



(21)申請案號：102127237

(22)申請日：中華民國 99 (2010) 年 03 月 26 日

(51)Int. Cl. : **H04N7/24 (2011.01)**

(30)優先權：2009/04/03 日本 2009-091245

(71)申請人：新力股份有限公司 (日本) SONY CORPORATION (JP)
日本

(72)發明人：鈴木輝彥 SUZUKI, TERUHIKO (JP)；服部信夫 HATTORI, SHINOBU (JP)

(74)代理人：陳長文

(56)參考文獻：

| | | | |
|----|-----------|----|---------|
| TW | I287403 | TW | I289997 |
| TW | I293228 | TW | I299838 |
| EP | 1689193A1 | | |

審查人員：蔡鴻璟

申請專利範圍項數：3 項 圖式數：32 共 62 頁

(54)名稱

編碼方法及編碼裝置

ENCODING METHOD AND ENCODING APPARATUS

(57)摘要

本發明係關於一種可容易地檢測 Dependent stream 之畫面之位元串流上之邊界的圖像信號解碼裝置、圖像信號解碼方法、圖像信號編碼裝置、圖像信號編碼方法、及程式。於位元串流上在各 AU(Access Unite)之開始位置處配置有 AD(AU Delimiter)，且於 Dependent stream 之畫面之邊界處配置有 DD(Dependent Delimiter)。藉由檢測起動碼而檢測 NAL unit 之開頭，並檢查檢測出開頭之 nal_unit_type 之值為 18，藉此可檢測 DD，即 Dependent stream 之畫面之邊界(開頭)。本發明可應用於處理立體圖像信號之裝置。

指定代表圖：

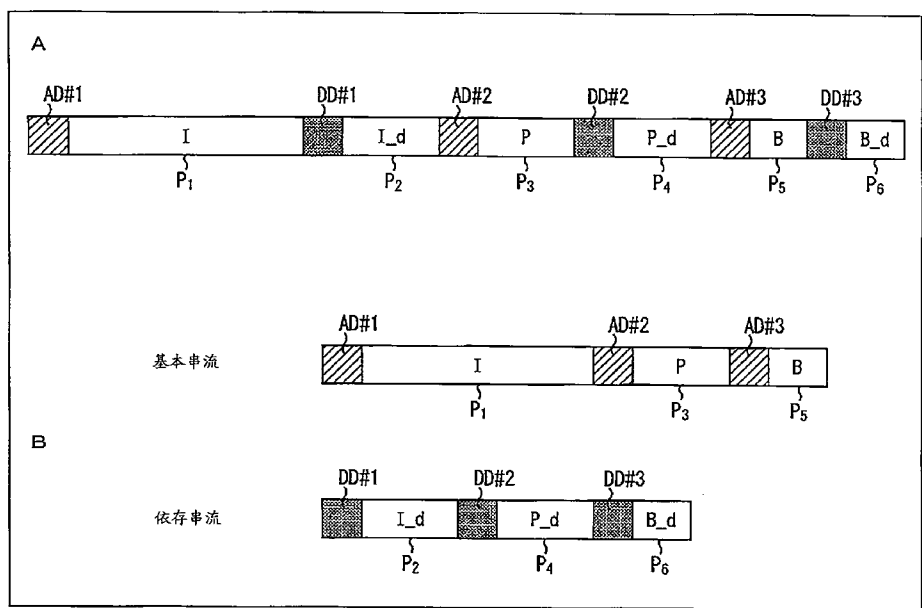


圖 16

發明摘要

※ 申請案號：102127237 (由99109239分割)

※ 申請日：99.7.26 ※IPC 分類：

H04N 7/24 (2011.01)

【發明名稱】(中文/英文)

編碼方法及編碼裝置

Encoding method and encoding apparatus

【中文】

本發明係關於一種可容易地檢測Dependent stream之畫面之位元串流上之邊界的圖像信號解碼裝置、圖像信號解碼方法、圖像信號編碼裝置、圖像信號編碼方法、及程式。於位元串流上在各AU(Access Unite)之開始位置處配置有AD(AU Delimiter)，且於Dependent stream之畫面之邊界處配置有DD(Dependent Delimiter)。藉由檢測起動碼而檢測NAL unit之開頭，並檢查檢測出開頭之nal_unit_type之值為18，藉此可檢測DD，即Dependent stream之畫面之邊界(開頭)。本發明可應用於處理立體圖像信號之裝置。

【英文】

無

【代表圖】

【本案指定代表圖】：第（ 16A ）圖。

【本代表圖之符號簡單說明】：

（無元件符號說明）

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

（無）

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

編碼方法及編碼裝置

Encoding method and encoding apparatus

【技術領域】

本發明係關於一種圖像信號解碼裝置、圖像信號解碼方法、圖像信號編碼裝置、圖像信號編碼方法、及程式，本發明特別是關於一種可容易地檢測Dependent stream(依存串流)之畫面之位元串流上之邊界的圖像信號解碼裝置、圖像信號解碼方法、圖像信號編碼裝置、圖像信號編碼方法、及程式。

【先前技術】

近年來，以高效之資訊傳輸、儲存為目的，利用圖像資訊特有之冗餘性，且依據藉由離散餘弦轉換等正交轉換及運動補償而壓縮的MPEG(Motion Picture Experts Group，動畫專家群)等方式之裝置，正於廣播站等之資訊發送、及普通家庭中之資訊接收此兩者中不斷普及。

尤其是，MPEG2(ISO/IEC(International Standardization Organization/Electrotechnical Commission，國際標準化組織/國際電工委員會) 13818-2)係作為通用圖像編碼方式而定義，其係包含交錯掃描圖像及順序掃描圖像此兩者、以及標準解像度圖像及高清晰度圖像之標準規格。目前係廣泛用於專業用途及消費者用途之大範圍的應用中。

藉由使用MPEG2，例如，只要為具有720×480像素之標準解像度之交錯掃描圖像，則可藉由分配4~8 Mbps之碼量(位元率)而實現高壓

縮率與良好的畫質。又，只要為具有 1920×1088 像素之高解像度之交錯掃描圖像，則可藉由分配18~22 Mbps之碼量而實現高壓縮率與良好的畫質。

MPEG2主要將適合放送用之高畫質編碼作為對象，並不對應於碼量低於MPEG1，即壓縮率更高之編碼方式。

因手機終端之普及，今後此種編碼方式之需求將變大，對應於此而進行MPEG4編碼方式之標準化。關於圖像編碼方式，其規格係於1998年12月作為ISO/IEC 14496-2而被認可為國際標準。

進而，進行H.264/AVC(Advanced Video Coding，高級視訊編碼)(MPEG-4 part 10、ISO/IEC 14496-10|ITU (International Telecommunication Union，國際電信聯盟)-T H.264)之標準之規格化。於ITU-T與ISO/IEC之間設立共同進行視訊編碼之標準化之稱為JVT(Joint Video Team，聯合視訊組)的團體，並由該團體來推進規格化。

已知與MPEG2或MPEG4等先前之編碼方式相比，H.264/AVC雖然其編碼、解碼要求更多之運算量，但可實現更高之編碼效率。

[H.264/AVC]

圖1係表示藉由離散餘弦轉換或卡忽南-拉維(Karhunen-Loeve)轉換等正交轉換與運動補償而實現圖像壓縮之圖像資訊編碼裝置之構成例的方塊圖。

1為A/D(analog/digital，類比/數位)轉換部，2為畫面重排序緩衝器，3為加算部，4為正交轉換部，5為量化部，6為可逆編碼部，7為儲存緩衝器，8為逆量化部、9為逆正交轉換部，10為訊框記憶體，11為運動預測補償部、12為速率控制部。

首先，於A/D轉換部1中，將作為輸入之圖像信號轉換為數位信號。

其次，根據作為輸出之圖像壓縮資訊之GOP(Group of Pictures，圖群)結構，而於畫面重排序緩衝器2中進行訊框之重排序。

關於進行框內編碼之圖像，係將訊框整體之圖像資訊輸入至正交轉換部4，並於此處實施離散餘弦轉換、卡忽南-拉維轉換等正交轉換。

於量化部5中，對作為正交轉換係數4之輸出之轉換係數實施量化處理。

將作為量化部5之輸出之經量化的轉換係數輸入至可逆編碼部6，於此處實施可變長度編碼、算術編碼等可逆編碼後，將其儲存至儲存緩衝器7中，並作為圖像壓縮資訊而輸出。量化部5之行爲係由速率控制部12而控制。

同時，將作為量化部5之輸出之經量化的轉換係數輸入至逆量化部8，進而，於逆正交轉換部9中實施逆正交轉換處理，而形成解碼圖像資訊，並將該資訊儲存至訊框記憶體10中。

關於進行框間編碼之圖像，首先，係自畫面重排序緩衝器2輸入至運動預測·補償部11。

同時，自訊框記憶體10中取出作為參照之圖像資訊，並實施運動預測·補償處理，而生成參照圖像資訊。

將參照圖像資訊傳送至加算部3，並於此處轉換為與該圖像資訊之差分信號。

運動補償·預測部11同時將運動向量資訊輸出至可逆編碼部6。對運動向量資訊實施可變長度編碼、算術編碼等可逆編碼處理，並將其插入至圖像壓縮資訊之標頭部。其他處理係與實施框內編碼之圖像壓縮資訊相同。

圖2係表示圖像資訊解碼裝置之構成例之方塊圖。

21為儲存緩衝器，22為可逆編碼解碼部，23為逆量化部，24為

逆正交轉換部，25為加算部，26為畫面重排序緩衝器，27為D/A(digital/analog，數位/類比)轉換部，28為訊框記憶體、29為運動預測·補償部。

首先，將作為輸入之圖像壓縮資訊(位元串流)儲存於儲存緩衝器21中，其後，傳輸至可逆編碼解碼部22。

於可逆編碼解碼部22中，根據規定之圖像壓縮資訊之格式而進行可變長度解碼、算術解碼等處理。

同時，當該訊框係經框間編碼者時，於可逆編碼解碼部22中，亦對圖像壓縮資訊之標頭部所儲存之運動向量資訊進行解碼，並將該經解碼之資訊輸出至運動預測·補償部29。

將作為可逆編碼解碼部22之輸出之經量化的轉換係數輸入至逆量化部23，並於此處作為轉換係數而輸出。

於逆正交轉換部24中，根據規定之圖像壓縮資訊之格式，而對轉換係數實施逆離散餘弦轉換、逆卡忽南-拉維轉換等逆正交轉換。

當該訊框係經框內編碼者時，將實施了逆正交轉換處理之圖像資訊儲存於畫面重排序緩衝器26中，並於D/A轉換處理之後加以輸出。

當該訊框係經框間編碼者時，根據實施了可逆解碼處理之運動向量資訊、及訊框記憶體28中所儲存之圖像資訊而生成參照圖像。於加算部25中將該參照圖像與逆正交轉換部24之輸出加以合成。其他處理與經框內編碼之訊框相同。

與MPEG2或MPEG4同樣地，之前敘述之由JVT而規格化之AVC係包含運動補償及離散餘弦轉換之混合編碼方式。

離散餘弦轉換亦可為與實數之離散餘弦轉換近似的整數轉換。離散餘弦轉換之轉換方法係使用4×4塊大小之整數係數者，有時運動補償時之塊大小可變等，詳細方式雖不同，但基本方式係與利用圖1

之構成所實現之編碼方式相同。

然而，近年來，由於立體圖像之拍攝、顯示技術之進步，使H.264/AVC向立體圖像信號之編碼擴展之研究不斷推進。

對利用複數個攝像裝置拍攝之多視點的圖像進行編碼之MVC(Multiview Video Coding，多視角視訊編碼)之標準化不斷推進。

將假設2個視點之拍攝、顯示之圖像稱為立體圖像。裸眼立體顯示器可對應於多視點之顯示。

以下，主要對2個視點之立體圖像之例進行說明，同樣地亦可適用於3個視點以上之多視點之圖像。

[MVC]

圖3係表示多視點編碼裝置之圖。

於多視點編碼裝置41中，對自2個攝像裝置、即攝像裝置31與攝像裝置32所供給之視訊信號進行編碼，並輸出藉由編碼而生成之位元串流。包含2個視點之圖像之資料之位元串流可多工於1條串流而輸出，亦可作為2個或2個以上之位元串流而輸出。

圖4係表示圖3之多視點編碼裝置41之構成例的方塊圖。

於多視點編碼裝置41中，將多視點之圖像中的1個視點之圖像作為基本串流(Base stream)而編碼，將其他圖像作為Dependent stream(依存串流)而編碼。

於立體圖像之情形時，將L圖像(左視點圖像)與R圖像(右視點圖像)中之任一圖像作為Base stream而編碼，將另一圖像作為Dependent stream而編碼。

Base stream係藉由H.264 AVC/High Profile等所編碼之與現有AVC相同的位元串流。因此，Base stream成為可藉由與H.264 AVC/High Profile相對應之現有AVC之解碼器而解碼的串流。

將作為Base stream而解碼之圖像輸入至重排序緩衝器51，以適當

的順序重排序從而將其編碼為I畫面(Intra picture, 節點畫面)、P畫面(Predicted picture, 預測畫面)、B畫面(Bi-predictive picture, 前後預測畫面)。將重排序後之圖像輸出至視訊編碼部52。

視訊編碼部52具有與圖1之圖像資訊編碼裝置相同之構成。於視訊編碼部52中, 例如依據H.264 AVC/High Profile而進行編碼, 並將所得之位元串流輸出至多工部57。又, 將局部解碼圖像保存於訊框記憶體53中, 並用作下一畫面或Dependent stream之畫面編碼時之參照圖像。

另一方面, 將作為Dependent stream而編碼之圖像輸入至重排序緩衝器54, 並以適當的順序重排序從而將其編碼為I畫面、P畫面、B畫面。將重排序後之圖像輸出至依存串流編碼部55。

於依存串流編碼部55中, 在通常之AVC之編碼中, 追加進行將訊框記憶體53中所記憶之Base stream之局部解碼圖像用作參照圖像的編碼, 並將位元串流輸出至多工部57。又, 將局部解碼圖像保存於訊框記憶體56中, 並用作下一畫面編碼時之參照圖像。

於多工部57中, 將Base stream及Dependent stream多工於1條位元串流並輸出。亦可將Base stream與Dependent stream分別作為不同的位元串流而輸出。

圖5係表示MVC之參照圖像之例的圖。

與通常之AVC同樣地, Base stream僅執行時間方向之預測而進行編碼。

Dependent stream在與通常之AVC相同之同一視點圖像內之時間方向之預測中, 追加進行將同時刻的Base stream之圖像作為參照圖像之預測而進行編碼。即便在無法適當地進行時間方向之預測時, 亦可藉由設為可參照同時刻之其他視點的圖像, 而改善編碼效率。

圖6係表示生成Base stream之圖4之視訊編碼部52之構成與訊框記

憶體53的方塊圖。

圖6所示之構成除了依存串流編碼部55參照訊框記憶體53中所保存之圖像方面以外，與圖1之圖像資訊編碼裝置之構成相同。

圖7係表示生成Dependent stream之圖4之依存串流編碼部55之構成與訊框記憶體56的方塊圖。

圖7所示之構成除了可參照訊框記憶體53中所保存之圖像方面以外，與圖1之圖像資訊編碼裝置之構成相同。將自訊框記憶體53讀出之參照圖像輸入至運動預測·補償部90，用於運動預測、運動補償。

圖8係表示多視點解碼裝置101之構成例之方塊圖。

將經由網路或記錄媒體而自多視點編碼裝置41供給之Base stream輸入至緩衝器111，將Dependent stream輸入至緩衝器114。當多工於1條串流而進行供給時，係將其分離為Base stream與Dependent stream，而分別輸入至緩衝器111與緩衝器114。

將在緩衝器111中延遲特定時間之Base stream輸出至視訊解碼部112。

於視訊解碼部112中，依據AVC而進行Base stream之解碼，並將所得之解碼圖像保存於訊框記憶體113中。將訊框記憶體113中所保存之解碼圖像用作下一畫面或Dependent stream之畫面解碼時之參照圖像。

以特定之時序，將藉由視訊解碼部112所獲得之解碼圖像作為視訊信號而輸出至3D(three-dimension，三維)顯示器102。

另一方面，將在緩衝器114中延遲特定時間之Dependent stream輸出至依存串流解碼部115。

於依存串流解碼部115中，進行Dependent stream之解碼，並將所得之解碼圖像保存於訊框記憶體116中。將訊框記憶體116中所保存之解碼圖像用作下一畫面解碼時之參照圖像。

於依存串流解碼部115中，適當地根據位元串流中之資訊(旗標等)而將訊框記憶體113中所保存之圖像用作參照圖像。

以特定之時序，將藉由依存串流解碼部115所獲得之解碼圖像作為視訊信號而輸出至3D顯示器102。

於3D顯示器102中，根據自視訊解碼部112所供給之視訊信號、及自依存串流解碼部115所供給之視訊信號，而進行立體圖像之顯示。

圖9係表示進行Base stream之解碼之圖8之視訊解碼部112之構成與訊框記憶體113的圖。

圖9所示之構成除了依存串流解碼部115參照訊框記憶體113中所保存之圖像方面以外，與圖2之圖像資訊解碼裝置之構成相同。

圖10係表示對Dependent stream進行解碼之圖8之依存串流解碼部115之構成與訊框記憶體116的方塊圖。

圖10所示之構成除了可參照訊框記憶體113中所保存之圖像方面以外，與圖2之圖像資訊解碼裝置之構成相同。將自訊框記憶體113讀出之參照圖像輸入至運動預測·補償部148，用於運動預測、運動補償。

[先前技術文獻]

[專利文獻]

[專利文獻1]日本專利特開2007-208917號公報

【發明內容】

[發明所欲解決之問題]

於AVC及MVC中，為防止緩衝器溢流、下溢，而定義模仿解碼裝置側之緩衝器之動作的虛擬緩衝器模型。

於編碼裝置側，以不使虛擬緩衝器溢流、下溢之方式而進行編碼。只要係以此方式而編碼之位元串流，則解碼裝置可不使緩衝器出

現故障而進行解碼。

對虛擬緩衝器模型進行說明。

圖11係表示虛擬緩衝器模型之例的圖。

將輸入之位元串流首先供給至緩衝器151。利用被稱為AU((Access Unite, 存取單元)之單元而自緩衝器151瞬時地讀出位元串流，並將其供給至視訊解碼部152。

