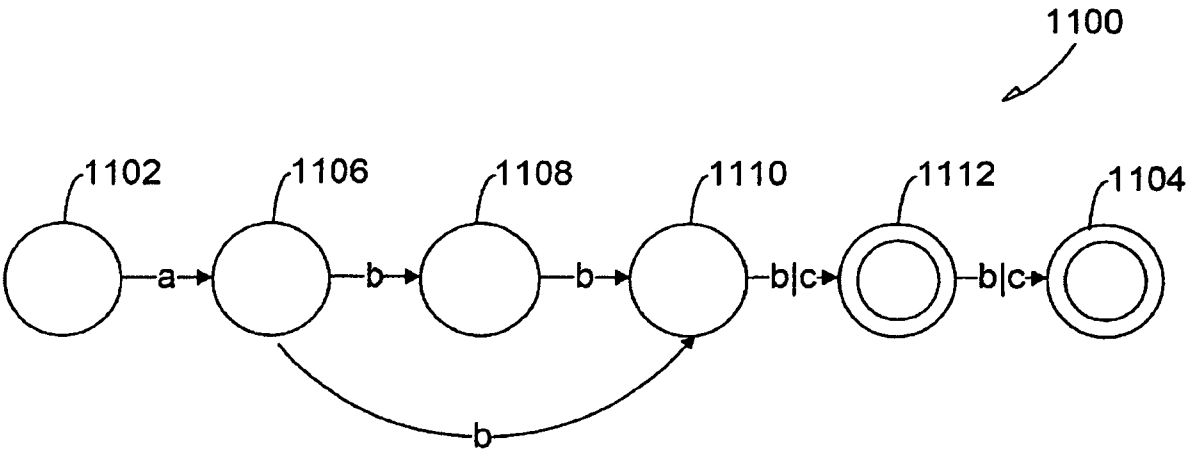
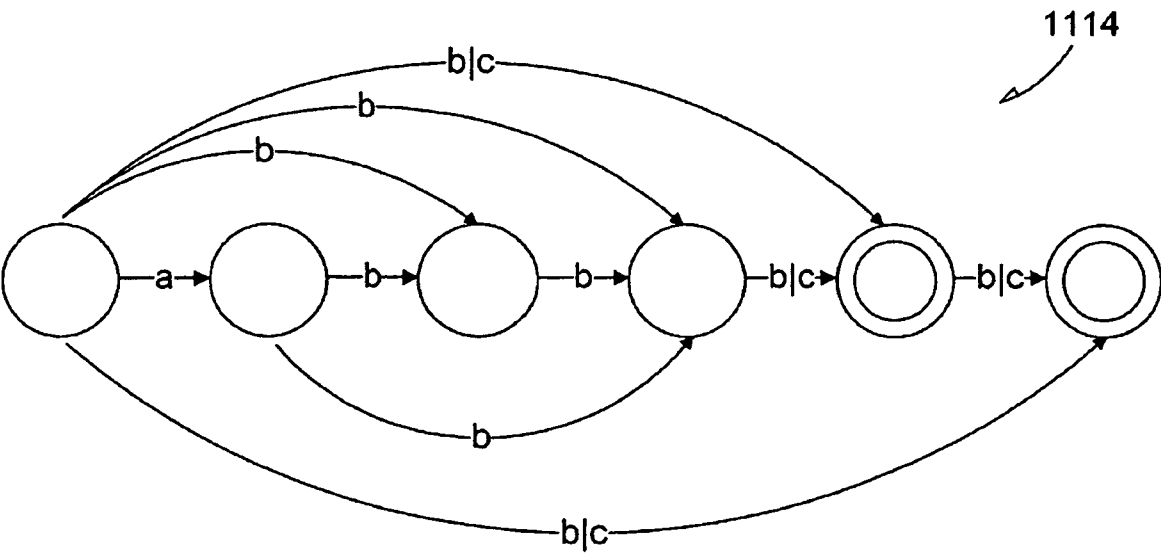


