สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โครงการพัฒนาโปรแกรมบีบอัดรูปภาพตามมาตรฐาน H.263 บน TMS320C6713

Development of image compression standard H.263 on TMS320C6713

นาย คมกริช มาเที่ยง

นางสาว อโนทัย จิรเพียงทอง

เลขหมู่.....
เลขทะเบียน.....61890
วัน,เดือน,ปี.....24 ก.ค. 2549

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพัฒนาโปรแกรมบีบอัดรูปภาพตามมาตรฐาน H.263 บน TMS320C6713

Development of image compression standard H.263 on TMS320C6713

โดย

นาย คมกริช มาเที่ยง

นางสาว อโนทัย จิรเพียงทอง

อาจารย์ที่ปรึกษา

ผศ.ดร. อรฉัตร จิตต์โสภักตร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โครงการพัฒนาโปรแกรมบีบอัดรูปภาพตามมาตรฐาน H.263 บน TMS320C6713

Development of image compression standard H.263 on TMS320C6713

คณะผู้จัดทำ

1. นาย คมกริช มาเที่ยง รหัสประจำตัว 45015356
2. นางสาว อโนทัย จิรเพียงทอง รหัสประจำตัว 44010580

อรฉัตร จิตต์โสภักตร์ .....อาจารย์ที่ปรึกษา

(ผศ.ดร. อรฉัตร จิตต์โสภักตร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพัฒนาโปรแกรมบีบอัครูปภาพตามมาตรฐาน H.263 บน TMS320C6713

นาย คมกริช มาเที่ยง 45015356
นางสาว อโนทัย จิรเพียงทอง 44010580
ผศ.ดร. อรฉัตร จิตต์โสภักตร์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการเข้ารหัสและถอดรหัส ไฟล์รูปภาพบนมาตรฐาน H.263 และ algorithm standard รวมทั้งได้ประยุกต์ให้โปรแกรมสามารถทำงานได้บน บอร์ด DSP ซึ่งบอร์ด DSP นี้มีคุณสมบัติในการจัดการหน่วยความจำอย่างชาญฉลาด มีประสิทธิภาพ และยังช่วยในการจัดการข้อมูลที่เป็น ไฟล์ stream เพื่อให้การประมวลผลไฟล์ภาพ เป็นไปอย่างต่อเนื่องและยังช่วยลดอัตราข้อมูลที่จะส่ง แต่ยังคง ให้ไฟล์ภาพมีคุณสมบัติหลังจากประมวลผลแล้วใกล้เคียงกับของเดิมมากที่สุด

โดยในปริญญานิพนธ์ฉบับนี้ได้รวมทฤษฎี H.263 algorithm standard และ ผลการทดลองดังกล่าว เพื่อให้เป็นประโยชน์ต่อผู้สนใจที่จะนำไปใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Development of image compression standard H.263 on TMS320C6713

Mr. Khomkris Mathaing 45015356
Miss Anothai Jirapiengthong 44010580
Asst.Prof.Dr. Orachat Chitsobhuk Advisor
Academic Year 2004

ABSTRACT

This project develops a computer program for encoding and decoding a video stream based on H.263 standard. The code is developed based on TI algorithm standard which enables the program to work efficiently on DSP board and can manage the memory intelligently and effectively. Moreover, it can manage 'file stream' data in order to perform the picture-processing process continuously and it also helps in reducing a data ratio, which will be transferred. After the picture-processing process the picture's characteristic should be as close as possible to that of the old one.

This thesis contains H.263 algorithm standard theory and the testing which can be used result to work effectively in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษาและการดูแลจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาที่ให้โอกาสข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ คอยให้ความเอาใจใส่แนะนำและความช่วยเหลือเสมอมา คือ ผศ.ดร. อรฉัตร จิตต์โสภักดร์ ซึ่งต้องขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็ว รวมทั้งยังมีอินเทอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าหาความรู้ต่างๆซึ่งท้ายที่สุดแล้วได้ประกอบกันเป็นส่วนหนึ่งของโครงงานนี้

ขอขอบคุณ พี่ๆ เพื่อนๆ น้องๆในห้องปฏิบัติการฮาร์ดแวร์ที่คอยสร้างความคึกครื้นยามอยู่ในห้องและเป็นกำลังใจเสมอมา และที่ขาดไม่ได้ต้องขอขอบคุณห้องฮาร์ดแวร์ที่ให้ที่ผักผ่อน และช่วยให้ห้องหายหิว มีแรงกายในการทำงาน

และสุดท้ายต้องขอขอบคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ นั่นคือ บิดา มารดา และบุคคลในครอบครัว อันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดู คอยอบรมสั่งสอนข้าพเจ้าเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และคอยให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

คมกริช มาเที่ยง
อโนทัย จิรเพียงทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1. หลักการและเหตุผล	1
1.2. วัตถุประสงค์	1
1.3. ผลที่คาดว่าจะได้รับ	2
1.4. ขอบเขตของการพัฒนา	2
1.4.1. คุณลักษณะของโปรแกรมเข้า/ถอดรหัสสัญญาณวิดีโอตามมาตรฐาน H.263	2
บทที่ 2 VIDEO CODING FOR LOW BIT RATE COMMUNICATION	3
2.1. ขอบเขต	3
2.2. ส่วนประกอบต่างๆของ การทำการเข้ารหัส ถอดรหัส ไฟล์รูปภาพ	3
2.2.1. ไฟล์อินพุทและไฟล์เอาต์พุท	4
2.2.2. ไฟล์บิตอินพุทและไฟล์บิตเอาต์พุท	4
2.3. รูปแบบของซอร์ทโคท (Source format)	7
2.4. Video source coding algorithm	10
2.4.1. GOBs, slices, macroblocks and blocks	10
2.4.2. การคาดเคาตำแหน่งของภาพ (Prediction)	12
2.4.3. Motion compensation	13
2.4.4. Quantization	14
2.4.5. การควบคุมการเข้ารหัสและถอดรหัส (Coding control)	15
2.5. Syntax and Semantics	16
2.5.1. Picture layer	23
2.5.2. Group of Blocks Layer	35
2.5.3. Macroblock layer	37
2.5.5. Block Layer	45
2.6. Decoding process	50
2.6.1. Motion compensation	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2. Coefficients decoding 52
2.6.3. Reconstruction of blocks 53
บทที่ 3 Interface Algorithm (IALG) 54
3.1. บทนำ 54
3.2. The IALG Interface 54
3.2.1. algAlloc() 55
3.2.2. algInit() 56
3.2.3. algFree() 57
3.3. Module-Specific Interface 57
3.3.1 V-table 58
3.3.2 โครงสร้างอัลกอริทึม 59
3.4. คำนิยามของ Dynamic System 60
3.4.1. The Create Function 60
3.4.2. The Algorithm-Specific Create Function 61
3.4.3. ความต้องการหน่วยความจำ (Memory Requirements) 62
3.4.4. ฟังก์ชัน Activate 62
3.4.5. ฟังก์ชัน Move 63
3.5. ทำไมต้องเป็น Dynamic System 64
3.6. Dynamic System ทำงานอย่างไร 64
3.6.1. สถานะตอน Power-On 65
3.6.2. การเริ่มต้นการโทรออก 66
3.6.3. การสลับไปใช้งานการเข้ารหัสเสียง 67
3.6.4. การวางสายโทรศัพท์ 69
3.7. บทสรุปท้ายบท 69
บทที่ 4 การทดสอบและผลการทดสอบ 71
บทที่ 5 วิจารณ์และบทสรุป 81
บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 ไดอะแกรมของการเข้ารหัส ถอดรหัสไฟล์รูปภาพ	3
รูปที่ 2-2 แสดงตำแหน่งของ บล็อคของความสว่างและบล็อคของสี	8
รูปที่ 2-3 การเข้ารหัสและถอดรหัสไฟล์ภาพ	10
รูปที่ 2-4 การจัดเรียง GOB ของ picture ที่มีรูปแบบ CIF	11
รูปที่ 2-5 แสดงถึงการจัดเรียง block ใน macroblock	12
รูปที่ 2-6 โครงสร้างของไฟล์ภาพ	18
รูปที่ 2-7โครงสร้างของไฟล์ภาพ	19
รูปที่ 2-8 โครงสร้างของไฟล์ภาพ	20
รูปที่ 2-9 โครงสร้างของไฟล์ภาพ	21
รูปที่ 2-10 โครงสร้างของไฟล์ภาพ	22
รูปที่ 2-11 แสดงถึง โครงสร้างของ รูปภาพ	23
รูปที่ 2-12 แสดงถึงส่วนประกอบของ GOB	36
รูปที่ 2-13 แสดงถึงโครงสร้างของ macroblock	37
รูปที่ 2-14 แสดงถึงโครงสร้างของ block layer	45
รูปที่ 2-15 รูปแบบการ predict motion vector	51
รูปที่ 2-16 แสดงถึงการจัดเรียงค่า quantized coefficient ตามลำดับ	52
รูปที่ 3-1 Application, implementation และ the V-table	55
รูปที่ 3-2 แสดงตัวอย่าง algAlloc() ที่ใช้จองหน่วยความจำ	55
รูปที่ 3-3 แสดงส่วนของ Object ที่ implement v-table ไว้ภายใน	56
รูปที่ 3-4 แสดงการ initialize ตัวแปร ของ algInit()	56
รูปที่ 3-5 แสดงส่วนฟังก์ชัน algFree()	57
รูปที่ 3-6 แสดง โครงสร้างของ SPI	57
รูปที่ 3-7 แสดงฟังก์ชัน G723ENC_TI_encode	58
รูปที่ 3-8 แสดง code ที่ใช้ในการประกาศ v-table	59
รูปที่ 3-9 แสดง โครงสร้างของ TMS320 DSP Algorithm Standard	59
รูปที่ 3-10 แสดงตัวอย่าง v-table ของ XMP Algorithm	60
รูปที่ 3-11 แสดงหน่วยความจำของ XMP Algorithm	61
รูปที่ 3-12 แสดง High-level function ที่ชื่อว่า XMP_create ()	61
รูปที่ 3-13 แสดงส่วนของ Code เมื่อผ่านค่าพารามิเตอร์เข้ามาเป็น NULL	62
รูปที่ 3-14 แสดงส่วนการเรียก ALG_create () ทั่วไปเพื่อใช้งาน	62
รูปที่ 3-15A แสดงข้อมูลในหน่วยความจำก่อนจัดการเรียงข้อมูล	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-15B แสดงข้อมูลในหน่วยความจำหลังจากจัดเรียงข้อมูลแล้ว 63
รูปที่ 3-16 แสดงระบบตัวอย่างที่ใช้ในการอธิบาย Dynamic System 65
รูปที่ 3-17 แสดงหน่วยความจำของระบบตัวอย่างเมื่อตอน Power-on 65
รูปที่ 3-18 แสดงหน่วยความจำเมื่อ Voice-recognition ขึ้นมาทำงาน 66
รูปที่ 3-19 แสดงหน่วยความจำที่ไม่เพียงพอสำหรับ DTMF อัลกอริทึม 66
รูปที่ 3-20 แสดงการย้ายข้อมูลของ Voice-recognition ไปไว้ที่หน่วยความจำภายนอก 67
รูปที่ 3-21 แสดงหน่วยความจำไม่เพียงพออีกครั้ง 68
รูปที่ 3-22 แสดงการจัดเรียงข้อมูลใหม่ในระบบตัวอย่าง 68
รูปที่ 4-1 แสดงรูปภาพตัวอย่างจากสัญญาณภาพวิดีโอที่ใช้ในการทดสอบ 71
รูปที่ 4-2 ไม่มีการเซตโหมดใดๆ 75
รูปที่ 4-3 เซต advanced prediction mode 75
รูปที่ 4-4 เซต syntax-base arithmetic coding 75
รูปที่ 4-5 เซต unrestricted motion vector 75
รูปที่ 4-6 เซต pb-frames 75
รูปที่ 4-7 เซต advanced prediction mode และ Pb-frame mode 76
รูปที่ 4-8 เซต advanced prediction mode 76
รูปที่ 4-9 เซต syntax-base arithmetic coding และ PB frame mode 77
รูปที่ 4-10 เซต unrestricted motion vector mode และ advanced prediction mode 77
รูปที่ 4-11 เซต unrestricted motion vector mode และ PB frame mode 77
รูปที่ 4-12 เซต unrestricted motion vector mode และ syntax-base arithmetic coding mode 77
รูปที่ 4-13 เซต unrestricted motion vector mode และ Syntax-base arithmetic coding mode และ Advanced prediction mode 77
รูปที่ 4-14 เซต unrestricted motion vector mode และ syntax-base arithmetic coding mode และ PB frame mode 77
รูปที่ 4-15 เซต syntax-base arithmetic coding และ Advanced prediction mode และ PB frame mode 78
รูปที่ 4-16 เซต unrestricted motion vector mode และ advanced prediction mode และ advanced prediction mode 78
รูปที่ 4-17 เซต unrestricted motion vector mode และ Syntax-base arithmetic coding mode และ Advanced prediction mode และ PB frame mode 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2-1 ค่า BPPmaxKb ที่น้อยที่สุดที่จะเป็นไปได้สำหรับไฟล์ภาพที่แตกต่างกันไป	6
ตารางที่ 2-2 จำนวน pixel per line และจำนวน lines สำหรับแต่ละรูปแบบของภาพ	8
ตารางที่ 2-3 การกำหนดค่า aspect ratio เอาเอง (custom pixel aspect ratio)	9
ตารางที่ 2-4 ค่า k สำหรับ แต่ละขนาดของ GOB	11
ตารางที่ 2-5 ความหมายของ PAR	30
ตารางที่ 2-6 แสดงถึงความสัมพันธ์ระหว่าง QUANT และ BQuant	34
ตารางที่ 2-7 แสดงถึง ค่า variable length code สำหรับ MCBPC (สำหรับ I –picture)	39
ตารางที่ 2-8 แสดงถึงค่า variable length code สำหรับ MCBPC (สำหรับ I –picture)	39
ตารางที่ 2-9 แสดงถึง macroblock data type และ องค์ประกอบของ normal picture	40
ตารางที่ 2-10 แสดงถึง macroblock data type และ องค์ประกอบของ PB frame	40
ตารางที่ 2-11 แสดงถึงค่า variable length code สำหรับ MODB	41
ตารางที่ 2-12 ค่า DQUANT และค่าความแตกต่างของ QUANT	42
ตารางที่ 2-13 แสดงถึงค่า variable length code ของ CBPY	42
ตารางที่ 2-14 แสดงถึงค่า variable length code สำหรับ MVD	44
ตารางที่ 2-14 (ต่อ) แสดงถึงค่า variable length code สำหรับ MVD	45
ตารางที่ 2-15 แสดงถึงระดับการ reconstruction สำหรับ INTRA DC Coefficient	46
ตารางที่ 2-16 แสดงถึง variable length code ของ TCOEF	47
ตารางที่ 2-16 (ต่อ) แสดงถึง variable length code ของ TCOEF	48
ตารางที่ 2-16 (ต่อ) แสดงถึง variable length code ของ TCOEF	49
ตารางที่ 2-17 แสดงถึงรูปแบบการทำนาย motion vectors	49
ตารางที่ 2-18 แสดงถึงการแก้ไข ขนาดของ chrominance vectors	51
ตารางที่ 4-1 ตารางแสดงผลการทดลองจากการทดสอบที่ 1	72
ตารางที่ 4-2 แสดงผลการทดลองจากการทดสอบที่ 2	73
ตารางที่ 4-3 แสดงผลการทดลองจากการทดสอบ decode ที่ 3	74
ตารางที่ 4-4 แสดงผลการทดลองจากการทดสอบที่ 4	76
ตารางที่ 4-5 แสดงการกำหนดค่าพารามิเตอร์ที่ใช้ในการประมวลผล	79
ตารางที่ 4-6 แสดงการใช้งานหน่วยความจำของแต่ละ Instance object ที่ถูกสร้างส่วนเข้ารหัส	79
ตารางที่ 4-7 แสดงขนาดของหน่วยความจำของ Library ที่ใช้งาน	79
ตารางที่ 4-8 แสดงขนาดหน่วยความจำของตาราง Code Book ที่ใช้งานในการเข้ารหัส	79
ตารางที่ 4-9 แสดงการใช้งานหน่วยความจำของแต่ละ Instance object ที่ถูกสร้างส่วนถอดรหัส	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1. หลักการและเหตุผล

เนื่องจากปัจจุบันมีความก้าวหน้าทางเทคโนโลยีในปัจจุบันเป็นไปอย่างรวดเร็ว ความต้องการติดต่อสื่อสารระหว่างกันเพื่อการรับรู้ข้อมูลข่าวสารให้ทันท่วงทีจากซีกโลกหนึ่งไปยังอีกซีกโลกหนึ่ง จากการส่งข้อมูลติดต่อกันทางจดหมาย ก็เปลี่ยนเป็นการติดต่อกันทางโทรศัพท์ที่มีแต่การพูดอย่างเดียว ด้วยความต้องการที่ไม่หยุดหยั้งของมนุษย์ การที่ได้สามารถเห็นหน้าตาของฝ่ายตรงข้ามระหว่างการสนทนาทำให้มีการพัฒนาวิธีการต่างๆอย่างมากมาย เพื่อตอบสนองต่อความต้องการนี้