於視訊解碼部152中，瞬時地對自緩衝器151讀出之資料進行解碼，並生成解碼圖像。此種假定係於虛擬緩衝器模型中完成。

再者，圖11之緩衝器151係與圖8之多視點解碼裝置101中的緩衝器111及緩衝器114相對應。又，圖11之視訊解碼部152係與圖8之多視點解碼裝置101中的視訊解碼部112及依存串流解碼部115相對應。

於AVC之情形時，1個AU中儲存有1個畫面之資料。

於MVC之情形時，1個AU中儲存有同一時刻之所有view(Base stream及Dependent stream)之資料。此處，所謂view係指各視點之圖像。

圖12表示此種虛擬緩衝器模型中之緩衝器151之動作。圖12之縱軸表示緩衝器佔有量，橫軸表示時刻。

如圖12所示，以特定的位元率向緩衝器151中供給位元串流。又，於各AU之讀出時刻，自緩衝器151抽出該AU中所儲存之位元串流。時刻 t_1 、 t_2 、 t_3 ，...分別係自緩衝器151之讀出時刻。

圖13係表示利用MVC而編碼之位元串流之AU之構成的圖。

於圖13之上段表示有AU#1、#2、#3。

AU#1包含作為同一時刻的畫面之Base stream之畫面 P_1 與Dependent stream之畫面 P_2 (儲存有畫面之資料)。

AU#2包含作為同一時刻的畫面之Base stream之畫面 P_3 與Dependent stream之畫面 P_4 。

AU#3 包含作為同一時刻的畫面之 Base stream 之畫面 P₅ 與 Dependent stream 之畫面 P₆。

於圖13之下段表示位元串流之構成。

圖13之位元串流係對 Base stream 與 Dependent stream 進行多工而構成之1條串流。於各 AU 之開頭插入有被稱為 AD(AU Delimiter, 存取單元定界符)之 NAL unit(Network Abstraction Layer unit, 網路抽象層單元)。AD 係某個固有之資料行。

AD 表示同時刻之 Base stream 之畫面與 Dependent stream 之畫面的資料之開始位置。

如圖13所示，於位元串流上，AU#1 包含 AD#1、畫面 P₁、及畫面 P₂。

AU#2 包含 AD#2、畫面 P₃、及畫面 P₄。

AU#3 包含 AD#3、畫面 P₅、及畫面 P₆。

藉由檢索 AD，可找出 AU 之開頭，而容易地進行對特定的畫面之存取。

圖14係表示位元串流之結構之例的圖。

圖14之A表示將 Base stream 與 Dependent stream 多工於1條位元串流時之結構。圖14之A之位元串流的結構與圖13之位元串流的結構相同。

圖14之B係表示 Base stream 與 Dependent stream 分別包含於不同之合計2條位元串流中時的結構。對與圖14之A所示之部分相對應之部分附上相同之符號。

如圖14之B所示，Base stream 係藉由依序配置 AD#1、畫面 P₁、AD#2、畫面 P₃、AD#3、及畫面 P₅ 而構成。

另一方面，Dependent stream 係藉由依序配置畫面 P₂、畫面 P₄、及畫面 P₆ 而構成。

使用圖11之構成，對以圖14之位元串流作為對象之解碼處理進行說明。

例如，於供給圖14之A之位元串流時，圖11之視訊解碼部152在檢測出AD#1之後，依序讀出Base stream之畫面P₁、Dependent stream之畫面P₂，並分別進行解碼。

又，於供給圖14之B之位元串流時，視訊解碼部152針對Base stream之第1個AU而檢測AD#1，讀出畫面P₁並進行解碼。又，視訊解碼部152針對第2個AU檢測AD#2，讀出畫面P₃並進行解碼。

由於Dependent stream中不存在AD，故而在讀出各畫面時，必須解析位元串流中之語法，並判定畫面P₂、P₄、P₆之邊界。因此，處理變得非常複雜。

此處，為了可容易地檢測畫面之開頭，而考慮與Base stream同樣地，在Dependent stream之各畫面之開頭單純地追加AD的情形。

於此情形時，例如將Base stream與Dependent stream多工於1條位元串流時，視訊解碼部152會將Base stream之AU與Dependent stream之AU認作為其他AU。由此，無法準確地再現虛擬緩衝器模型之動作。

本發明係鑒於上述情況研究而成者，其可容易地檢測Dependent stream之畫面之於位元串流上之邊界。

[解決問題之技術手段]

本發明之一型態之圖像信號解碼裝置，其對包含複數個視點之圖像信號的立體圖像信號經編碼而成之位元串流進行解碼，且其包括如下解碼機構：對1個視點之圖像中所編碼之、表示特定時刻之所有視點之圖像編碼位元串流開始的第1固有資料行進行解碼，並識別上述特定時刻之編碼資料之開始；對其他視點之圖像之編碼位元串流之開頭分別編碼之、表示上述特定時刻之上述其他視點之圖像編碼位元

串流開始的第2固有資料行進行解碼，並識別上述其他視點之圖像信號之編碼資料之開始，從而對位元串流進行解碼。

上述複數個視點之圖像信號係2個視點之圖像信號，其可作為1條位元串流而構成。

複數個視點之圖像信號係2個視點之圖像信號，其可作為2條位元串流而構成。

於上述解碼機構中，可對一個視點之圖像中所編碼之、表示上述特定時刻之所有視點之圖像編碼位元串流開始的上述第1固有資料行進行解碼，並識別上述特定時刻之編碼資料之開始，對另一個視點之圖像之編碼位元串流之開頭所編碼之、表示上述特定時刻之上述另一個視點之圖像編碼位元串流開始的上述第2固有資料行進行解碼，並識別上述另一個視點之圖像信號之編碼資料之開始，從而使位元串流解碼。

於對可準確地開始解碼之位置進行存取時，上述解碼機構可對1個視點之圖像中所編碼之、表示上述特定時刻之所有視點之圖像編碼位元串流開始的上述第1固有資料行進行解碼，並識別上述1個視點之上述特定時刻之編碼資料之開始，對上述其他視點之圖像之編碼位元串流之開頭分別編碼之、表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的上述第2固有資料行進行解碼，並識別上述其他視點之圖像信號之編碼資料之開始，從而自識別出之開始位置起使位元串流解碼。

本發明之一型態之圖像信號解碼方法，其對包含複數個視點之圖像信號的立體圖像信號經編碼而成之位元串流進行解碼，且其包含如下步驟：對1個視點之圖像中所編碼之、表示特定時刻之所有視點之圖像編碼位元串流開始的第1固有資料行進行解碼，並識別上述特定時刻之編碼資料之開始；對其他視點之圖像之編碼位元串流之開頭

分別編碼之、表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行進行解碼，並識別上述其他視點之圖像信號之編碼資料之開始，從而對位元串流進行解碼。

本發明之一型態之程式使電腦執行對包含複數個視點之圖像信號的立體圖像信號經編碼而成之位元串流進行解碼之處理，其使電腦執行包含如下步驟之處理：對1個視點之圖像中所編碼之、表示特定時刻之所有視點之圖像編碼位元串流開始的第1固有資料行進行解碼，並識別上述特定時刻之編碼資料之開始；對其他視點之圖像之編碼位元串流之開頭分別編碼之、表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行進行解碼，並識別上述其他視點之圖像信號之編碼資料之開始，從而對位元串流進行解碼。

本發明之另一型態之圖像信號編碼裝置，其對包含複數個視點之圖像信號的立體圖像信號進行編碼，且其包括如下編碼機構：對1個視點之圖像中表示特定時刻之所有視點之圖像編碼位元串流開始之第1固有資料行進行編碼，並對其他視點之圖像之編碼位元串流之開頭表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行分別進行編碼。

於上述編碼機構中，可將所有視點之編碼資料多工於1條位元串流中。

於上述編碼機構中，可使所有視點之編碼資料分別包含於2條位元串流中。

可使上述複數個視點之圖像信號為2個視點之圖像信號。

本發明之另一型態之圖像信號編碼方法，其對包含複數個視點之圖像信號的立體圖像信號進行編碼，且其包含如下步驟：對1個視點之圖像中表示特定時刻之所有視點之圖像編碼位元串流開始進行第1固有資料行，並對其他視點之圖像之編碼位元串流之開頭表示上述

特定時刻之上述其他視點之圖像編碼位元串流開始分別進行編碼。

本發明之另一型態之程式使電腦執行對包含複數個視點之圖像信號的立體圖像信號進行編碼之處理，且其使電腦執行包含如下步驟之處理：對1個視點之圖像中表示特定時刻之所有視點之圖像編碼位元串流開始之第1固有資料行進行編碼，並對其他視點之圖像之編碼位元串流之開頭表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行分別進行編碼。

於本發明之一型態中，對1個視點之圖像中所編碼之、表示特定時刻之所有視點之圖像編碼位元串流開始的第1固有資料行進行解碼，並識別上述特定時刻之編碼資料之開始。又，對其他視點之圖像之編碼位元串流之開頭分別編碼之、表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行進行解碼，並識別上述其他視點之圖像信號之編碼資料之開始，從而將位元串流解碼。

於本發明之另一型態中，對1個視點之圖像中表示特定時刻之所有視點之圖像編碼位元串流開始之第1固有資料行進行編碼，並對其他視點之圖像之編碼位元串流之開頭表示上述特定時刻之上述其他視點之圖像編碼位元串流開始的第2固有資料行分別進行編碼。

[發明之效果]

根據本發明之一型態，可容易地檢測Dependent stream之於畫面之位元串流上的邊界。

【圖式簡單說明】

圖1係表示圖像資訊編碼裝置之構成例之方塊圖。

圖2係表示圖像資訊解碼裝置之構成例之方塊圖。

圖3係表示多視點編碼裝置之圖。

圖4係表示圖3之多視點編碼裝置之構成例之方塊圖。

圖5係表示MVC之參照圖像之例之圖。

圖6係表示圖4之視訊編碼部之構成與訊框記憶體之方塊圖。

圖7係表示圖4之依存串流編碼部之構成與訊框記憶體之方塊圖。

圖8係表示多視點解碼裝置之構成例之方塊圖。

圖9係表示圖8之視訊解碼部之構成與訊框記憶體的圖。

圖10係表示圖8之依存串流解碼部之構成與訊框記憶體的方塊圖。

圖11係表示虛擬緩衝器模型之例之圖。

圖12係表示圖11之緩衝器之動作之圖。

圖13係表示利用MVC而編碼之位元串流之AU之構成的圖。

圖14A、圖14B係表示位元串流之結構之例的圖。

圖15係表示作為本發明之一實施形態之多視點編碼裝置的一部分之依存串流編碼部之構成例的方塊圖。

圖16A、圖16B係表示位元串流之結構之例之圖。

圖17係表示NAL unit之構成例之圖。

圖18係表示NAL unit之語法之圖。

圖19係表示nal_unit_type之定義之圖。

圖20係表示nal_unit_type之定義之圖。

圖21係簡化表示NAL unit之語法之圖。

圖22係表示包含NAL unit之語法之圖。

圖23係簡化表示圖22之語法之圖。

圖24係表示作為本發明之一實施形態之多視點解碼裝置的一部分之依存串流解碼部之構成例的方塊圖。

圖25係表示虛擬緩衝器模型之例之圖。

圖26係表示對AU之開頭進行存取之方法之例的圖。

圖27係對語法解析部之動作進行說明之流程圖。

圖28係對先前之解析處理進行說明之流程圖。

圖29係表示利用MVC編碼3個視點之影像而獲得之位元串流之AU之構成的圖。

圖30A、圖30B係表示位元串流之結構之例的圖。

圖31A、圖31B係表示位元串流之結構之例的圖。

圖32係表示電腦之構成例之方塊圖。

【實施方式】

[多視點編碼裝置之構成]

圖15係表示作為本發明之一實施形態之多視點編碼裝置的一部分，即依存串流編碼部之構成例的方塊圖。

於本發明之一實施形態之多視點編碼裝置中，對Base stream中表示AU之邊界的資訊進行編碼，對Dependent stream中表示Dependent stream之畫面間之邊界的資訊進行編碼。

藉此，多視點解碼裝置於對立體圖像之位元串流進行解碼時，可藉由簡單之處理而判定AU之邊界等，且可根據虛擬緩衝器模型而進行解碼動作。

多視點編碼裝置具有與圖4之構成相同之構成。圖15表示圖4之依存串流編碼部55之構成例。本發明之一實施形態之多視點編碼裝置41之視訊編碼部52，係具有與圖6之構成相同的構成。對相同之構成附上相同之符號。

除了設有DD編碼部201之方面、以及將DD編碼部201之輸出供給至可逆編碼部86之方面以外，圖15所示之依存串流編碼部55之構成係與圖7之構成為相同之構成。

DD編碼部201於Dependent stream之各畫面之開頭插入被稱為DD(Dependent delimiter，依存定界符)(View and dependency representation delimiter NAL unit)之NAL unit。DD與AD同樣為固有資

料行，但具有與AD不同之值。

可逆編碼部86將藉由DD編碼部201而生成之DD之NAL unit插入至各畫面之開頭後進行編碼，並輸出插入有DD之Dependent stream。

圖16係表示藉由包含圖15之依存串流編碼部55之多視點編碼裝置41而生成之位元串流之結構之例的圖。

圖16之A表示將Base stream與Dependent stream多工於1條位元串流時之結構之例。

圖16之B表示使Base stream與Dependent stream分別包含於不同之合計2條位元串流中時之結構之例。對與圖16之A所示部分相對應的部分附上相同之符號。

生成圖16之A之位元串流時，多視點編碼裝置41於對AD#1進行編碼之後，對Base stream之畫面P₁進行編碼。其次，多視點編碼裝置41於對Dependent stream之畫面進行編碼之前對DD#1進行編碼，其後對Dependent stream之畫面P₂進行編碼。

將處理對象切換為下一AU時，多視點編碼裝置41對AD#2進行編碼，並對Base stream之畫面P₃進行編碼。其次，多視點編碼裝置41於對Dependent stream之畫面進行編碼之前對DD#2進行編碼，其後對Dependent stream之畫面P₄進行編碼。

同樣地，多視點編碼裝置41於對Dependent stream之畫面進行編碼之前對DD進行編碼，並於DD之後立即配置Dependent stream之畫面。

於該例中，亦於AU之開頭配置AD。第1個AU包含AD#1、畫面P₁、DD#1、及畫面P₂，第2個AU包含AD#2、畫面P₃、DD#2、及畫面P₄。第3個AU包含AD#3、畫面P₅、DD#3、及畫面P₆。

生成圖16之B之位元串流時，多視點編碼裝置41針對Base stream而對AD#1進行編碼之後，對Base stream之畫面P₁進行編碼。將處理

對象切換為下一AU時，多視點編碼裝置41對AD#2進行編碼，並對Base stream之畫面P₃進行編碼。

又，多視點編碼裝置41針對Dependent stream而對DD#1進行編碼之後，對Dependent stream之畫面P₂進行編碼。將處理對象切換為下一AU時，多視點編碼裝置41對DD#2進行編碼，並對Dependent stream之畫面P₄進行編碼。

同樣地，多視點編碼裝置41於對Dependent stream之畫面進行編碼之前對DD進行編碼，並於DD之後立即配置Dependent stream之畫面。

如圖16之B所示，Base stream係藉由依序配置AD#1、畫面P₁、AD#2、畫面P₃、AD#3、及畫面P₅而構成。

另一方面，Dependent stream係藉由依序配置DD#1、畫面P₂、DD#2、畫面P₄、DD#3、及畫面P₆而構成。

於圖16之B之例中，亦為第1個AU包含AD#1、畫面P₁、DD#1、及畫面P₂，第2個AU包含AD#2、畫面P₃、DD#2、及畫面P₄。第3個AU包含AD#3、畫面P₅、DD#3、及畫面P₆。

AD表示同時刻之Base stream之畫面與Dependent stream之畫面的資料之開始位置，即特定時刻之所有視點之圖像編碼位元串流的開始位置。又，DD表示Dependent stream之畫面之資料的邊界位置，即Dependent stream之圖像編碼位元串流之開始位置。

[關於語法]

對DD之語法進行說明。

包含AD或各畫面之所有編碼資料係儲存於被稱為NAL unit之單元中，並依特定的順序重排序而構成位元串流。詳細內容係記載於ISO/IEC 14496-10 | ITU-T H.264中。

DD亦儲存於NAL unit中。

圖17係表示NAL unit之構成例的圖。

以位元組單位來處理編碼資料或標頭。NAL_BytesInNALunit係以位元組單位來表示其NAL unit之尺寸之參數，且於識別NAL unit之邊界時為必需者。

作為識別NAL unit之邊界之一個方法，存在對起動碼進行檢索之方法。除此以外之情形時，任何方法均必須對解碼器傳輸NAL unit之尺寸。rsbp_byte[]係NAL unit中所儲存之編碼資料或標頭。

圖18係表示NAL unit之語法的圖。

forbidden_zero_bit為1位元之資料，且始終為0。

nal_ref_idc為2位元之資料，其值為0以外之值時，表示其他NAL unit參照該NAL unit之資料。當nal_ref_idc之值為0時，其他NAL unit並不進行參照。

nal_unit_type為5位元之資料，且表示該NAL unit之內容為何者。

forbidden_zero_bit、nal_ref_idc、nal_unit_type構成圖17之標頭(Header)。

圖19及圖20係表示nal_unit_type之定義的圖。

如圖19所示，nal_unit_type=9表示該NAL unit之內容為AD。

如圖20所示，nal_unit_type=18表示該NAL unit之內容為DD。

若簡化圖18之NAL unit之語法，則成為如圖21所示者。

NAL unit根據圖22所示之語法，附加起動碼，構成被稱為位元組串流之串流。該位元組串流相當於上述位元串流。

圖 22 之 3 byte(0x000001) 之 startcode_prefix (start_code_prefix_one_3bytes)係附加於NAL unit之開頭之起動碼。startcode_prefix為固有碼，於位元組串流中不可出現於其他部位。

藉由檢索位元組串流中之startcode_prefix，而可檢測NAL unit之開頭。又，對檢測出開頭之NAL unit之標頭中之nal_unit_type進行檢

查，藉此可對所需之NAL unit進行存取。

若簡化圖22之語法，則成爲如圖23所示者。

如此一來，無論Dependent stream是與Base stream一併多工於1條位元串流時，還是包含於不同的位元串流時，於Dependent stream之各畫面之邊界處均配置有DD。

於儲存DD之NAL unit之開頭附加起動碼，設定表示內容爲DD之值之18來作爲該nal_unit_type之值。

多視點解碼裝置可藉由檢測起動碼而檢測NAL unit之開頭。又，檢查所檢測出開頭之nal_unit_type之值爲18，藉此可檢測DD即Dependent stream之畫面之邊界(開頭)。