圖10



$abb?(b|c)\{1, 2\}$

圖 11A



$SS(abb?(b|c)\{1, 2\})$

圖 11B

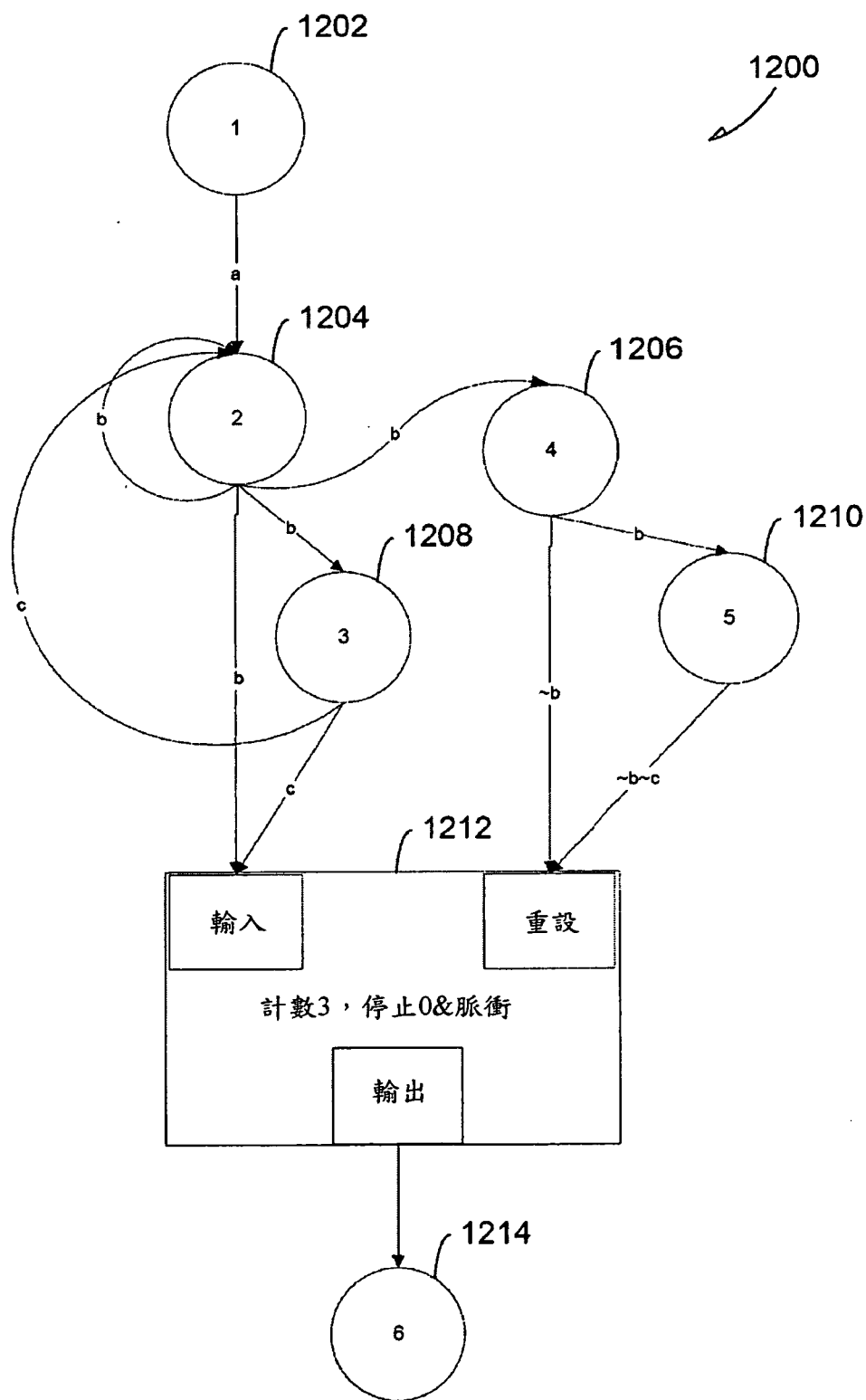


