

ÖZET

PCM İLE KODLANMIŞ BLOKLAR İÇİN KODLAMA ÜNİTESİ BÖLÜMLÜ SİNYALİZASYONU

5

Bir video kodlama aygıtının, kodlama ağacı ünitesine ait tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden kodlama ağacının tepesindeki ctu_pcm_flag sözdizimini çoğullamasına rağmen, kodlama ağacı birimine ait tüm resim blokları PCM blokları olduğu zaman, kodlama ağacı ünitesinin ayırma düzeninin göstergesi olan en azından cu_split_flag sözdizimi, PCM verilerini verimli bir şekilde iletmek için çoğullanmamaktadır.

İSTEMLER

1. Aşağıdakileri ihtiva etmekte olan bir video kodlama cihazı:

bir görüntü bloğunu dönüştürmek amaçlı olan dönüştürme aracı;

5 dönüşüm aracı tarafından dönüştürülen görüntü bloğu üzerindeki dönüştürülmüş verinin entropi-kodlanması amaçlı olan entropi-kodlama aracı;
görüntü bloğunun PCM-kodlaması için PCM-kodlama aracı;

her bir görüntü bloğu için, ya entropi kodlama aracının çıktı verisini ya da PCM kodlama aracının çıktı verilerini seçmek amaçlı olan çoğullanmış veri seçim aracı; ve

10 çoğullamalı veri seçim aracı tarafından seçilen bir görüntü blokları kümesi olarak bir kodlama ağacı birimine ait olan tüm görüntü bloklarının, PCM blokları olup olmadığını gösteren kodlama ağacı biriminin tepesindeki `ctu_pcm_flag` sözdizimini çoğullama amaçlı olan çoğullama aracı,

15 burada kodlama ağacı birimine ait tüm görüntü blokları PCM bloklardır, çoğullama aracı kodlama ağacı ünitesine ait bir bölme düzeninin modelinin göstergesi olan en azından `cu_split_flag` sözdizimini çoğullamaz ve kodlama ağacı birimine ait tüm görüntü blokları PCM blokları olmadığında, çoğullama aracı, kodlama ağacı ünitesindeki bir görüntü karesinin bir pozisyonuna ve boyutuna, kodu çözülecek olan bir görüntü çerçevesinin bir boyutuna ve bir maksimum PCM blok boyutuna göre `cu_split_flag` sözdizimini belirler ve `cu_split_flag` sözdizimini çoğullar.

2. Aşağıdakileri ihtiva etmekte olan bir video kod çözme cihazı:

kodu çözülecek kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden `ctu_pcm_flag` sözdizimi ihtiva eden bir bit akışının çoğulluğunun çözülmesi için çoğullama çözme aracı;

25 PCM kod çözme aracı, bit akışında bulunan bir görüntü bloğu üzerindeki PCM verisinin PCM-kod çözülmesi amaçlı olan PCM kod çözme aracı; bit akışında bulunan görüntü bloğu üzerinde yer alan dönüştürülmüş verinin entropi- kod çözülmesi amaçlı olan için entropi-kod çözme aracı; ve

kodu çözülecek kod çözme ağacı biriminin `ctu_pcm_flag` sözdiziminin bir değerinin, 30 kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olduğunu belirttiği zaman, kodlama ağacı birimindeki görüntü bloğunun bir pozisyonu ve bir boyutuna, kodu çözülecek olan bir görüntü çerçevesinin bir boyutuna ve bir maksimum PCM blok boyutuna dayanan bir hesaplamanın sonucuna göre kodlama ağacı biriminin bir

bölme düzeninin bir göstergesi olan `cu_split_flag` sözdizimini belirleyen kod çözme kontrol aracı.

3. Aşağıdakileri ihtiva etmekte olan bir video kodlama yöntemi:

bir görüntü bloğunun dönüştürülmesi;

5 dönüştürülmüş görüntü bloğu üzerinde yer alan dönüştürülmüş verilerin entropi-kodlanması;

görüntü bloğunun PCM-kodlanması;

her bir görüntü bloğu için, entropi kodlu veri veya PCM-kodlanmış verilerin seçilmesi; ve

10 bir küme görüntü bloğu olarak kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden kodlama ağacı biriminin tepesinde yer alan `ctu_pcm_flag` sözdiziminin çoğullanması,

burada kodlama ağacı birimine ait tüm görüntü blokları PCM bloklarıdır, çoğullama aracı kodlama ağacı ünitesine ait bir bölme düzeninin göstergesi olan en azından

15 `cu_split_flag` sözdizimini çoğullamaz ve kodlama ağacı birimine ait tüm görüntü blokları PCM blokları olmadığında, `cu_split_flag` sözdizimi, kodlama ağacı ünitesindeki bir görüntü karesinin bir pozisyonuna ve boyutuna, kodu çözülecek olan bir görüntü çerçevesinin bir boyutuna ve bir maksimum PCM blok boyutuna göre belirlenir ve `cu_split_flag` sözdizimi çoğullanır.

20 4. Aşağıdakileri ihtiva eden bir video kod çözme yöntemi:

kodu çözülecek olan kodlama ağacı birimine ait olan tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden `ctu_pcm_flag` sözdizimini içeren bir bit akışının çoğullamasının çözülmesi;

25 bit akışında bulunan bir görüntü bloğu üzerindeki PCM verilerinin PCM-kod çözülmesi;

bit akışında bulunan görüntü bloğu üzerinde yer alan dönüştürülmüş veriyi entropi-kod çözme; ve

30 kodu çözülecek kodlama ağacı biriminin `ctu_pcm_flag` sözdizimi değeri, kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olduğuna işaret ettiğinde, kodlama ağacı biriminin bir bölme düzenini belirten `cu_split_flag` sözdizimini kodlama ağacı birimindeki görüntü bloğunun bir konumu ve büyüklüğüne, kodlanacak bir görüntü çerçevesinin büyüklüğüne ve bir maksimum PCM blok boyutuna bağlı olarak belirlenmesi.

5. Bir bilgisayarın aşağıdakileri çalıştırmasına sebep olan bir video kodlama programı:
bir görüntü bloğunu dönüştürme işlemi;

dönüştürülmüş görüntü bloğu üzerinde yer alan dönüştürülmüş edilmiş veriyi entropi kodlayan bir işlem;

5 görüntü bloğunu kodlayan bir PCM-kodlama süreci;

her bir görüntü bloğu için ya entropiyle kodlanmış verileri ya da PCM-kodlanmış verileri seçme işlemi; ve

bir dizi görüntü bloğu olarak kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığını gösteren kodlama ağacı biriminin tepesindeki `ctu_pcm_flag` sözdiziminin çoğullanması işlemi,

10 burada kodlama ağacı birimine ait tüm görüntü blokları PCM bloklarıdır, bilgisayar kodlama ağacı ünitesine ait bir bölme düzeninin göstergesi olan en azından `cu_split_flag` sözdizimini çoğullamaz ve kodlama ağacı birimine ait tüm görüntü blokları PCM blokları olmadığında, `cu_split_flag` sözdizimi, kodlama ağacı ünitesindeki bir görüntü karesinin bir pozisyonuna ve boyutuna, kodu çözülecek olan bir görüntü çerçevesinin bir boyutuna ve bir maksimum PCM blok boyutuna göre belirlenir ve `cu_split_flag` sözdizimi çoğullanır.

15 kodlayıcı ağaç birimine ait tüm görüntü bloklarının PCM blokları olduğu durumlarda, bilgisayarın, kodlama ağacı birimine ait bir ayırma düzeninin göstergesi olan en azından `cu_split_flag` sözdizimini çoğaltamamasına ve kodlama ağacı birimine ait tüm görüntü blokları olmadığına karar verilir. PCM blokları, `cu_split_flag` sözdizimi, kodlama ağacı birimindeki bir görüntü bloğunun bir konumuna ve büyüklüğüne, kodlanacak bir görüntü çerçevesinin bir büyüklüğüne ve bir maksimum PCM blok boyutuna ve `cu_split_flag` sözdiziminin çoğullanmasına göre belirlenir.

25 6. Bir bilgisayarın aşağıdakileri çalıştırmasına sebep olan bir video kod çözme programı:
kodu çözülecek kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığını belirten `ctu_pcm_flag` sözdizimi ihtiva eden bir bit akışının çoğullamasının çözülme işlemi;

30 bit akışında bulunan bir görüntü bloğu üzerindeki PCM verilerinin bir PCM-kod çözme işlemi;

bit akışında bulunan görüntü bloğu üzerinde yer alan dönüştürülmüş verilerin entropi-kod çözme işlemi; ve

kodu çözülecek kod çözme ağacı biriminin `ctu_pcm_flag` sözdiziminin bir değerinin, kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olduğunu belirttiği zaman, kodlama ağacı birimindeki görüntü bloğunun bir pozisyonu ve bir boyutuna, kodu çözülecek olan bir görüntü çerçevesinin bir boyutuna ve bir maksimum PCM blok boyutuna dayanan bir hesaplamanın sonucuna göre kodlama ağacı biriminin bir bölme düzeninin bir göstergesi olan `cu_split_flag` sözdiziminin belirlenmesi.

TARİFNAME

PCM İLE KODLANMIŞ BLOKLAR İÇİN KODLAMA ÜNİTESİ BÖLÜMLÜ SİNYALİZASYONU

5

Mevcut buluş, bir kodlama ağacı ünitesini otomatik olarak ayırmak amaçlı olan bir video kodlama cihazı ve bir video kod çözme cihazı ile ilgilidir.

Patentli-Olmayan Literatürde (NPL) (1) tarif edilen bir sisteme dayanan bir video kodlama sistemi, sayısallaştırılmış videonun her bir çerçevesini kodlama ağacı birimlerine (CTU'lar) böler ve her bir CTU, ızgara tarama sırasına göre kodlanır. Her bir CTU, dördün ağaç bir yapıda içerisinde kodlama birimlerine (CU'lar) bölünür ve kodlanır. Her bir CU, kestirim birimlerine (PU'lar) bölünür ve kestirimi yapılır. Buna ilave olarak, her bir CU'nun bir kestirim hatası, bir dördün ağaç yapısındaki dönüşüm birimlerine (TU'lar) bölünür ve dönüştürülür.

15 CU, çerçeve-içi kestirim/çerçeveler-arası kestirimine ait bir kodlama ünitesidir. Çerçeve-içi kestirim ve çerçeveler-arası kestirim aşağıda açıklanacaktır.

Çerçeve-içi kestirim, kodlanacak bir çerçevenin yeniden oluşturulmuş görüntüsünden kaynaklanan kestirimdir. NPL l'de tarif edilen sistem içerisinde, kodlanacak bir bloğun etrafındaki yeniden oluşturulmuş bir piksel, bir çerçeve-içi kestirim sinyalinin üretilmesi için ekstrapole edilmiştir. Buradan sonra, çerçeve-içi kestirimini kullanan bir CU, bir çerçeve-içi CU olarak adlandırılır. NPL l'de çerçeve-içi CU'nun pred_mode_flag sözdizimi değerinin 1 olduğuna dikkat ediniz.

Buna ilave olarak, çerçeve-içi kestirimi kullanmayan bir çerçeve-içi CU, bir I_PCM (Çerçeve-İçi Darbe Kodlamalı Modülasyonu) CU olarak adlandırılır. I_PCM CU'da, 25 CU'nun bir kestirim hatasını göndermek yerine, CU'nun bir görüntüsü bozulmadan iletilir. NPL l'de I_PCM CU'nun pcm_flag sözdiziminin değerinin 1 olduğuna dikkat ediniz.