[多視點解碼裝置]

圖24係表示作爲本發明之一實施形態之多視點解碼裝置的一部分之依存串流解碼部之構成例的方塊圖。

本發明之一實施形態之多視點解碼裝置具有與圖8之構成相同之構成。圖24表示圖8之依存串流解碼部115之構成例。本發明之一實施形態之多視點解碼裝置101之視訊解碼部112具有與圖9之構成相同之構成。

圖24所示之依存串流解碼部115之構成除了自可逆編碼解碼部142輸出DD方面以外，與圖10之構成爲相同之構成。

可逆編碼解碼部142對Dependent stream中所含之DD進行解碼，並將其輸出。根據所輸出之DD，藉由上位之應用程式等，而確定隨機存取時之解碼之開始位置等。

對MVC之虛擬緩衝器模型進行說明。

圖25係表示MVC之位元串流係由Base stream與Dependent stream此2條而構成之情形時之虛擬緩衝器模型之例的圖。

將所輸入之Base stream儲存於緩衝器151A中，將Dependent

stream儲存於緩衝器151B中。利用AU單元自緩衝器151A、151B瞬時地讀出位元串流，於視訊解碼裝置152中瞬時地進行解碼。

再者，圖25之緩衝器151A係與圖8之多視點解碼裝置101中之緩衝器111相對應，圖25之緩衝器151B係與圖8之多視點解碼裝置101中之緩衝器114相對應。又，圖25之視訊解碼部152係與圖8之多視點解碼裝置101中之視訊解碼部112及依存串流解碼部115相對應。

例如，於圖16之B之Base stream儲存於緩衝器151A中，且Dependent stream儲存於緩衝器151B中之情形時，於某個時序中，根據AD#1而自緩衝器151A讀出畫面P₁，根據DD#1而自緩衝器151B讀出畫面P₂。所讀出之畫面係藉由視訊解碼部152而進行解碼。

又，於下一時序中，根據AD#2而自緩衝器151A讀出畫面P₃，根據DD#2而自緩衝器151B讀出畫面P₄。

MVC之位元串流由Base stream與Dependent stream之1條而構成之情形時之虛擬緩衝器模型與圖11所示者相同。

例如，於圖16之A之位元串流儲存於圖11之緩衝器151中時，於某個時序中，根據AD#1而自緩衝器151讀出畫面P₁與P₂。

又，於下一時序中，根據AD#2而自緩衝器151讀出畫面P₃與P₄。

其次，參照圖26，對隨機存取時或產生錯誤時等對特定之AU之開頭進行存取之方法的例子進行說明。

首先，將所輸入之位元串流供給至緩衝器211，暫時記憶後供給至起動碼檢測部212與解碼器214。

起動碼檢測部212對具有固有資料行之起動碼進行檢測。於檢測出起動碼時，起動碼檢測部212將其後續資料輸出至語法解析部213。

語法解析部213對自起動碼檢測部212所供給之資料進行解析。由於係起動碼後續之資料，故而自起動碼檢測部212所供給之資料為NAL unit之資料。

於語法解析部213藉由進行NAL unit之標頭等之解析而識別出AU之開始位置時，對解碼器214與緩衝器211輸出表示其係AU之開始位置之信號。

於AD存在時(供給NAL unit之資料時)，語法解析部213檢查是否nal_unit_type=9。於nal_unit_type=9時，其NAL unit之內容如上所述為AD，藉此而檢測出其係AU之開頭。

同樣地，於AD存在時，語法解析部213檢查是否nal_unit_type=18。於nal_unit_type=18時，其NAL unit之內容如上所述為DD，藉此檢測出其係Dependent stream之畫面之開頭。

解碼器214與緩衝器211根據來自語法解析部213之信號，之後利用AU單元如通常般進行各畫面之解碼。

參照圖27之流程圖，對檢測Dependent stream之畫面之開頭的動作進行說明。

此處，如圖16之B所示，Base stream與Dependent stream分別構成不同之位元串流，於Dependent stream中編碼有DD。

於步驟S1中，起動碼檢測部212進行起動碼之檢索。

檢測出起動碼之後，於步驟S2中，語法解析部213檢查是否nal_unit_type=18。

於步驟S2中判定為nal_unit_type=18時，於步驟S3中，語法解析部213檢測附加有所檢測出之起動碼之NAL unit係儲存DD之單元，且係Dependent stream之畫面之開頭。

圖28表示如圖14之B所示由2條位元串流而構成、且如先前之MVC般於Dependent stream中不存在DD之情形時的動作。

如圖28所示，於Dependent stream中不存在DD時，可解碼至條帶頭(slice header)為止，經由多個條件判定，而最終於步驟S23中檢測出其係AU之開頭(畫面之開頭)。

於圖28中，直至步驟S23為止之各判定中所使用之值係儲存於NAL unit之RBSP(Raw Byte Sequence Payload，原始位元組序列負載)(圖17)中之條帶頭所記述之資訊。由於其如上述nal_unit_type=18般，並非為NAL unit之標頭中所記述之資訊，因此解析處理變得複雜。

[3個視點之位元串流]

對3個視點之位元串流進行說明。

圖29係表示利用MVC對3個視點之影像進行編碼而獲得之AU之構成的圖。

如上所述，於MVC之情形時，1個AU中儲存同一時刻之所有view之資料。視點之數量為3時亦相同。

圖29之AU#11包含Base stream之畫面P₁₁、Dependent stream1之畫面P₁₂、及Dependent stream2之畫面P₁₃。

AU#12包含Base stream之畫面P₁₄、Dependent stream1之畫面P₁₅、及Dependent stream2之畫面P₁₆。

AU#13包含Base stream之畫面P₁₇、Dependent stream1之畫面P₁₈、及Dependent stream2之畫面P₁₉。

圖30係表示位元串流之結構之例的圖。

於圖30所示之位元串流中並不包含DD。

圖30之A表示將Base stream、Dependent stream1與Dependent stream2多工於1條位元串流時之結構之例。

圖30之B表示使Base stream、Dependent stream1與Dependent stream2分別包含於不同之合計3條位元串流中時之結構之例。對與圖30之A所示之部分相對應的部分附上相同之符號。

例如，於圖30之A之供給有位元串流之情形時，視訊解碼部152檢測出AD#11之後，依序讀出Base stream之畫面P₁₁、Dependent

stream1之畫面 P_{12} 、Dependent stream2之畫面 P_{13} ，並分別進行解碼。

又，於圖30之B之供給有位元串流之情形時，視訊解碼部152針對Base stream之第1個AU而檢測AD#11，讀出畫面 P_{11} 並進行解碼。又，視訊解碼部152針對第2個AU而檢測AD#12，讀出畫面 P_{14} 並進行解碼。

於Dependent stream1、2中並不存在DD。因此，讀出各畫面時，必須依據圖28之處理而解析位元串流中之語法，以檢測畫面 P_{12} 、 P_{13} 、 P_{15} 、 P_{16} 、 P_{18} 、 P_{19} 之邊界。

圖31係表示藉由本發明之一實施形態之多視點編碼裝置101而生成之位元串流之結構之例的圖。

圖31之A表示將Base stream、Dependent stream1與Dependent stream2多工於1條位元串流時之結構之例。

圖31之B表示使Base stream、Dependent stream1與Dependent stream2分別包含於不同之合計3條位元串流中時之結構之例。

於圖31之A之供給有位元串流之情形時，視訊解碼部152檢測出AD#11之後，依序讀出Base stream之畫面 P_{11} 、Dependent stream1之畫面 P_{12} 、Dependent stream2之畫面 P_{13} ，並分別進行解碼。

於圖31之B之供給有位元串流之情形時，視訊解碼部152針對Base stream之第1個AU而檢測出AD#11之後，對畫面 P_{11} 進行解碼。視訊解碼部152針對第2個AU而檢測出AD#12之後，對畫面 P_{14} 進行解碼。

又，視訊解碼部152針對Dependent stream1而檢測出DD#11之後，對畫面 P_{12} 進行解碼。又，視訊解碼部152檢測出DD#12之後，對畫面 P_{15} 進行解碼。

進而，視訊解碼部152針對Dependent stream2而檢測出DD#21之後，對畫面 P_{13} 進行解碼。又，視訊解碼部152檢測出DD#22之後，對

畫面P₁₆進行解碼。

如此，於多視點解碼裝置101中，根據作為表示AU之邊界的資訊之AD、及作為表示Dependent stream之畫面間之邊界的資訊之DD，而檢測所需之AU並進行解碼，從而顯示立體圖像。

於檢測Dependent stream之某個畫面之開始位置時，無需進行條帶頭之解析，故可實現高速之隨機存取。

上述一系列處理可藉由硬體而執行，亦可藉由軟體而執行。於藉由軟體而執行一系列處理時，構成該軟體之程式係自程式記錄媒體而安裝至專用硬體中組裝之電腦、或通用之個人電腦等中。

圖32係表示藉由程式而執行上述一系列處理之電腦的硬體之構成例的方塊圖。

CPU(Central Processing Unit，中央處理單元)301、ROM(Read Only Memory，唯讀記憶體)302、RAM(Random Access Memory，隨機存取記憶體)303係藉由匯流排304而相互連接。

匯流排304上進而連接有輸入輸出介面305。輸入輸出介面305上連接有包括鍵盤、滑鼠等之輸入部306，及包括顯示器、揚聲器等之輸出部307。又，匯流排304上連接有包括硬碟或非揮發性記憶體等之記憶部308、包括網路介面等之通信部309、及驅動可移動媒體311的驅動器310。

於以如上所述之方式而構成之電腦中，CPU301例如將記憶部308中所記憶之程式經由輸入輸出介面305及匯流排304而加載至RAM303並執行，藉此進行上述一系列處理。

CPU301執行之程式例如記錄於可移動媒體311中，或者經由區域網路、網際網路、數位廣播等有線或無線之傳輸媒體而提供，並被安裝於記憶部308中。

再者，電腦執行之程式可為按照本說明書中說明之順序而時間

序列地進行處理之程式，亦可為並列地、或以進行調用時等之必要的
時序而進行處理之程式。

本發明之實施形態並不限定於上述實施形態，可於不脫離本發
明之主旨之範圍內進行各種變更。

【符號說明】

- 41 多視點編碼裝置
- 51 重排序緩衝器
- 52 視訊編碼部
- 53 訊框記憶體
- 54 重排序緩衝器
- 55 依存串流編碼部
- 56 訊框記憶體
- 57 多工部
- 101 多視點解碼裝置
- 102 3D顯示器
- 111 緩衝器
- 112 視訊解碼部
- 113 訊框記憶體
- 114 緩衝器
- 115 依存串流解碼部
- 116 訊框記憶體
- 201 DD編碼部

申請專利範圍

1. 一種編碼方法，其係用於對第1視點圖像(view)之第1圖像信號及第2視點圖像之第2圖像信號進行編碼，該方法包含如下步驟：
對上述第1圖像信號進行編碼以產生基本串流；及
對上述第2圖像信號進行編碼以產生依存串流，且於上述依存串流中之一畫面之開頭插入表示上述依存串流中之畫面之間的畫面邊界之依存定界符，該依存定界符係作為網路抽象層單元(Network Abstraction Layer unit, NAL unit)，該網路抽象層單元包括具有一值的nal_unit_type，該值不同於存取單元定界符(access unit delimiter)之nal_unit_type之值，該存取單元定界符表示上述基本串流中之存取單元(access unit)之間的邊界。
2. 一種編碼裝置，其對第1視點圖像之第1圖像信號及第2視點圖像之第2圖像信號進行編碼，該裝置包含：
用於對上述第1圖像信號進行編碼以產生基本串流之機構；及
用於對上述第2圖像信號進行編碼以產生依存串流，且於上述依存串流中之一畫面之開頭插入表示上述依存串流中之畫面之間的畫面邊界之依存定界符的機構，該依存定界符係作為網路抽象層單元，該網路抽象層單元包括具有一值的nal_unit_type，該值不同於存取單元定界符之nal_unit_type之值，該存取單元定界符表示上述基本串流中之存取單元之間的邊界。
3. 一種編碼裝置，其對第1視點圖像之第1圖像信號及第2視點圖像之第2圖像信號進行編碼，該裝置包含：
處理單元，其構成為對上述第1圖像信號進行編碼以產生基本串流，對上述第2圖像信號進行編碼以產生依存串流，及於上述依存串流中之一畫面之開頭插入表示上述依存串流中之畫面之

間的畫面邊界之依存定界符，該依存定界符係作為網路抽象層單元，該網路抽象層單元包括具有一值的nal_unit_type，該值不同於存取單元定界符之nal_unit_type之值，該存取單元定界符表示上述基本串流中之存取單元之間的邊界。