圖12

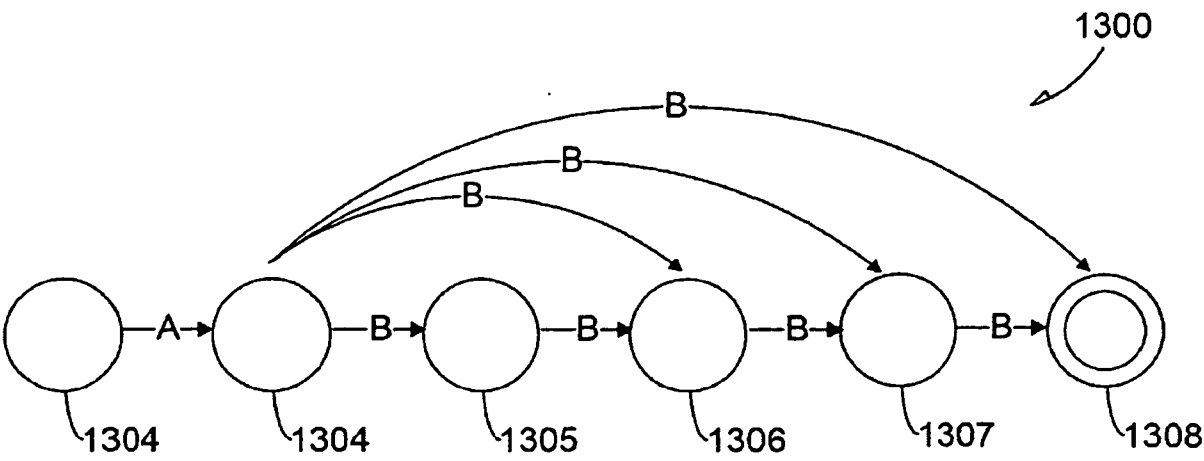


圖 13A

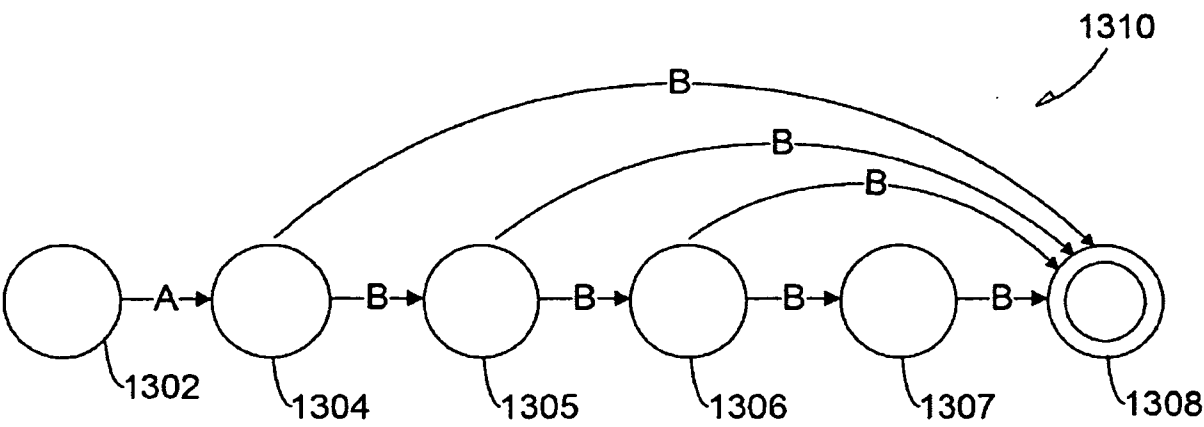