Çerçeveler-arası kestirim, görüntülenme zamanında kodlanmış bir çerçeveden farklı olan yeniden yapılandırılmış bir çerçevenin (referans resim) görüntüsüne dayanan kestirimdir. Buradan sonra, çerçeveler-arası kestirim de ayrıca ara kestirim olarak 30 adlandırılacaktır. Şekil 10, bir çerçeveler-arası kestirim örneğini tasvir etmekte olan bir açıklayıcı diyagramdır. Bir hareket vektörü $MV = (mv_x, mv_y)$, kodlanacak olan bir bloğa göre olan bir referans resminin yeniden oluşturulmuş bir görüntü bloğunun öteleme

miktarına işaret eder. Ara-kestirim içerisinde, bir referans resminin yeniden oluşturulmuş bir görüntü bloğuna (eğer gerekirse piksel enterpolasyonunu kullanarak) dayanan bir ara kestirim sinyali üretilir. Buradan sonra, ara kestirim kullanan bir CU, bir ara CU olarak adlandırılacaktır. NPL 1'de ara CU'nun pred_mode_flag sözdizimi değerinin 0 olduğuna dikkat ediniz.

Hareket vektörü farkı bilgisini ve bir CU kestirim hatasını göndermeme amaçlı olan çerçeveler-arası kestirim kullanan bir ara CU, bir Atlama CU olarak adlandırılır. NPL 1'de, Atlama CU'nun skip_flag sözdizimi değeri 1'dir.

Sadece yukarıda belirtilen iç CU'larla kodlanmış bir çerçeveye bir I çerçevesi (veya bir I resmi) denir. İç CU'lar kadar ara CU'ları içererek kodlanmış bir çerçeveye bir P çerçevesi (veya bir P resmi) denir. Bir blokun ara kestirimi için sadece bir adet referans resmin değil, iki adet referans resmin aynı anda kullanıldığı ara CU'ları ihtiva eden bir çerçeve, bir B çerçevesi (veya bir B resmi) olarak adlandırılır.

İç kestirim ve çerçeveler-arası kestirim yukarıda açıklandığı gibidir.

Şekil 11'e atıfta bulunulduğunda, bir giriş görüntüsü olarak sayısallaştırılmış videonun her bir çerçevesinin her bir CU'sunu alan ve bir bit akışı veren tipik bir video kodlama cihazının konfigürasyonu ve çalışması tarif edilecektir.

Şekil 11'de tasvir edilen video kodlama cihazı, bir dönüştürücü/nicemleyici (102), bir entropi kodlayıcı (103), bir ters dönüştürücü/ters nicemleyici (104), bir tampon (105), bir kestirimci (106), bir PCM kodlayıcısı (107), bir PCM kod çözücüsü (108), bir çoğullanmış veri seçicisi (109), bir çoğullayıcı (110), bir anahtar (121) ve bir anahtar (122) ihtiva etmektedir.

Şekil 12'de gösterildiği üzere, bir çerçeve LCU'lardan (En Büyük Kodlama Üniteleri) oluşur. Bir LCU, CU'lardan (Kodlama Üniteleri) oluşur. Şekil 12, bir t çerçevesinin CTU bölünmesinin bir örneğini ve çerçevenin uzaysal çözünürlüğü CIF (Genel Ara Biçim) ve CTU büyüklüğü de 64 olduğunda t çerçevesindeki CTU8'in bir CU bölünme örneğini gösteren bir açıklayıcı diyagramdır. CTU8'in bir dördün ağaç yapısı, bir 64 x 64 bölgesinin bölünmüş olduğunu gösterecek biçimde CUDepth = 0'da cu_split_flag = 1 ile temsil edilebilir; ilk üç 32x32 CU'nun (CU0, CU1 ve CU2) bölünmediğini gösterir biçimde CUDepth = 1'de üç cu_split_flag = 0 ile temsil edilebilir, son 32x32 CU'nun bölündüğüne işaret eder biçimde CUDepth = 1'de cu_split_flag = 1 ile temsil edilebilir, CUDepth = 2'de ilk üç 16x16 CU'nun (CU3, CU4 ve CU5) bölünmediğine işaret edecek biçimde üç cu_split_flag = 0 ile temsil edilebilir ve CUDepth = 2'de son 16x16 CU'nun ayrıldığına işaret

edecek biçimde $cu_split_flag = 1$ ile temsil edilebilir ve $CUDepth = 3$ 'te olduğu, tüm 8×8 CU'ların (CU6, CU7, CU8 ve CU9) tümünün bölünmediğini işaret edecek biçimde dört $cu_split_flag = 0$ ile temsil edilebilir.

Şekil 11'de gösterilen video kodlama cihazı, ızgara tarama sırasında LCU'ları kodlar ve z-tarama sırasındaki her bir LCU'yu oluşturan CU'ları kodlar. Bir CU'nun boyutu, 64×64 , 32×32 , 16×16 ve 8×8 'den herhangi birisidir. En küçük CU, en küçük bir kodlama birimi (SCU) olarak adlandırılır.

Dönüştürücü/nicemleyici (102), bir kestirim hatası görüntüsünün bir frekans dönüştürme katsayısını elde etmek amacıyla bir kestirim sinyalinin çıkarıldığı bir görüntüyü (kestirim hatası görüntüsü) frekans-dönüştürür.

Dönüştürücü/nicemleyici (102) buna ilave olarak, önceden belirlenmiş bir nicemleme adım büyüklüğü Q_s ile frekans dönüştürme katsayısını nicemler. Buradan sonra, nicemlenmiş frekans dönüşümü katsayısı, bir katsayı nicemleme değeri veya bir nicemleme seviyesi değeri olarak adlandırılacaktır.

Entropi kodlayıcı (103) bir kestirim parametresini ve nicemleme seviyesi değerini entropi-kodlar. Kestirim parametresi, yukarıda belirtilen bir CU'nun kestirim tipi (iç kestirim veya ara kestirim) ve CU'da yer alan PU'ler (Kestirim Üniteleri) hakkındaki bilgilerle ilgili bilgilerdir.

Ters dönüştürücü/ters nicemleyici (104), nicemleme seviyesi değerini niceleme adımı büyüklüğü Q_s ile ters-nicemleştirir. Ters dönüştürücü/ters nicemleştirici (104) buna ilave olarak, ters nicemleme ile elde edilen frekans dönüşüm katsayısını ters-frekans-dönüştürür. Kestirim sinyali, ters dönüşümle elde edilen yeniden oluşturulmuş bir kestirim hatası görüntüsüne eklenir ve yeniden yapılandırılmış kestirim hatası görüntüsü, anahtara (122) beslenir.

Çoğullanmış veri seçicisi (109) kodlanacak bir CU'ya karşılık gelmekte olan entropi kodlayıcının (103) girdi verisinin miktarını izler. Entropi kodlayıcı (103), girdi verilerini CU'nun işlem süresi içinde entropi-kodlayabildiği zaman, çoğullamalı veri seçici (109), entropi kodlayıcısının (103) çıkış verilerini seçer ve çıkış verilerini, çoğullayıcıya (110) anahtar aracılığıyla (121) gönderir. Çoğullanmış veri seçici (109) buna ilave olarak, ters dönüştürücü/ters nicemleyicinin (104) çıktı verilerini seçer ve çıkış verisini anahtar (122) aracılığıyla tampona (105) gönderir.

Entropi kodlayıcı (103), girdi verilerini CU'nun işlem süresi içinde entropi-kodlayamadığında, çoğullanmış veri seçicisi (109), PCM kodlayıcısından (107)

gelen çıkış verisini seçer ve çıkış verisini, çoğullayıcıya (110) anahtar aracılığıyla (121) gönderir. Çoğullanmış veri seçicisi (109), buna ilave olarak PCM kodlayıcının (107) çıkış verilerinin PCM-kodunu çözen PCM kod çözücüsü (108) tarafından elde edilen çıktı verilerini seçer ve çıkış verisini anahtar (122) aracılığıyla tampona (105) gönderir.

- 5 Tampon (105), anahtar (122) aracılığıyla sağlanan yeniden oluşturulmuş bir görüntüyü kayıt altına alır. Bir çerçeve için yeniden oluşturulmuş bir görüntü, yeniden oluşturulmuş bir resim olarak adlandırılır.

Çoğullayıcı (110), entropi kodlayıcının (103) ve PCM kodlayıcının (107) çıktı verilerini çoğullar ve çoğullanan çıktı verisini çıktı olarak verir.

- 10 Yukarıda-bahsedilen işleme dayanarak, video kodlama cihazındaki çoğullayıcı (110) bir bit akışı üretir.

Ahntı Listesi

Patentli-Olmayan Literatür

- NPL 1: Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, ve Thomas
15 Wiegand, "Yüksek Verimlilikteki Video Kodlama (HEVC) metin şartnamesi taslak 9," JCTVC-K1003_v10, (JCT-VC) ITU-T SG16 WP3 ve ISO/IEC JTC1/SC29/WG11 6. Toplantısındaki Video Kodlama Birleşik İşbirliği Ekibi : Şangay, CN, 10-19, Ekim 2012.

- Şekil 11'de gösterilen video kodlama cihazında, bir CU'nun blok başlığı entropi kodlanmıştır. Başka bir deyişle, bir I_PCM CU'nun blok başlığı (I_PCM modu başlığı)
20 gönderileceği zaman, iki süreç gereklidir, yani, entropi kodlayıcının (103) bir kodlama motorunun çıktı-olmayan sembollerini çıktı olarak verme işlemi (pcm_flag entropi-kodlandıktan sonra) ve kodlama motorunu sıfırlama işlemi. 7.3.9.4 Kodlama dördün ağacı sözdizimi ve Bölüm 7.3.9.5 NPL 1'deki Kodlama birimi sözdiziminden anlaşıldığı gibi, I_PCM mod başlığı genellikle CU bölünmüş bilgisinden (cu_split_flag sözdizimi), CU
25 kestirim türünden (skip_flag sözdizimi, pred_mode_flag sözdizimi ve pred_mode sözdizimi, part_mode sözdizimi) ve bir PU için bir PCM işaretinden oluşmaktadır.