圖式

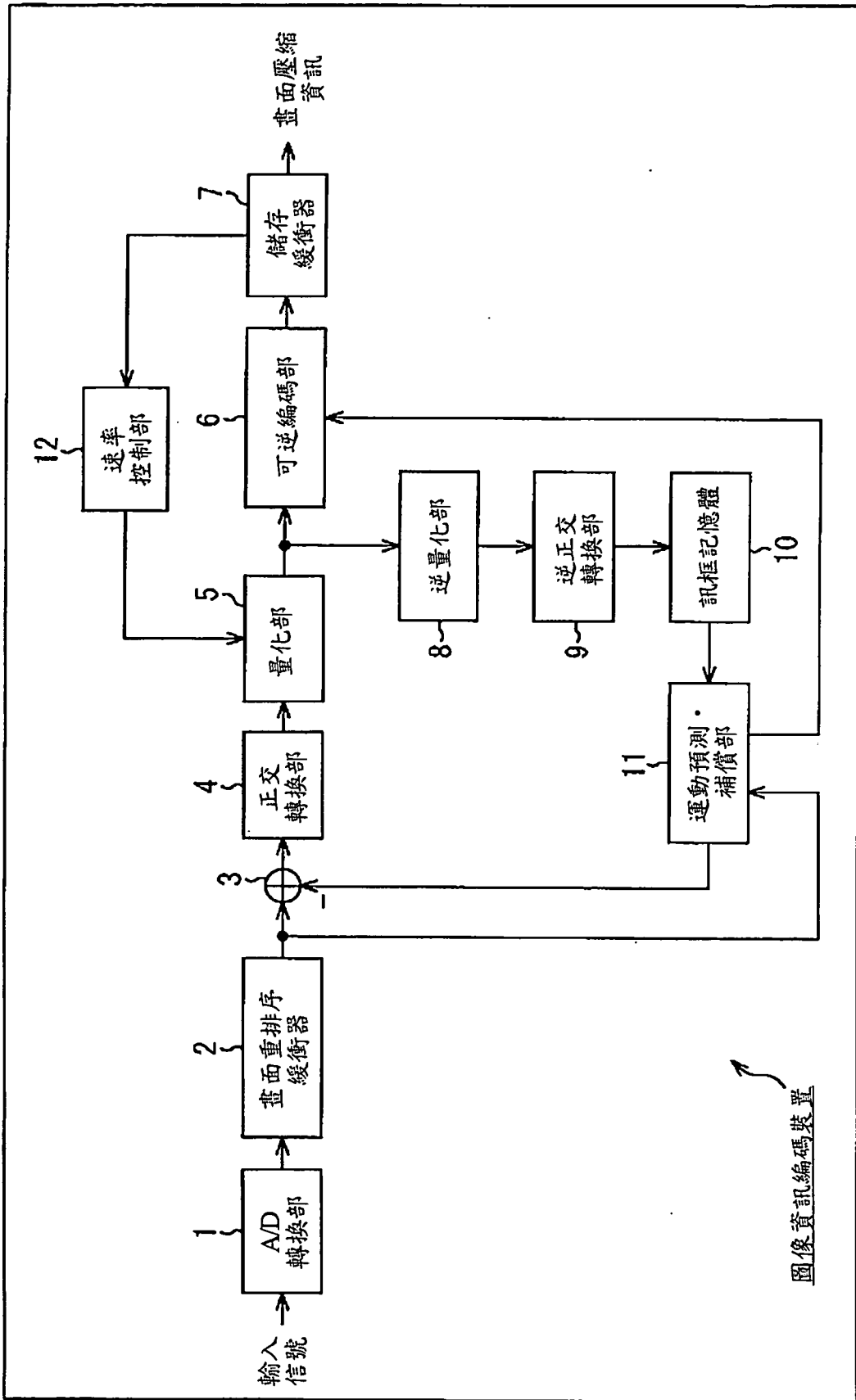


圖 1

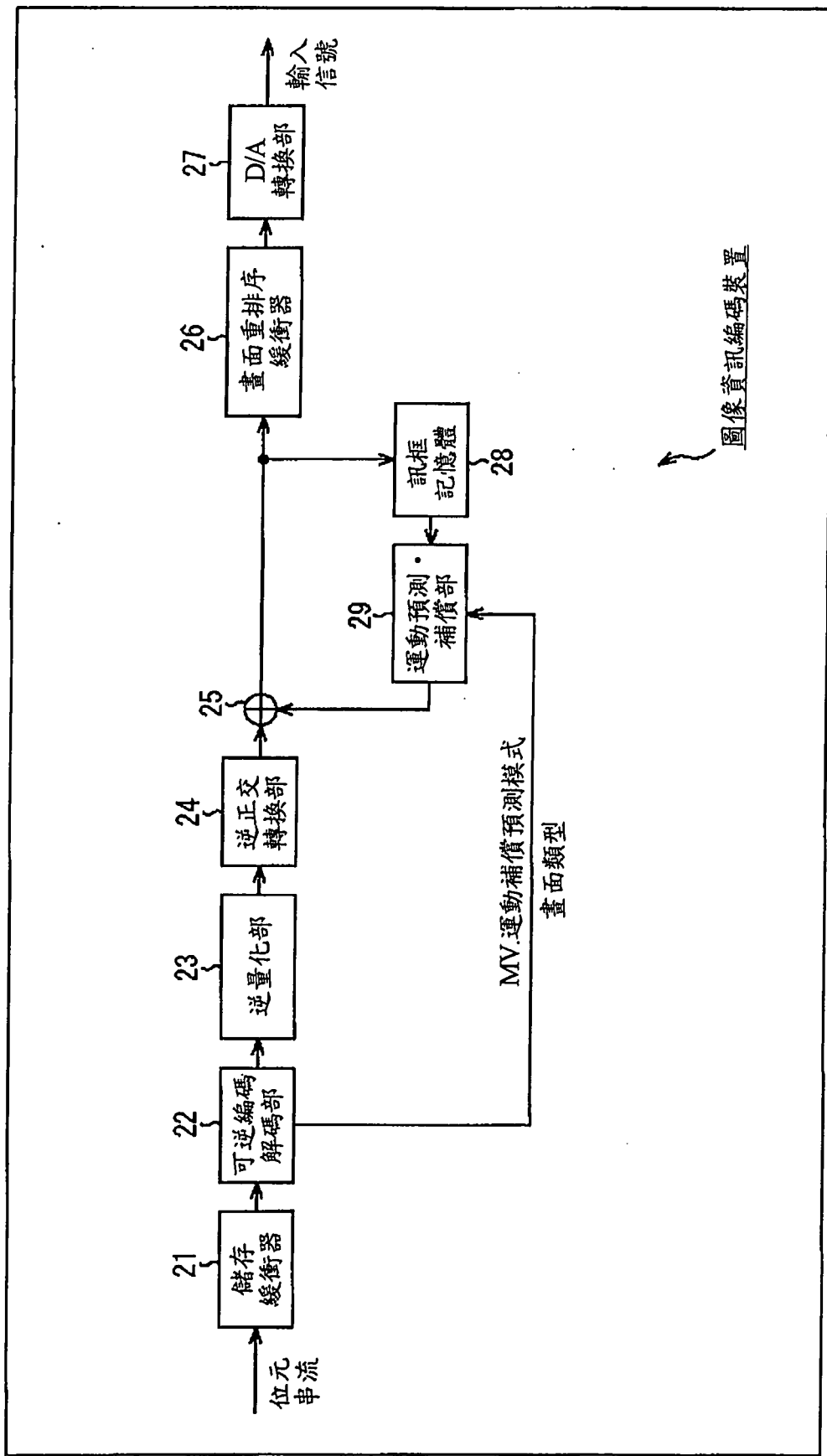


圖 2

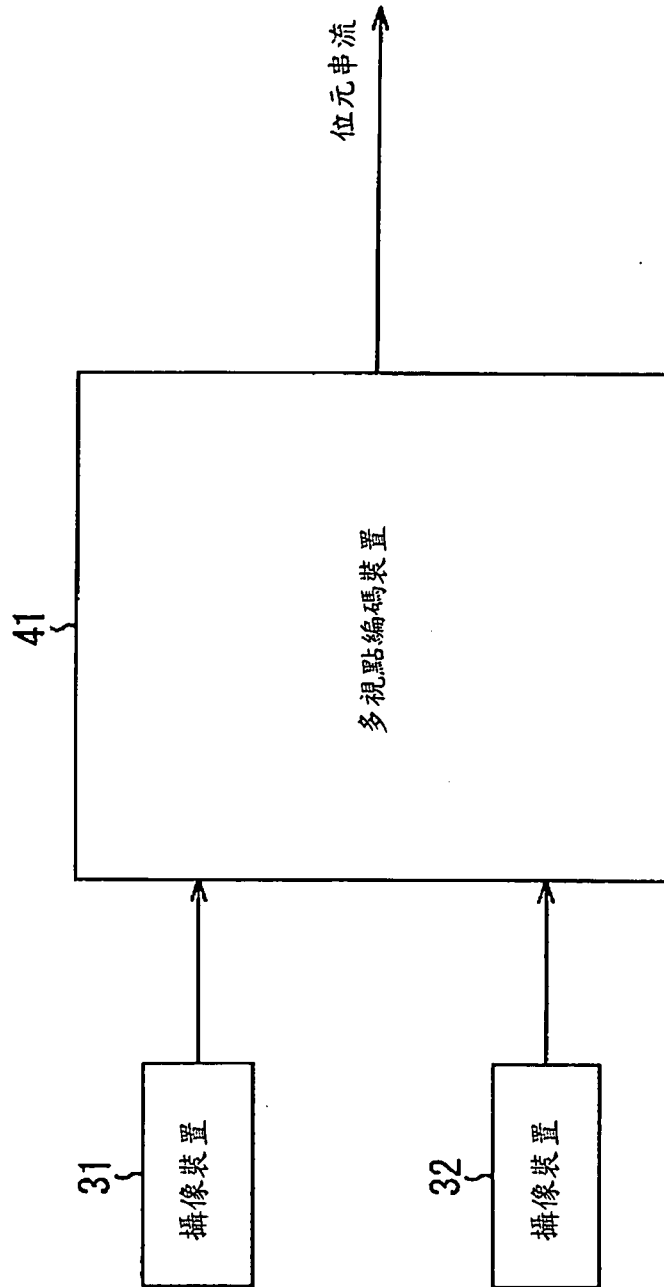


圖 3

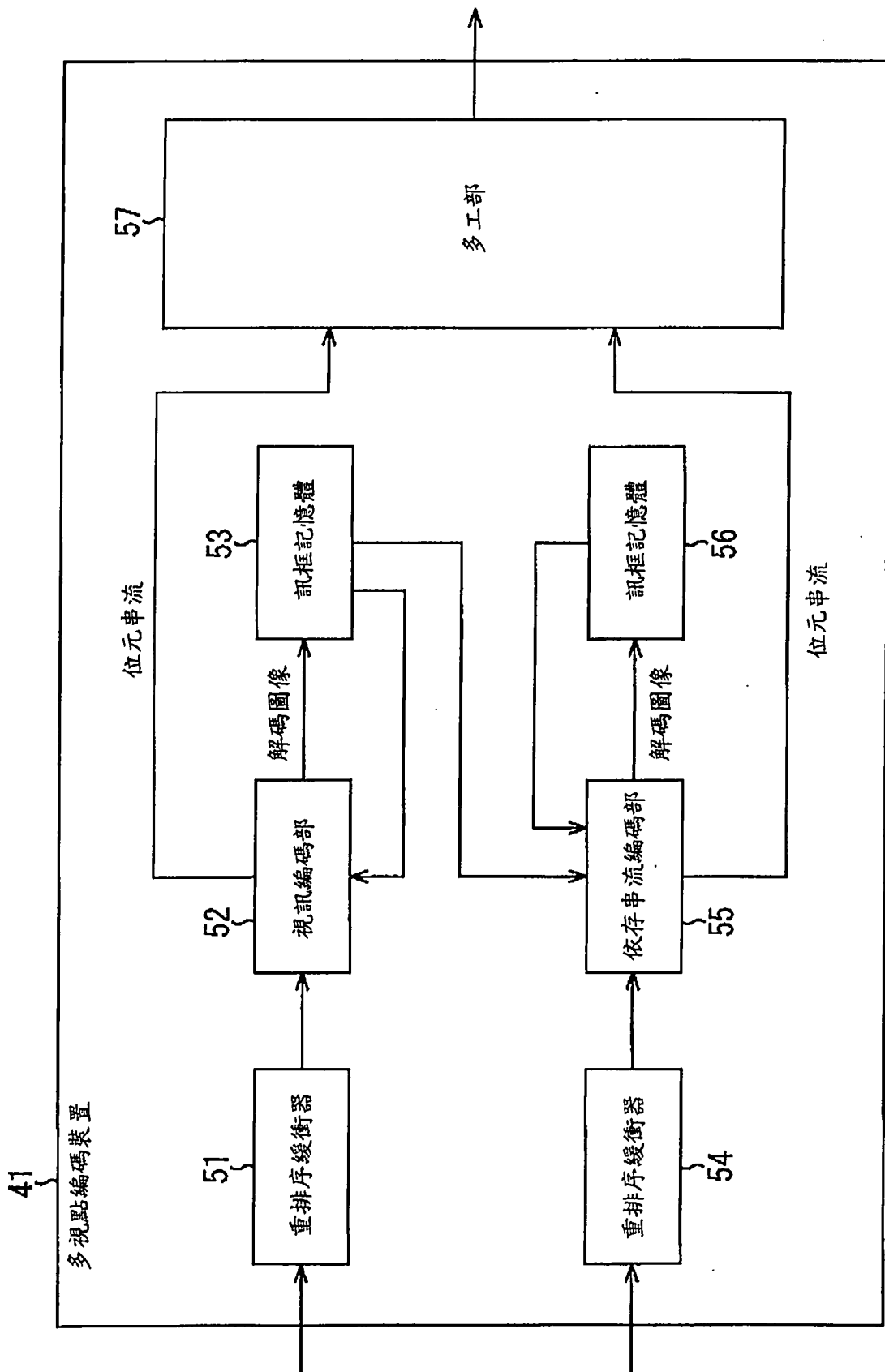


圖 4

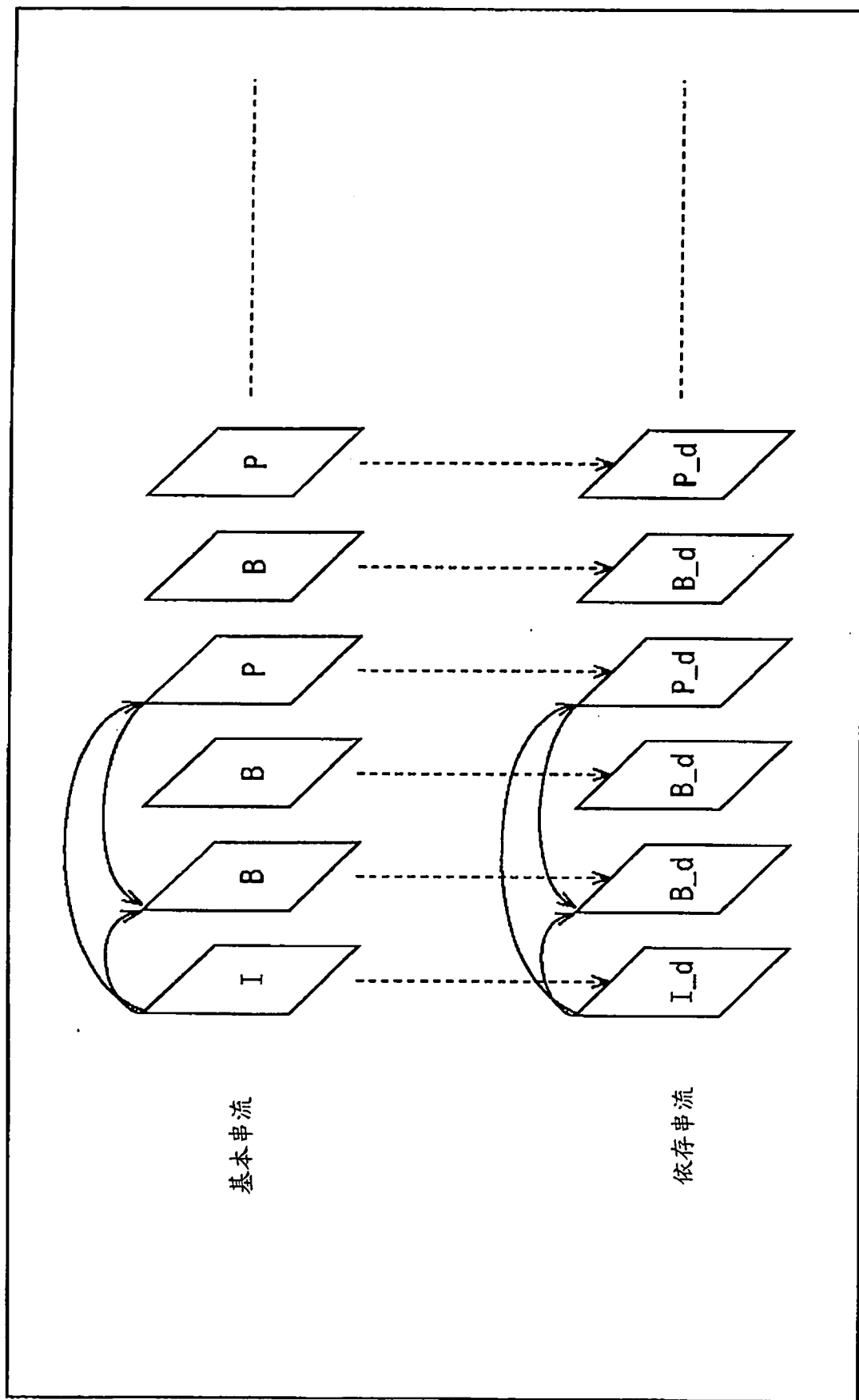


圖 5

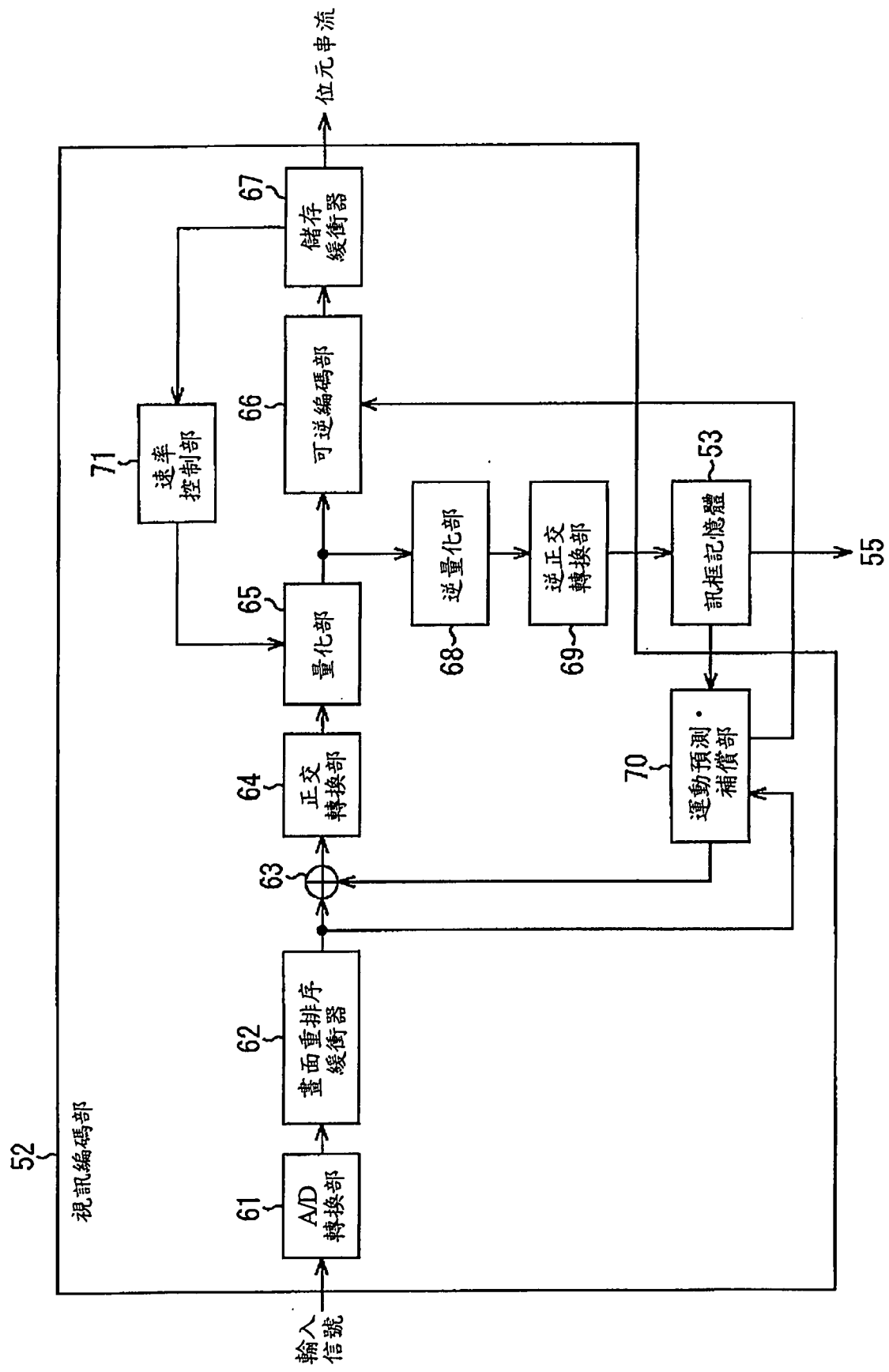


圖 6

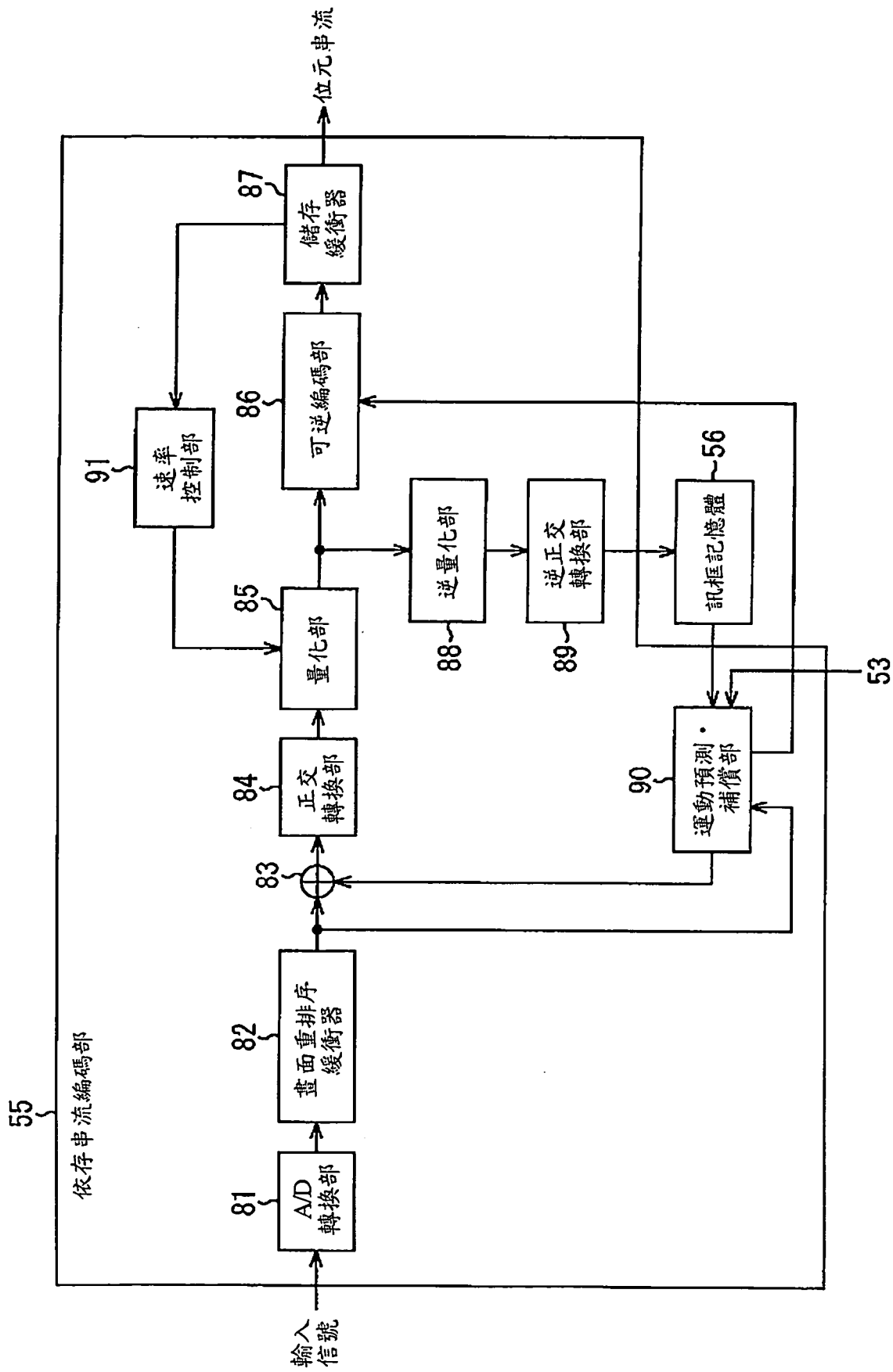


圖 7

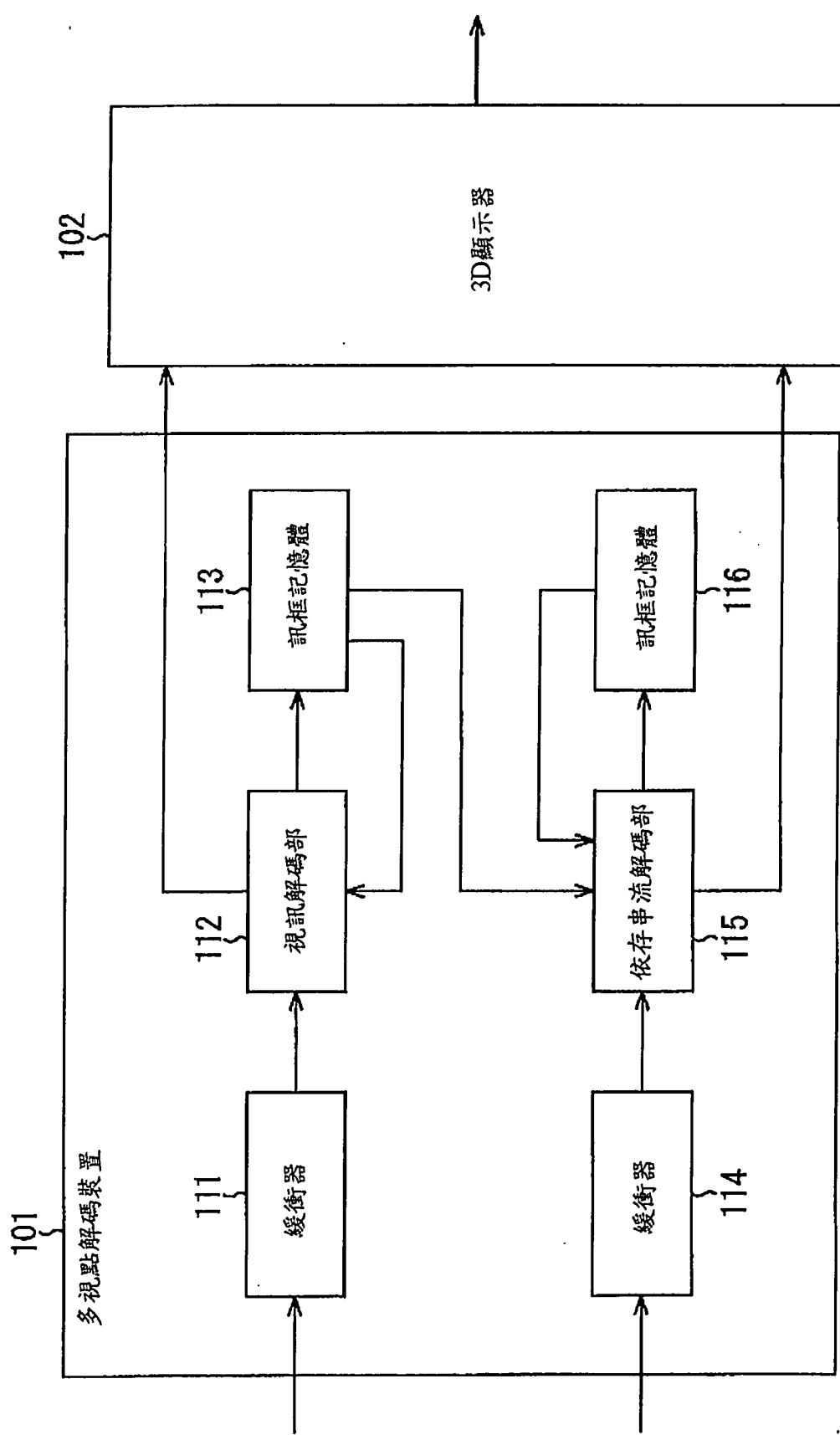


圖8

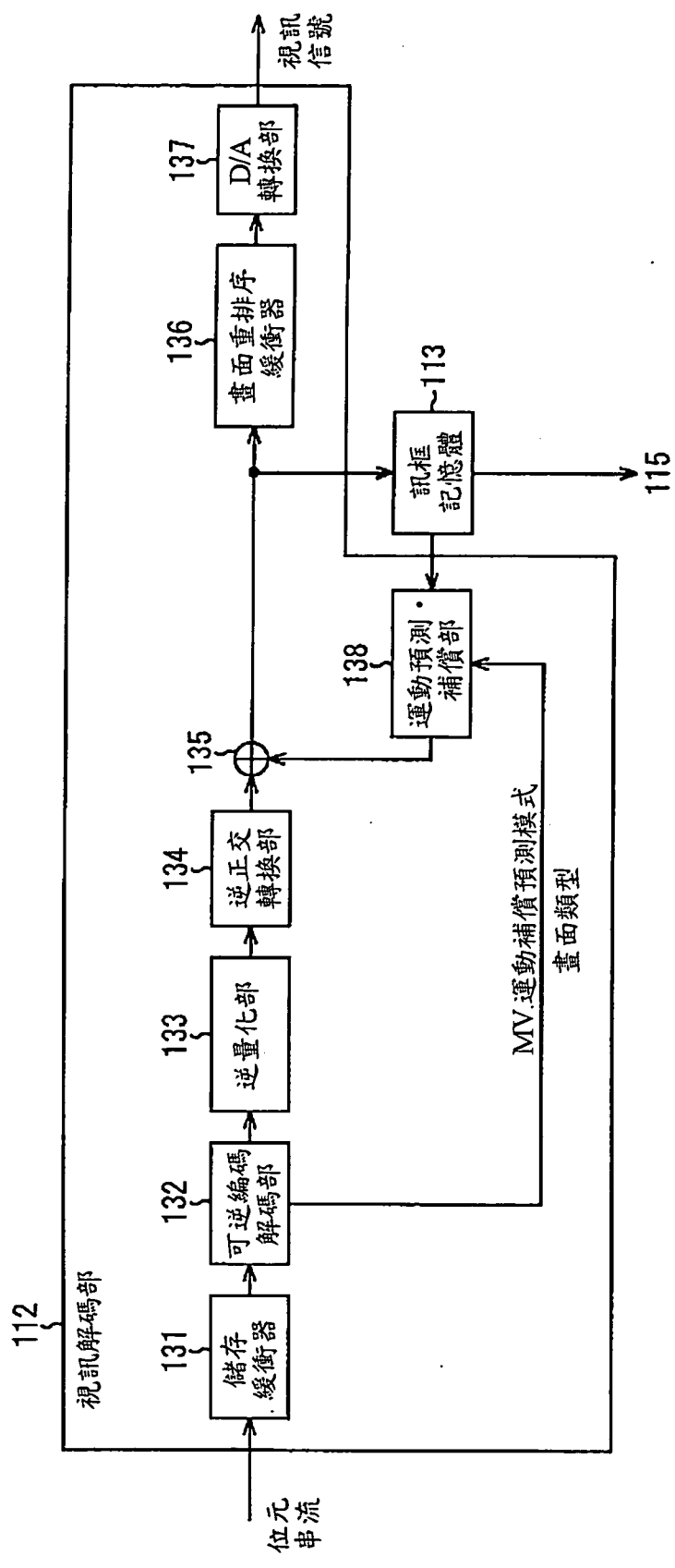


圖 9

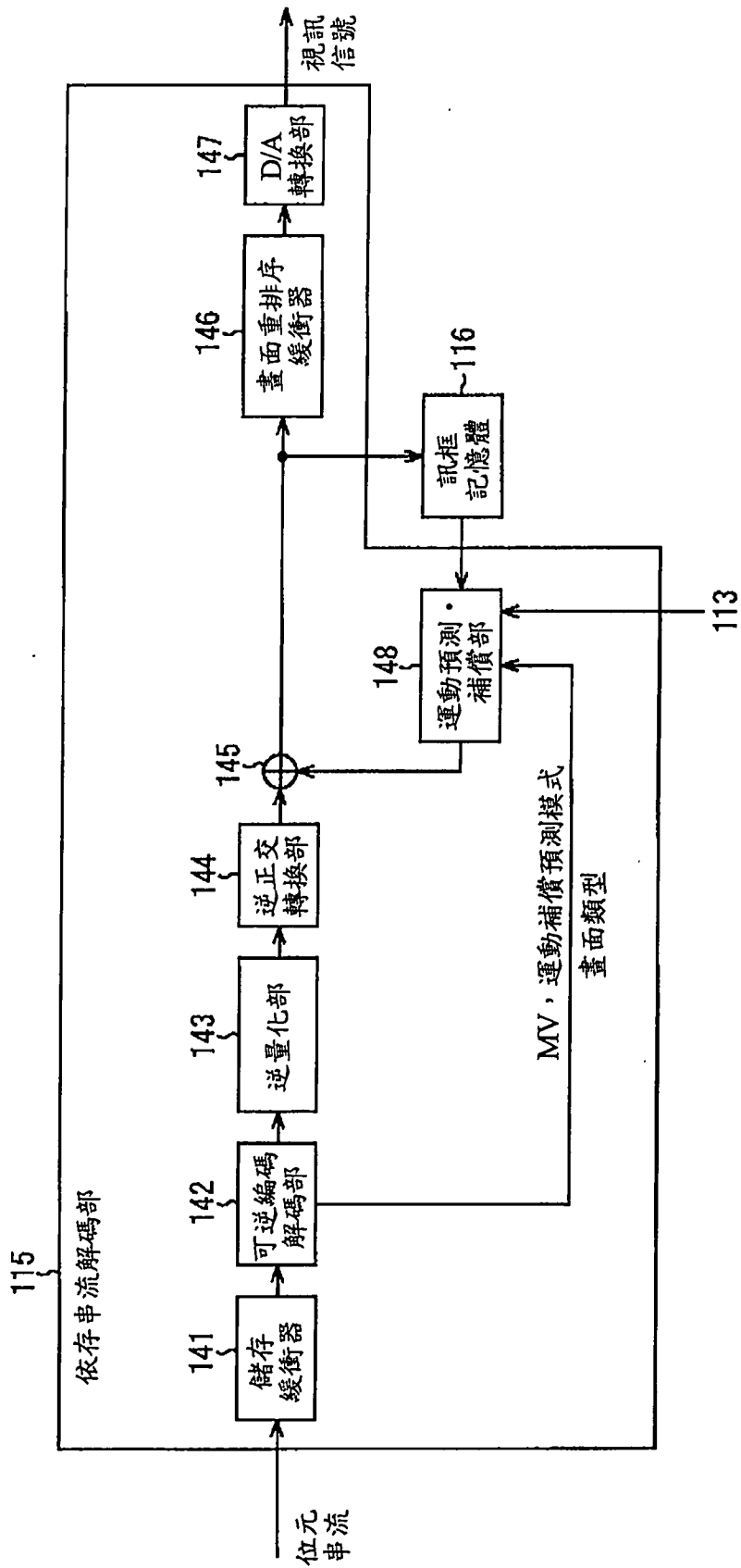


圖 10

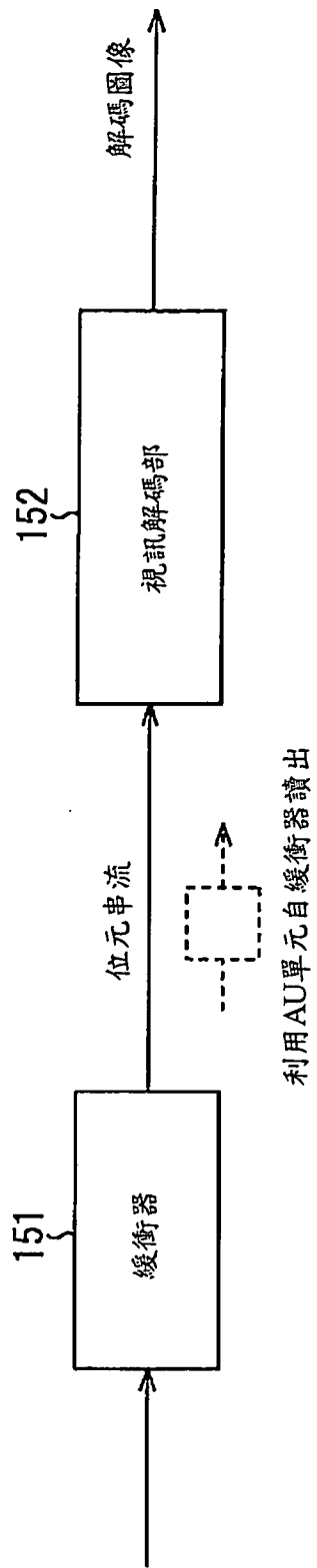


圖 11

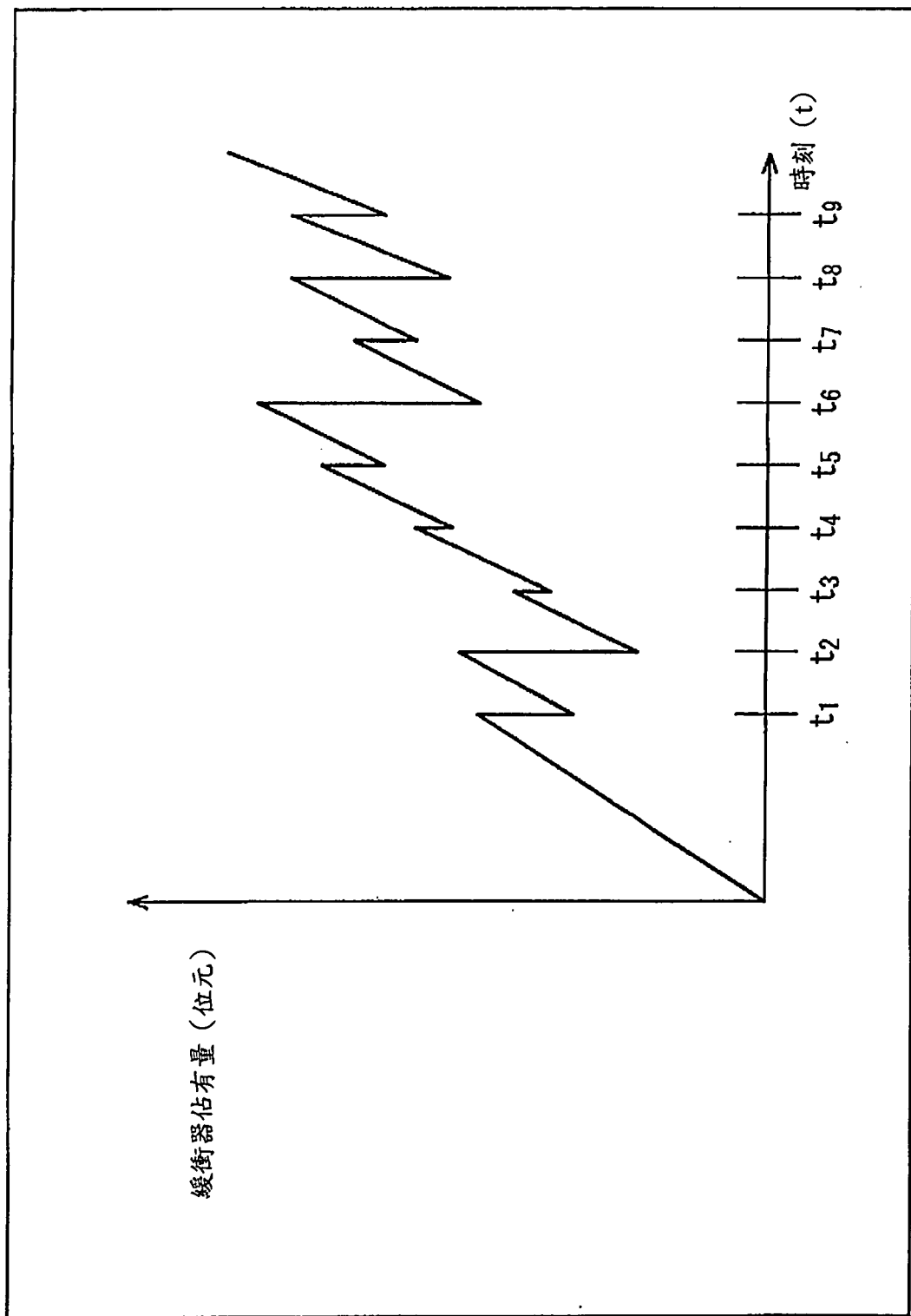


圖 12

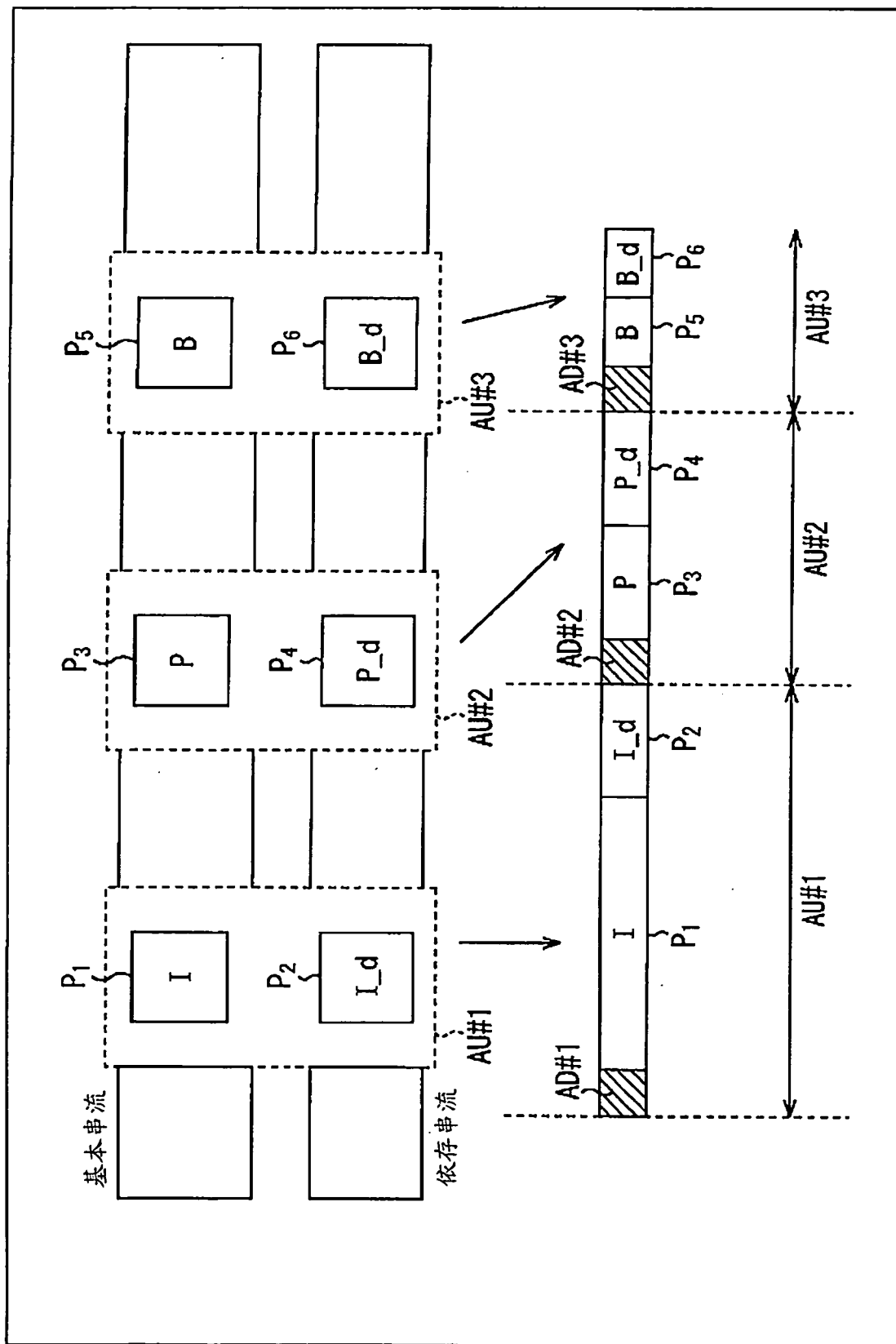