圖 13B

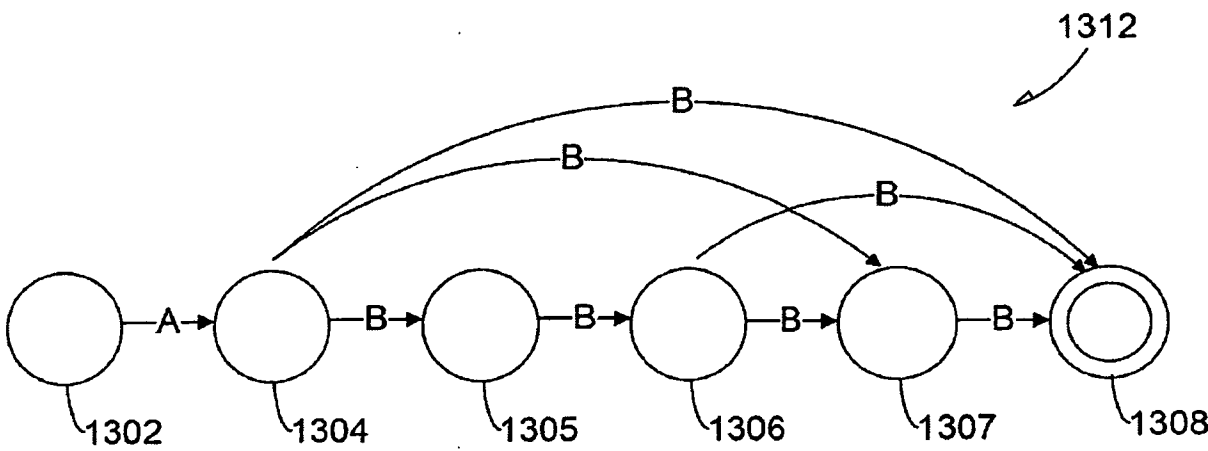


圖13C

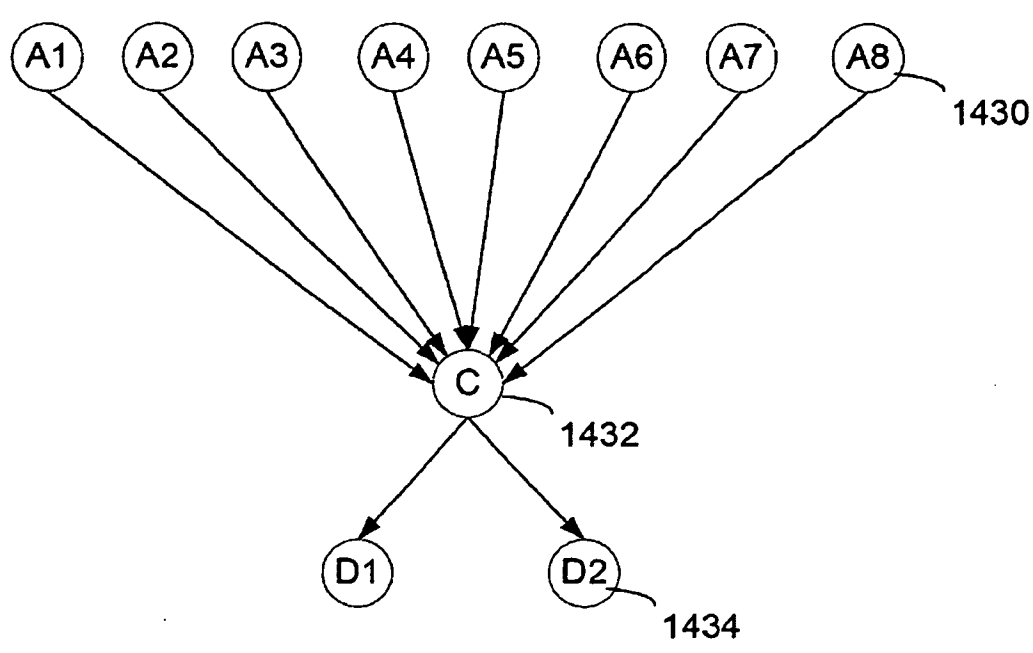