- Şekil 13'e atıfta bulunulduğunda, bir CUT'deki kodlama sırasına göre ardışık olan I_PCM CU'larının kodlanması açıklanacaktır. Şekil 13'teki (A) 'da da gösterildiği üzere, kodlanacak bir CUT'de, kodlama sırasında 0, 1, 2 ve 3. olan CU'ların (CU (0), CU (1), CU
30 (2) ve CU (3)) I_PCM CU olduğu varsayılmıştır. Her bir CU'nun I_PCM modu başlığının bir video bit akışına, hizalama verilerinin (pcm_alignment_zero_bit) ve PCM verilerinin (pcm_sample_luma/pcm_sample_chroma) araya sokulmasıyla karşılık gelen bir CTU bit akışı oluşturulur. Başka bir deyişle, Şekil 13'teki (A) 'da gösterilen CTU bit akışı,

aşağıdakilerden oluşur: $cu_split_flag = 1$, $cu_split_flag = 0$, $skip_flag = 0$, $pred_mode_flag = 1$, $part_mode = 2N \times 2N$, $pcm_flag = 1$, $pcm_alignment_zero_bit$, ve

CU(0)'nun $pcm_sample_luma/pcm_sample_chroma$ 'sı; $cu_split_flag=0$, $skip_flag=0$, $pred_mode_flag=1$, $part_mode=2N \times 2N$, $pcm_flag=1$, $pcm_alignment_zero_bit$, ve

5 CU(1)'nin $pcm_sample_luma/pcm_sample_chroma$ 'sı ; $cu_split_flag=0$, $skip_flag=0$, $pred_mode_flag=1$, $part_mode=2N \times 2N$, $pcm_flag=1$, $pcm_alignment_zero_bit$, ve

CU(2)'nin $pcm_sample_luma/pcm_sample_chroma$ 'sı; ve

$cu_split_flag=0$, $skip_flag=0$, $pred_mode_flag=1$, $part_mode=2N \times 2N$, $pcm_flag=1$, $pcm_alignment_zero_bit$, ve

10 CU(3)'sı $pcm_sample_luma/pcm_sample_chroma$ 'sı. 7.3.9.5 Kodlama birimi sözdizimi ve NPL 1'deki bölüm 7.3.9.7 PCM örnek sözdiziminden açıkça görüldüğü üzere, $pcm_alignment_zero_bit$, pcm_sample_luma ve pcm_sample_chroma , entropi-harici kodlanacak olan sözdizimleridir.

CTU'daki kodlama sırasında art arda gelen I_PCM'in CU (0), CU (1), CU (2) ve CU

15 (3) 'sunun I_PCM mod üstbilgisi entropi-kodlandığından dolayı, kodlama motorunun çıktı olmayan sembollerin çıktı olarak verilmesi işlemi ve kodlama motorunun sıfırlanmasının işlemi tekrarlanır. Bu nedenle, CU (0), CU (1), CU (2) ve CU (3) CTU'da kodlama sırasına göre I_PCM ardışık olsa bile, CU'lar, çıktı-olmayan sembollerin işleminin ve her bir CU üzerindeki PCM verilerinin çoğullanmasından önce her bir I_PCM modu başlığı üzerinde

20 yürütülen yukarıda bahsedilen kodlama motorunun sıfırlanması işleminin tamamlanmasını beklemek durumundadır. Tipik teknolojiye, CTU'daki kodlama sırasına göre art arda olan I_PCM CU'lar üzerindeki PCM verilerinin verimli bir şekilde bir bit-akışına çoğullanamaması, yani CTU'daki kodlama sırasındaki I_PCM CU'ları üzerindeki PCM verilerinin verimli bir şekilde gönderilememesi sorunu vardır.

25 Benzer şekilde, Şekil 13'teki (B) 'de gösterildiği üzere, aynı zamanda, görüntünün sağ sınırında eksik bir CTU'da kodlama düzeninde art arda olan I_PCM CU'lar ile ilgili olarak da aynı problem vardır. Şekil 13'teki (B) 'de, "Çıkarım", değerin, işlenecek bir CU'nun pozisyonuna (x_0 ve y_0) ve blok büyüklüğüne ($\log_2 CbSize$), görüntü çerçevesinin boyutuna ($pic_width_in_luma_samples$ genişliği ve $pic_height_in_luma_samples$ yüksekliği) ve

30 minimum CU boyutuna ($\log_2 MinCbSizeY$) bağlı olan cu_split_flag 'i göndermeden kod çözücü tarafından belirlendiğini gösterir (Ayrıntılar için bölüm 7.3.9.5 Kodlama birimi sözdizimi ve bölüm 7.4.9.4 NPL 1'deki dördün ağacı semantik kodlamaya bakınız). Buna ilave olarak, Şekil 13'teki (B) 'de "N", ilgili pozisyonda herhangi bir CU bulunmadığını ve

dolayısıyla gereksiz cu_split_flag'ın aktarılmadığını gösterir.

Şekil 13'ün (B)'sindeki CTU bit akışı aşağıdakilerden oluşmaktadır: cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit, ve

- 5 CU(0)'nun pcm_sample_luma/pcm_sample_chroma'sı; cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit, ve
CU(1)'nun pcm_sample_luma/pcm_sample_chroma'sı; cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit, ve
CU(2)'nun pcm_sample_luma/pcm_sample_chroma'sı; cu_split_flag=0, skip_flag=0,
10 pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit, ve
CU(3)'nun pcm_sample_luma/pcm_sample_chroma'sı; cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1, pcm_alignment_zero_bit, ve
CU(4)'nun pcm_sample_luma/pcm_sample_chroma'sı; ve
cu_split_flag=0, skip_flag=0, pred_mode_flag=1, part_mode=2Nx2N, pcm_flag=1,
15 pcm_alignment_zero_bit, ve
CU(5)'nun pcm_sample_luma/pcm_sample_chroma'sı.

- Mevcut buluşun bir amacı, I_PCM CU'ları bir CTU'da kodlama sırasına göre art arda oluşturulduğunda bile I_PCM CU'lar üzerindeki PCM verilerini verimli bir şekilde gönderebilen bir video kodlama cihazı, bir video kod çözme cihazı, bir video kodlama
20 yöntemi, bir video kod çözme yöntemi, bir video kodlama programı ve bir video kod çözme programı sağlamaktır.

- Mevcut buluşa göre bir video kodlama cihazı şunları ihtiva etmektedir: bir görüntü bloğunun dönüştürülmesi için dönüştürme aracı; dönüştürme aracı tarafından dönüştürülen görüntü bloğu üzerindeki dönüştürülmüş verinin entropi-kodlanması için entropi kodlama
25 aracı; görüntü bloğunu PCM-kodlamak için PCM-kodlama aracı, her bir görüntü bloğu için, ya entropi-kodlama aracının çıktı verisini ya da PCM-kodlama aracının çıktı verisini seçmek için çoğullamalı veri seçim aracı; ve çoğullamalı veri seçim aracı tarafından seçilen bir görüntü blokları kümesi olarak bir kodlama ağacı birimine ait olan tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden kodlama ağacı ünitesinin tepesindeki
30 ctu_pcm_flag sözdiziminini çoğullama amaçlı olan çoğullama aracı, burada kodlama ağacı birimine ait olan tüm görüntü blokları PCM bloklarıdır, çoğullama aracı kodlama ağacı biriminin ayırma modelinin göstergesi olan en azından cu_split_flag sözdizimini

çoğullamaz.

Mevcut buluşa göre olan bir video kod çözme cihazı aşağıdakileri ihtiva eder: deşifre edilmesinin, kodu çözülecek kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığını gösteren `ctu_pcm_flag` sözdizimini içeren bir bit akışının
5 çoğulluğunun çözülmesi için çoğullama çözücü araç; bit akışında bulunan bir görüntü bloğu üzerindeki PCM verisini PCM-kod çözme amaçlı olan PCM kod çözme aracı; bit akışında bulunan görüntü bloğu üzerindeki dönüştürülmüş verinin entropi-kod çözülmesi için entropi kod çözme aracı; ve kodu çözülecek olan kodlama ağacı biriminin `ctu_pcm_flag` sözdiziminin değeri 1 olduğunda kodlama ağacı birimindeki görüntü bloğunun pozisyonuna,
10 kodu çözülecek bir görüntü çerçevesinin boyutuna ve maksimum PCM blok boyutuna bağlı olarak kodlama ağacı biriminin ayırma modelinin göstergesi olan `cu_split_flag` sözdizimini belirleyen kod çözme kontrol aracı.

Mevcut buluşa göre olan bir video kodlama yöntemi şunları ihtiva eder: bir görüntü bloğunun dönüştürülmesi; dönüştürülmüş görüntü bloğu üzerinde dönüştürülmüş verinin
15 entropi-kodlanması; görüntü bloğunun PCM-kodlanması; her bir görüntü bloğu için, ya entropi-kodlanmış verinin veya PCM-kodlanmış verinin seçilmesi; ve bir kodlama ağacı birimine ait tüm görüntü bloklarının bir dizi görüntü bloğu olarak PCM blokları olup olmadığına işaret eden kodlama ağacı ünitesinin tepesindeki `ctu_pcm_flag` sözdiziminin çoğullanması, burada kodlama ağacı birimine ait tüm görüntü bloklarının PCM bloklardır,
20 kodlama ağacı biriminin ayırma düzeninin göstergesi olan en azından `cu_split_flag` sözdizimi çoğullanmamaktadır.

Mevcut buluşa göre olan bir video kod çözme yöntemi aşağıdakileri içerir:, bir kodlama ağacı birimine ait olan tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden `ctu_pcm_flag` sözdizimini içeren bir bit akışının çoğullanmasının çözülmesi; bit
25 akışında bulunan bir görüntü bloğu üzerindeki PCM verilerinin PCM-kod çözülmesi; bit akışında bulunan görüntü bloğu üzerinde yer alan dönüştürülmüş verinin entropi-kod çözülmesi; ve kodu çözülecek kodlama ağacı biriminin `ctu_pcm_flag` sözdiziminin değeri 1 olduğunda, kodlama ağacı birimindeki görüntü bloğunun pozisyonuna, görüntünün boyutuna ve maksimum PCM blok boyutuna bağlı olarak kodlama ağacı biriminin ayırma
30 düzeninin göstergesi olan `cu_split_flag` sözdiziminin belirlenmesi.

Mevcut buluşa göre bir video kodlama programı, bir bilgisayarın aşağıdakileri çalıştırmasını sağlar: bir görüntü bloğunu dönüştürme işlemi; dönüştürülmüş görüntü bloğu üzerinde yer alan dönüştürülmüş veriyi entropi-kodlama işlemi; görüntü bloğunu

PCM-kodlayan bir PCM süreci; her bir görüntü bloğu için ya entropiyle kodlanmış verileri ya da PCM-kodlanmış verileri seçme işlemi; ve bir kodlama ağacı birimine ait tüm görüntü bloklarının bir dizi görüntü bloğu olarak PCM blokları olup olmadığına işaret eden kodlama ağacı biriminin tepesindeki `ctu_pcm_flag` sözdiziminin çoğullama işlemi, burada kodlama ağacı birimine ait olan tüm görüntü blokları PCM bloklarıdır, bilgisayar kodlama ağacı ünitesinin ayırma düzenini gösteren en azından `cu_split_flag` sözdizimini çoğullamamaktadır.

Mevcut buluşa göre olan bir video kod çözme programı, bir bilgisayarın aşağıdakileri çalıştırmasına neden olur; kodu çözülecek bir kodlama ağacı birimine ait tüm görüntü bloklarının PCM blokları olup olmadığına işaret eden `ctu_pcm_flag` sözdizimini içeren bir bit akışının çoğullamasının çözülmesine dair bir süreç; bit akışında bulunan görüntü bloğu üzerindeki PCM verilerinin PCM-kod çözülmesine dair bir süreç; bit akışında bulunan görüntü bloğu üzerinde entropi-kod çözmeyle dönüştürme süreci; ve kodlama ağacı biriminin `ctu_pcm_flag` sözdiziminin değeri 1 olduğunda, kodlama ağacı birimindeki görüntü bloğunun konumuna bağlı olarak kodlama ağacı biriminin ayırma düzeninin göstergesi olan `cu_split_flag` sözdizimini belirlemeye dair süreç, kodu çözülecek bir görüntü çerçevesinin boyutu ve maksimum PCM blok boyutu.