圖 13

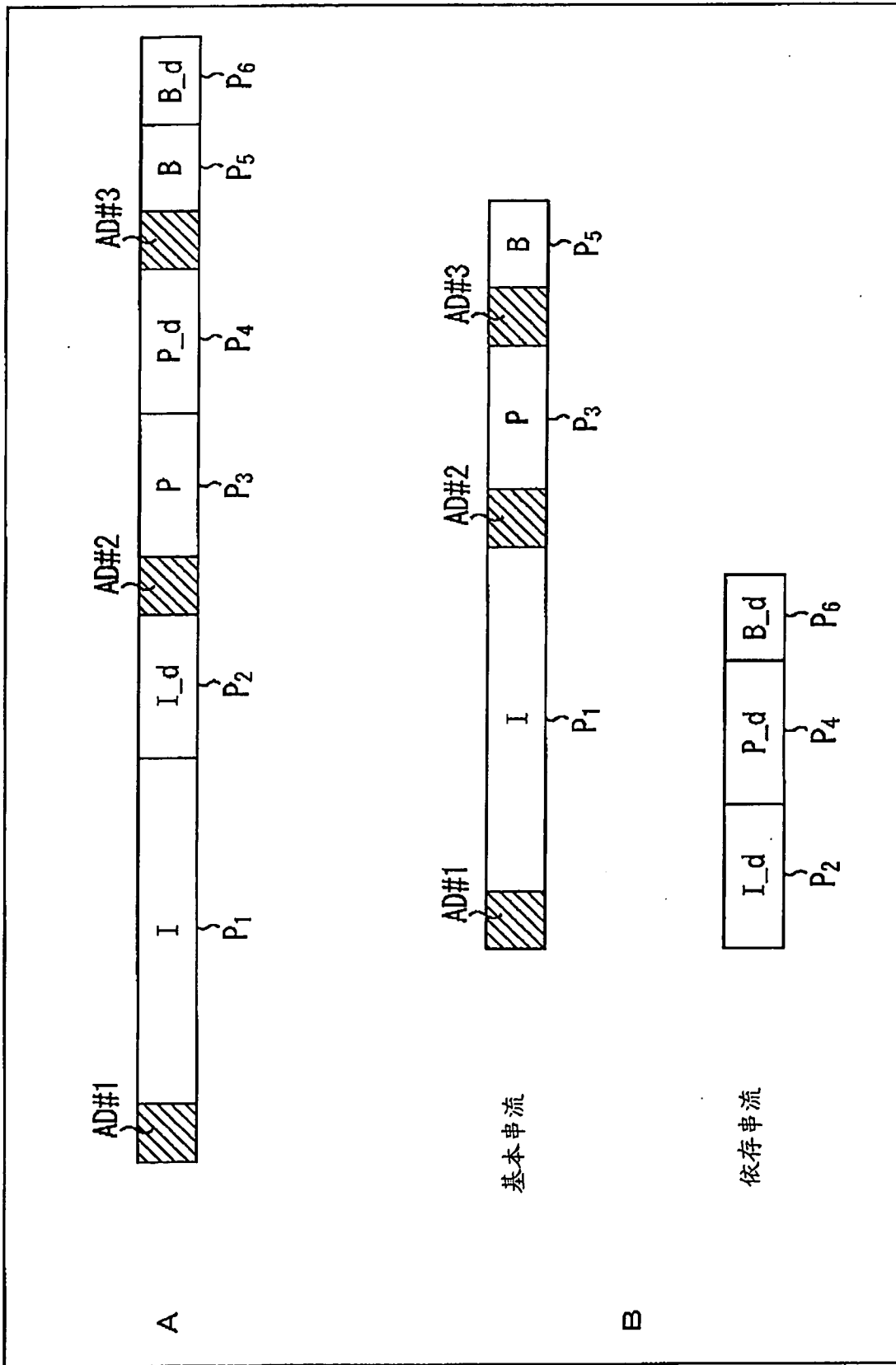


圖 14

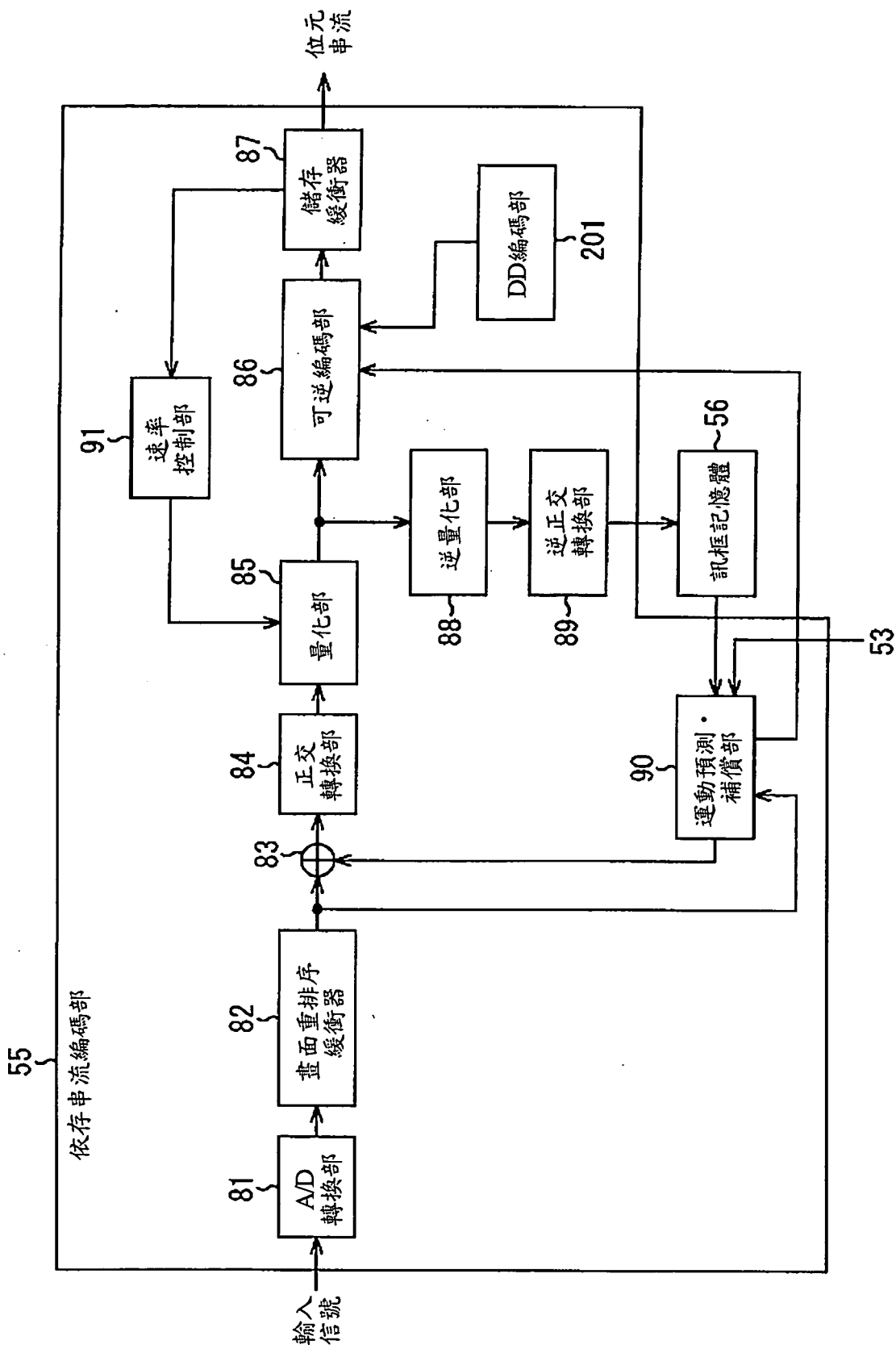


圖 15

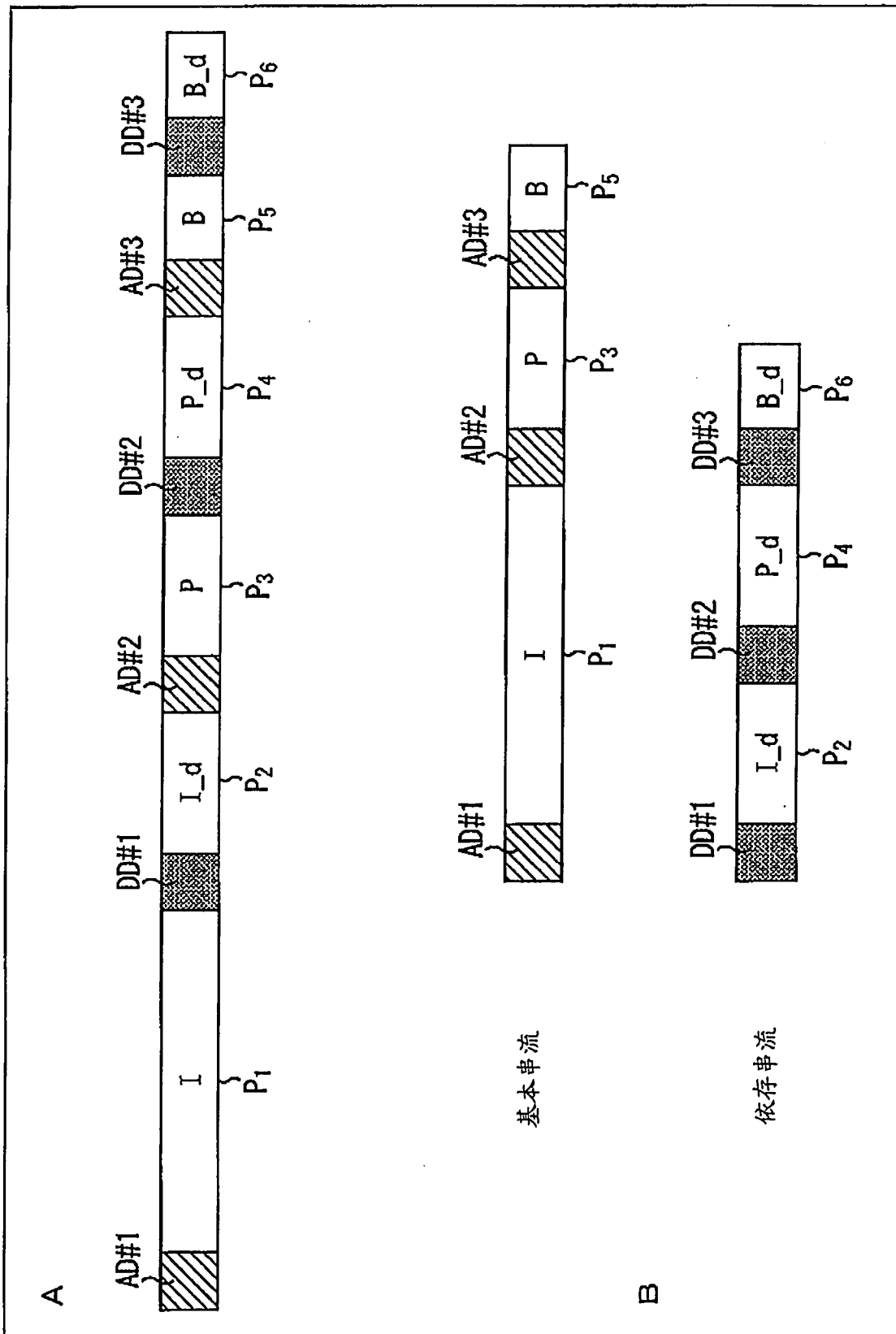


圖 16

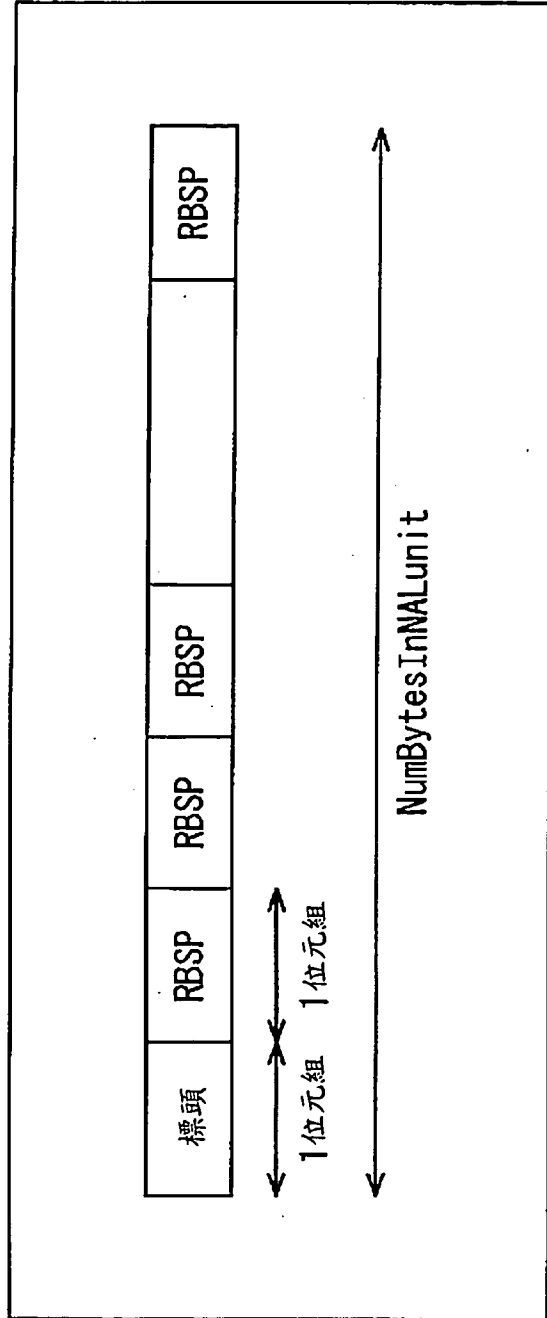


圖 17

| nal_unit(NumBytesInNALunit) { | C | Descriptor |
|------------------------------------------------------------|-----|------------|
| forbidden_zero_bit | All | f(1) |
| nal_ref_idc | All | u(2) |
| nal_unit_type | All | u(5) |
| NumBytesInRBSP = 0 | | |
| nalUnitHeaderBytes = 1 | | |
| if(nal_unit_type == 14 nal_unit_type == 20) { | | |
| svc_extension_flag | All | u(1) |
| if(svc_extension_flag) | | |
| nal_unit_header_svc_extension() /* specified in Annex G */ | All | |
| else | | |
| nal_unit_header_mvc_extension() /* specified in Annex H */ | All | |
| nalUnitHeaderBytes+=3 | | |
| } | | |
| for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) { | | |
| if(i+2 < NumBytesInNALunit && next_bits(24) == 0x000003) { | | |
| rbsp_byte[NumBytesInRBSP++] | All | b(8) |
| rbsp_byte[NumBytesInRBSP++] | All | b(8) |
| i+=2 | | |
| emulation_prevention_three_byte /* equal to 0x03 */ | All | f(8) |
| } else | | |
| rbsp_byte[NumBytesInRBSP++] | All | b(8) |
| } | | |
| } | | |