圖 14A

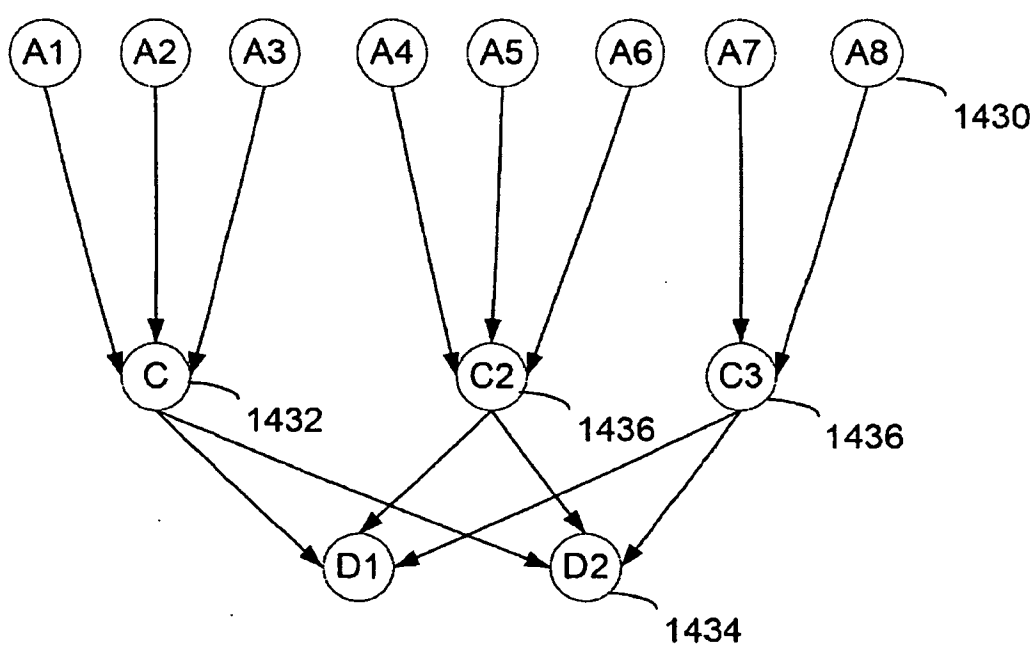


圖 14B

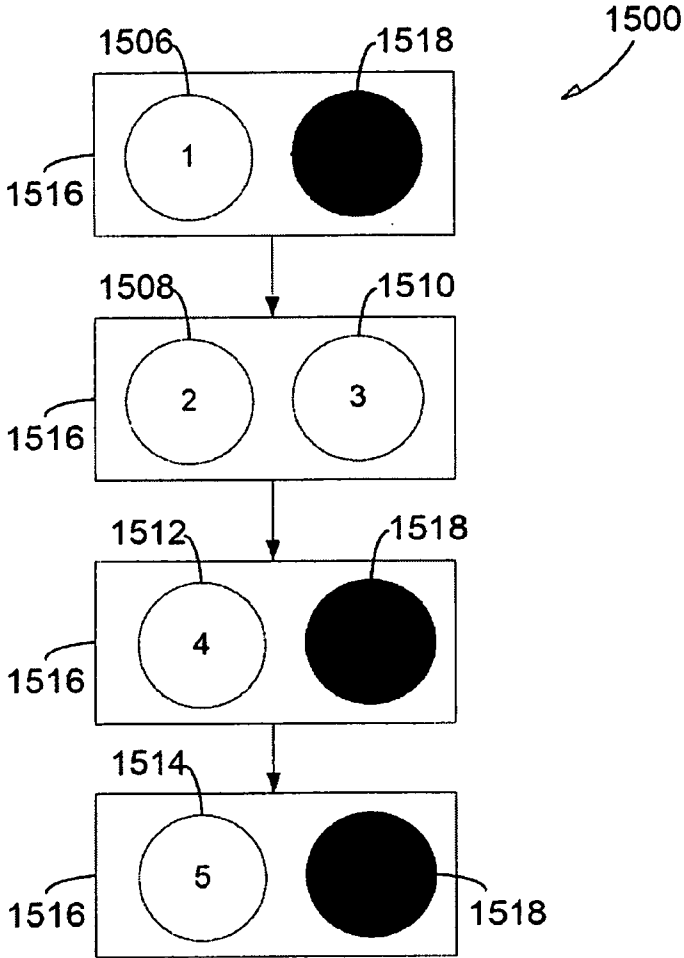