จากความต้องการข้างต้นเป็นเหตุให้จำเป็นต้องมีการส่งข้อมูลที่เป็นสัญญาณวิดีโอเข้าไปที่ช่องทางการสื่อสารพร้อมกับข้อมูลอื่นๆอีก หากเราส่งข้อมูลที่เป็นสัญญาณวิดีโอเข้าไปโดยไม่ทำการใดๆ เลย ข้อมูลที่ส่งไปนั้น จะมีขนาดที่ใหญ่มากและด้วยช่องทางในการสื่อสารมีความสามารถในการตอบสนองต่อปริมาณข้อมูลที่ไม่มากนักเป็นเหตุให้มีการคิดค้นวิธีให้สามารถส่งข้อมูลสัญญาณวิดีโอเข้าไปที่สื่อกลางได้

จากที่ได้กล่าวมาหนึ่งในวิธีการนั้นได้แก่ การบีบอัดสัญญาณวิดีโอมาตรฐาน H.263 ที่ International Telecommunication Union (ITU) ได้กำหนดเป็นมาตรฐานไว้

และด้วยเหตุนี้ปริญญานิพนธ์ฉบับนี้ได้จัดทำขึ้นมาเพื่อพัฒนาโปรแกรมที่สามารถบีบอัดข้อมูลที่เป็นสัญญาณวิดีโอตามมาตรฐาน H.263 โดยได้พัฒนาโปรแกรมขึ้นบน Digital Signal Processor (DSP) ตระกูล TMS320C6000 โดยเลือกใช้งาน DSP เบอร์ TMS320C6713 ขึ้นมาที่มีความสามารถในการประมวลผลรวดเร็ว และมีประสิทธิภาพสูง

1.2. วัตถุประสงค์

1. เพื่อศึกษากระบวนการเข้ารหัสและการถอดรหัสสัญญาณวิดีโอ ตามมาตรฐาน H.263
2. เพื่อศึกษาการใช้งาน Board TMS320C6713 DSP starter kit
3. เพื่อศึกษาการทำงานของ DSP ตระกูล TMS320C6713
4. เพื่อสร้างโปรแกรมการเข้าและถอดรหัสสัญญาณวิดีโอที่ทำงานบน DSP ตระกูล TMS320C6713
5. เพื่อศึกษาการเขียนโปรแกรมที่เป็นไปตามมาตรฐานอัลกอริทึม (standard algorithm) ของ TI (Texas Instruments)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3. ผลที่คาดว่าจะได้รับ

1. ได้รับความรู้ ความเข้าใจ กระบวนการเข้า/ถอดรหัสสัญญาณวิดีโอตามมาตรฐาน H.263
2. ได้รับความรู้ ความเข้าใจ การใช้งาน Board DSP starter kit
3. ได้รับความรู้ ความเข้าใจ DSP ตระกูล TMS320C6713
4. ได้รับความรู้ ความเข้าใจในด้านการเขียนโปรแกรมที่เป็นไปตามมาตรฐานอัลกอริทึม
5. โปรแกรมที่สามารถทำงานได้บน Board DSP starter kit ที่เป็นไปตามมาตรฐานอัลกอริทึม

1.4. ขอบเขตของการพัฒนา

ในการพัฒนาโครงงานนี้จะดำเนินการสร้างโปรแกรมบีบอัดสัญญาณวิดีโอตามมาตรฐาน H.263 เฉพาะในส่วนของอัลกอริทึมในการเข้า/ถอดรหัสสัญญาณวิดีโอเท่านั้น ประกอบด้วยส่วนการทำงาน 2 ส่วน ได้แก่ ส่วนของการเข้ารหัสสัญญาณวิดีโอ และส่วนของการถอดรหัสสัญญาณวิดีโอ สำหรับส่วนฮาร์ดแวร์นั้นจะใช้งาน Board TMS320C6713 DSP starter kit ของ บริษัท Spectrum digital ในการทดสอบการทำงานโปรแกรม

สำหรับส่วนของซอฟท์แวร์จะทำการ Implement โดยใช้ Code Composer Studio (C6713 DSK CCS) ในการพัฒนาโปรแกรมที่เป็นไปตามมาตรฐานอัลกอริทึมและสร้างเป็นไลบรารี (Library)

1.4.1. คุณลักษณะของโปรแกรมเข้า/ถอดรหัสสัญญาณวิดีโอตามมาตรฐาน H.263

1. โปรแกรมสามารถใช้งาน Option ต่างๆ ได้ดังนี้
   - 1.1. Unrestricted Motion Vector Mode
   - 1.2. Syntax-base Arithmetic Coding mode
   - 1.3. Advance Prediction mode
   - 1.4. PB-frame mode
2. โปรแกรมเป็นไปตามมาตรฐานการเขียนโปรแกรมของ DSP
3. โปรแกรมสามารถเข้า/ถอดรหัสสัญญาณวิดีโอตามมาตรฐาน H.263 ได้
4. โปรแกรมสามารถทำงานรองรับการทำงานแบบ Multi-channel ได้
5. โปรแกรมสามารถทำงานแบบ Re-entrancy ได้
6. โปรแกรมมีความสามารถ Code re-locatability

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

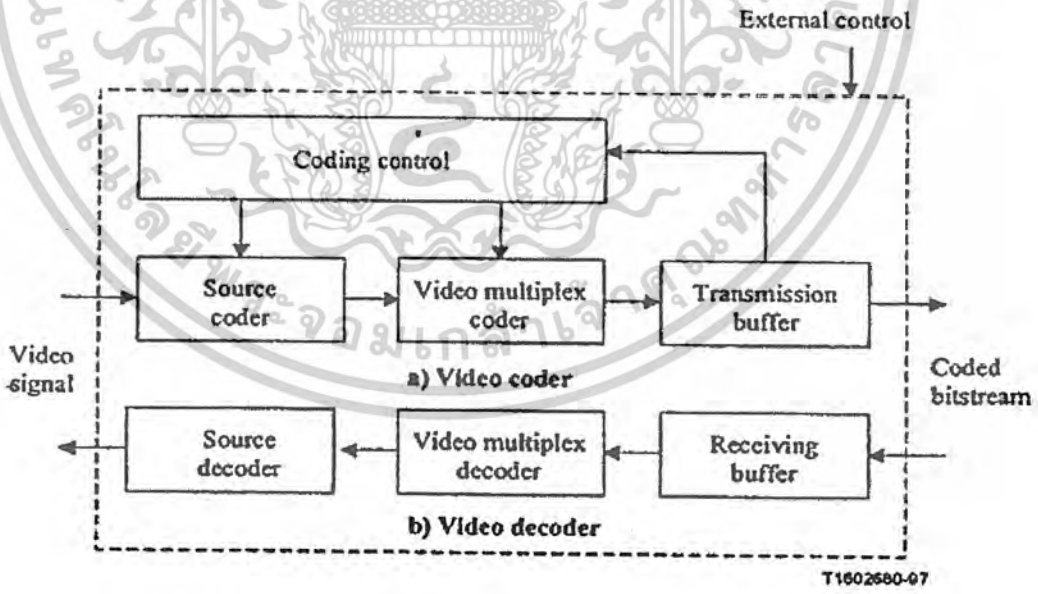
บทที่ 2

VIDEO CODING FOR LOW BIT RATE COMMUNICATION

2.1. ขอบเขต

ส่วนของการแนะนำนี้ กล่าวระบุถึง การเข้ารหัส ถอดรหัส (Coding) ไฟล์รูปภาพซึ่งการ coding นี้สามารถใช้ในการย่อส่วนประกอบของภาพเคลื่อนไหวที่บิตเรทต่ำๆได้พื้นฐานการเซตค่าต่างๆ (config) ของรูปแบบการเข้ารหัส/ถอดรหัสของไฟล์รูปภาพ มีพื้นฐานมาจาก รูปแบบมาตรฐาน H.261 โดยมีการรวม โหมดการทำงาน 16โหมด ที่สามารถปรับเปลี่ยนค่าได้เอาไว้ เพื่อปรับปรุงคุณสมบัติให้ดีขึ้นและเพื่อเพิ่มฟังก์ชั่นให้มากขึ้น

2.2. ส่วนประกอบต่างๆของ การทำการเข้ารหัส ถอดรหัส ไฟล์รูปภาพ



รูปที่ 2-1 ไดอะแกรมของการเข้ารหัส ถอดรหัสไฟล์รูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.ไฟล์อินพุทและไฟล์เอาต์พุท

การจะทำให้ เป็น รูปแบบมาตรฐานที่ใช้งานได้ในช่วง 625 – 525 ขนาดของ ไฟล์รูปภาพ (source format) จะต้องอยู่ในรูปแบบของ 352 พิกเซล x 288 พิกเซล (common Intermediate Format: CIF) แต่ทั้งนี้ เราก็สามารถกำหนดค่าเองได้เพื่อที่จะสามารถใช้ใน ช่วงที่กว้างกว่านี้ได้

2.2.2.ไฟล์บิตอินพุทและไฟล์บิตเอาต์พุท

การเข้ารหัส จะกระทำการเข้ารหัสให้อยู่ในรูปแบบของ ไฟล์บิต ซึ่งไฟล์บิตนี้ นี้อาจจะถูกนำไปรวมกับสัญญาณรูปแบบอื่นๆได้ส่วนถอดรหัสสามารถทำได้ในทางกลับกัน (ย้อนกระบวนการ)

1. ความถี่ของการสุ่มตัวอย่าง

รูปภาพจะถูกสุ่มขึ้นมาในลักษณะจำนวนเท่าของเลขจำนวนเต็มของอัตราการแสดงภาพ. โดยที่ความถี่ของ สัญญาณนาฬิกาของเครื่องประมวลผล และ สัญญาณนาฬิกาของเครือข่ายจะมีลักษณะเป็นแบบ ไม่ต่อเนื่อง ทั้งคู่

2. อัลกอริทึมของการเข้ารหัสและถอดรหัส

การนำเทคนิคอัลกอริทึมมาใช้นี้เพื่อการนำความซ้ำซ้อนที่มีมากเกินไปมาใช้ให้เกิดประโยชน์โดยการลดความซับซ้อนของข้อมูลซึ่งสามารถทำได้โดยการผสมผสานของการเข้ารหัสถอด/รหัสไฟล์ภาพที่อ้างอิงกับการเข้ารหัสและภอดรหัสของไฟล์ภาพก่อนหน้า (inter-picture prediction) ในส่วนของการถอดรหัสนั้นก็จะมีความสามารถในการชดเชยความซ้ำซ้อนที่ถูกลดไปจากการเข้ารหัส ซึ่งในส่วนนี้เองได้ทำให้เกิด ทางเลือกใหม่ ๆ ทำให้เกิดเทคนิคใน การเข้ารหัสถอดรหัส หลักการพิจารณาครึ่งบิต (Half pixel precision) ถูกนำมาใช้สำหรับเทคนิคการชดเชยความซ้ำซ้อนของข้อมูล การเข้ารหัสถอดรหัสแบบใช้บิตไบนารี่ (Variable length coding or VLC) ถูกใช้แทนสำหรับสัญลักษณ์ เพื่อที่จะถูกส่งผ่านข้อมูลต่อไป

นอกจาก อัลกอริทึมที่ได้กล่าวมานี้ยังมีโหมดการทำงานอื่นอีกที่สามารถนำมาใช้ได้ โดยใช้ได้ทั้งแบบใช้ร่วมกันหรือใช้แยกกัน (ขึ้นอยู่กับข้อจำกัดที่แน่นอน) ข้อมูลเพิ่มเติมอาจถูกรวมไว้ใน บิตไฟล์เพื่อเพิ่มความสามารถในการ แสดงไฟล์ภาพ รายละเอียดการใช้โหมดการทำงานจะมีดังต่อไปนี้

- โหมดการทำงานที่ไม่จำกัดโมชั่นเวคเตอร์ (Unrestricted Motion Vector mode)

ในโหมดการทำงานนี้, โมชั่นเวคเตอร์ (motion vectors) จะถูกอนุญาตให้ชี้ไปยังบริเวณที่อยู่ภายนอกภาพได้โดยจะมี พิกเซลที่ขอบ (edge pixels) จะถูกใช้เป็นตัวทำนาย สำหรับการทำนายพิกเซลที่ไม่มีอยู่จริง โดยผลที่ได้รับคือ สามารถมีการเคลื่อนที่ข้าม/ผ่านพิกเซลที่ขอบของรูปไปได้ โดยเฉพาะอย่างยิ่ง สำหรับรูปแบบภาพที่มีขนาดเล็ก ในโหมดนี้ มีการรวมไปถึงการขยายขอบเขต ของโมชั่นเวคเตอร์ ดังนั้นโมชั่นเวคเตอร์ ที่มีขนาดใหญ่กว่านั้น ๆ จะถูกนำมาใช้ได้ ส่วนนี้จะมีประโยชน์อย่างมากในกรณีของไฟล์ภาพเคลื่อนไหวของเครื่องบันทึกภาพ และ ไฟล์ภาพที่มีขนาดใหญ่ๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมดการทำงานที่ใช้การคำณวนช่วยในการเข้ารหัสและถอดรหัส(Syntax-based Arithmetic Coding mode)

ในโหมดการทำงานนี้การเข้ารหัส และถอดรหัสแบบใช้ผลลัพธ์ที่ได้จากการคำณวนจะถูกนำมาใช้แทนที่ การเข้ารหัสและถอดรหัสแบบใช้บิตไบนารี่

- โหมดการเข้ารหัสและถอดรหัสแบบก้าวหน้า(Advanced Prediction mode)

โหมดการทำงานนี้, การทดแทนข้อมูลในส่วนที่ถูกซ้อนทับ(Overlapped Block Motion Compensation หรือ OBMC) จะถูกนำมาใช้สำหรับบล็อคข้อมูลที่แทนความสว่าง(luminance part) ของภาพที่อ้างอิงกับภาพก่อนหน้า(P-pictures) จะมีการใช้ โมชั่นเวคเตอร์ ขนาด 8 x 8 จำนวน 4 โมชั่นเวคเตอร์ แทนที่ โมชั่นเวคเตอร์ ขนาด 16 x 16 จำนวน 1 เวคเตอร์ สำหรับ กลุ่มของบล็อคข้อมูล (macroblocks) บางอัน ในรูปนั้นๆ. โดยที่ฝั่งของการเข้ารหัส จะเป็นฝั่งที่ตัดสินว่าจะใช้เวคเตอร์ รูปแบบ/ลักษณะ (type ไหน.ซึ่งเวคเตอร์ แบบ 4 เวคเตอร์ จะใช้ จำนวนบิตมากกว่าแต่ก็จะสามารถเข้ารหัสและถอดรหัส (prediction) ได้ดีกว่า. การใช้ mode นี้ โดยปกติจะทำให้เกิดผลที่ดีขึ้น. แต่ก็จะมีผลเสียที่ เกิดขึ้นเนื่องจาก การทดแทนข้อมูลในส่วนที่ถูกซ้อนทับ ทำให้เกิด การสูญหายของข้อมูลในบางส่วนไปได้

- โหมดการเข้ารหัสและถอดรหัสแบบสลับ(PB-frames mode)

โหมดนี้จะ ประกอบด้วยรูปภาพ 2 รูป ซึ่งถูก เข้ารหัสให้ กลายเป็นหนึ่งภาพ ชื่อ โหมดการเข้ารหัสแบบสลับ (PB) มาจากชื่อของ รูปแบบของรูปภาพ คือ การเข้ารหัสและถอดรหัสแบบอ้างอิงกับไฟล์ภาพก่อนหน้า(P-pictures) และ การเข้ารหัสและถอดรหัสที่อ้างอิงกับไฟล์ภาพก่อนหน้าและไฟล์ภาพที่กำลังถูกกระทำอยู่ (B-pictures) ดังนั้น โหมดการเข้ารหัสและถอดรหัสแบบสลับนั้น จะประกอบด้วย ก) หนึ่ง P-picture ซึ่ง หนึ่ง P-picture อันนี้ถูกทำนายมาจาก P-picture ที่ถูก decoded มาแล้วก่อนหน้านี้ ข) หนึ่ง B-picture ซึ่ง หนึ่ง B-picture นี้ ถูกทำนายมาจาก P-picture ทั้ง 2 ส่วน คือ 1) P-picture ที่ถูกถอดรหัสมาแล้วก่อนหน้านี้ และ 2) P-picture ที่กำลังถูกถอดรหัสอยู่ในปัจจุบัน สาเหตุที่ถูกตั้งชื่อว่า B-picture เนื่องจากในหลาย ๆ ส่วนของ B-pictures อาจจะถูกอ้างอิงจากทั้ง 2 ทาง คือถูกอ้างอิงจากภาพในอดีต และจากภาพในอนาคต. ด้วยวิธีการเข้ารหัสและถอดรหัสในลักษณะนี้ ทำให้สามารถเพิ่มอัตราการส่งข้อมูล (frame rate) ได้โดยที่ไม่ต้องเพิ่ม อัตราความเร็วในการส่งบิตข้อมูล (bit rate)