圖 18

| nal_unit_type | Content of NAL unit and RBSP syntax structure | C | Annex A NAL unit type class | Annex G and Annex H NAL unit type class |
|---------------|------------------------------------------------------------------------------|---------|-----------------------------|-----------------------------------------|
| 0 | Unspecified | | non-VCL | non-VCL |
| 1 | Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp() | 2, 3, 4 | VCL | VCL |
| 2 | Coded slice data partition A slice_data_partition_a_layer_rbsp() | 2 | VCL | not applicable |
| 3 | Coded slice data partition B slice_data_partition_b_layer_rbsp() | 3 | VCL | not applicable |
| 4 | Coded slice data partition C slice_data_partition_c_layer_rbsp() | 4 | VCL | not applicable |
| 5 | Coded slice of an IDR picture slice_layer_without_partitioning_rbsp() | 2, 3 | VCL | VCL |
| 6 | Supplemental enhancement information (SEI) sei_rbsp() | 5 | non-VCL | non-VCL |
| 7 | Sequence parameter set seq_parameter_set_rbsp() | 0 | non-VCL | non-VCL |
| 8 | Picture parameter set pic_parameter_set_rbsp() | 1 | non-VCL | non-VCL |
| 9 | Access unit delimiter access_unit_delimiter_rbsp() | 6 | non-VCL | non-VCL |
| 10 | End of sequence end_of_seq_rbsp() | 7 | non-VCL | non-VCL |
| 11 | End of stream end_of_stream_rbsp() | 8 | non-VCL | non-VCL |
| 12 | Filler data filler_data_rbsp() | 9 | non-VCL | non-VCL |

圖 19

| nal_unit_type | Content of NAL unit and RBSP syntax structure | C | Annex A NAL unit type class | Annex G and Annex H NAL unit type class |
|---------------|------------------------------------------------------------------------------------------------------------|---------|-----------------------------|-----------------------------------------|
| 13 | Sequence parameter set extension seq_parameter_set_extension_rbsp() | 10 | non-VCL | non-VCL |
| 14 | Prefix NAL unit prefix_nal_unit_rbsp() | 2 | non-VCL | suffix dependent |
| 15 | Subset sequence parameter set subset_seq_parameter_set_rbsp() | 0 | non-VCL | non-VCL |
| 16..17 | Reserved | | non-VCL | non-VCL |
| 18 | Dependent delimiter dependent_delimiter_rbsp() | | non-VCL | non-VCL |
| 19 | Coded slice of an auxiliary coded picture without partitioning slice_layer_without_partitioning_rbsp() | 2, 3, 4 | non-VCL | non-VCL |
| 20 | Coded slice extension slice_layer_extension_rbsp() | 2, 3, 4 | non-VCL | VCL |
| 21..23 | Reserved | | non-VCL | non-VCL |
| 24..31 | Unspecified | | non-VCL | non-VCL |

圖 20

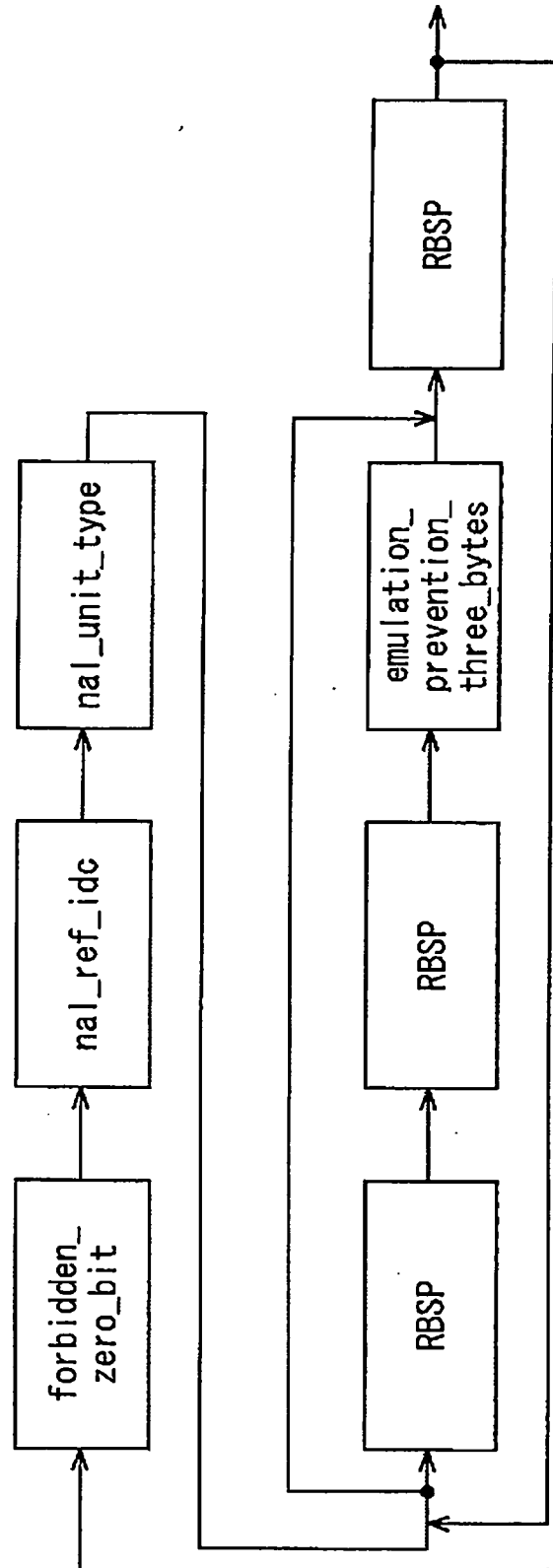


圖 21

| byte_stream_nal_unit(NumBytesInNALunit) { | C | Descriptor |
|--------------------------------------------------------------------------------------------------------|---|------------|
| while(next_bits(24) != 0x000001 && next_bits(32) != 0x00000001) | | |
| leading_zero_8bits /* equal to 0x00 */ | | f(8) |
| if(next_bits(24) != 0x000001) | | |
| zero_byte /* equal to 0x00 */ | | f(8) |
| start_code_prefix_one_3bytes /* equal to 0x000001 */ | | f(24) |
| nal_unit(NumBytesInNALunit) | | |
| while(more_data_in_byte_stream() && next_bits(24) != 0x000001 && next_bits(32) != 0x00000001) | | |
| trailing_zero_8bits /* equal to 0x00 */ | | f(8) |
| } | | |