圖15A

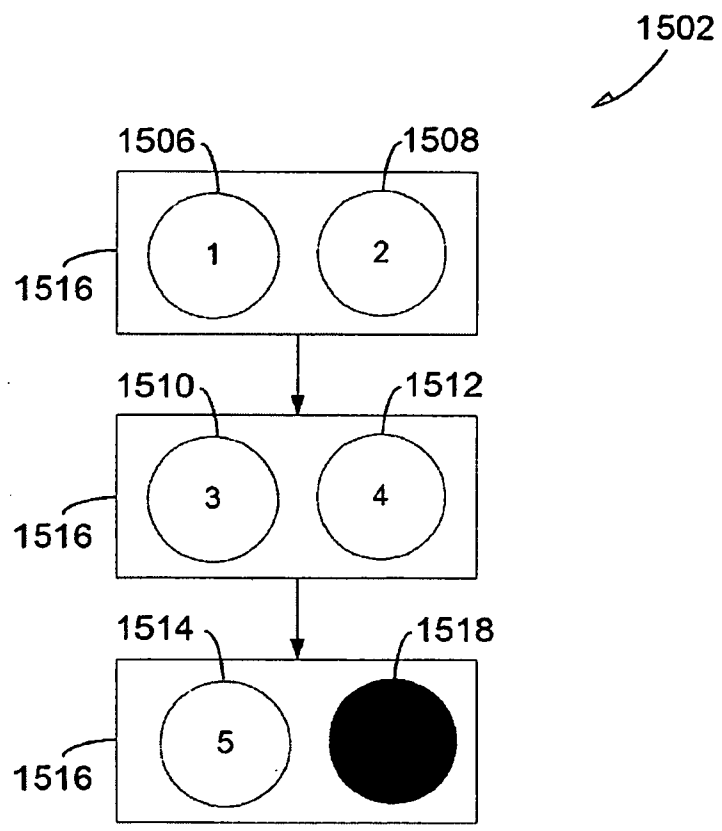


圖 15B

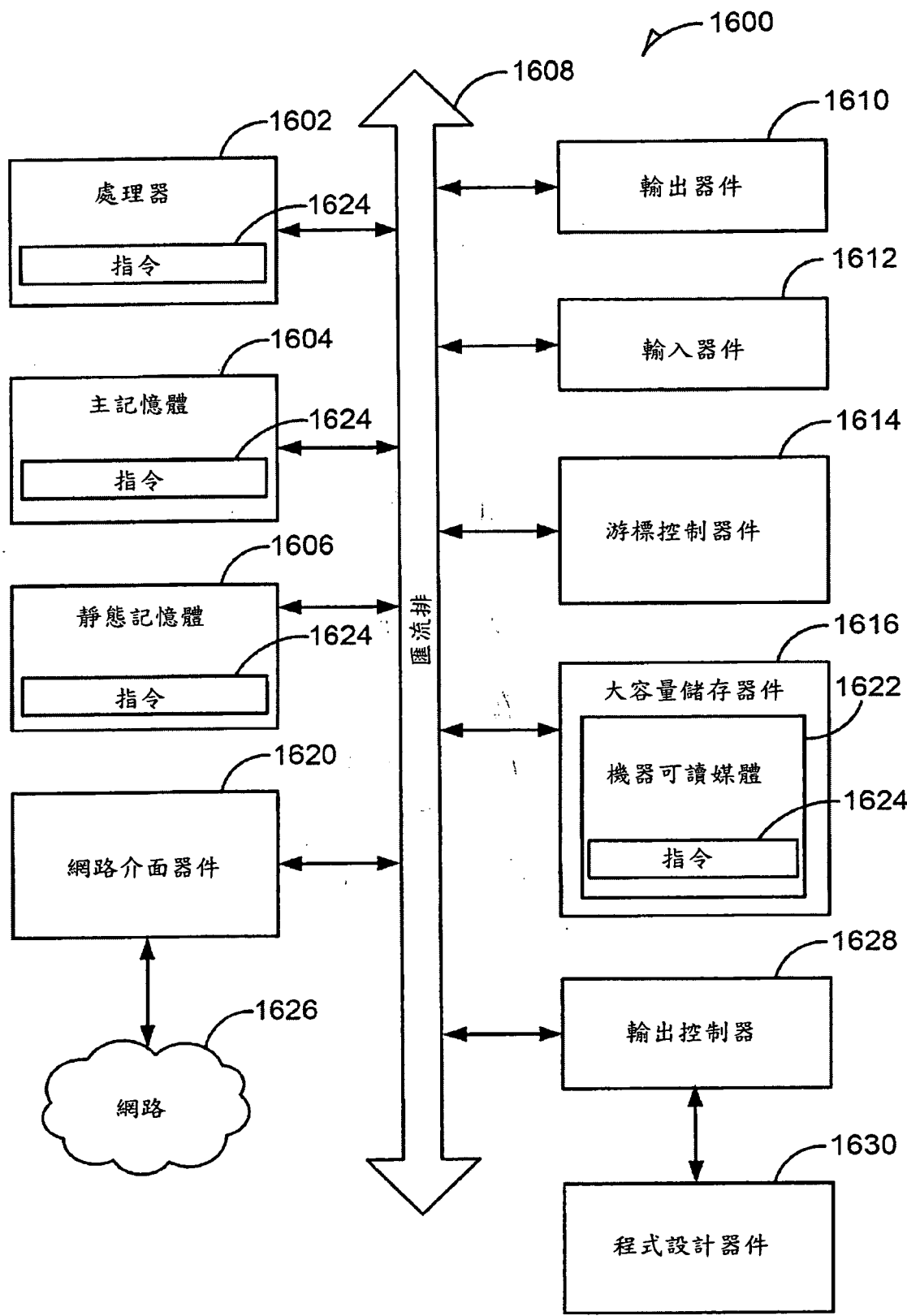


圖16