- Modified Quantization mode

ใน optional mode นี้ จะช่วยพัฒนาในเรื่องของความสามารถในการควบคุมบิตเรท,ลดความผิดพลาดของการปรับระดับความละเอียดของข้อมูล(Chrominance quantization error), ขยายขอบเขตของตัวเลขสัมประสิทธิ์ (discreate cosine transfroem หรือ DCT), และมีการกำหนด/จำกัดค่าตัวเลขสัมประสิทธิ์ที่แน่นอนไว้ โหมด นี้ มีการเปลี่ยนแปลงความหมายของความแตกต่างของการปรับระดับความละเอียด (differential quantization step size parameter) ของ ไฟล์ข้อมูล โดยการขยายช่วง ของการเปลี่ยนแปลง ความแตกต่างของการปรับระดับความละเอียด (ซึ่งสามารถกำหนดค่าได้) ให้กว้างขึ้น และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันยังช่วยลดการปรับระดับความละเอียดซึ่งถูกใช้สำหรับ บล็อคข้อมูลแสดงส่วนความสว่าง. และได้มีการขยาย ช่วง ของระดับตัวเลขสัมประสิทธิ์ เพื่อให้แน่ใจว่าค่าของตัวเลขสัมประสิทธิ์ใด ๆ นี้จะสามารถถูกเข้ารหัสและถอดรหัส ให้มีความถูกต้องตามความละเอียดของข้อมูลได้อย่างถูกต้อง ตัวเลขสัมประสิทธิ์จะถูกจำกัดค่าแน่นอนไว้ เพื่อเพิ่มความสามารถในการตรวจจับข้อผิดพลาดและช่วยลดความซับซ้อนในการทำ การถอดรหัสด้วย

3. อัตราการส่งข้อมูล(Bit rate)

อัตราการส่งข้อมูลถูกจัดให้อยู่ภายนอก.และอัตราการแสดงภาพ (Video bit rete) จะสามารถมีการเปลี่ยนแปลงค่าได้ ใน ส่วนทฤษฎีนี้ไม่มีข้อบังคับใด ๆ เกี่ยวกับ อัตราการแสดงภาพ โดยข้อบังคับต่าง ๆ จะถูกกำหนดโดย ผู้รับและผู้ส่งเท่านั้น

4. การจัดเก็บข้อมูล(Buffering)

ฝั่งของการเข้ารหัสจะคอยควบคุมไฟล์ภาพเอาท์พุท ของมันเองให้เป็นไปตามข้อกำหนดของ ฝั่งถอดรหัส

ด้านจำนวนของบิต ที่ถูกสร้างขึ้นโดยวิธีการเข้ารหัสและถอดรหัสรูปภาพรูปเดียวใด ๆ ก็ตามจำนวนบิตจะไม่เกินกว่าค่าสูงสุดที่ถูกระบุไว้โดยพารามิเตอร์ BPPmaxKb, ซึ่งถูกวัดอยู่ในหน่วยของ 1024 บิต ค่า BPPmaxkb ที่น้อยที่สุดที่สามารถเป็นได้จะขึ้นอยู่กับ ขนาดของรูปที่ใหญ่ที่สุดที่ถูกนำมาใช้ในไฟล์ข้อมูลนั้น (ดูตารางที่ 2-1) โดยขนาดของ ภาพ จะถูกวัดเป็นจำนวนเท่าของความกว้างและความสูงของรูป สำหรับ บล็อคของความสว่าง (luminance or (Y)), วัดอยู่ในหน่วยของ พิกเซล ตัว ฝั่งของการถอดรหัสอาจจะใช้ค่า BPPmaxKb มากกว่าค่าที่ระบุไว้ใน ตารางที่ 2-1 ได้, ซึ่งการใช้ค่า BPPmaxKb ในลักษณะนี้จะใช้ในกรณีอื่น ๆ โดยสามารถดูตัวอย่างการใช้ได้ใน Recommendation H.245

เมื่อมีการนำ Temporal, SNR, และ Spatial Scalability Mode มาใช้, จำนวนบิตที่ส่งออกไปในแต่ละ enhancement layer ของแต่ละ picture จะไม่เกินค่าสูงสุดที่ถูกกำหนดไว้โดย BPPmaxKb

Y picture size in pixels	Minimum BPPmaxKb
Up to 25344 (or QCIF)	64
25360 to 101376 (or CIF)	256
101392 to 405504 (or 4CIF)	512
405520 and above	1024

ตารางที่ 2-1 ค่า BPPmaxKb ที่น้อยที่สุดที่จะเป็นไปได้สำหรับไฟล์ภาพที่แตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ความสมมาตรของการส่งข้อมูล(Symmetry of Transmission)

การเข้ารหัสและถอดรหัสข้อมูลอาจจะถูกใช้สำหรับได้ทั้งแบบทิศทางเดียวหรือสองทิศทางก็ได้

2.3.รูปแบบของซอร์ทโคท (Source format)

การเข้ารหัสและถอดรหัส ที่ดำเนินการบน รูปภาพที่ไม่ต่อเนื่อง(Non-interlaced pictures) จะมีรายละเอียดของ รูปแบบของซอร์ทโคท ดังต่อไปนี้

1) the picture format, ซึ่งถูกกำหนดโดยจำนวนของ pixels per line, จำนวนของ lines per picture, และ the pixel aspect ratio
2) The timing between pictures, ซึ่งถูกกำหนดโดย the Picture Clock Frequency (PCF). ตัวอย่างเช่น the Common Intermediate Format (CIF) มี 352 pixels per line, มี 288 lines, มี pixel aspect ratio ที่ 12:11, และมี picture clock frequency 30000/1001 pictures per second

ส่วนการเข้ารหัสและถอดรหัสที่ดำเนินการบน รูปภาพที่ไม่ต่อเนื่องและเกิดขึ้นที่ PCF เท่ากับ 30000/1001 (ประมาณ 29.97) ครั้งต่อวินาทีจะถูกเรียกว่า the CIF PCF นอกจากนั้นยังสามารถปรับการใช้งาน PCF ได้โดยปัจจัยภายนอก (คือสามารถ ปรับแต่งค่าเอง ไม่ใช้ตามของเดิม) โดย custom PCF มีค่าอยู่ที่ I800000/ (clock divisor * clock conversion factor) ซึ่ง clock advisor มีค่าได้ตั้งแต่ 1 ถึง 127 ส่วน clock conversion factor มีค่าเท่ากับ 1000 หรือ 1001 เพียงค่าใดค่าหนึ่งเท่านั้น. ค่า Tolerance บน picture clock frequency เท่ากับ บวกลบ 50 ppm

รูปภาพ จะถูกเข้ารหัส แบบ บล็อคความสว่าง 1 บล็อค และบล็อคของสีอีกสองบล็อค (Y, Cb and Cr) โดยที่ ส่วนประกอบเหล่านี้และการเข้ารหัสจะเป็นตัวแทนในการบอกถึงค่า sampled values ของพวกมันเอง ดังเช่นที่นิยามไว้ใน ITU-R Recommendation BT.601-5.

- Black = 16
- White = 235
- Zero colour difference = 128
- Peak colour difference = 16 and 240

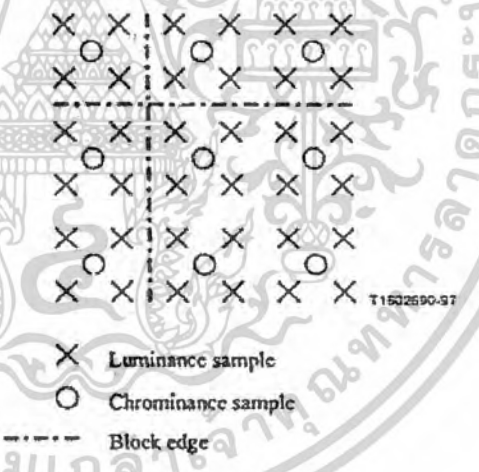
มาตรฐานของ รูปแบบของภาพ(Picture format)มี 5 รูปบบด้วยกัน คือ sub-QCIF, QCIF, CIF, 4CIF และ 16CIF. และมีความเป็นไปได้ในการปรับ ค่าได้เองตามต้องการ สำหรับ รูปแบบของภาพทั้งหมดนี้ จะมีโครงสร้างเป็น dx pixels per line, dy lines per picture ตามข้อตกลงของการตั้งฉาก การสุ่มตัวอย่าง ของแต่ละความแตกต่างของสี 2 สี (Cr และ Cb) ได้ผลเป็น dx/2 pixels per line, dy/2 lines per picture, ในรูปแบบของ orthogonal. สำหรับมาตรฐานทั้ง 5 รูปแบบของ picture formats, ค่าของ dx, dy, dx/2 และ dy/2 ถูกกำหนดให้ดัง ตารางที่ 2-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture format	Number of pixels for luminance (dx)	Number of lines for luminance (dy)	Number of pixels for chrominance (dx/2)	Number of lines for chrominance (dy/2)
sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

ตารางที่ 2-2 จำนวน pixel per line และจำนวน lines สำหรับแต่ละรูปแบบของภาพ

สำหรับทุกรูปแบบของภาพ, ตัวอย่างของความแตกต่างสี จะถูกเลื่อนตำแหน่งจนกระทั่งขอบของบล๊อคตรงกับขอบของบล๊อค luminance ดังแสดงในรูปที่ 2-2 pixel aspect ratio จะเป็นค่าเดียวกับแต่ละ picture formats และยังเป็นค่าเดียวกับ QCIF และ CIF ซึ่งมีการนิยามไว้ใน Recommendation H.261: (288/3):(352/4), ซึ่งค่าดังกล่าว สามารถลดทอนให้เป็นจำนวนเฉพาะได้เท่ากับ 12:11 ส่วน The picture area ที่ถูกคลอบคลุมโดยมาตรฐาน picture formats ทั้งหมดจะมี aspect ratio 4:3 ยกเว้นมาตรฐาน sub-QCIF ตัวเดียวที่จะไม่ใช้ค่า aspect ratio นี้



รูปที่ 2-2 แสดงตำแหน่งของ บล็อคของความสว่างและบล็อคของสี

ถ้ามีการปรับ custom pixel aspect ratio ตั้งแต่เริ่มแรกโดยกรณีอื่น ๆ ปัจจัยภายนอก เราสามารถดูค่า custom pixel aspect ratio ของ custom picture formats ได้ใน ตารางที่ 2-3 Custom picture formats สามารถที่จะมีจำนวนของ lines และจำนวนของ pixels per line ใด ๆ ก็ได้ ซึ่งแสดงให้เห็นว่า จำนวนของ lines สามารถแบ่งได้เป็น 4 ส่วนและอยู่ในช่วง [4, ..., 1152], และแสดงให้เห็นว่า จำนวนของ pixels per line ก็สามารถแบ่งได้เป็น 4 ส่วน เช่นกัน และอยู่ในช่วง [4, ..., 2048]. สำหรับ picture formats ซึ่งมีความกว้างและความสูงที่ไม่สามารถแบ่งเป็น 16 ส่วนได้, picture ดังกล่าวก็จะถูก decode ไปตามปกติ ราวกับว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความกว้างและความสูงของมันมีค่าเพิ่มมากขึ้นจนกระทั่งสามารถแบ่งเป็น 16 ส่วนได้ แต่จะเกิดผลลัพธ์คือ picture นั้นจะถูกตัดส่วนขวาและส่วนล่างออกไป เพื่อแสดงความกว้างและความสูงของรูปได้อย่างเหมาะสม

Pixel aspect ratio	Pixel width:Pixel height
Square	1:1
CIF	12:11
525-type for 4:3 picture	10:11
CIF for 16:9 picture	16:11
525-type for 16:9 picture	40:33
Extended PAR	m:n, m and n are relatively prime

ตารางที่ 2-3 การกำหนดค่า aspect ratio เอาเอง (custom pixel aspect ratio)

สำหรับ ผู้ถอดรหัส และผู้เข้ารหัส ทุก ๆ คน จะสามารถปฏิบัติการโดยการใช้ CIF picture clock frequency ได้ ซึ่ง ผู้ถอดรหัสและผู้เข้ารหัสบางคนอาจจะรองรับ custom picture clock frequencies ได้ด้วย สำหรับ ผู้ถอดรหัส ทุก ๆ คน จะสามารถปฏิบัติการโดยการใช้ sub-QCIF picture format ได้ และ ผู้ถอดรหัส ทุก ๆ คนนี้ยังจะสามารถเลือกโดยใช้ QCIF picture format ได้ด้วย ผู้ถอดรหัสบางคนอาจจะปฏิบัติการด้วย CIF, 4CIF หรือ 16CIF, หรือปฏิบัติการโดยใช้ custom picture formats. (สรุปคือ ผู้ถอดรหัส และผู้เข้ารหัส ทุก ๆ คน สามารถปฏิบัติการโดยใช้ CIF picture clock frequency, sub-QCIF picture format, QCIF picture format ได้ ส่วน ผู้ถอดรหัส และผู้เข้ารหัส บางคนจะสามารถ support custom picture clock frequencies ได้ และบางคนนี่แหละอาจจะปฏิบัติการด้วย CIF, 4CIF หรือ 16CIF ได้)

ผู้เข้ารหัส จะสามารถเลือกใช้ Picture formats หลายแบบ ซึ่ง 1 ในแบบนั้นเป็นการรวมกันของ sub-QCIF และ QCIF โดย ผู้เข้ารหัส จะเป็นผู้ตัดสินว่าจะใช้รูปแบบไหน แต่ไม่จำเป็นว่าจะต้องปฏิบัติการโดยใช้ทั้งคู่ ผู้เข้ารหัส บางคนอาจเลือกปฏิบัติการด้วย CIF, 4CIF, 16CIF หรือ custom picture formats.การจะเลือกใช้ optional formats และ picture clock frequencies แบบใดนั้นจะถูกตัดสินโดย ผู้ถอดรหัส ซึ่งสิ่งที่มีผลกับการตัดสินของ ผู้ถอดรหัส คือปัจจัยภายนอก/กรณีที่มีความต้องการ – ดูตัวอย่างได้ใน Recommendation H.245. สามารถดูภาพรวมโดยสมบูรณ์ของ picture formats และ video coding algorithms ที่เป็นไปได้, มีการอ้างถึงการอธิบาย terminal ได้ โดยดูตัวอย่างดังกล่าวได้ที่ Reccommendation H.324.

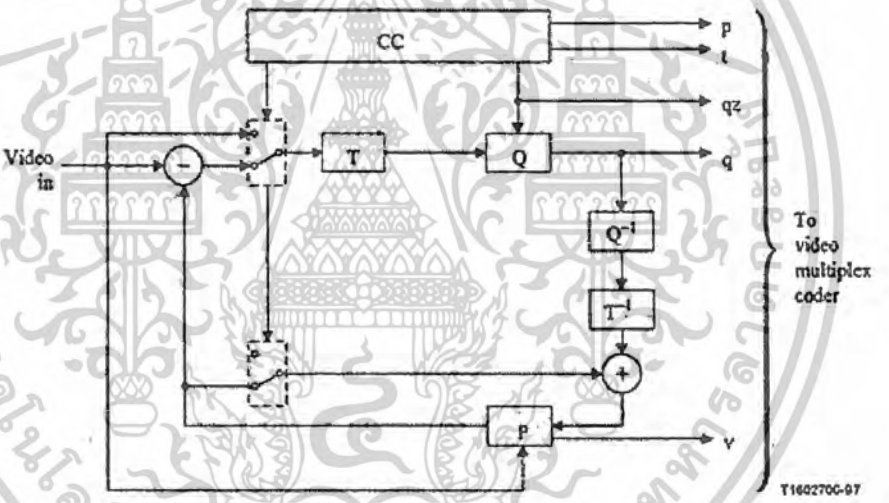
NOTE – สำหรับ CIF, จำนวนของ pixels per line ที่เหมาะสมในทางปฏิบัติ ได้มาโดยการ sampling ส่วนที่ใช้งานได้ของ luminance และ colour difference signals จาก 525- หรือ 625- line sources ที่ความถี่ 6.75 และ 3.375 MHz. ตามลำดับ. โดยที่ความถี่เหล่านี้มีความสัมพันธ์แบบ simple กับ those ที่อยู่ใน ITU-R Recommendation BT.601-5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการส่งครั้งหนึ่ง ๆ ปัจจัย/สิ่งที่กำหนดการใช้งาน (Means) จะเป็นตัวจำกัด อัตราการส่งภาพ (picture rate) สูงสุดของ ผู้เข้ารหัส โดยจะใช้จำนวนที่น้อยที่สุดของ รูปภาพที่ไม่สามารถถูกส่งไปได้เป็นเงื่อนไข. ซึ่งการเลือกจำนวนที่น้อยที่สุดนี้จะถูกกำหนดโดยปัจจัยภายนอก (ดูตัวอย่างใน Recommendation H.245). สำหรับการคำนวณจำนวนที่น้อยที่สุดของ รูปภาพที่ไม่สามารถถูกส่งผ่านไปได้ใน โหมดการเข้ารหัสและถอดรหัสแบบสลับ(PB-frames mode), the P-picture และ B-picture ของโหมดการเข้ารหัสและถอดรหัสแบบสลับ จะอยู่ในรูป 2 รูปที่แยกจากกัน.