Mevcut buluşa göre, bir CTU'daki tüm CU'lar I_PCM olduğunda, bu PCM veri parçaları verimli bir şekilde gönderilebilir.

[Şekil 1] Bir birinci örnek uygulamanın bir video kodlama cihazını tasvir eden bir blok diyagramı tasvir etmektedir.

[Şekil. 2] I_PCM CU'ları kodlamanın açıklayıcı bir diyagramını betimler.

[Şekil 3] Birinci örnek uygulamanın video kodlama cihazının çalışmasını tasvir eden bir akış şemasını gösterir.

[Şekil 4] Bir ikinci örnek uygulamanın bir video kod çözme cihazını tasvir eden bir blok diyagramı göstermektedir.

[Şekil 5], ikinci örnek uygulamanın video kod çözme cihazının çalışmasını tasvir eden bir akış şemasını göstermektedir.

[Şekil 6] Kodlama ağacı birimi sözdizimini temsil eden bir listenin açıklayıcı bir diyagramını betimlemektedir.

[Şekil 7] PCM dördün ağacı sözdizimini temsil eden bir listenin açıklayıcı bir diyagramını betimlemektedir.

[Şekil 8] PCM örnek sözdizimini temsil eden bir listenin açıklayıcı bir diyagramını betimlemektedir.

[Şekil 9] Bir program kullanarak bilgi işleme sistemini tasvir eden bir blok diyagramı betimlemektedir.

5 [Şekil 10] Bir çerçeveler- arası kestirim örneğini gösteren açıklayıcı bir diyagramı betimlemektedir.

[Şekil 11] Tipik bir video kodlama cihazını tasvir eden bir blok diyagramı betimlemektedir.

[Şekil 12] Bir t çerçevesinin CTU ayrılması örneğini gösteren ve t çerçevesinde CTU8'in CU ayrılmasına dair bir örneği gösteren açıklayıcı bir diyagramı betimlemektedir.

10 [Şekil 13] I_PCM CU'ları kodlamanın açıklayıcı bir diyagramını betimlemektedir.

Örnek Uygulama 1

Şekil 1, bir birinci örnek uygulamanın bir video kodlama cihazını gösteren bir blok diyagramdır. Örnek uygulamanın video kodlama cihazı, Şekil 11'de gösterilen video kodlama cihazında yer alan çoğullayıcı (110) yerine bir çoğullayıcı (1100) içerir. Şekil 11'de
15 gösterilen çoğullayıcıdan (110) farklı olarak, örnek uygulamada yer almakta olan çoğullayıcı (1100), kodlanacak bir CTU içerisindeki tüm CU'ların I_PCM olup olmadığını belirleyen bir PCM belirleyicisini (1101) ihtiva eder. PCM belirleyicisinin (1101) çoğullayıcıda (1101) mevcut olduğu yere dair bir örnek Şekil l'de gösterilmektedir, ancak bu zorunlu değildir. PCM belirleyicisi (1101), çoğullayıcıdan (1100) başka bir parçada
20 sağlanabilir veya Şekil l'de gösterilen her bir birimden bağımsız olarak sağlanabilir.

Örnek uygulamanın video kodlama cihazının, her CTU'daki gönderim sırasına göre I_PCM CU'leri ardarda iletildiği varsayılmaktadır. Örnek uygulama içerisinde, Şekil 2'deki (A) 'da (Şekil 13'teki (A) 'da olduğu üzere) gösterildiği üzere, kodlama sırasında ardışık olan CU (0), CU (1), CU (2) ve CU (3) 'ün PCM-kodlanacak bloklar olduğu kabul edilecektir.

25 Video kodlama cihazının çalışması, Şekil 2'deki (A) 'da gösterilen I_PCM CU'ları örnek olarak alarak Şekil 3'teki bir akış şemasına atıfta bulunularak açıklanacaktır.

Örnek uygulamanın video kodlama cihazı, Şekil 3'ün akış şemasında gösterildiği gibi çalışır. Başka bir deyişle, PCM belirleyicisi (1101), S101 basamağında, kodlanacak bir CTU içerisindeki tüm CU'ların I_PCM olup olmadığını belirler. Tüm CU'lar I_PCM olduğunda,
30 ctu_pcm_flag sözdizimi 1 olarak ayarlanır. Aksi takdirde, ctu_pcm_flag sözdizimi 0 olarak ayarlanır. Şekil 2'deki (A) 'da gösterilen örnek kullanıldığında, ctu_pcm_flag değeri 1 olur. Bu durumda, entropi kodlayıcısı (103), S102 basamağında ctu_pcm_flag'i 1 olarak

entropi-kodlar. Entropi kodlayıcı (103), buna ilave olarak henüz çıktı olarak verilmemiş olan sembollerini çıktı olarak verir. `ctu_pcm_flag` değeri 0 olduğunda, işlem Şekil 3'ün akış şemasında tamamlanmış olmasına rağmen, örnek uygulamanın video kodlama cihazı, entropi kodlayıcı (103) `ctu_pcm_flag`'i 0 olarak entropi kodlandıktan sonra tipik video kodlama cihazı ile aynı şekilde her bir CU'yu kodlar.

İşlenecek bir CU'nun pozisyonuna (x_0, y_0) ve büyüklüğüne ($\log_2 \text{CbSize}$), I_PCM CU'nun bir görüntü çerçevesinin boyutuna (`pic_width_in_luma_samples` genişliği ve `pic_height_in_luma_samples` yüksekliği) ve maksimum boyutuna (minimum CU boyutu $\log_2 \text{MaxIpcmCbSizeY}$ 'den büyük veya eşit olduğunda $\log_2 \text{MaxIpcmCbSizeY}$) bağlı olarak, çoğullayıcı (1100), işlenecek kod çözücü tarafından belirlenen CU'nun S103 basamağında `cu_split_flag`'ini hesaplar. Aşağıdaki koşullardan herhangi biri veya daha fazlası karşılandığında, çoğullayıcı (1100), `cu_split_flag`'i 1'e ayarlar. Başka bir deyişle, işlenecek CU bölünür.

[Koşullar]

1. $x_0 + (1 \ll \log_2 \text{CbSize}) > \text{pic_width_in_luma_samples}$
2. $y_0 + (1 \ll \log_2 \text{CbSize}) > \text{pic_height_in_luma_samples}$
3. $\log_2 \text{CbSize} > \log_2 \text{MaxIpcmCbSizeY}$

Yukarıdaki koşullardan hiçbiri karşılanmadığında, çoğullayıcı (1100) `cu_split_flag`'i 0 yapar. Başka bir deyişle, işlenecek CU bölünmez.

Çoğullayıcı (1100), basamak S104'te `cu_split_flag = 0` olup olmadığını belirler. `cu_split_flag = 0` olduğunda, prosedür S105 basamağına ilerler. Aksi takdirde, CU bölünür ve prosedür S103 basamağına geri döner.

Basamak S105'te, çoğullayıcı (1100), CU (n) 'nin `pcm_alignment_zero_bit`'ini entropi-dışında kodlar.

S106 basamağında, PCM kodlayıcı (107), CU (n) üzerindeki PCM verilerini entropi-dışında kodlar.

Basamak S107'de, çoğullayıcı (110), kodlanacak CTU'daki tüm CU'ların kodlanıp kodlanmadığını belirler. CTU'daki kodlanacak tüm CU'lar kodlandığında, prosedür S108 basamağına ilerler. Aksi takdirde, prosedür, işlenecek CU'ları kodlamak için S103 basamağına geri döner.

S108 basamağında, entropi kodlayıcı (103) kodlama motorunu sıfırlar. Daha sonra, prosedür sonraki CTU'yu kodlama işlemine ilerler.

Çoğullayıcı (1100), PCM kodlayıcı (107) ve entropi kodlayıcı (103) tarafından kodlanan verileri çoğullayarak elde edilen bir bit akışını çıktı olarak verir.

Örnek uygulamanın video kodlama cihazının çalışması yukarıda açıklandığı gibidir. Çoğullayıcının (1100), çoğullanmış veri seçici (109) tarafından seçilen bir görüntü blokları kümesine karşılık gelen bir kodlama ağacı birimine ait olan tüm görüntü bloklarının, PCM blokları olup olmadığını gösteren kodlama ağacı ünitesinin tepesindeki `ctu_pcm_flag` sözdizimini çoğullamasına rağmen, çoğullayıcının (1100) kodlama ağacı birimine ait tüm görüntü blokları PCM blokları olduğunda, kodlama ağacı biriminin ayırma modelini gösteren en azından `cu_split_flag` sözdizimini çoğullamadığına dikkat ediniz.

Örnek uygulamanın video kodlama cihazının yukarıda-tarif edilen çalışmasına göre, kodlanacak bir CTU içerisindeki tüm CU'lar I_PCM olduğunda, entropi-kodlanacak bir I_PCM mod başlığı, ardışık I_PCM CU'lar üzerindeki PCM verileri arasına eklenmez. Başka bir deyişle, Şekil 2'deki (A) 'da görüntülenen CTU bit akışı, aşağıdakilerden oluşur:

`ctu_pcm_flag` = 1; `pcm_alignment_zero_bit` ve CU (0)'nun `pcm_sample_luma/pcm_sample_chroma'sı`; CU(1)'nin `pcm_sample_luma/pcm_sample_chroma'sı`; CU (2)'nin `pcm_sample_luma/pcm_sample_chroma'sı` ve CU(3)'nin `pcm_sample_luma/pcm_sample_chroma'sı`. Tek bir parça PCM verisinin bit sayısı, kaçınılmaz olarak 8 bitin bir katıdır. Bu nedenle, bir I_PCM CU ilk olarak gönderildikten sonra, takip etmekte olan I_PCM CU'nun başlangıç pozisyonu kaçınılmaz olarak 8 bitin bir katı olur. Diğer bir deyişle, takip etmekte olan I_PCM CU'nun başlangıç pozisyonu kaçınılmaz olarak bayt-hizalanmış bir adres haline gelir. Bu nedenle, `pcm_alignment_zero_bit`, CU (1), CU (2) ve CU (3) 'de gönderilmez.

Benzer şekilde, Şekil 2'nin (B) 'de gösterilen CTU bit akışı, aşağıdakilerden oluşur:

`ctu_pcm_flag` = 1; `pcm_alignment_zero_bit` ve CU (0)'nun `pcm_sample_luma/pcm_sample_chroma'sı`; CU(1)'nin `pcm_sample_luma/pcm_sample_chroma'sı`; CU (2)'nin `pcm_sample_luma/pcm_sample_chroma'sı`; CU(3)'nin `pcm_sample_luma/pcm_sample_chroma'sı`; CU(4)'nin `pcm_sample_luma/pcm_sample_chroma'sı`; ve CU(5)'nin `pcm_sample_luma/pcm_sample_chroma'sı`. Başka bir deyişle, kodlanacak bir CTU'daki tüm CU'lar I_PCM olduğunda, entropi kodlanacak olan I_PCM mod başlığı ardışık I_PCM CU'lar üzerindeki PCM verileri arasına eklenmez.