圖 22

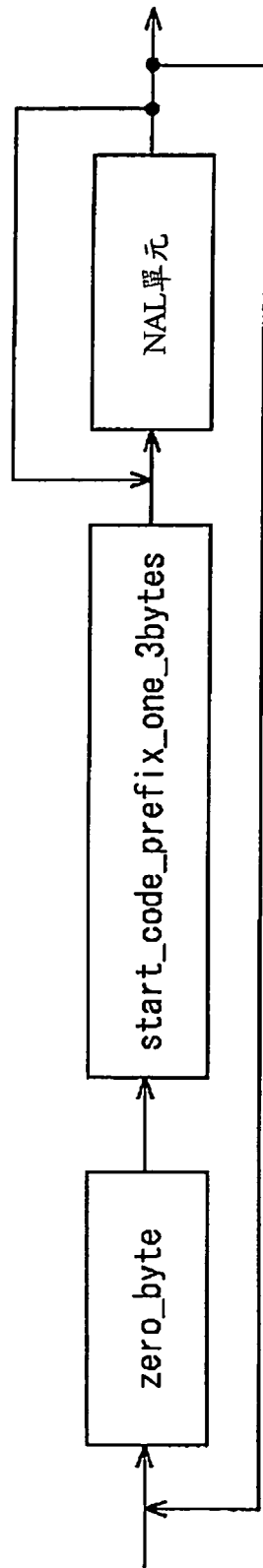


圖 23

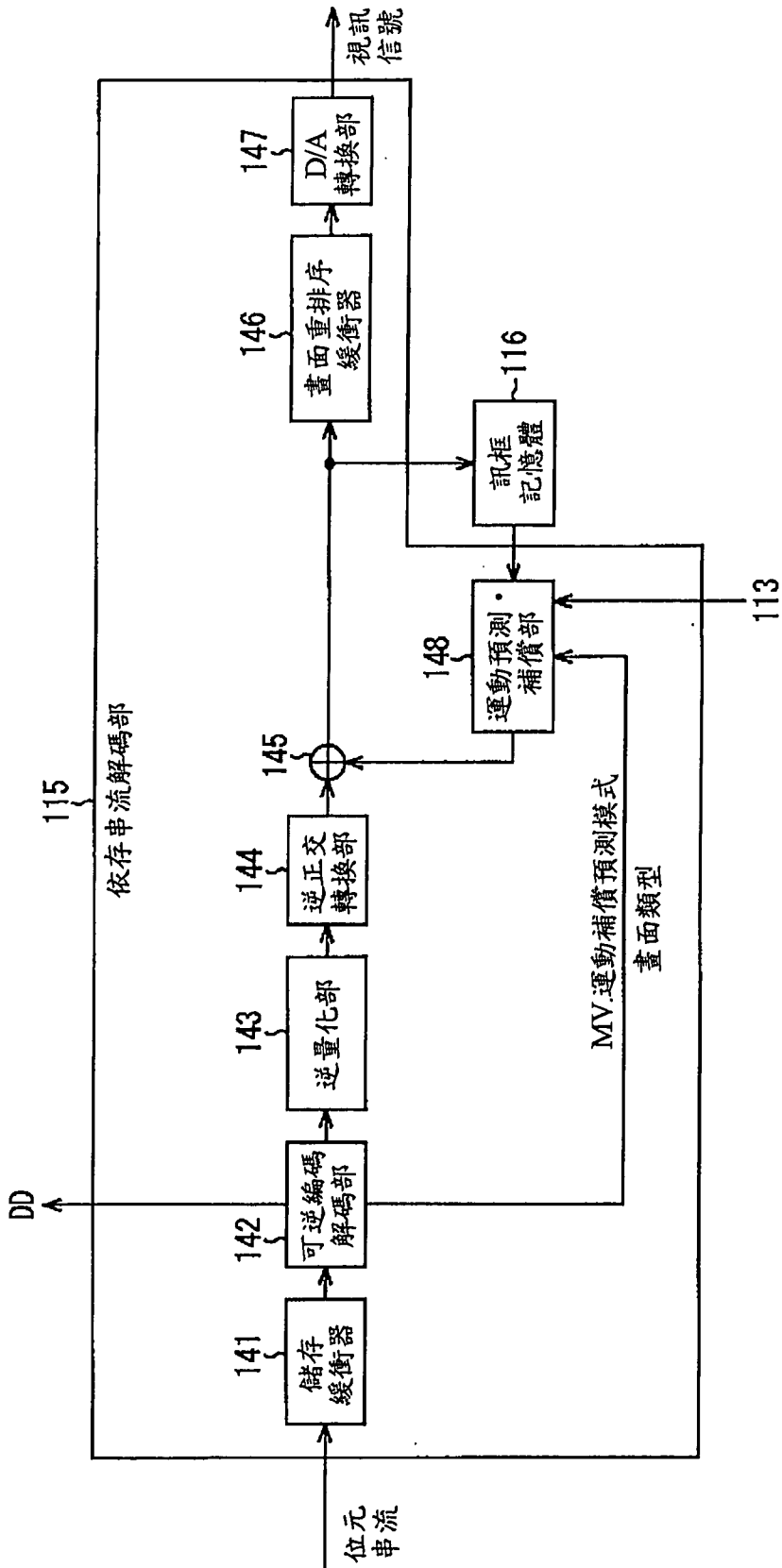


圖 24

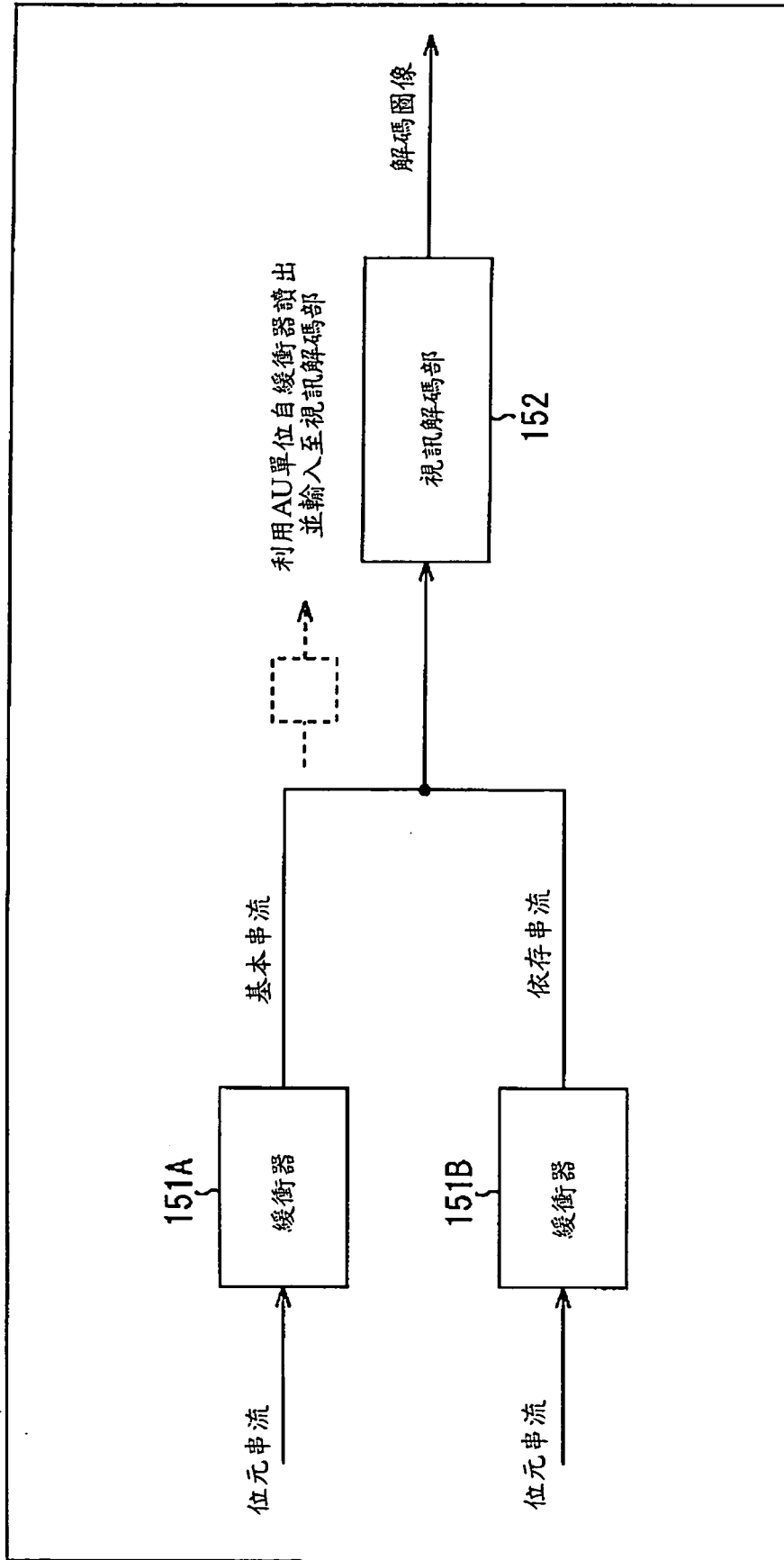


圖 25

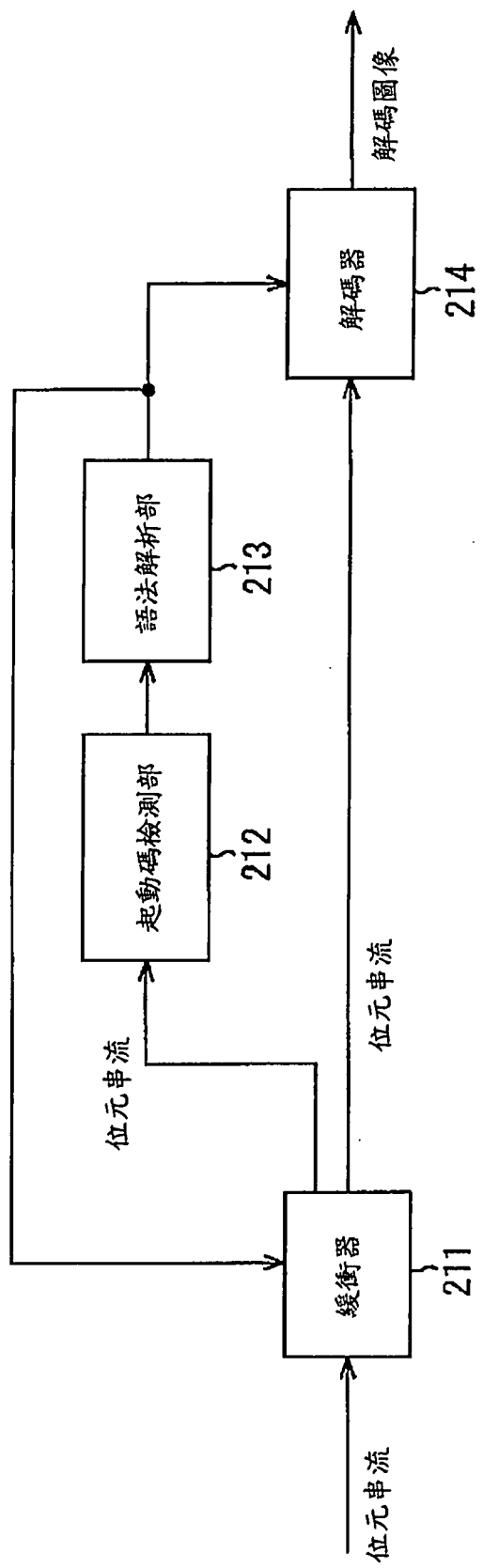


圖 26

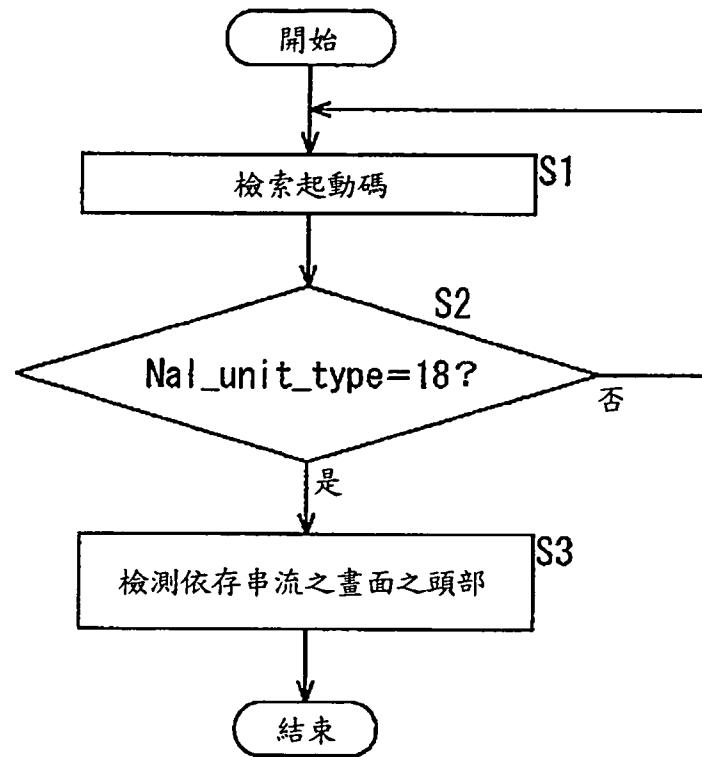


圖 27

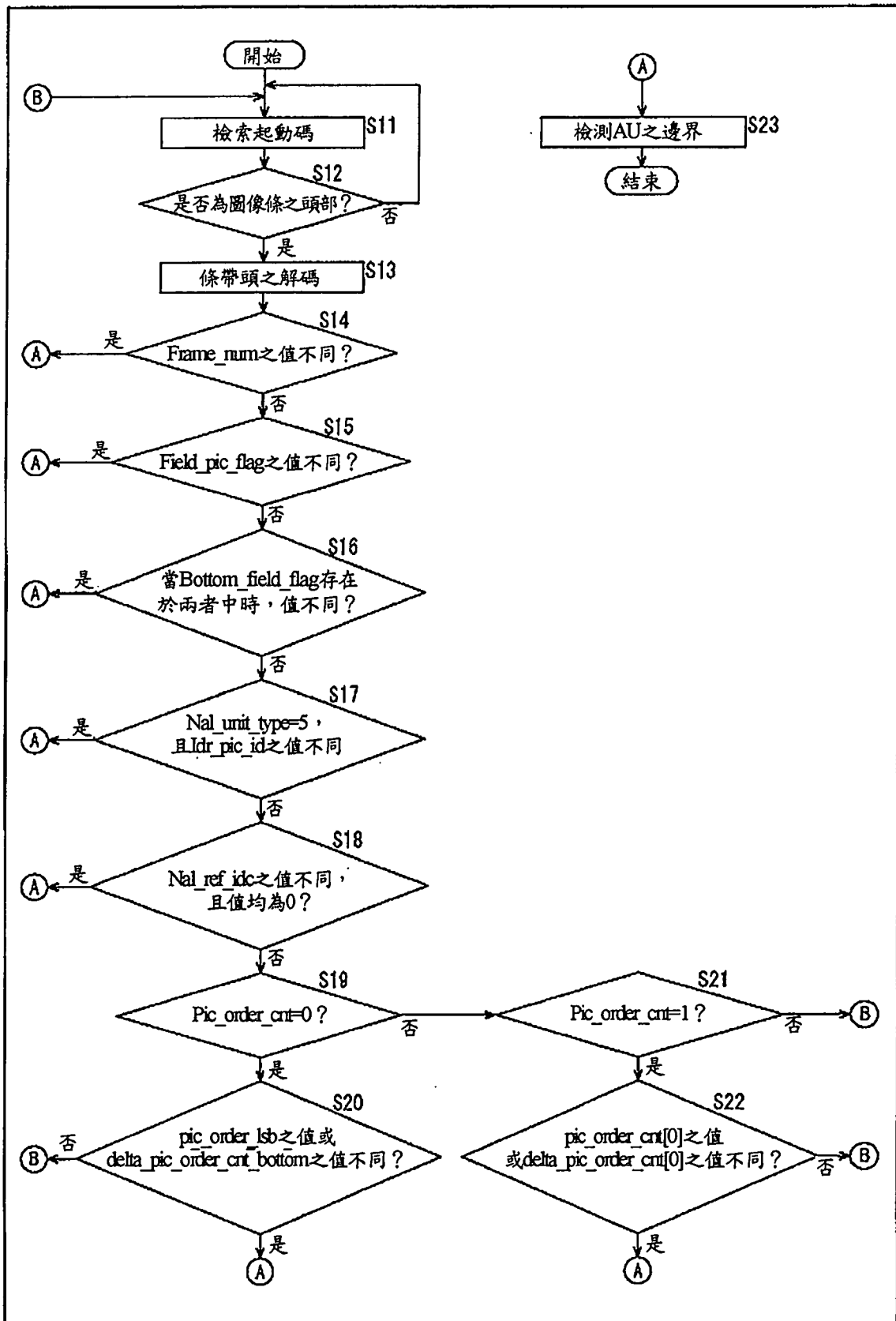


圖 28

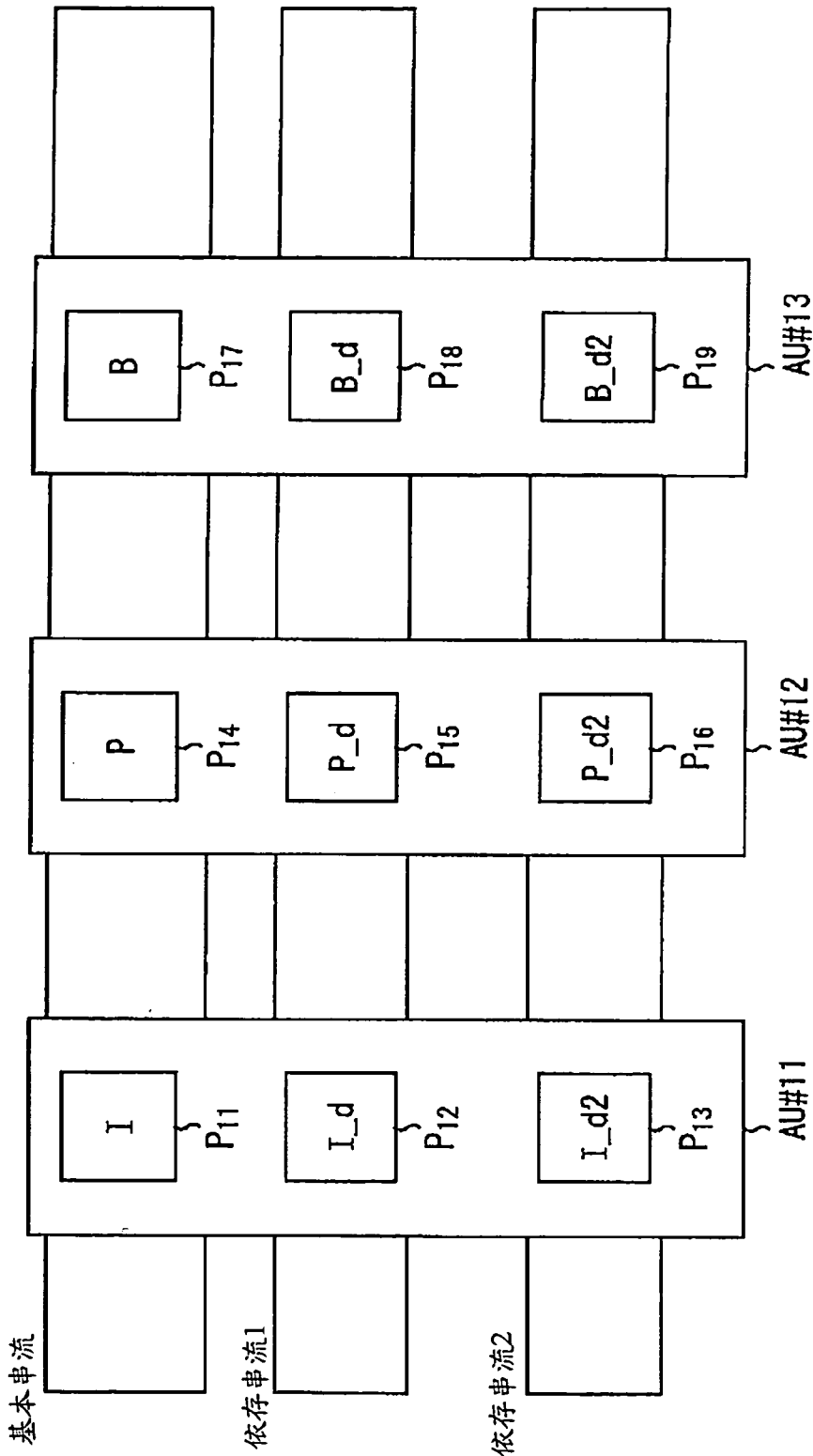


圖 29

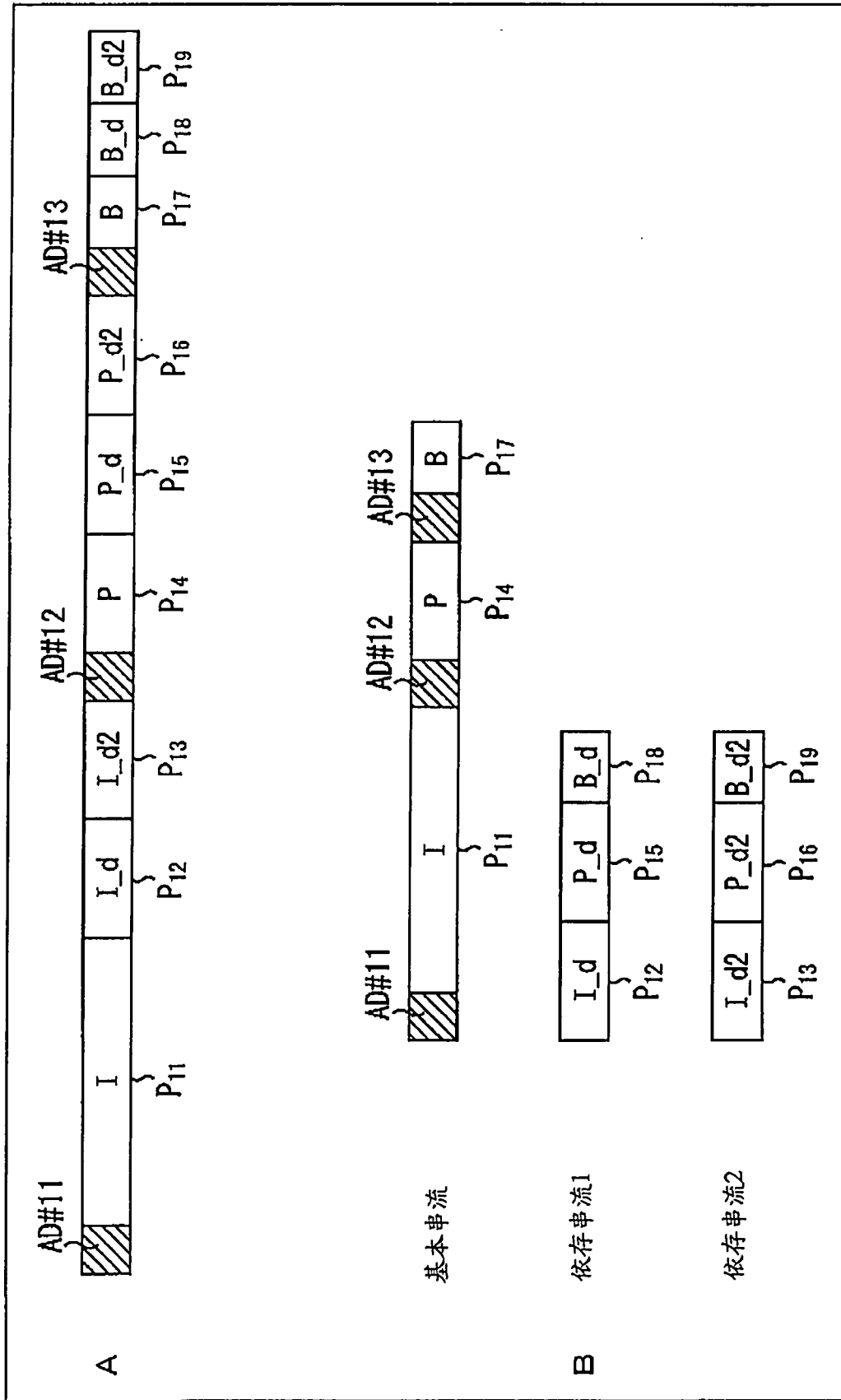


圖 30

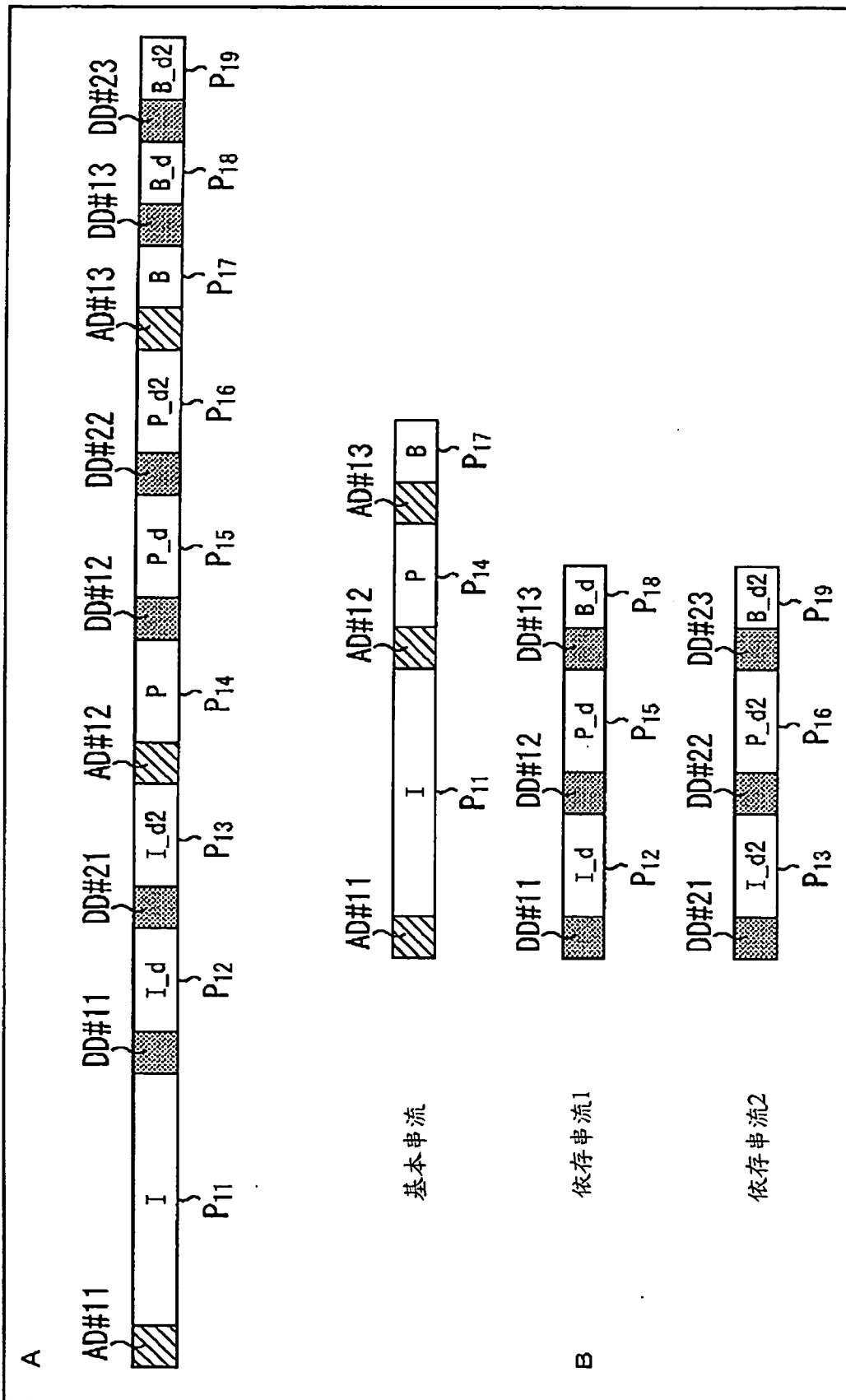


圖 31

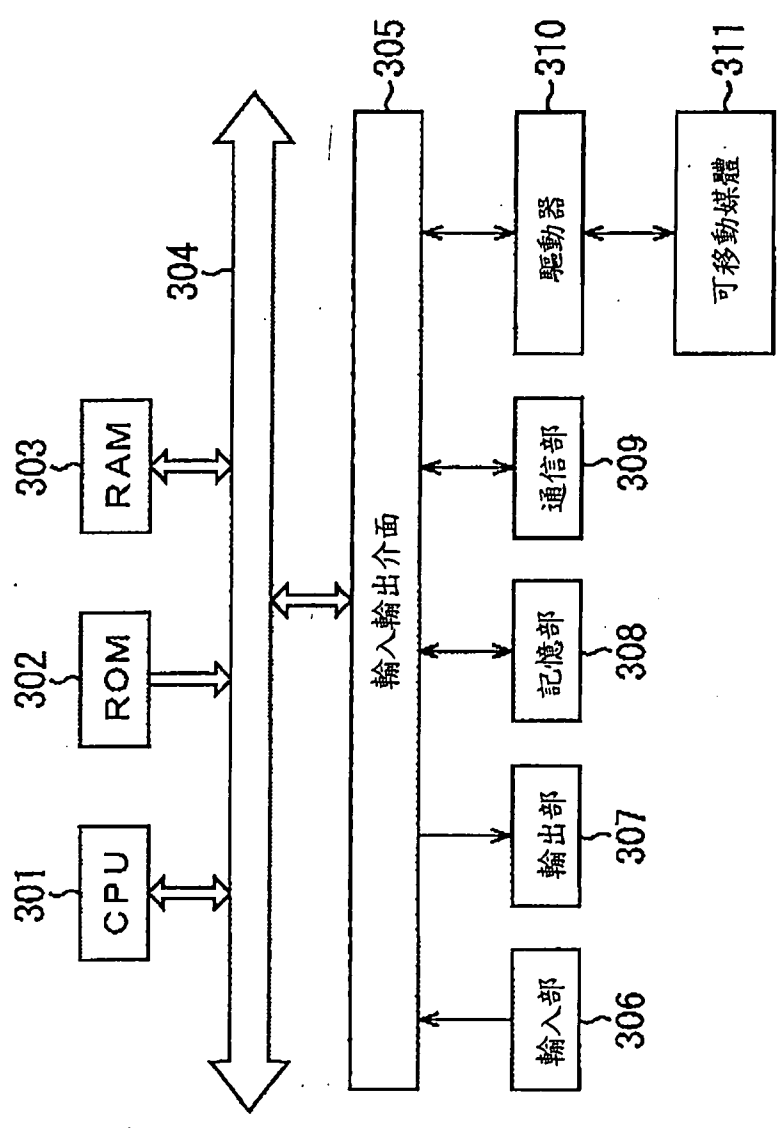


圖 32