2.4. Video source coding algorithm

การเข้ารหัสและถอดรหัส ถูกแสดงในลักษณะทั่ว ๆ ไปในรูป รูปที่ 2-3 ซึ่ง ส่วนประกอบหลัก ๆ คือ การทำนาย (prediction), การแปลงข้อมูลภาพให้เป็นบิต(block transformation) และ การปรับระดับความละเอียดของข้อมูล(quantization.)



T Transform
Q Quantizer
P Picture Memory with motion compensated variable delay
CC Coding control
p Flag for INTRA/INTER
t Flag for transmitted or not
qz Quantizer indication
q Quantizing index for transform coefficients
v Motion vector

รูปที่ 2-3 การเข้ารหัสและถอดรหัสไฟล์ภาพ

2.4.1. GOBs, slices, macroblocks and blocks

รูปภาพแต่ละรูป จะถูกแบ่งออกเป็น Groups Of Blocks (GOBs) หรือ slices อย่างใดอย่างหนึ่ง

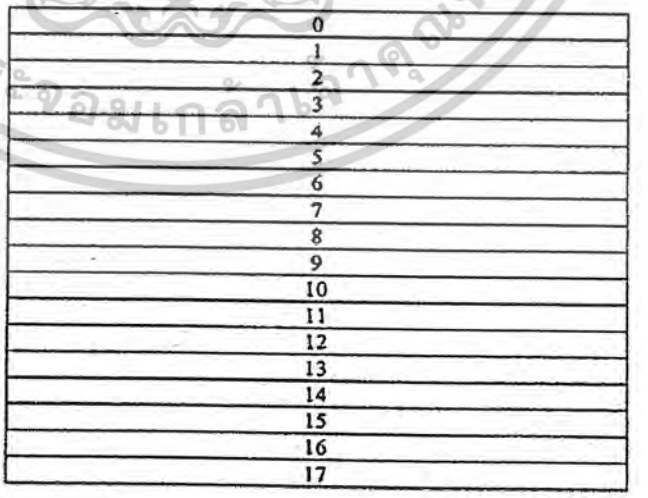
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ GOB จะมีค่าเท่ากับ k * 16 แถว, ซึ่งค่าของ k ขึ้นอยู่กับจำนวนแถวที่อยู่ใน picture format ดูสรุปในตารางที่ 2-4 นี้

Number of lines dy	Value of k when not in RRU mode
4,...,400	1
404,...800	2
804,...1152	4

ตารางที่ 2-4 ค่า k สำหรับ แต่ละขนาดของ GOB

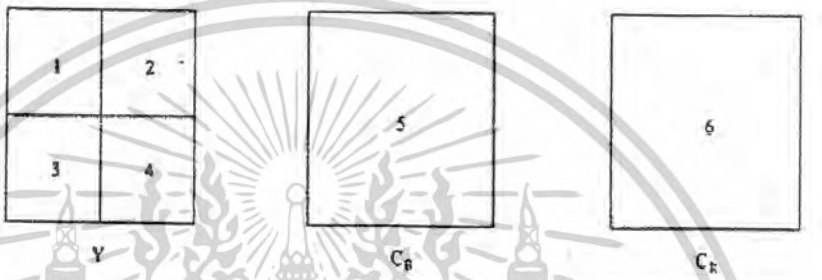
เมื่อมีการใช้ custom picture sizes, จำนวนของแถวที่อยู่ด้านล่างสุดของ GOB อาจจะมีค่าน้อยกว่า k * 16 ในกรณีที่จำนวนแถวในรูป ไม่สามารถแบ่งได้เป็น k * 16 ส่วน (คือหารด้วย k * 16 ไม่ลงตัวนั่นเอง) อย่างไรก็ตาม ทุก ๆ GOB ในแต่ละมาตรฐาน picture formats จะมีค่า k * 16 แถว, ขณะเดียวกันด้านจำนวนแถวในแต่ละมาตรฐาน picture formats จะมีค่าเป็นจำนวนเต็มคูณด้วย k * 16. ตัวอย่างเช่น (ไม่มีการใช้ RRU mode) จำนวนของ GOBs per picture มีค่า 6 สำหรับ sub-QCIF, มีค่า 9 สำหรับ QCIF, และมีค่า 18 สำหรับ CIF, 4CIF และ 16CIF การนับจำนวน GOB สามารถทำได้โดย vertical scan (แนวดิ่ง) ของ GOBs, โดยเริ่มต้นที่ GOB ด้านบนสุด (number 0) และสิ้นสุดการ scan ที่ด้านล่างสุดของ GOB ดูตัวอย่างการจัดการ GOB ดังกล่าวสำหรับมาตรฐานแบบ CIF picture format ได้ในรูป รูปที่ 2-4 โดย ข้อมูลของแต่ละ GOB จะประกอบไปด้วย GOB header (ซึ่งอาจจะว่างก็ได้) ตามด้วยข้อมูลสำหรับ macroblocks เมื่อมีการเพิ่มจำนวนของ GOB จะส่งผลให้ข้อมูลของ GOBs ถูกส่งผ่านไป 1 GOB (ส่งผ่านไปยัง GOB ถัดไป).



รูปที่ 2-4 การจัดเรียง GOB ของ picture ที่มีรูปแบบ CIF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GOB แต่ละ GOB ก็จะถูกแบ่งย่อยเป็น macroblocks โดยโครงสร้างของ macroblock แต่ละ macroblock จะมีความสัมพันธ์อยู่ในรูปแบบ 16 pixels by 16 lines ของ Y และ 8 pixels by 8 lines ของ Cb และ Cr นอกจากนั้น macroblock ยังประกอบด้วย 4 luminance blocks และ 2 colour difference blocks ดังในรูปที่ 2-5 ซึ่งแต่ละ luminance block หรือ chrominance block จะมีความสัมพันธ์อยู่ในรูปแบบ 8 pixels by 8 lines ของ Y, Cb หรือ Cr. โดย GOB จะประกอบด้วย 1 macroblock row สำหรับ sub-QCIF, QCIF และ CIF, จะประกอบด้วย 2 macroblock rows สำหรับ 4CIF, และจะประกอบด้วย 4 macroblock rows สำหรับ 16CIF.



รูปที่ 2-5 แสดงถึงการจัดเรียง block ใน macroblock

การนับจำนวน macroblock สามารถทำได้โดย Horizontal scan (แนวนอน) ของ macroblock rows ทิศทางจากซ้ายไปขวา, โดยเริ่มต้นที่ upper macroblock row และสิ้นสุดด้วย lower macroblock row. เมื่อมีการเพิ่มของ Macroblock number จะส่งผลให้ data ของ macroblocks ถูกส่งผ่านไป 1 macroblock (ส่งผ่านไปยัง macroblock ถัดไป).

2.4.2. การคาดเดาตำแหน่งของภาพ (Prediction)

รูปแบบดั้งเดิมของ prediction จะอยู่ในรูป Inter-picture และอาจจะถูกขยายทำให้เพิ่มขึ้นได้โดยเทคนิค การทดแทนตำแหน่งของข้อมูล (motion compensation) (ดูหัวข้อ 3.2.3). การเข้ารหัสและถอดรหัสที่มีการทำนายแบบ อ้างอิงไฟล์ภาพจะถูกเรียกว่า INTER ส่วน การเข้ารหัสและถอดรหัสที่ไม่มีการทำนายแบบ อ้างอิงไฟล์ภาพ จะถูกเรียกว่า INTRA การเข้ารหัส และถอดรหัส แบบ INTRA จะสามารถสังเกตได้จาก picture level (INTRA สำหรับ I-pictures หรือ INTER สำหรับ P-pictures) หรือสังเกตได้จาก macroblock level ใน P-pictures สำหรับ PB-frames mode, B-pictures จะถูก code แบบ INTER mode เสมอ ซึ่ง B-pictures จะถูก predict เป็นส่วน ๆ แบบ bidirectionally

โดยสรุป, H.263 มี basic picture types อยู่ 7 types (2 types แรก คือ INTRA และ INTER มีความสำคัญมาก) โดยทั้ง 7 types นี้ถูกแยกออกจากกันตาม prediction structure.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) INTRA : เป็น รูปภาพที่ไม่มีการใช้ reference picture(s) ในการ prediction (INTRA เรียกได้อีกชื่อหนึ่งว่า I-picture);
2) INTER : เป็น รูปภาพที่มีการใช้ temporally previous reference picture (ใช้ reference picture อันก่อนหน้า) ในการ prediction (INTER เรียกได้อีกชื่อหนึ่งว่า P-picture);
3) PB : เป็น frame ที่แสดงถึง picture จำนวน 2 pictures และมีการใช้ temporally previous reference picture
4) Improved PB : เป็น frame ที่มีรูปแบบเหมือน PB-frame แต่มีข้อดีกว่าในบางส่วน
5) B : เป็น รูปภาพที่มี 2 reference pictures, reference picture แรก มาก่อน B-picture (อดีต) ส่วนอีก reference picture นึง มาหลัง B-picture (คือไม่มี reference picture ณ ปัจจุบัน) และ reference picture ทั้ง 2 มีขนาดเท่ากัน
6) EI : เป็น รูปภาพที่มี reference picture เป็น reference picture ที่ถูกดึงมาจากหลาย ๆ ที่ในเวลาเดียวกัน โดยมีขนาด picture เท่ากัน หรือเล็กกว่า เพียงอย่างใดอย่างหนึ่ง
7) EP : เป็น รูปภาพที่มี 2 reference pictures, reference picture แรก มาก่อน EP-picture (อดีต) ส่วนอีก reference picture นึงเป็น reference picture ที่ถูกดึงมาจากหลาย ๆ ที่ในเวลาเดียวกัน โดยมีขนาด picture เท่ากัน หรือเล็กกว่า เพียงอย่างใดอย่างหนึ่ง

2.4.3. Motion compensation

การถอดรหัส จะยอมรับเพียง 1 vector ต่อ 1 macroblock แต่จะมีการเปลี่ยนแปลงได้ตามเงื่อนไขต่อไปนี้

1) ถ้ามีการใช้ the Advanced Prediction mode หรือ Deblocking Filter mode, จะยอมรับได้ที่ 1 หรือ 4 vectors ต่อ 1 macroblock
2) ถ้ามีการใช้ PB-frames mode, ใน 1 macroblock จะสามารถส่ง delta vector เพิ่มเข้าไปได้อีก 1 vector ทั้งนี้เนื่องจากเป็นการปรับตัวของ motion vectors ให้เข้ากับการ prediction ของ B-macroblock.
3) ถ้ามีการใช้ Improved PB-frame macroblock กรณีนี้จะเหมือนกับกรณีที่ใช้ PB-frames mode แต่จะสามารถเพิ่ม forward motion vector เพิ่มเข้าไปได้อีก 1 vector.
4) สำหรับ B-picture macroblocks จะถูกส่งผ่านไปกับ 1 forward motion vector และ 1 backward motion vector.
5) สำหรับ EP-pictures จะถูกส่งผ่านไปกับ 1 forward motion vector.

ทั้ง Horizontal components และ vertical components ของ the motion vectors จะมีค่าอยู่ในรูปของจำนวนเต็ม หรือครึ่งหนึ่งของจำนวนเต็ม (พวก 0.5 ทั้งหลาย) สำหรับ prediction mode แบบ default ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านี้จะถูกจำกัดอยู่ใน ช่วง [-16, 15.5] (ซึ่งเงื่อนไขนี้สามารถใช้ได้กับ forward motion vector components และ backward motion vector components สำหรับ B-pictures)

ใน Unrestricted Motion Vector mode, maximum range สำหรับ vector components จะถูกเพิ่มให้มี range ที่กว้างขึ้นกว่าเดิมอีกโดย

- ถ้าไม่มี PLUSPTYPE, range คือ [-31.5, 31.5], ภายใต้เงื่อนไขที่ว่า ถ้า predictor อยู่ใน ช่วง [-15.5, 16], เฉพาะค่าที่อยู่ใน ช่วง ของ [-16, 15.5] บริเวณใกล้ ๆ กับ predictor สำหรับแต่ละ motion vector component เท่านั้นที่จะสามารถเข้าถึงได้
- แต่ถ้าไม่มี PLUSPTYPE, และ predictor อยู่นอก range [-15.5, 16], การเข้าถึง จะสามารถเข้าถึงทุก ๆ ค่าที่อยู่ในช่วง [-31.5, 31.5] ที่มี sign เหมือนกับ the predictor รวมกับ the zero value
- ถ้ามี PLUSPTYPE, ข้อจำกัดต่าง ๆ สำหรับค่าของ the motion vector จะมีน้อยกว่า
- ใน RRU mode, range ของ the motion vector จะถูกขยายจนมีค่าเกือบเป็น 2 เท่า, และแต่ละ vector component จะถูกจำกัดให้มีเพียงรูปแบบ half-integer หรือ zero value. ดังนั้น ช่วงของแต่ละ Motion vector component จะเป็น [-31.5, 30.5] ใน the default RRU mode และสามารถขยาย range ให้กว้างขึ้นได้ถ้ามีการใช้ the Unrestricted Motion Vector mode
- ค่าบวกของ Horizontal และ vertical component ของ the motion vector แสดงให้เห็นว่า การ prediction นี้ถูกดำเนินการจาก pixels ใน referenced picture ซึ่งมีที่ว่างทางด้านขวาหรือด้านล่าง

โดยปกติ Motion vectors จะถูกจำกัดขอบเขตไว้ระดับนึง ดังนั้น all pixels ที่ถูก reference โดย motion vectors เหล่านี้จะมีขอบเขตอยู่ภายใน the coded picture area, ยกเว้นจะมีการใช้ mode อื่น ๆ ดังต่อไปนี้

- The Unrestricted Motion Vector mode, the Advanced Prediction mode, หรือ the Deblocking Filter mode
- B- และ EP-pictures ของ the Temporal, SNR, และ Spatial Scalability mode

2.4.4. Quantization

Number ของ quantizers จะเป็น 1 สำหรับตัวเลขสัมประสิทธิ์ตัวแรกของ INTRA (DC-coefficient) blocks และจะเป็น 31 สำหรับตัวเลขสัมประสิทธิ์ตัวอื่น ๆ นอกเสียจากว่าจะมีการใช้ Advanced INTRA Coding mode หรือ Modified Quantization mode (คือถ้ามีการใช้ 2 mode นี้ number ของ quantizers จะไม่เป็นดังข้างต้น). ภายใน Macroblock, quantizer ตัวเดิมจะถูกใช้สำหรับตัวเลขสัมประสิทธิ์ทุก ๆ ตัว ยกเว้นตัวเลขสัมประสิทธิ์ของ INTRA block อันแรก แต่ไม่มีการนิยามระดับของการตัดสินใจไว้ ตัวเลขสัมประสิทธิ์ตัวแรกของ INTRA blocks เป็นค่าที่บอกถึงการ transform dc value ให้อยู่ในรูปแบบเดียวกัน โดยให้อยู่ในรูป quantized ด้วย step size of 8 ส่วนแต่ละ Quantizers ทั้ง 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

quantizers จะใช้ spaced reconstruction levels เท่า ๆ กัน โดยมี central dead-zone ใกล้ ๆ zero และมี step size ของเลขคู่ที่อยู่ในช่วง 2 ถึง 62

NOTE – สำหรับ smaller quantization step sizes, full dynamic range ของ the transform coefficients จะไม่สามารถถูกแทนได้ นอกเสียจากว่าจะมีการใช้ Modified Quantization mode.