Bu nedenle, birinci I_PCM CU'nun ardından I_PCM CU üzerindeki PCM verisi bir bit akışına çoğullanıldığında, kodlama motorunun çıkış-harici sembolleri çıktı olarak verme sürecinin ve yukarıda bahsedilen kodlama motorunu sıfırlama işleminin tamamlanmasına kadar beklemeye gerek yoktur. Bu nedenle, örnek teşkil eden uygulamanın video kodlama cihazı, ardışık I_PCM CU'lar üzerindeki PCM verilerini bir bit akışına verimli bir şekilde çoğullayabilir. Başka bir deyişle, ardışık I_PCM CU'lar üzerindeki PCM verileri verimli bir şekilde gönderilebilir.

Örnek Uygulama 2

Şekil 4, birinci örnek uygulamanın video kodlama cihazına karşılık gelmekte olan bir video kod çözme cihazını gösteren bir blok diyagramdır. Örnek uygulamanın video kod çözme cihazı bir çok çoğullama çözücü (201), bir kod çözme kontrolcüsü (202), bir PCM kod çözücüsü (203), bir entropi kod çözücüsü (204), bir ters dönüştürücü/ters nicemleyici (206), bir kestirimci (207), bir tampon (208) ve bir anahtar (221) ve bir anahtar (222) ihtiva eder.

Çoğullama çözücü (201), CTU içindeki kodlanacak olan tüm CU'ların I_PCM olup olmadığını belirleyen bir PCM belirleyicisi (2011) içerir. PCM belirleyicisinin (2011) çoğullama çözücünün (201) içerisinde nerede olduğu Şekil 4'te gösterilmektedir, ancak bu zorunlu değildir. PCM belirleyicisi (2011), çoğullama çözücünün (201) dışında başka bir parçada sağlanabilir veya Şekil 4'te gösterilen her bir birimden bağımsız olarak sağlanabilir.

Çoğullama çözücü (201), bir girdi bit akışının çoğullamasını çözer ve bir video bit akışını çıkarır.

Kodu çözülecek CTU'nun video bit akışının entropi-kod çözülmesiyle elde edilen ctu_pcm_flag 1 olduğunda (yani, PCM belirleyicisi (2011) CTU'daki kodu çözülecek tüm CU'ların I_PCM olduğunu belirlediğinde), kod çözücü kontrolcüsü (202) kodu çözülecek CTU içerisindeki işlenecek bir CU'nun cu_split_flag değerini hesaplar. Özel olarak, işlenecek CU'nun pozisyonuna (x0, y0) ve büyüklüğüne (log2CbSize) bir görüntü çerçevesinin boyutuna (pic_width_in_luma_samples genişliği ve pic_height_in_luma_samples yüksekliği) ve I_PCM CU'nun maksimum boyutuna (Log2MaxIpcmCbSizeY'nin minimum CU boyutu Log2MinCbSizeY'dan daha büyük olduğu veya eşit olduğu durumda Log2MaxIpcmCbSizeY) bağlı olarak, aşağıdaki koşullardan herhangi biri veya daha fazlasının karşılandığı durumda, cu_split_flag = 1 olarak ayarlanır. Başka bir deyişle, işlenecek CU bölünmüştür.

[Koşullar]

1. $x0 + (1 \ll \log2CbSize) > pic_width_in_luma_samples$
2. $y0 + (1 \ll \log2CbSize) > pic_height_in_luma_samples$
3. $\log2CbSize > Log2MaxIpcmCbSizeY$

Yukarıdaki koşullardan hiçbiri karşılanmadığında, kod çözme kontrolcüsü (202)

5 cu_split_flag '1 0'a ayarlar. Başka bir deyişle, işlenecek CU bölünmez.

Buna ilave olarak, $cu_split_flag = 0$ olduğunda, kod çözücü kontrolcüsü (202) video bit akışından, kodu çözülecek bir I_PCM CU'nun $pcm_alignment_zero_bit$ sözdizimini okur ve entropi kod çözücüsünün (204) halen kodu çözülecek bir CU'nun CU/PU başlığını entropi-kod çözmesine neden olmadan PCM kod çözücüsüne (203) takip eden PCM verisini
10 sağlar. PCM kod çözücü (203), anahtar (221) aracılığıyla sağlanan PCM verilerini okur ve PCM verilerinin PCM-kodunu çözer. Kod çözücü kontrolcüsü (202), PCM kod çözücüsünden (203) tampona (208) sağlanan yeniden yapılandırılmış bir görüntüyü sağlamak için anahtarın (222) üzerinde değişir. $cu_split_flag = 1$ durumunda, kod çözme kontrolcüsü (202), kodu çözülecek CTU'da işlenecek bir sonraki CU'nun kodunu çözme
15 işlemine ilerler.

Kodu çözülecek CTU'nun video bit akışının entropi-kod çözülmesiyle elde edilen ctu_pcm_flag değeri 0 olduğunda, kod çözücü kontrolcüsü (202), entropi kod çözücüsünün (204), halihazırda kodu çözülecek olan bir CU'nun CU/PU başlığını entropi-kod çözülmesine neden olur. Entropi kod çözücüsü (204), 1 değerine sahip olan pcm_i
20 entropi-kod çözdüğü zaman, kod çözücü kontrolcüsü (202), video bit akışından, kodu çözülecek olan I_PCM CU'nun $pcm_alignment_zero_bit$ sözdizimini okur ve takip etmekte olan PCM verilerini PCM kod çözücüsüne (203) gönderir. PCM kod çözücüsü (203), anahtar (221) aracılığıyla sağlanan PCM verilerini okur ve PCM verilerinin PCM-kodunu çözer. Kod çözücü kontrolcüsü (202), PCM kod çözücüsünden (203) tampona (208)
25 sağlanan yeniden yapılandırılmış bir görüntüyü sağlamak için anahtarın (222) üzerinde değişir. Entropi kod çözücüsü (204), 0 değerine sahip olan pcm_flag 'ı entropi-kod çözdüğünde veya pcm_flag olmadığında entropi kod çözücüsü (204), buna ilave olarak kestirim parametresini ve kodu çözülecek bir CU'nun nicemleme seviyesi değerini entropi-kod çözer eder ve bunları ters dönüştürücüye/ters nicemleyiciye (206) ile
30 kestirimciye (207) besler.

Ters dönüştürücü/ters nicemleyici (206), nicemleme seviyesi değerini ters-frekans-nicemleştirir ve buna ilave olarak ters nicemleme ile elde edilen frekans dönüşüm katsayısını ters-frekans-dönüştürür.

Ters dönüşümden sonra, kestirimci (207), entropiyle-kodu çözülmüş kestirim parametresine dayanarak tamponda (208) kayıt altına alınmış bir yeniden oluşturulmuş resmin görüntüsünü kullanarak bir kestirim sinyali üretir. Kestirim sinyali üretildikten sonra, kestirimci (207) temin edilen kestirim sinyali ters-dönüştürücü/ters nicemleyici (206) tarafından ters-frekans-dönüştürülmüş yeniden yapılandırılmış bir kestirim hatası görüntüsüne eklenir ve yeniden yapılandırılmış kestirim hatası görüntüsü anahtara (222) beslenir. Kestirim sinyali eklendikten sonra, kod çözücü kontrolcüsü (202), kestirim sinyalinin eklendiği yeniden oluşturulmuş kestirim hata görüntüsünü yeniden yapılandırılmış bir görüntü olarak tampona (208) tedarik etmek üzere anahtarın (222) üzerinde değişir. Daha sonra, prosedür sonraki CU'nun kodunu çözme işlemine ilerler.

Örnek uygulamanın video kod çözme cihazının çalışması, Şekil 5'in bir akış şemasına atıfta bulunularak açıklanacaktır.

S201 basamağında, PCM belirleyicisi (2011), CTU içinde kodlanacak olan tüm CU'ların I_PCM olup olmadığını belirler. CTU'daki kodu çözülecek tüm CU'lar I_PCM olduğu zaman, prosedür basamak S202'ye ilerler. Aksi halde, işlem Şekil 5'in akış şemasında sona ermesine rağmen, örnek teşkil eden uygulamanın video kod çözme cihazı, tipik video kodlama cihazı ile aynı şekilde her bir CU'nun kodunu çözmektedir.

S202 basamağında, kod çözme kontrolcüsü (202), kodu çözülecek CU (n) 'nin cu_split_flag değerini hesaplar.

S203 basamağında, kod çözme kontrolcüsü (202), hesaplanan cu_split_flag'in 0 olup olmadığını belirler. 0 olduğu zaman, prosedür S204 basamağına ilerler. Aksi takdirde (cu_split_flag 1 olduğunda), kodu çözülecek CU bölünür ve prosedür S202 basamağına döner.

S204 basamağında, kod çözme kontrolcüsü (202), bir video bit akışından, kodu çözülecek I_PCM'nin CU (n)'nin pcm_alignment_zero_bit sözdizimini okur. PCM kod çözücüsünün (203) entropi-harici kodlanacak pcm_alignment_zero_bit sözdizimini okuyabileceğine dikkat ediniz.

S205 basamağında, kod çözme kontrolcüsü (202), video bit akışından, kodu çözülecek I_PCM'nin CU(n)'nin üzerindeki PCM verilerini okur. PCM kod çözücü (203) CU (n) üzerindeki PCM verilerinin entropi-harici olarak kodunu çözer. PCM kod çözücüsünün (203) ayrıca, entropi -harici biçimde kod çözülecek olan PCM verilerini de okuyabileceğine dikkat ediniz. Daha sonra, yukarıda belirtildiği üzere, yeniden yapılandırılmış bir CU (n) görüntüsü, PCM verilerine dayanarak belirlenir.

S206 basamağında, kod çözme kontrolcüsü (202), CTU içindeki kod çözülecek olan tüm CU'ların kodunun çözülüp çözülmediğini belirler. Tüm CU'ların kodu çözüldüğünde, prosedür S207 basamağına ilerler. Aksi takdirde, prosedür bir sonraki CU'nun kodunu çözmek için S202 basamağına döner.

- 5 S207 basamağında, entropi kod çözücüsü (204) bir kod çözme motorunu sıfırlar ve daha sonra kodu çözülecek CTU'nun kod çözme işlemini sona erdirir.

Örnek uygulamanın video kod çözme cihazının çalışması yukarıda açıklandığı gibidir.

- Yukarıda tarif edilen örnek uygulamaya ait olan video kod çözme cihazının çalışmasına göre, bit akışından bir birinci I_PCM CU'yu takiben I_PCM CU üzerindeki
10 PCM verisini okumak için kod çözme motorunu sıfırlama işleminin tamamlanmasını beklemeye gerek yoktur. Bu nedenle, örnek teşkil eden uygulamanın video kod çözme cihazı, bit akışından verimli bir şekilde ardışık I_PCM CU'lar üzerindeki PCM verilerini okuyabilir. Başka bir deyişle, ardışık I_PCM CU'lar üzerindeki PCM verileri etkin bir şekilde alınabilir.