2.4.5. การควบคุมการเข้ารหัสและถอดรหัส (Coding control)

ค่าพารามิเตอร์หลาย ๆ ค่าอาจจะมีค่าที่ไม่คงที่/มีการแปรเปลี่ยนไป ทั้งนี้เพื่อที่จะควบคุมอัตราของการเกิดของ video data ที่ถูก code รวมถึงการประมาลก่อน the source coder, the quantizer, block significant criterion and temporal subsampling ซึ่งสัดส่วนของแต่ละ such measures ในภาพรวมของการควบคุม strategy ไม่ได้มีการกล่าวถึงไว้ใน recommendation

ตัวถอดรหัสสามารถที่จะบ่งการ tradeoff ที่แน่นอนระหว่าง spatial และ temporal resolution ของ the video signal ได้ ส่วนตัวเข้ารหัสจะบ่งการ tradeoff แบบ default ได้ที่จุดเริ่มต้นของการ call และจะแสดงการตอบสนองไปยังตัวถอดรหัสเพื่อร้องขอให้ทำการเปลี่ยนรูปแบบการ tradeoff นี้วิธีการส่งเหล่านี้ขึ้นกับปัจจัยภายนอก

1. Forced updating

ฟังก์ชั่นนี้สามารถทำได้โดยการ force การใช้ INTRA mode ของ the coding algorithm. แต่ไม่มีการนิยามในส่วน update pattern ไว้ ในการควบคุมการเพิ่มขึ้นของ inverse transform mismatch error, แต่ละ macroblock จะถูก code แบบ INTRA mode อย่างน้อยที่สุด 1 ครั้งในทุก ๆ 132 ครั้ง ที่ตัวเลขสัมประสิทธิ์ถูกส่งผ่านสำหรับ macroblock นี้ใน P-pictures. และสำหรับ EP-pictures ก็มี requirement แบบเดียวกัน ซึ่งแต่ละ macroblock จะถูก code แบบ INTRA หรือถูก code ด้วย mode ที่สูงกว่า INTRA อย่างน้อยที่สุด 1 ครั้งในทุก ๆ 132 ครั้ง เมื่อตัวเลขสัมประสิทธิ์ถูกส่งผ่านสำหรับ macroblock

2. Byte alignment of start codes

การจัดเรียง ของ start codes สามารถทำได้โดยการใส่บิต 0 เข้าไป (โดยที่การ ใส่บิตเข้าไปจะประกอบด้วยจำนวน บิต 0 ที่น้อยกว่า 8 บิต) เข้าไปก่อน start code ดังนั้น bit แรกของ start code จะเป็น bit แรกของ byte (เป็น most significant bit นั่นเอง) start code จะกลายเป็น byte ที่ถูกจัดเรียงถ้าตำแหน่งของ most significant bit เป็นผลคูณของ 8 bits จาก bit แรกใน H.263 bitstream. Start codes ของ picture, slice, และ EOSBS จะมีการจัดเรียง byte, ส่วน start code ของ GOB และ EOS อาจจะมีการจัดเรียง byte

NOTE 1 – จำนวนบิตทั้งหมด ที่ใช้สำหรับ certain picture จะมีค่าได้หลายค่า แต่จะเป็นค่าที่คูณด้วย 8 bits เสมอ

NOTE 2 – H.324 ต้องการ H.263 encoders ในการจัดเรียง picture start codes ด้วยจุดเริ่มของ logical information units ที่ถูกส่งไปยัง the Adaptation Layer (AL_SDU's)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5. Syntax and Semantics

The video syntax สามารถถูกจัดเรียงตามโครงสร้างลำดับขั้น (hierarchical structure) ได้เป็น 4 ชั้นหลัก ๆ (four primary layers) โดยมีการเรียง Layer จากชั้นบนไปยังชั้นล่างดังต่อไปนี้

- Picture;
- Group of Blocks, หรือ slice, หรือ video picture segment;
- Macroblock;
- Block.

The syntax diagram ถูกแสดงอยู่ในรูป รูปที่ 2-6 ข้อแนะนำในการทำความเข้าใจ diagram ดังกล่าว มีดังนี้

1) Arrow paths แสดง flows ที่เป็นไปได้ของ syntax elements. Syntax element ใด ๆ ก็ตามที่มีความยาวเป็น 0 จะถูกพิจารณาว่าไม่มีตัวตนในความตั้งใจจะสื่อความหมายด้วย arrow path diagramming นี้ (ดังตัวอย่าง, จะเห็นว่ามี arrow path อ้อมผ่าน PSTUF ไป ทั้ง ๆ ที่โดยปกติแล้วน่าจะมี arrow path วิ่งผ่านเลย ทั้งนี้เนื่องจาก length ของ PSTUF field อาจจะมีค่าเป็น 0).
2) ตัวย่อ/การย่อ และความหมายสำหรับแต่ละ syntax element จะอธิบายในข้อต่อ ๆ ไป
3) ส่วนของ syntax elements และ arrow paths ที่แสดงในรูปของ thick solid lines (เส้นหนาทึบ) หมายถึง syntax flow ของ "baseline" mode of operation, โดยที่ไม่มีการใช้ optional enhancements ใด ๆ. (Syntax นี้มีการกล่าวถึงแล้วใน Version 1 ของ Recommendation นี้ และยังคงมีเนื้อความคงเดิม ไม่มีการเปลี่ยนแปลงใด ๆ.)
4) ส่วนของ syntax elements และ arrow paths ที่แสดงในรูปของ thick dotted lines (เส้นประทึบ) หมายถึง elements ที่ถูกเพิ่มเข้าไปใน syntax flow ของ optional enhancements, ซึ่งมีการกล่าวถึงแล้วในทั้ง Version 1 และ Version 2 ของ Recommendation นี้. (โดย syntax นี้ยังคงมีเนื้อความคงเดิม ไม่มีการเปลี่ยนแปลงใด ๆ.)
5) ส่วนของ syntax elements และ arrow paths ที่แสดงในรูปของ thin solid lines (เส้นบาง) หมายถึง new elements ที่ถูกเพิ่มเข้าไปใน syntax flow ของ optional enhancements โดยตรงกับ optional features ที่เพิ่มเข้าไป โดยมีการระบุไว้ใน Version 2. (Syntax นี้ ไม่มีการกล่าวไว้ใน Version 1.)
6) Syntax element fields ที่ปรากฏอยู่ในรูปของ square-edged boundaries (เหลี่ยม ๆ) มีความหมายว่าเป็น fixed-length fields, Syntax element fields ที่ปรากฏในรูปของ rounded boundaries มีความหมายว่าเป็น variable-length fields. ส่วน syntax element อีกรูปแบบหนึ่งคือ DQUANT ซึ่งมี boundaries ทั้ง 2 รูปแบบ (square-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

edged รวมกับ rounded) เนื่องจาก DQUANT นี้สามารถเป็น variable length หรือ fixed length ก็ได้อย่างใดอย่างหนึ่ง.

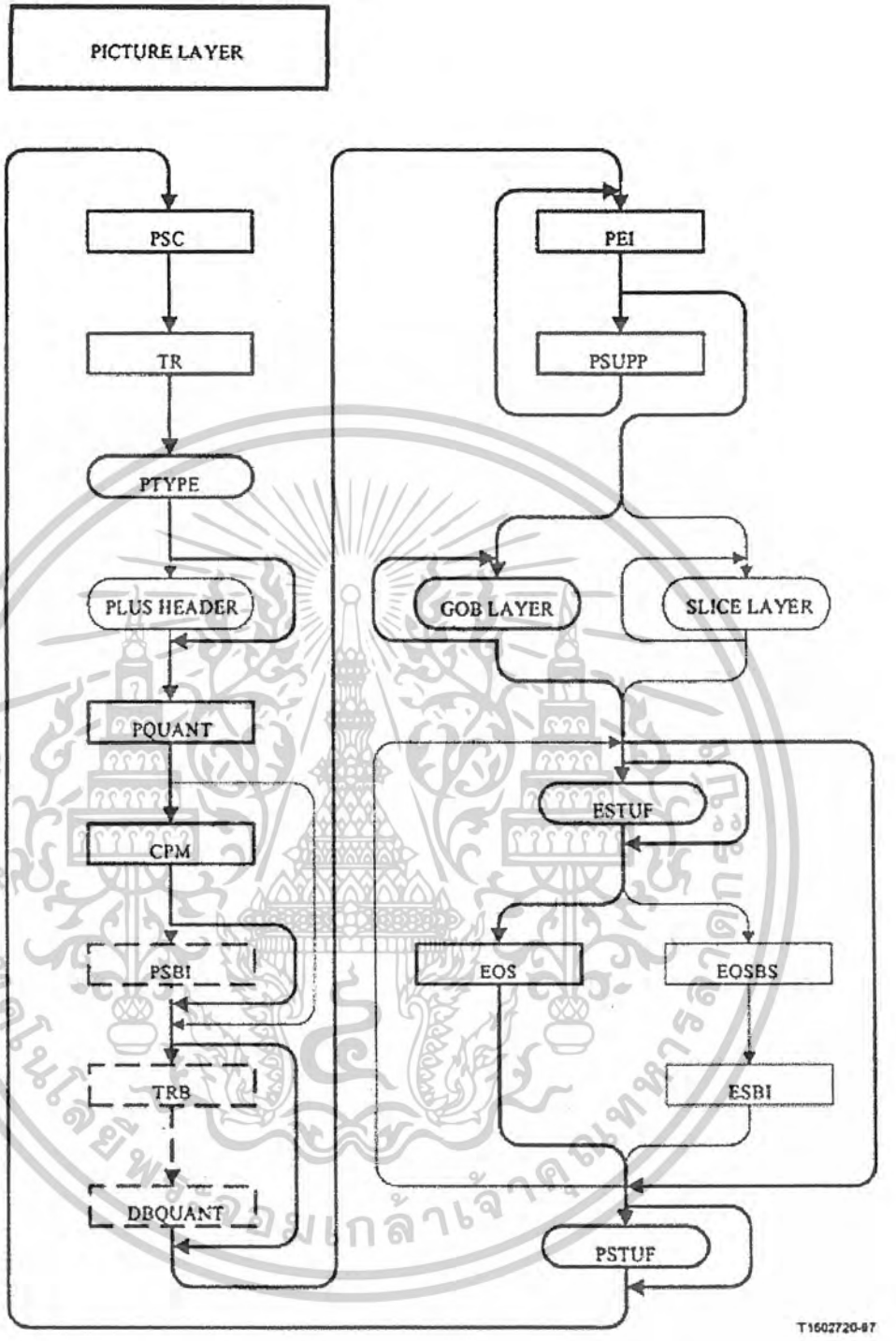
7) นิยามความหมายของ Fixed-length field คือ field ที่ค่า length ไม่ขึ้นกับ data ใน content ของตัว field นั้น ๆ เอง ค่าความยาวของ field นี้จะมีค่าได้ 2 ลักษณะ คือ มีค่าเท่ากับค่าเดิมเสมอ หรือมีค่าขึ้นกับ prior data ใน syntax flow อย่างใดอย่างหนึ่ง

8) คำว่า "Layer" ถูกใช้เพื่ออ้างถึงส่วนใด ๆ ก็ตามของ syntax ที่สามารถจะถูกเข้าใจและถูก diagram ได้อย่างชัดเจน

นอกจากจะมีการระบุให้ใช้เงื่อนไขอย่างอื่น ให้ใช้เงื่อนไขดังต่อนี้

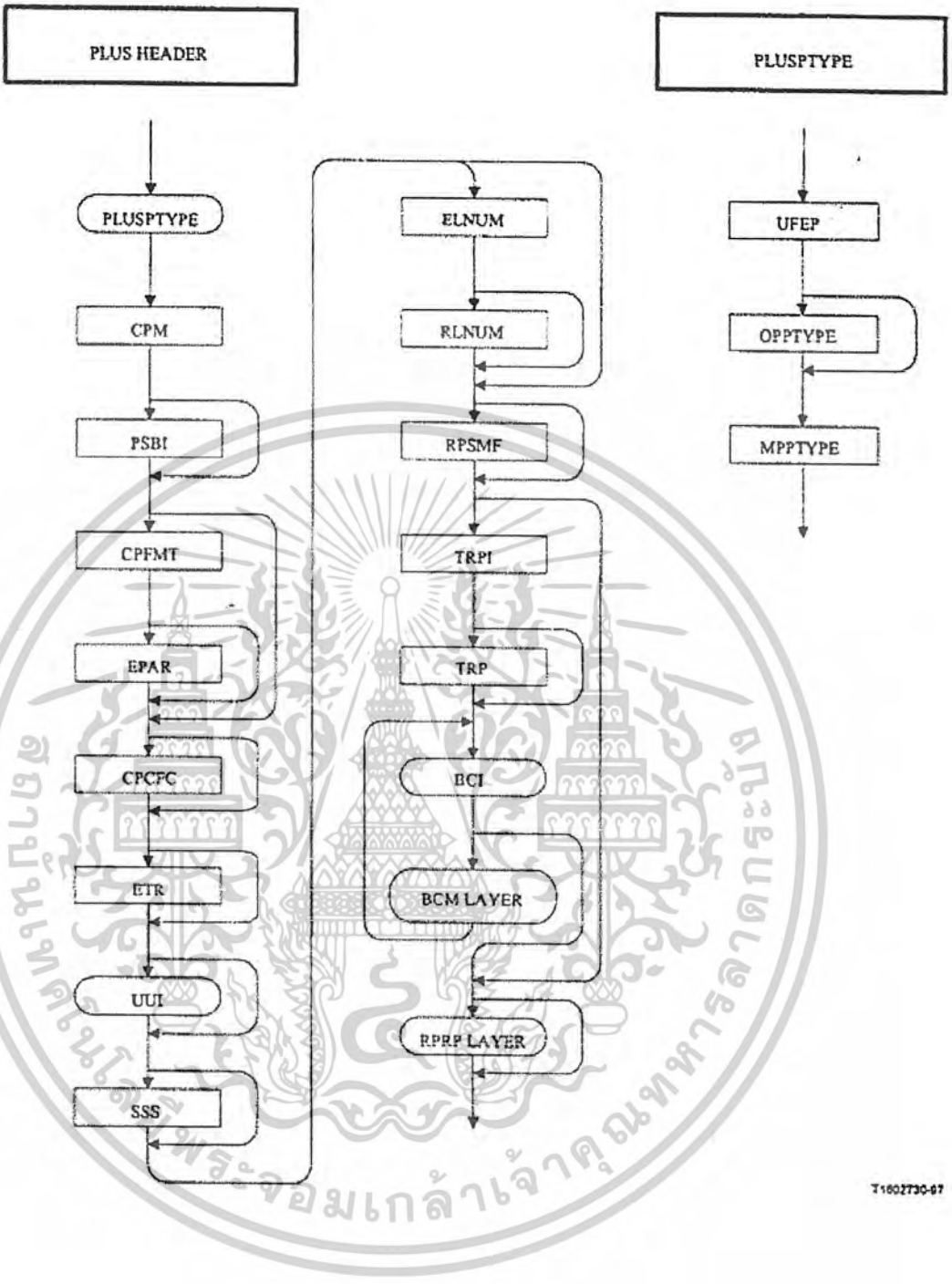
- The most significant bit จะถูกส่งก่อน. ซึ่งเป็นบิตที่ 1 และเป็น bit ที่อยู่ทางด้านซ้ายสุดใน code tables ใน Recommendation นี้
- ทุก ๆ unused bits หรือ spare bits ถูกกำหนดให้มีค่าเท่ากับ "1". Spare bits จะไม่ถูกใช้งานจนกระทั่ง ITU-T ทำการสั่งให้ functions ของ spare bits นั้น ๆ ทำงาน

61890
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



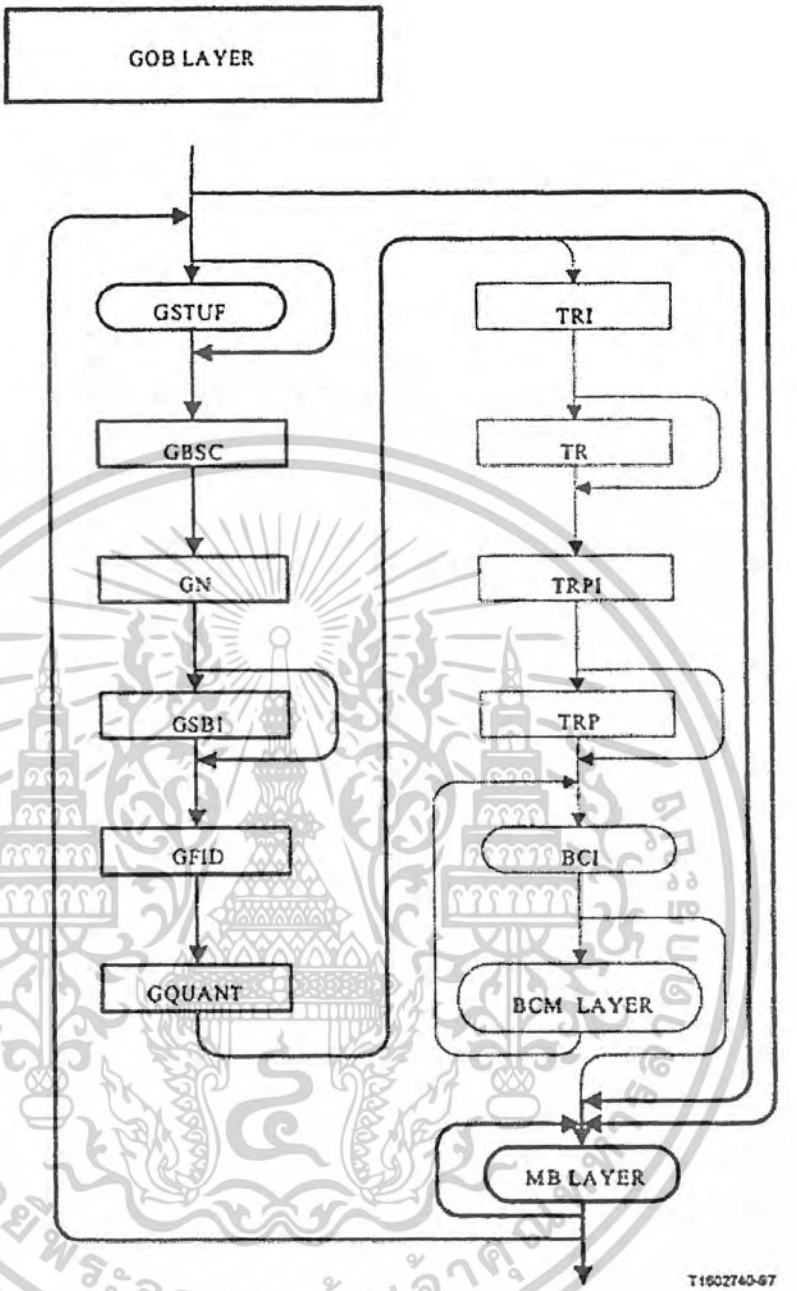
รูปที่ 2-6 โครงสร้างของไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