- Kodlama ağacı sözdizimi, PCM dördün ağacı sözdizimi ve yukarıda bahsedilen örnek
15 uygulamaların her birine karşılık gelen PCM örnek sözdizimi, sırasıyla Şekil 6, Şekil 7 ve Şekil 8'de gösterilmektedir. Şekil 6, Şekil 7 ve Şekil 8'e istinaden, bir ctu_pcm_flag sözdizimi gönderimi durumu Şekil 6'dan görülmektedir. Kodlama ağacı biriminin bir görüntü bloğunun pozisyonuna, kodu çözülecek olan bir görüntü çerçevesinin boyutuna ve maksimum PCM blok boyutuna (Log2MaxIpcmCbSizeY) bağlı olarak, kodlama ağacı
20 ünitesinin bölünme düzeninin göstergesini belirten bir cu_split_flag sözdizimini belirleme durumu, Şekil 7'den görülmektedir. pcm_alignment_zero_bit ve PCM verilerinin (pcm_sample_luma/pcm_sample_chroma) kodlama durumu Şekil 8'den görülmektedir. Başka bir deyişle, kod çözme motorunu sıfırlama işleminin, bir bit akışından ardışık I_PCM CU'lar üzerindeki PCM verilerini okumak için gerekli olmadığı bulunmuştur.

- 25 Yukarıda belirtilen örnek uygulamaların her birinde, bir CTU içerisindeki tüm CU'ların I_PCM olduğu varsayılır, fakat mevcut buluş bir CTU'daki tüm CU'ların Atlama olması durumuna da uygulanabilir. Örnek olarak, bir CTU'daki tüm CU'ların I_PCM olup olmadığını belirleyen tüm-PCM belirleyicisi, sadece CTU'daki tüm CU'ların Atlama olup olmadığını belirleyen bir tüm-Atlama belirleyicisi ile değiştirilmelidir, böylece tüm CU'lar
30 Atlama olduğunda, ctu_skip sözdizimi ctu_pcm sözdizimi gibi kodlanacak/kodu çözülecektir. Buna ilave olarak, 7.3.9.6 NPL 1'in Kestirim birimi sözdiziminin bölümünde tanımlanan merge_idx, sadece ctu_skip işaretini takiben kodlanmalıdır/kodu çözülmelidir.

Yukarıda belirtilen örnek uygulamaların her biri donanımda konfigüre edilebilir,

ancak örnek uygulamanın bir bilgisayar programı ile uygulanması da mümkündür.

Şekil 9'da tasvir edilen bir bilgi işleme sistemi, bir işlemci (1001), bir program belleği (1002), video verilerinin saklanması için bir kayıt ortamı (1003) ve bir bit akışını kayıt altına almak için bir kayıt ortamını (1004) ihtiva etmektedir. Kayıt ortamı (1003) ve kayıt ortamı 5 (1004), ayırık kayıt ortamları veya aynı kayıt ortamı üzerindeki kayıt alanları olabilir. Bir sabit disk gibi bir manyetik kayıt ortamı, bir kayıt ortamı olarak kullanılabilir.

Şekil 9'da gösterilen bilgi işleme sisteminde, sırasıyla Şekil 1'de ve Şekil 4'te gösterilen her bir bloğun fonksiyonunu gerçekleştirmeye yönelik bir program, program belleğinde (1002) kayıt altına alınır. Daha sonra işlemci (1001), video kodlama cihazının 10 fonksiyonlarını veya sırasıyla Şekil 1 veya Şekil 4'te gösterilen video kod çözme cihazının fonksiyonlarını gerçekleştirmek için program belleğinde (1002) kayıt altında tutulan programa göre işlemleri yürütür.

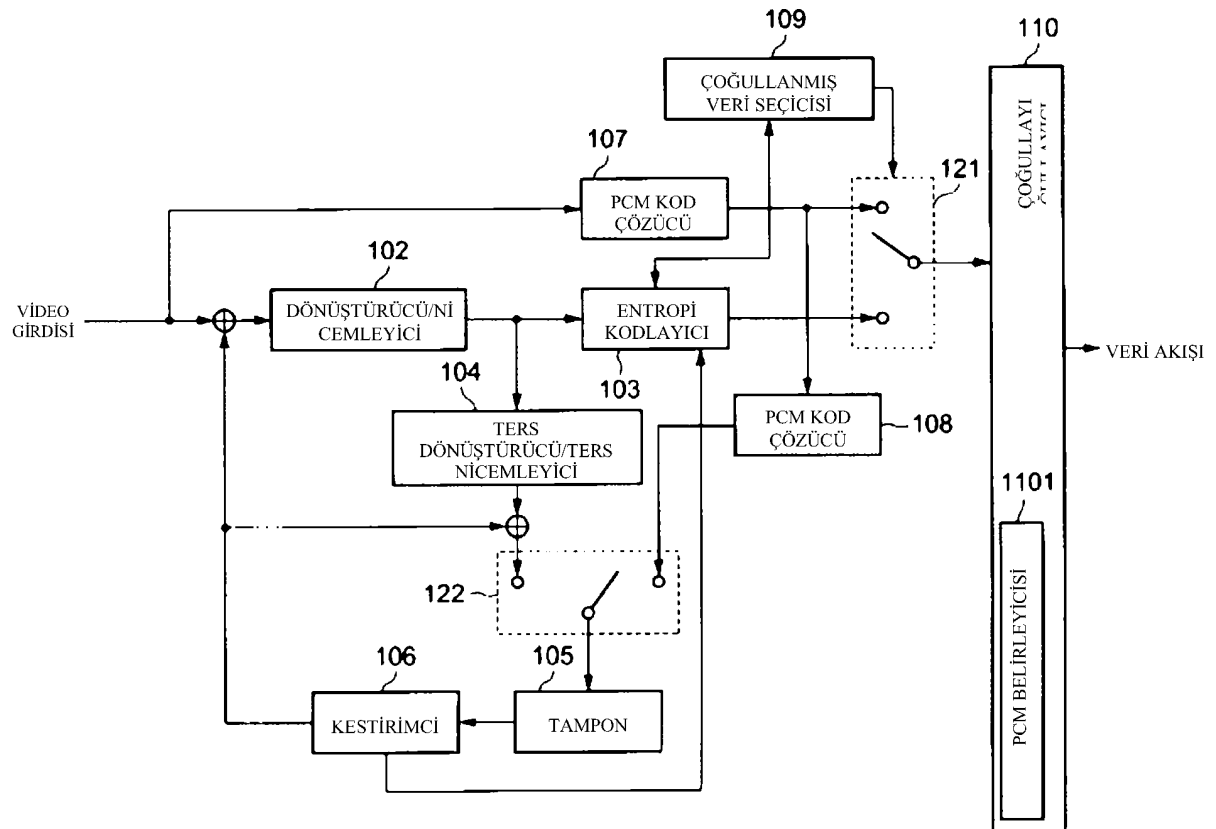
Mevcut buluş, yukarıda belirtilen örnek uygulamalara ve örneklerle referansla tarif edilmiş olmakla birlikte, mevcut buluş, yukarıda belirtilen örnek uygulamalar ve örnekler ile 15 sınırlı değildir. Mevcut buluşun kapsamı dahilinde, tekniğinde tecrübeli kişilerce anlaşılabilen çeşitli değişiklikler, mevcut buluşun konfigürasyonlarına ve ayrıntılarına uygulanabilir.

Referans İşaretleri Listesi

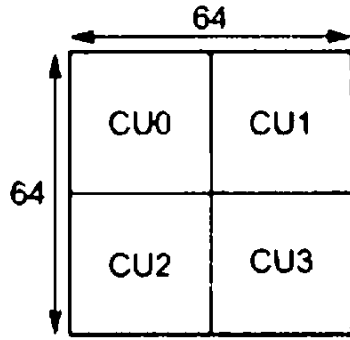
102	dönüştürücü/nicemleyici
20 103	entropi kodlayıcı
104	ters dönüştürücü/ters nicemleyici
105	tampon
106	kestirimci
107	PCM kodlayıcısı
25 108	PCM kod çözücüsü
109	çoğullanmış veri seçicisi
110, 1100	çoğullayıcı
1101	PCM belirleyici
121	anahtar
30 122	anahtar
201	çoğullama çözücü

	2011	PCM belirleyici
	202	kod çözme kontrolcüsü
	203	PCM kod çözücüsü
	204	entropi kod çözücüsü
5	205	ters dönüştürücü/ters nicemleyici
	206	kestirimci
	207	tampon
	221	anahtar
	222	anahtar
10	1001	işlemci
	1002	program belleği
	1003	kayıt ortamı
	1004	kayıt ortamı

ŞEKİL 1

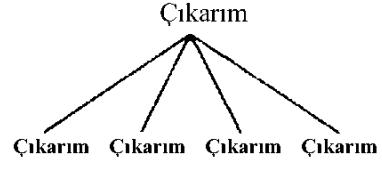


ŞEKİL 2

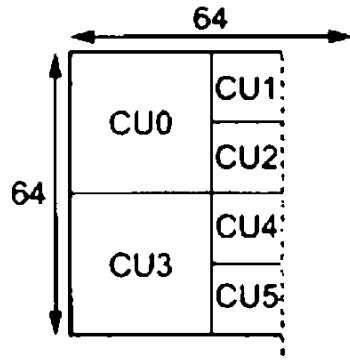


cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1



(A) TÜM CTU



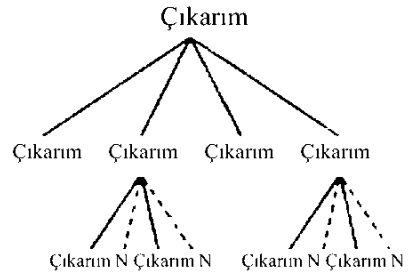
GÖRÜNTÜNÜN
SINIRI
(SAĞ KENAR)

cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1

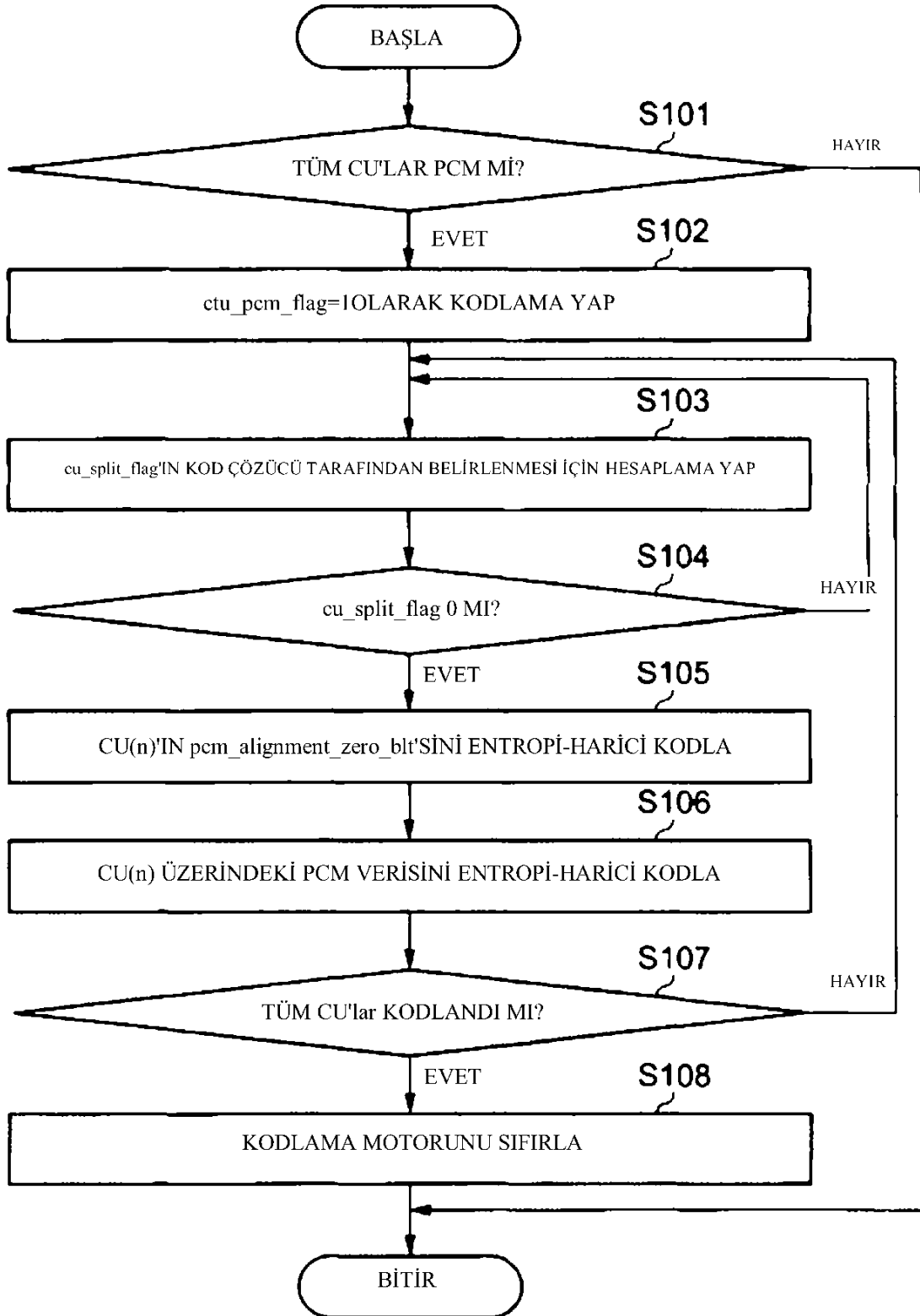
cu_split_flag
@CUDepth=2

cu_split_flag
@CUDepth=3

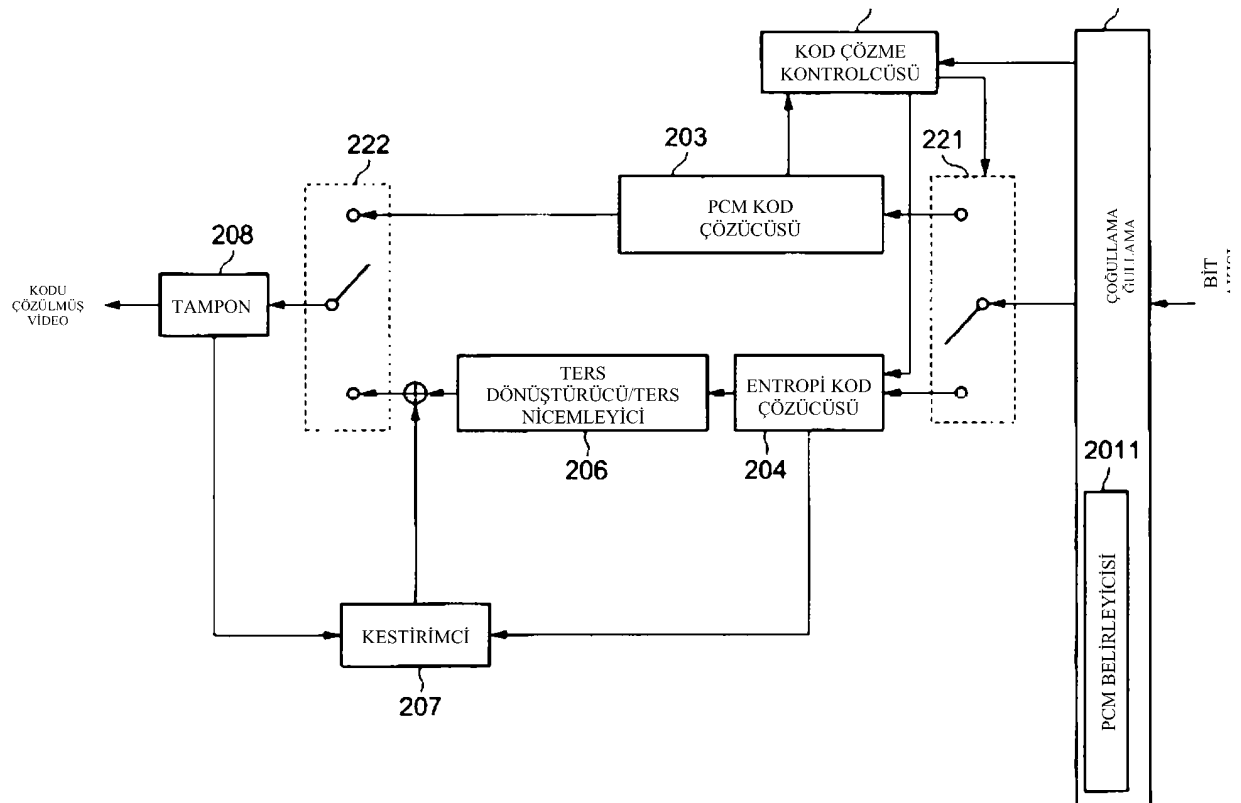


(B) EKSİK CTU

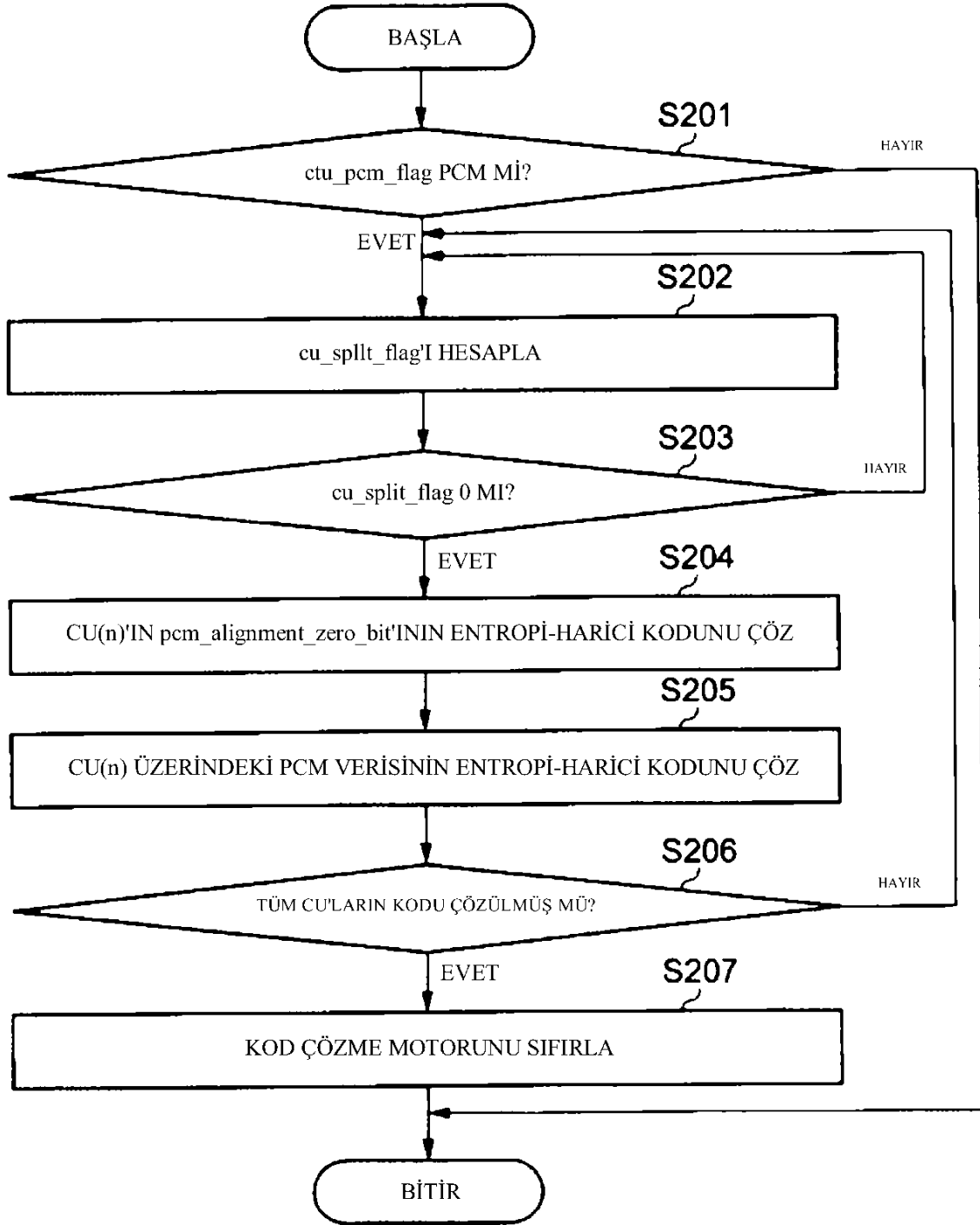
ŞEKİL 3



ŞEKİL 4



ŞEKİL 5



ŞEKİL 6

(Kodlama ağacı ünitesi sözdizimi)

coding_tree_unit() {	Tanımlayıcı
xCtb={ CtbAddrRS % PicWidthInCtbsY) « Log2CtbSizeY	
yCtb=(CtbAddrRS/PicWidthInCtbsY) « Log2CtbSizeY	
CtbAddrInSliceSeg = CtbAddrInRS - slice_segment_address	
if(slice_sao_luma_flag slice_sao_chroma_flag)	
sao(xCtb >> Log2CtbSizeY, yCtb >> Log2CtSizeY)	
if(ctu_pcm_enabled_fia9)	
ctu_pcm_flag	ae(v)
if(!ctu_pcm_flag)	
coding_quadtree(xCtb, yCtb, Log2CtbSizeY, 0)	
Else	
pcm_quadtree(xCtb, yCtb, Log2CtbSizeY, 0)	
}	

ŞEKİL 7

(PCM dördün ağacı sözdizimi)

pcm_quadtree(x0, y0, log2CbSize, ctDepth) {	Tanımlayıcı
if(x0 + (1 << log2CbSize) > pic_width_in_luma_samples y0 + (1 << log2CbSize) > pic_height_in_luma_samples log2CbSize > Log2MaxIpcmCbSizeY)	
cu_split_flag[x0][y0] = 1	
Else	
cu_split_flag[x0][y0] = 0	
if(cu_split_flag[x0][y0]) {	
x1 = x0 + ((1 << log2CbSize) >> 1)	
y1 = y0 + ((1 << log2CbSize) >> 1)	
pcm_quadtree(x0, y0, log2CbSize - 1, ctDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
pcm_quadtree(x1, y0, log2CbSize - 1, ctDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
pcm_quadtree(x0, y1, log2CbSize - 1, ctDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
pcm_quadtree(x1, y1, log2CbSize - 1, ctDepth + 1)	
} else	
pcm_sample(x0, y0, log2CbSize)	
}	