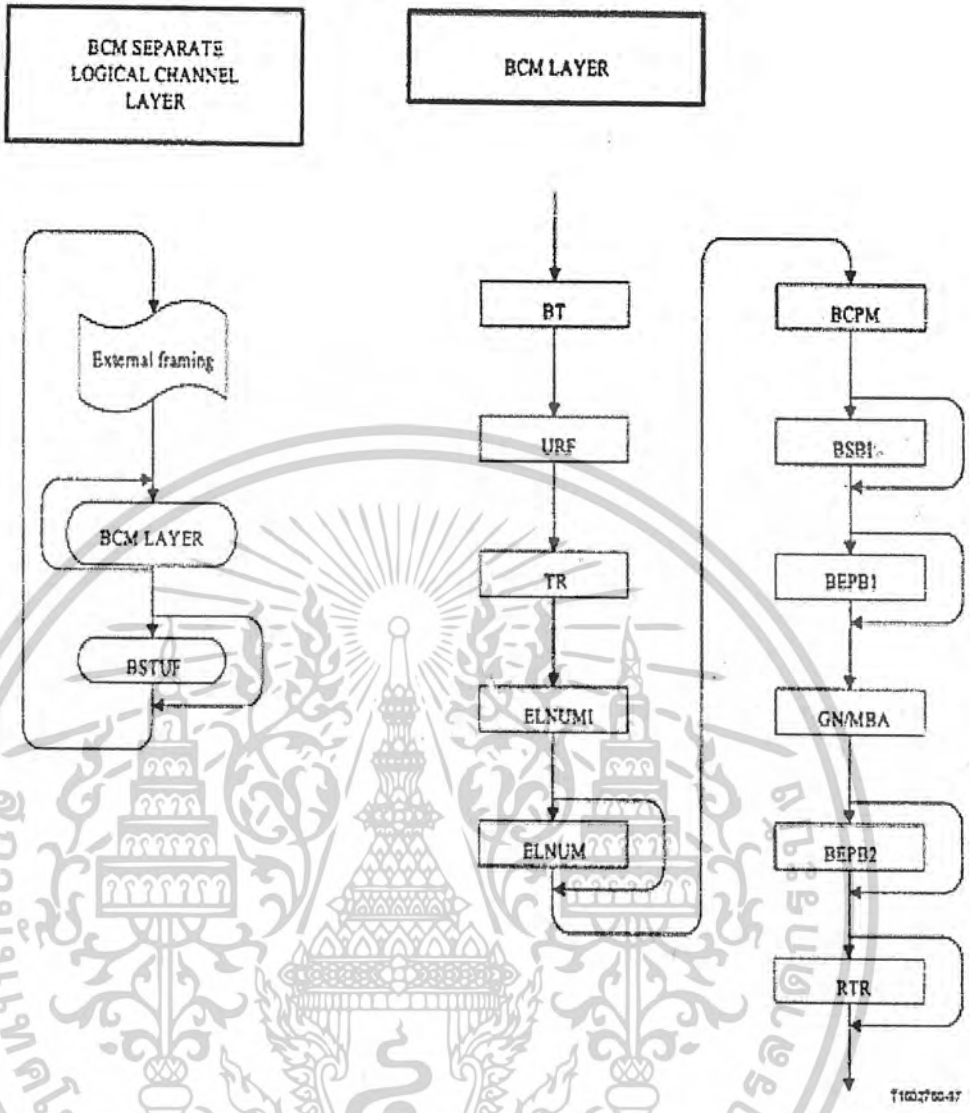
รูปที่ 2-7โครงสร้างของไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



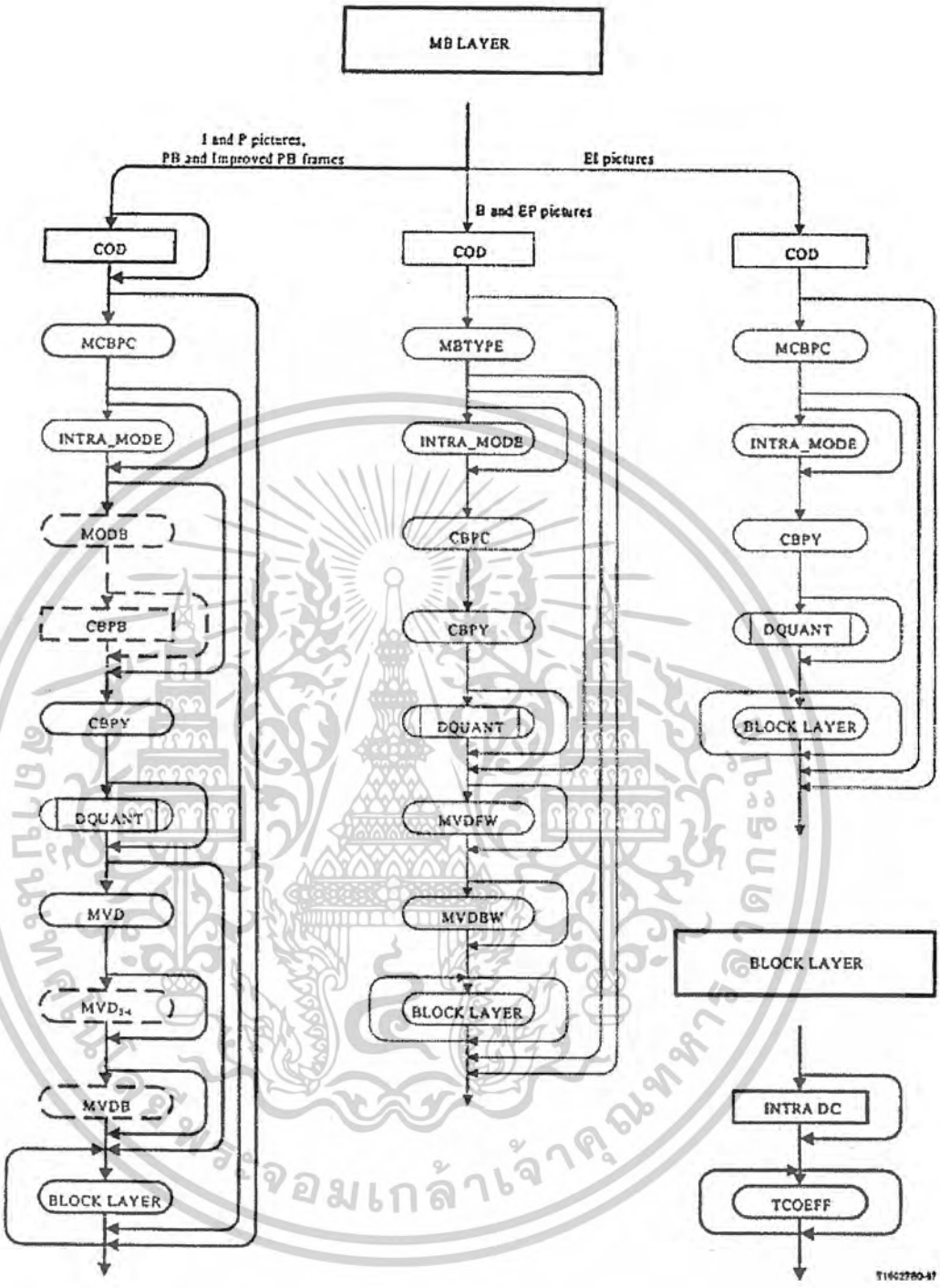
รูปที่ 2-8 โครงสร้างของไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 โครงสร้างของไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



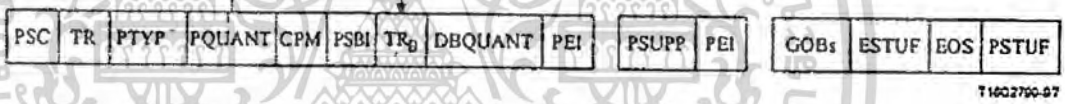
รูปที่ 2-10 โครงสร้างของไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1. Picture layer

ข้มูลของแต่ละ รูปภาพจะประกอบด้วย picture header และตามด้วยข้อมูล สำหรับ Group of Blocks, และสุดท้ายจะตามด้วย end-of-sequence code (เป็น optional) และ stuffing bits สำหรับโครงสร้างที่แสดงอยู่ในรูปที่ 2-7 เป็นรูปแบบที่ไม่มี PLUSPTYPE data field. PSBI จะมีอยู่กรณีเดียวเท่านั้นคือกรณีที่ถูกระบุโดย CPM TRb และ DBQUANT จะมีอยู่กรณีเดียวเท่านั้นคือกรณีที่ PTYPE มีการระบุใช้ "PB-frame" mode (ถ้ามี PLUSPTYPE field และมีการระบุใช้งาน DBQUANT, เงื่อนไขจะไม่เป็นดังข้างต้น)

PLUSPTYPE data field จะมีอยู่เมื่อมีการแสดง bits 6-8 ของ PTYPE. เมื่อมี PLUSPTYPE, set of data จะถูกเพิ่มเข้าไปใน bitstream ด้วย โดยแทรกอยู่ระหว่าง PTYPE และ PQUANT (ตามหลัง PTYPE + อยู่หน้า PQUANT). และการมี PLUSPTYPE ยังส่งผลให้ CPM และ PSBI fields ถูกเลื่อนมาด้านหน้า เลื่อนเข้ามาหา picture header. ดังนั้น ตำแหน่งของ 2 ตัวนี้จะปรากฏอยู่หลัง PLUSPTYPE ทันที แทนที่เดิมจะอยู่หลัง PQUANT ดูรูปแบบสำหรับข้อมูลที่ถูกเพิ่มเติมเข้าไปเมื่อมี PLUSPTYPE ได้ดังรูปที่ 2-8 ทุก ๆ fields ใน picture header data ที่ถูกเพิ่มเข้าไปถ้าอยู่หลัง PLUSPTYPE จะถือว่าเป็น optional ขึ้นอยู่กับว่า PLUSPTYPE ระบุการมีอยู่ของตัวไหนไว้. เมื่อมีการใช้ Slice Structured mode slices จะไปแทนที่ตำแหน่งของ GOBs ในรูปที่ 2-7



รูปที่ 2-11 แสดงถึง โครงสร้างของ รูปภาพ

1. Picture Start Code (PSC) (22 bits)

PSC มีค่า 0000 0000 0000 0000 1 00000 (22 bits) Start codes ของทุก ๆ รูปจะมีการจัดเรียง byte ซึ่งสามารถทำได้โดยการแทรก PSTUF bits ตามความจำเป็นเข้าไปก่อน start code ดังนั้น bit แรกของ start code จะเป็น bit แรก (และเป็น most significant bit) ของ byte

2. Temporal Reference (TR) (8 bits)

ค่าของ TR ถูกสร้างขึ้นมาโดยการเพิ่มค่าของมันเองใน temporally previous reference picture header โดย 1 บวกกับจำนวนของ skipped pictures หรือ non-reference pictures

3. Type Information (PTYPE) (Variable Length)

ข้อมูลเกี่ยวกับ complete picture มีดังนี้

- Bit 1 มีค่าเป็น "1" เสมอ, เพื่อมิให้ซ้ำกับ start code
- Bit 2 มีค่าเป็น "0" เสมอ, สำหรับค่าอื่น ๆ ที่ไม่ใช่ 0 ดูรายละเอียดใน Recommendation H.261.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Bit 3 Split screen indicator, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 4 Document camera indicator, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 5 Full Picture Freeze Release, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 6-8 Source Format, "000" forbidden, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" reserved, "111" extended PTYPE.

ถ้า บิต 6-8 ไม่เท่ากับ "111", ซึ่ง "111" นี้แสดงถึงรูปแบบ extended PTYPE (PLUSPTYPE), 5 bits ต่อไปนี้ (bit 9-13) จะเป็น PTYPE

- Bit 9 Picture Coding Type, เป็น "0" เป็นแบบ INTRA (I-picture), เป็น "1" เป็นแบบ INTER (P-picture).
- Bit 10 Optional Unrestricted Motion Vector mode เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 11 Optional Syntax-based Arithmetic Coding mode เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 12 Optional Advanced Prediction mode เป็น "0" หมายถึง off, เป็น "1" หมายถึง on.
- Bit 13 Optional PB-frames mode "0" เป็นแบบ normal I- หรือ P-picture, "1" เป็นแบบ PB-frame.

Split screen indicator เป็น signal ที่บอกว่า upper half (ครึ่งบน) และ lower half (ครึ่งล่าง) ของ picture ที่ถูก decode จะถูกแสดงเป็นแบบ side by side. Bit นี้ไม่มีผลกระทบโดยตรงในการ encoding หรือ decoding ของ picture.

Full Picture Freeze Release เป็น signal จาก encoder ที่ response ไปยัง request สำหรับ packet retransmission (if not acknowledged) หรือ fast update request หรือ Picture freeze request และอนุญาตให้ ตัวถอดรหัส ออกจาก freeze picture mode และแสดง picture ที่ถูก decode ในสภาพปกติ

ถ้าบิต 6-8 แสดง source format ที่แตกต่างกันกว่าใน picture header อันก่อนหน้า จะรู้ว่า picture ปัจจุบันจะเป็น I-picture นอกเสียจากว่าในบิต 6-8 จะเป็น extended PTYPE และการตัดสินเพื่อใช้งาน (the capability to use) the optional Reference Picture Resampling mode ถูกกระทำจากภายนอก (ดูตัวอย่างใน Recommendation H.245)

Bits 10-13 อ้างถึง optional modes ที่ถูกใช้หลังจากการตกลงกันระหว่าง encoder และ decoder โดยจะมีเงื่อนไขว่า ถ้า bit 9 ถูก set เป็น "0", bit 13 จะถูก set เป็น "0".

Bits 6-8 จะไม่เป็น "111" ซึ่งเป็นค่าที่แสดงถึงการเป็น extended PTYPE (PLUSPTYPE) นอกเสียจากว่าจะมีการระบุให้ใช้งานจากภายนอก (ดูตัวอย่างใน Recommendation H.245) เพื่ออนุญาตให้มีการใช้ custom source format หรือ optional modes อื่น ๆ ได้กรณีเดียวคือในกรณีที่ถูกใช้โดย extended PTYPE. เมื่อไรก็ตามที่บิต 6-8 ไม่มีค่า "111", additional modes ทั้งหมดจะมีได้จากการใช้ของ extended PTYPE เท่านั้น และ modes เหล่านี้จะถูกพิจารณาว่าเป็น "off" state และจะถูกแสดงว่า "off" ไปจนจะมีการ switched on ใน bitstream ในภายหลัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Plus PTYPE (PLUSPTYPE) (Variable Length)

Codeword ของ 12 หรือ 30 bits จะถูกแสดงเฉพาะในกรณีที่การมีอยู่ของ extended PTYPE ถูกแสดงอยู่ใน bits 6-8 ของ PTYPE.

PLUSPTYPE ถูกประกอบขึ้นจาก 3 subfields: UFEP, OPPTYPE, และ MPPTYPE OPPTYPE จะมีอยู่เฉพาะในกรณีที่ UFEP มีค่า particular ขึ้นมา

1). Update Full Extended PTYPE (UFEP) (3 bits)

A fixed length codeword ของ 3 bits จะถูกแสดงเฉพาะในกรณีที่ extended PTYPE ถูกแสดงอยู่ใน PTYPE bits 6-8. เมื่อ set ไว้ที่ "000", แสดงว่าเฉพาะ extended PTYPE fields ที่ต้องการการ signal ในทุก ๆ picture header (MPPTYPE) และจะถูกรวมไว้ใน current picture header ด้วย. เมื่อ set ไว้ที่ "001", แสดงว่าทุก ๆ extended PTYPE fields ถูกรวมอยู่ใน current picture header. ถ้า picture type เป็นแบบ INTRA หรือ EI, field นี้จะถูก set เป็น "001"

ถ้า PLUSPTYPE ถูกแสดงในแต่ละลำดับที่ต่อเนื่องของ pictures, field นี้จะถูก set เป็น "001" อย่างน้อยบ่อยเท่ากับการถูกระบุถึงโดย a five-second หรือ five-picture timeout period ไม่ว่าจะมีช่วงของเวลามากกว่าก็ตาม. Timeout period ต้องการ UFEP = "001" เพื่อปรากฏอยู่ใน PLUSPTYPE field (ถ้า PLUSPTYPE มีอยู่ในทุก ๆ intervening picture) ของ picture header แรกด้วย temporal reference ที่แสดงช่วงเวลาที่มากกว่าหรือเท่ากับ 5 วินาที ตั้งแต่การเกิด UFEP = "001" ครั้งสุดท้าย, หรือของ picture ที่ 5 หลังจากการเกิด UFEP = "001" ครั้งสุดท้าย (requirement อนุญาตให้ period of time ยืดออกไปได้เท่ากับที่ถูกวัดโดย temporal reference).

Encoders ควรจะ set UFEP เท่ากับ "001" ให้บ่อยขึ้นใน error-prone environments. ส่วนค่าของ UFEP อื่น ๆ นอกเหนือจาก "000" และ "001" มีการถูกใช้งานแล้ว

2). The Optional Part of PLUSPTYPE (OPPTYPE) (18 bits)

ถ้า UFEP = "1" แล้ว bits ต่อไปนี้จะ present in PLUSPTYPE

- Bits 1-3 Source Format, "000" reserved, "001" sub-QCIF, "010" QCIF, "011" CIF, "100" 4CIF, "101" 16CIF, "110" custom source format, "111" reserved;
- Bits 4 Optional Custom PCF, "0" CIF PCF, "1" custom PCF;
- Bits 5 Optional Unrestricted Motion Vector (UMV) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 6 Optional Syntax-based Arithmetic Coding (SAC) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 7 Optional Advanced Prediction (AP) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 8 Optional Advanced INTRA Coding (AIC) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 9 Optional Deblocking Filter (DF) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Bits 10 Optional Slice Structured (SS) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 11 Optional Reference Picture Selection (RPS) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 12 Optional Independent Segment Decoding (ISD) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 13 Optional Alternative INTER VLC (AIV) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 14 Optional Modified Quantization (MQ) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 15 มีค่าเท่ากับ "1" เพื่อป้องกันการซ้ำกับ start code;
- Bits 16 Reserved, มีค่าเท่ากับ "0";
- Bits 17 Reserved, มีค่าเท่ากับ "0";
- Bits 18 Reserved, มีค่าเท่ากับ "0".

3). The mandatory part of PLUSPTYPE when PLUSPTYPE present (MPPTYPE) (9 bits)

Bits 9 bits ต่อไปนี้ present in PLUSPTYPE โดยไม่คำนึงถึงค่าของ UFEP

- Bits 1-3 Picture Type Code:
  - "000" I-picture (INTRA),
  - "001" P-picture (INTER),
  - "010" Improved PB-frame
  - "011" B-picture
  - "100" EI-picture
  - "101" EP-picture,
  - "110" Reserved,
  - "111" Reserved;
- Bits 4 Optional Reference Picture Resampling (RPR) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 5 Optional Reduced-Resolution Update (RRU) mode, เป็น "0" หมายถึง off, เป็น "1" หมายถึง on;
- Bits 6 Rounding Type (RTYPE) (ดูหัวข้อ 6.1.2);
- Bits 7 Reserved, มีค่าเท่ากับ "0";
- Bits 8 Reserved, มีค่าเท่ากับ "0";
- Bits 9 มีค่าเท่ากับ "1" เพื่อป้องกันการซ้ำกับ start code;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Encoder ควรจะควบคุม the rounding type เพื่อที่ว่า P-pictures, Improved PB-frames, และ EP-pictures จะมีค่า bit 6 ที่แตกต่างกัน (Rounding Type สำหรับ P-pictures) จาก reference pictures ของพวกมันเอง เพื่อการทำ motion compensation Bit 6 สามารถมี arbitrary value ได้ ถ้ามี reference picture เป็น I- หรือ EI-picture Bit 6 สามารถถูก set ค่าเป็น "1" ได้เฉพาะกรณีที่ bits 1-3 บอกว่าเป็น P-picture, Improved PB-frame, หรือ EP-picture. สำหรับ pictures type อื่น ๆ, Bit 6 นี้จะถูกกำหนดให้มีค่าเท่ากับ "0" เสมอ.

4). the semantics of PLUSPTYPE

สิ่งที่จำเป็นของ PLUSPTYPE คือ features ซึ่งมีแนวโน้มที่จะถูกเปลี่ยนตาม picture-by-picture basis. โดยปกติบิตเหล่านี้เป็นบิตที่บอกให้รู้ว่าเป็น picture type ไหนระหว่าง I, P, Improved PB, B, EI, และ EP (NOTE – PB-frames mode ไม่สามารถใช้กับ PLUSPTYPE ได้, จะใช้ Improved PB-frames mode แทน.) อย่างไรก็ตาม, มีการระบุถึงการใช้ RPR และ RRU modes เนื่องจากอาจจะมีการเปลี่ยนจาก picture to picture ได้

เมื่อมีการเปลี่ยนรูปภาพไป features ที่มีแนวโน้มที่จะยังคงมีการใช้งานอยู่มากกว่าที่จะถูกเปลี่ยนไป (ยกเว้นกรณีที่เห็นได้อย่างชัดเจน) จะถูกรวมอยู่ใน optional part ของ PLUSPTYPE ด้วย เมื่อ UFEP เป็น 000, the missing mode information จะถูกแสดงให้เห็นจาก type ของ picture และจาก mode information ที่ถูกส่งไปใน PLUSPTYPE อันก่อน ซึ่งมี UFEP เท่ากับ 001

ถ้ามี PLUSPTYPE แต่ UFEP เป็น 000 แล้ว

- สำหรับ P-picture หรือ Improved PB-frame, ค่า pixel aspect ratio, picture width, และ picture height จะไม่เปลี่ยนไปจาก reference picture.
- สำหรับ Temporal-scalability B-picture ใน enhancement layer, the Reference Layer Number (RLNUM) จะเหมือนกับ the Enhancement Layer Number (ELNUM) ถ้า picture สุดท้ายที่ถูกส่งไปใน enhancement layer เป็น EI- หรือ EP-picture ถ้ารูปภาพสุดท้ายที่ถูกส่งไปใน enhancement layer เป็น B-picture, reference layer number จะเหมือนกับของ B-picture อันล่าสุด ค่า pixel aspect ratio, picture width และ picture height จะคงเดิมไม่เปลี่ยนไปจากใน temporally reference layer picture ที่ตามมา
- สำหรับ SNR/Spatial scalability EP-picture ค่า pixel aspect ratio, picture width และ picture height จะคงเดิมไม่เปลี่ยนไปจากใน temporally reference picture อันก่อนที่อยู่ใน enhancement layer เดียวกัน

5). Mode restrictions for certain picture types and mode inference rules

Certain modes ไม่สามารถใช้กับ certain types ของ pictures ได้, โดยมีกฎดังนี้

5.1) Modes ต่อไปนี้ไม่สามารถใช้ภายใน I (INTRA) pictures ได้

- Unrestricted Motion Vector
- Advanced Prediction
- Alternative INTER VLC,
- Reference Picture Resampling,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Reduced-Resolution Update

5.2) Modes ต่อไปนี้ไม่สามารถใช้ภายใน B-pictures ได้

- Syntax-based Arithmetic Coding
- Deblocking Filter
- Advanced Prediction

5.3) Modes ต่อไปนี้ไม่สามารถใช้ภายใน EI-pictures ได้

- Unrestricted Motion Vector,
- Syntax-based Arithmetic Coding,
- Advanced Prediction ,
- Reference Picture Resampling,
- Reduced-Resolution Update
- Alternative INTER VLC

5.4) Modes ต่อไปนี้ไม่สามารถใช้ภายใน EP-pictures ได้

- Syntax-based Arithmetic Coding
- Advanced Prediction

Modes ที่ถูกระบุไว้ 4 ข้อข้างต้น อาจจะมี mode flag ที่มีค่า "1" ใน optional part ของ PLUSPTYPE ภายใน picture ของ type ที่ถูกห้ามสำหรับ mode นั้น ๆ (types I, B, EI, EP) เงื่อนไขนี้จะถูกอนุญาตและถูกทำให้เข้ากับ mode inference rules ซึ่งมีรายละเอียดอยู่ใน paragraph ต่อไป

Mode states จะเป็นไปตาม mode inference rules ต่อไปนี้

1) เมื่อ mode flag ถูก set เป็น "1" ใน optional part ของ PLUSPTYPE, current picture และแต่ละ picture ที่ตามมาใน bitstream จะถูกกำหนดให้มี state เป็น "on" สำหรับ mode นั้น ๆ
2) State "off" จะถูกกำหนดไปยัง mode ใด ๆ ก็ตามที่ไม่มีการใช้อยู่ภายใน picture ที่มี current picture type code. อย่างไรก็ตาม แต่ละรูปที่ตามมาใน Bitstream จะมี state "on" สำหรับ mode นั้น ๆ (นอกเสียจากว่าเงื่อนไขนี้ทำให้ผลลัพธ์จำนวนมาก เกิด conflict ที่ชัดเจน – ซึ่งจะถูกแก้ไขได้โดยวิธีเดียวกัน) ในกรณีของ layered scalable bitstreams mode state จะถูกแสดงจากภายใน layer เดียวกันของ bitstream เท่านั้น
3) The inference ของ state จะคงดำเนินการต่อเนื่องจนกระทั่ง picture ที่อยู่ใน layer เดียวกันถูกส่งผ่านไปโดยไม่คำนึงถึงว่ามี optional part ของ PLUSPTYPE หรือไม่ ถ้ารูปใหม่ที่มี optional part ของ PLUSPTYPE ถูกส่งผ่านไป, state ที่ถูกส่งไปพร้อมกับ new message จะแทนที่ state เดิมไป. แต่ถ้ารูปที่ถูกส่งผ่านไปไม่มี PLUSPTYPE (รูปที่ bits 6-8 ของ PTYPE ไม่ใช่ "111"), state "off" จะถูก assign ไปยังทุก ๆ mode ที่ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในสถานะ "on" ใน PTYPE field, และทุก ๆ mode จะยังคงมี state "off" จนกระทั่ง picture ใหม่ที่มี optional part ของ PLUSPTYPE ถูกส่งผ่านไป

4) Mode 2 modes คือ the Reference Picture Resampling mode และ the Reduced-Resolution Update mode จะไม่ต้องการ state inference เนื่องจาก mode flags สำหรับ modes เหล่านี้มีปรากฏอยู่ในส่วนสำคัญของ PLUSPTYPE. ซึ่ง mode flag ของแต่ละ mode นี้ จะไม่ถูก set นอกเสียจากว่า current picture มีการอนุญาตให้ใช้ mode นั้น. ตัวอย่างเช่น the Reduced-Resolution Update mode bit จะไม่ถูก set ใน INTRA picture.

5). Mode interaction restrictions certain modes ไม่สามารถใช้กับ certain modes อื่น ๆ ได้ ดังตัวอย่างต่อไปนี้

5.1) The Syntax-based Arithmetic Coding mode จะไม่ถูกใช้กับ Alternative INTER VLC mode หรือ the Modified Quantization mode

5.2) ถ้ามี PLUSPTYPE, the Unrestricted Motion Vector mode จะไม่ถูกใช้กับ the Syntax-based Arithmetic Coding mode

5.3) The Independent Segment Decoding mode จะไม่ถูกใช้กับ the Reference Picture Resampling mode

5.4) The Independent Segment decoding mode จะไม่ถูกใช้กับ the Slice Structured mode โดยที่ไม่มีการใช้ the Rectangular Slice submode ของ the Slice Structured mode ในเวลาเดียวกัน.

5.1.1.1 The picture header location of CPM (1 bit) (2 bits)

ตำแหน่งของ CPM และ PSBI field ใน picture header ขึ้นอยู่กับว่ามี PLUSPTYPE อยู่หรือไม่ ถ้ามี PLUSPTYPE, CPM จะอยู่ถัดจาก PLUSPTYPE ใน picture header ถ้าไม่มี PLUSPTYPE, CPM จะอยู่ถัดจาก PQUANT ใน picture header และ PSBI จะอยู่ถัดจาก CPM เสมอ (ถ้า CPM = "1")

5. Custom Picture Format (CPFMT) (23 bits)

A fixed length codeword ของ 23 bits จะมีอยู่เฉพาะในกรณีที่การใช้ custom picture format ถูก signaled ใน PLUSPTYPE และ UFEP เป็น "001". CPFMT จะประกอบด้วย

- Bits 1-4 Pixel Aspect Ratio Code รูปแบบ 4-bit ดูที่ค่า PAR ใน TABLE 5. สำหรับ extended PAR, ค่า pixel aspect ratio จริง ๆ จะถูกระบุใน EPAR
- Bits 5-13 Picture Width Indication: Range [0, ..., 511]; จะมีจำนวนของ pixel per line = (PWI + 1) * 4
- Bit 14 มีค่าเท่ากับ "1" เพื่อป้องกันการซ้ำกับ start code
- Bits 15-23 Picture Height Indication: Range [1, , 288]; จำนวนของ lines = PHI * 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAR code	Pixel aspect ratios
0000	Forbidden
0001	1:1 (Square)
0010	12:11 (CIF for 4:3 picture)
0011	10:11 (525-type for 4:3 picture)
0100	16:11 (CIF stretched for 16:9 picture)
0101	40:33 (525-type stretched for 16:9 picture)
0110-1110	Reserved
1111	Extended PAR

ตารางที่ 2-5 ความหมายของ PAR

6. Extended Pixel Aspect Ratio (EPAR) (16 bits)

A fixed length codeword ของ 16 bits จะมีอยู่เฉพาะในกรณีที่มี CPFMT และมีการระบุถึง extended PAR. EPAR จะประกอบด้วย

- Bits 1-8 PAR Width: จะมีค่าเป็น "0" ไม่ได้. เป็น natural binary representation ของ PAR width;
- Bits 9-16 PAR Height: จะมีค่าเป็น "0" ไม่ได้. เป็น natural binary representation ของ PAR height;

The PAR Width และ PAR Height จะมีความสัมพันธ์กันอย่างมาก

7. Custom Picture Clock Frequency Code (CPCFC) (8 bits)

A fixed length codeword ของ 16 bits จะมีอยู่เฉพาะในกรณีที่มี PLUSPTYPE และ UFEP เป็น 001 และ custom picture clock frequency ถูก signaled ใน PLUSPTYPE. CPCFC จะประกอบด้วย

- Bit 1 Clock Conversion Code: มีค่าเป็น "0" แสดง clock conversion factor 1000, มีค่าเป็น "1" แสดง 1001;
- Bits 2-8 Clock Divisor: จะมีค่าเป็น "0" ไม่ได้. เป็น natural binary representation ของค่าของ clock divisor.

ค่า custom picture clock frequency หาได้โดย 1800000/(clock divisor * clock conversation factor) Hz.