5

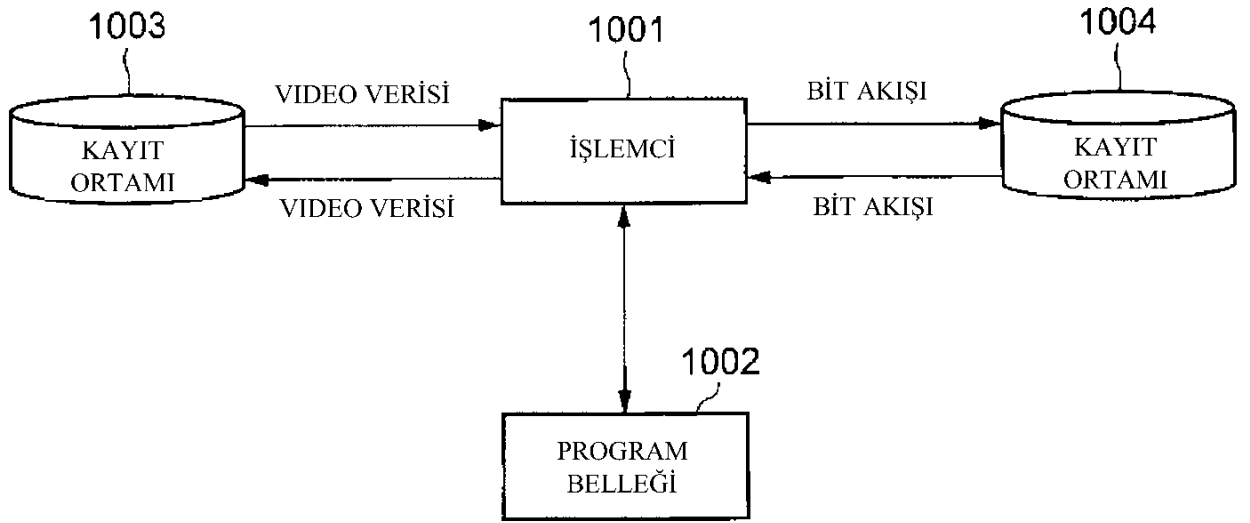
ŞEKİL 8

(PCM örnek sözdizimi)

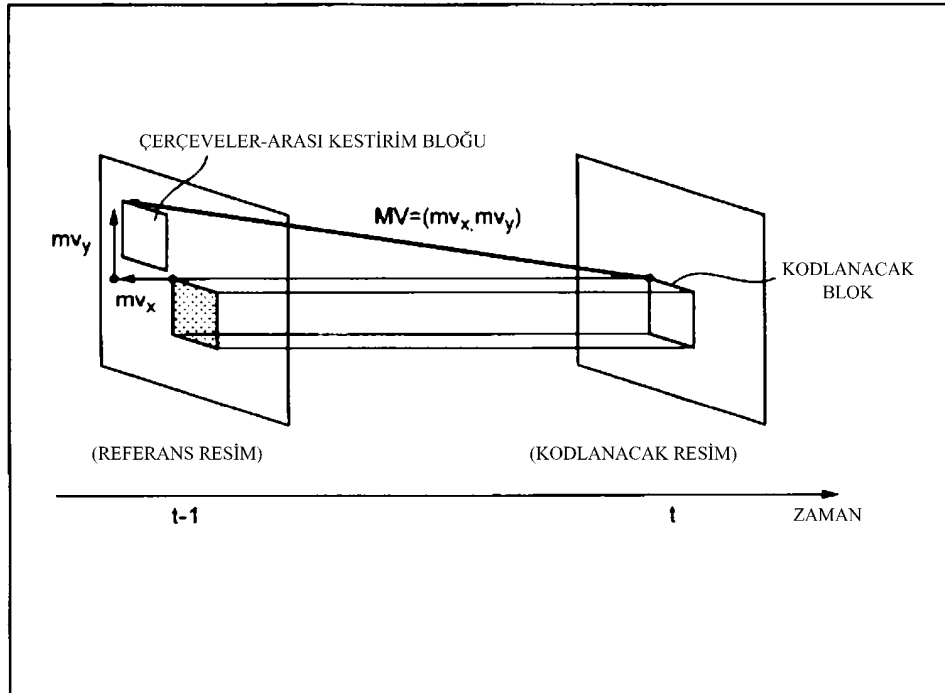
pcm_sample(x0, y0, log2CbSize) {	Tanımlayıcı
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
for(i =0; i < 1 << (log2CbSize << 1); i++)	
pcm_sample_luma[i]	u(v)
for(i =0; i < (1 << (log2CbSize << 1)) >> 1; i++)	
pcm_sample_chroma[i]	u(v)
}	

10

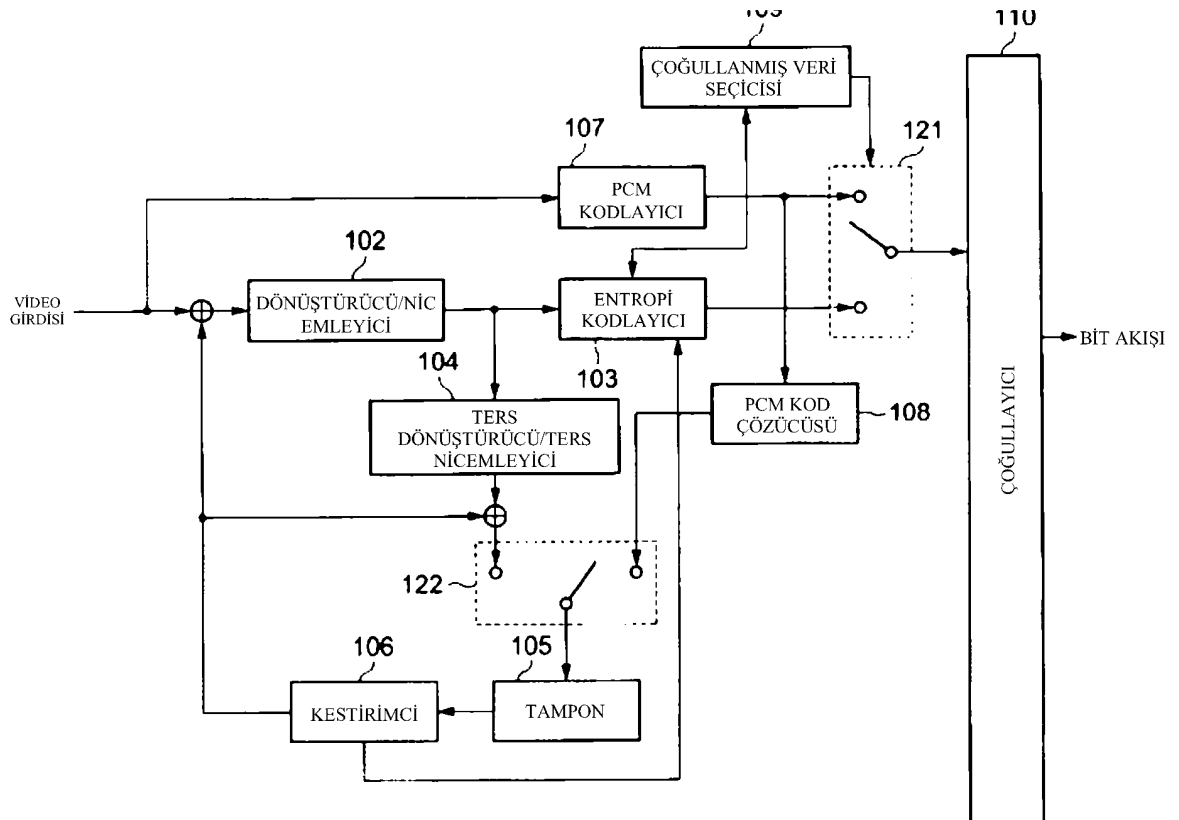
ŞEKİL 9



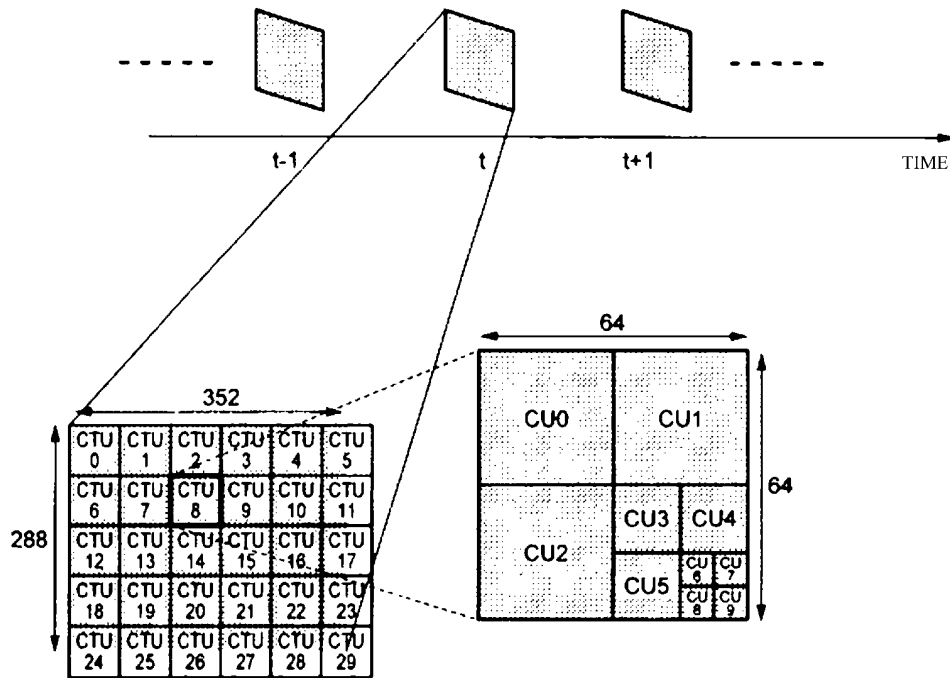
ŞEKİL 10



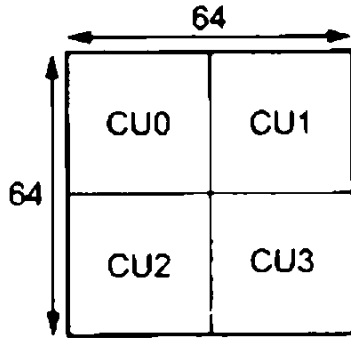
ŞEKİL 11



ŞEKİL 12

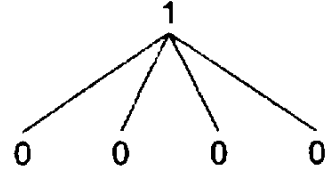


ŞEKİL 13

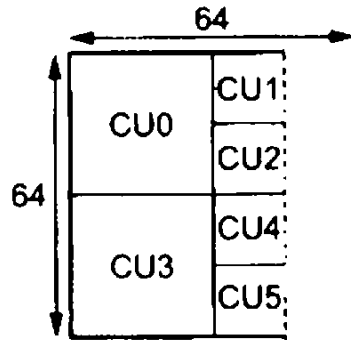


cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1



(A) TÜM CTU



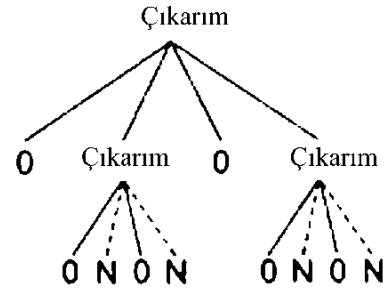
GÖRÜNTÜNÜN
SINIRI (SAĞ
KENAR)

cu_split_flag
@CUDepth=0

cu_split_flag
@CUDepth=1

cu_split_flag
@CUDepth=2

cu_split_flag
@CUDepth=3



(B) EKSİK CTU