The temporal reference counters จะนับ unit ของการ inverse ของ picture clock frequency, นับเป็นวินาที เมื่อ PCF เปลี่ยนจากที่เคยมีระบุไว้ใน picture ก่อนหน้านี้, temporal reference สำหรับ current

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

picture จะถูกวัดในเทอมของ PCF, ดังนั้น new PCF จะมีผลสำหรับ temporal reference ของ future pictures เท่านั้น

8. Extended Temporal Reference (ETR) (2 bits)

A fixed length codeword 2 bits จะมีอยู่เฉพาะในกรณีที่มีการใช้ custom picture clock frequency (โดยไม่คำนึงถึงค่าของ UFEP) ซึ่งก็คือ 2 MSBs ของตัวเลข 10 บิต

9. Unlimited Unrestricted Motion Vectors Indicator (UUI) (Variable length)

A variable length codeword 1 หรือ 2 bits จะมีอยู่เฉพาะกรณีที่ Unrestricted Motion Vector mode ถูกแสดงใน PLUSPTYPE และ UFEP เป็น 001 เมื่อมี UUI จะแสดงว่ามีการใช้ limitation ที่ effective ของ range ของ motion vectors

- UUI = "1" หมายถึง the motion vector range
- UUI = "01" หมายถึง the motion vector range ไม่ถูกจำกัด ยกเว้นโดย picture size

10. Enhancement Layer Number (ELNUM) (4 bits)

A fixed length codeword ของ 4bits จะมีอยู่เฉพาะกรณีที่มีการใช้ Temporal, SNR, และ Spatial Scalability mode (โดยไม่คำนึงถึงค่าของ UFEP) the particular enhancement layer ถูกระบุโดย an enhancement layer number, ELNUM Picture ที่มีความเหมือนกันระหว่าง layers ถูกทำได้โดยผ่านทาง temporal reference Picture size จะอยู่แสดงอยู่ภายในแต่ละ enhancement layer ที่มีการใช้ source format fields ที่มีอยู่แล้ว หรือถูกแสดงโดย the relationship ไปยัง the reference layer. Enhancement layer ชั้นแรกที่อยู่เหนือ base layer ถูกตั้งชื่อว่า Enhancement Layer Number 2, และ base layer เป็นหมายเลข 1

11. Reference Layer Number (RLNUM) (4 bits)

A fixed length codeword of 4 bits จะมีอยู่เฉพาะกรณีที่มีการใช้ Temporal, SNR, และ Spatial Scalability mode และ UFEP เป็น 001. The layer number สำหรับ picture ที่ถูกใช้เป็น reference anchors ถูกแสดงโดย Reference Layer Number (RLNUM). เวลาที่ตรงกันระหว่าง layer ถูกทำได้โดยผ่านทาง temporal reference

Note – สำหรับ B-pictures ใน enhancement layer ซึ่งมี temporally surrounding EI- or EP-pictures ที่ถูกแสดงใน enhancement layer เดียวกัน, RLNUM จะเท่ากับ ELNUM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. Reference Picture Selection Mode Flags (RPSMF) (3 bits)

A fixed length codeword ของ 3 bits จะมีอยู่ในกรณีที่มีการใช้ the Reference Picture Selection mode และ UFEP เป็น 001. RPSMF จะแสดงว่า type ใดของ back-channel messages ที่ encoder ต้องการ ถ้ามีการใช้ the Reference Picture mode แต่ไม่มี RPSMF, ค่าสุดท้ายของ RPSMF ที่ถูกส่งไปจะยังคงมีผลอยู่

- 100 : ไม่ต้องการทั้ง ACK และ NACK;
- 101: ต้องการ ACK signals เพื่อถูกส่งคืนมา;
- 110: ต้องการ NACK signals เพื่อถูกส่งคืนมา;
- 111: ต้องการทั้ง ACK และ NACK signals เพื่อถูกส่งคืนมา;
- 000-011: Reserved

13. Temporal Reference for Prediction Indication (TRPI) (1 bit)

A fixed length codeword ของ 1 bit จะมีอยู่ในกรณีที่มีการใช้ Reference Picture Selection mode (โดยไม่คำนึงถึงค่าของ UFEP) TRPI จะแสดงการมีอยู่ของ TRP field ดังนี้

- 0 : ไม่มี TRP field;
- 1 : มี TRP field;

TRPI จะเป็น 0 เมื่อไรก็ตามที่ picture header แสดง I- หรือ EI-picture

14. Temporal Reference for Prediction (TRP) (10 bits)

TRP จะแสดง Temporal Reference ซึ่งถูกใช้สำหรับ prediction ของ the encoding, ยกเว้นในกรณีของ B-pictures สำหรับ B-pictures, picture ที่มี temporal reference TRP จะถูกใช้สำหรับ prediction ใน forward direction. (การ Prediction ใน reverse temporal direction จะใช้กับ temporally picture รูปถัดมาทันที.) TRP เป็น 10-bit number ถ้า custom picture clock frequency ไม่ได้ถูกใช้สำหรับ reference picture, 2 MSBs ของ TRP จะมีค่าเป็น 0 และจะมี 8-bit TR อยู่ภายใน LSBs ซึ่งพบได้ใน picture header ของ reference picture ถ้ามีการใช้ custom picture clock frequency สำหรับ reference picture, TRP เป็น 10-bit number ที่ประกอบด้วย concatenation ของ ETR และ TR จาก reference picture header

เมื่อไม่มี TRP, temporally previous anchor picture อันล่าสุดจะถูกใช้สำหรับการ prediction, และเมื่อไม่ได้อยู่ใน Reference Picture Slection mode TRP จะใช้ได้จนกระทั่งมี PSC, GSC, หรือ SSC อันต่อไปเข้ามา

15. Back-Channel message Indication (BCI) (Variable length)

A variable length field ของ 1 หรือ 2 bits จะมีอยู่ในกรณีที่มีการใช้ Reference Picture Selection mode เมื่อ set เป็น "1" จะบอกถึงการมีอยู่ของ video Back-Channel Message (BCM) field อันถัดมา ค่า "01" แสดงถึงการหายไปหรือจุดสิ้นสุดของ video back-channel message field อาจจะไม่มีการรวมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BCM และ BCI , และอาจจะถูกทำซ้ำเมื่อมี BCI จะถูกกำหนดเป็น "01" ถ้าไม่มีการใช้ videomux submode ของ Reference Picture Selection mode

16. Back-Channel Message (BCM) (Variable length)

The Back-Channel message ที่มี Syntax ตามที่ระบุไว้ใน N.4.2, จะมีอยู่ในกรณีที่มี BCI field อันก่อนหน้า และถูก set ค่าเป็น "1"

17. Reference Picture Resampling Parameters (RPRP) (Variable length)

A variable length field จะมีอยู่ในกรณีที่มีการ set Reference Picture Resampling mode ใน PLUSPTYPE. ใน field นี้จะมี parameters ของ Reference Picture Resampling mode Note – the Reference Picture Resampling mode สามารถถูก invoked ได้อย่างชัดเจนโดยการเกิด picture header สำหรับ INTER coded picture ที่มี picture size แตกต่างไปจากใน picture ที่ถูก encoded อันก่อนหน้านี้, ในเงื่อนไขที่ไม่มี RPRP field และไม่มีการ set Reference Picture Resampling mode

18. Quantizer Information (PQUANT) (5 bits)

A fixed length codeword 5 bits ที่แสดงถึง การปรับค่าความละเอียดของข้อมูล QUANT จะถูกใช้สำหรับ picture จนกระทั่งมีการ update โดย GQUANT หรือ DQUANT ใด ๆ ที่ตามมา. ค่าของ QUANT ที่จะมี step sizes ครึ่งหนึ่ง, range จาก 1 ถึง 31

19. Continuous Presence Multipoint and Video Multiplex (CPM) (1 bit)

A codeword ของ 1 bit จะ signal การใช้ Continuous Presence Multipoint และ Video Multiplex mode (CPM); โดยค่า "0" หมายถึง off, ค่า 1 หมายถึง on. ดูการใช้ CPM ได้ที่ ถ้ามี PLUSPTYPE, CPM จะอยู่ถัดจาก PLUSPTYPE แต่ถ้าไม่มี PLUSPTYPE, CPM จะอยู่ถัดจาก PQUANT ใน picture header

20. Picture Sub-Bitstream Indicator (PSBI) (2 bits)

A fixed length codeword ของ 2 bits จะมีอยู่ในกรณีที่ Continuous Presence Multipoint และ Video Multiplex mode ถูกระบุโดย CPM The codewords เป็น natural binary representation ของ sub-bitstream number สำหรับ picture header และ information ที่ตามมา จนกระทั่ง Picture หรือ GOB start code อันต่อไปเข้ามา. ถ้า CPM เป็น "1", PSBI จะอยู่ถัดจาก CPM (ตำแหน่งของ CPM และ PSBI ใน picture header ขึ้นอยู่กับว่ามี PLUSPTYPE หรือไม่)

21. Temporal Reference for B-pictures in PB-frames (TRb) (3/5 bits)

TRb จะมีอยู่ถ้า PTYPE หรือ PLUSPTYPE ระบุถึง "PB-frame" หรือ "Improved PB-frame" และระบุถึงจำนวนของ non-transmitted หรือ non-reference pictures (ที่ custom picture clock frequency เท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

29.97 Hz หรือเท่ากับที่ระบุไว้ใน CPCFC) ตั้งแต่ P- หรือ I-picture หรือ P-part ของ PB- หรือ Improved PB-frame และก่อน B-picture part ของ PB- หรือ Improved PB-frame. Codeword เป็น natural binary representation ของจำนวนของ non-transmitted pictures บวก 1 สำหรับ standard CIF picture clock frequency จะมีลักษณะเป็น 3 bit long และจะถูกเพิ่มเป็น 5 bits เมื่อมีการใช้ custom picture clock frequency จำนวนสูงสุดของ Non-transmitted pictures เป็น 6 สำหรับ standard CIF picture clock frequency และจะเป็น 30 เมื่อมีการใช้ custom picture frequency

22. Quantization information for B-pictures in PB-frames (DBQUANT) (2 bits)

DBQUANT จะมีอยู่ถ้า PTYPE หรือ PLUSPTYPE ระบุถึง "PB frame" หรือ "Improved PB-frame" ในกระบวนการ decoding, แต่ละ macroblock จะได้รับ quantization parameter QUANT. สำหรับ PB-frame, QUANT ถูกใช้สำหรับ P-block, สำหรับ B-block จะมีการใช้ quantization parameter BQUANT ที่แตกต่างไป QUANT มี range ตั้งแต่ 1 ถึง 31 DBQUANT แสดงถึงความสัมพันธ์ระหว่าง QUANT และ BQUANT ตามที่ระบุไว้ใน ตารางที่ 6ใน table นี้ "/" มายถึงการแบ่งโดยการ truncation BQUANT มี range ตั้งแต่ 1 ถึง 31 ถ้าค่าของ BQUANT ที่ได้มาจาก ตารางที่ 6 มีค่ามากกว่า 31 จะถูกตัดให้เหลือเพียง 31.

DBQUANT	BQUANT
00	(5 × QUANT)/4
01	(6 × QUANT)/4
10	(7 × QUANT)/4
11	(8 × QUANT)/4

ตารางที่ 2-6 แสดงถึงความสัมพันธ์ระหว่าง QUANT และ BQuant

23. Extra Insertion Information (PEI) (1 bit)

เป็น bit ที่เมื่อ set เป็น "1" แล้วจะ signal ถึงการมีอยู่ของ optional data field ที่ตามมา,

24. Supplemental Enhancement Information (PSUPP) (0/8/16 ... bits)

ถ้า PEI ถูก set เป็น "1", แล้ว 9 bits ที่ตามมาประกอบด้วย 8 bits ของ data (PSUP) และอีก 1 bit เป็น PEI bit เพื่อแสดงการตามมาของ 9 bits ถัดไปและต่อ ๆ ไป ตัวเข้ารหัสจะใช้ PSUPP ตามที่ระบุ

ตัวถอดรหัสที่ไม่ support the extended capabilities จะถูกออกแบบให้ละทิ้ง PSUPP ถ้า PEI ถูกกำหนดเป็น 1 เหตุการณ์ดังกล่าวจะไป enable backward compatibility สำหรับ extended capabilities

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น bitstream ที่ใช้งาน extended capabilities จะสามารถถูกใช้โดยไม่มี alteration โดยตัวถอดรหัสที่ไม่ support capabilities เหล่านั้น

25. Stuffing (ESTUF) (Variable length)

A codeword ของ variable length บิต 0 น้อยกว่า 8 บิต ตัวเข้ารหัสอาจจะแทรก codeword นี้โดยตรงเข้าไปก่อน EOS codeword ตัวเข้ารหัสจะแทรก codeword นี้ตามจำเป็นเพื่อได้มาซึ่ง byte alignment ที่จำเป็นโดยตรงก่อน EOSBS codeword ถ้ามี ESTUF, bit สุดท้ายของ ESTUF จะเป็น bit สุดท้ายของ byte (least significant bit), ดังนั้น การเริ่มของ EOS หรือ EOSBS codeword จะเป็นแบบ byte aligned. Decoders จะถูกออกแบบเพื่อละทิ้ง ESTUF

26. End Of Sequence (EOS) (22 bits)

A codeword ของ 22 bits มีค่าเป็น 0000 0000 0000 0000 1 11111 โดยขึ้นอยู่กับตัวถอดรหัสว่าจะแทรก codeword นี้หรือไม่ EOS อาจจะเป็นแบบ byte aligned ซึ่งสามารถทำได้โดยการแทรก ESTUF เข้าไปอยู่ก่อน EOS code ดังนั้นบิตแรกของ EOS code จะเป็นบิตแรกของไบต์ (most significant bit) EOS จะไม่ถูกทำซ้ำนอกเสียจากว่ามี picture start code อย่างน้อยที่สุด 1 อัน ปรากฏอยู่ระหว่างแต่ละคู่ของ EOS codes

27. Stuffing (PSTUF) (Variable length)

A codeword ของ variable length ประกอบด้วย บิต 0น้อยกว่า 8บิต Encoder จะแทรก codeword นี้สำหรับการจัดเรียง byte ของ PSC อันต่อไป Bit สุดท้ายของ PSTUF จะเป็น bit สุดท้ายของ byte (least significant bit), ดังนั้น video bitstream รวมถึง PSTUF จะเป็นผลคูณของ 8 bits จาก bit แรกใน H.263 bitstream. Decoders จะถูกออกแบบเพื่อละทิ้ง PSTUF

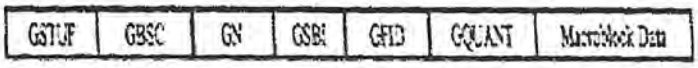
สำหรับบางเหตุผลที่ encoder ทำการหยุดการ encoding pictures เป็นระยะเวลา time-period ที่แน่นอน และจะทำการ encoding picture ต่อทีหลัง, PSTUF จะถูกส่งผ่านก่อนที่ encoder ทำการหยุด เพื่อป้องกันไม่ให้ bit สุดท้าย และถัดขึ้นมาถึง bit ที่ 7 ของ picture ก่อนหน้านี้ถูกส่งออกไป จนกระทั่ง the coders ทำการ coding ต่อ.

2.5.2. Group of Blocks Layer

Data สำหรับแต่ละ Group of Blocks (GOB) ประกอบด้วย GOB header ตามด้วย data สำหรับ macroblocks โครงสร้างถูกแสดงในรูปที่ 2-9 แต่ละ GOB จะมี rows of macroblocks (1 row หรือมากกว่า) อยู่ภายใน. สำหรับ GOB แรกในแต่ละ picture (with number 0), จะไม่มีการส่งของ GOB header ส่วนสำหรับ GOBs ที่เหลือทั้งหมด, GOB header อาจจะไม่มีข้อมูลอยู่เลย (empty) ขึ้นอยู่กับ encoder strategy Decoder จะ signal the remote encoder เพื่อทำการส่งเฉพาะ non-empty GOB headers โดยปัจจัยภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(external means), ดังเช่นตัวอย่างใน Recommendation H.245 GSTUF อาจจะมีอยู่เมื่อมี GBSC GN, GFID และ GQUANT จะมีอยู่เมื่อมี GBSC GSBI จะมีอยู่เมื่อ CPM มีค่าเป็น "1" ใน Picture header



รูปที่ 2-12 แสดงถึงส่วนประกอบของ GOB

- Stuffing (GSTUF) (Variable length)

A codeword ของ variable length ประกอบด้วยบิต 0 ที่น้อยกว่า 8 บิต ตัวเข้ารหัสอาจจะแทรก codeword นี้โดยตรง เข้าไปอยู่ก่อน GBSC codeword ถ้ามี GSTUF, bit สุดท้ายของ GSTUF จะเป็น bit สุดท้ายของ byte (least significant bit), ดังนั้น การเริ่มของ GBSC codeword จะเป็นแบบ byte aligned. Decoders จะถูกออกแบบให้ละทิ้ง GSTUF

- Group of Block Start Code (GBSC) (17 bits)

A word ของ 17 bits มีค่าเป็น 0000 0000 0000 0000 1 GOB start codes อาจจะเป็นแบบ byte aligned. ซึ่งสามารถถูกทำได้โดยการแทรก GSTUF เข้าไปอยู่ก่อน start code ดังนั้น bit แรกของ start code จะเป็น bit แรกของ byte (most significant bit)

- Group Number (GN) (5 bits)

A fixed length codeword 5 bits เป็นบิตฐานสองที่แสดงหมายเลขแทน Group of Blocks สำหรับ GOB ที่มี number 0, the GOB header รวมถึง GSTUF, GBSC, GN, GSBI, GFID และ GQUANT จะว่างเปล่า (empty); เนื่องจาก group number 0 ถูกใช้ใน PSC. Group numbers 1 ถึง 17 จะถูกใช้ใน GOB headers ของมาตรฐาน picture formats. Group numbers 1 ถึง 24 จะถูกใช้ใน GOB headers ของ custom picture formats. Group numbers 16 ถึง 28 จะถูก emulated ขึ้นมาใน the slice header เมื่อ CPM = "0", และ Group Numbers 25-27 และ 29 จะถูก emulated ขึ้นมาใน the slice header เมื่อ CPM = "1" Group number 31 ถูกใช้ใน EOS code, และ group number 30 ถูกใช้ใน EOSBS code

- GOB Sub-Bitstream Indicator (GSBI) (2 bits)

A fixed length codeword ของ 2 bits จะมีอยู่เฉพาะกรณีที่ CPM เป็น "1" ใน picture header The codewords มีลักษณะเป็น natural binary representation ของ the sub-bitstream number สำหรับ the GOB header และของ information ที่ตามมาทั้งหมด จนกระทั่งมี Picture start code หรือ GOB start code อันถัดไปเกิดขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- GOB Frame ID (GFID) (2 bits)

A fixed length codeword 2 bits GFID จะมีค่าเดิมในทุก ๆ GOB header (หรือ Slice header) ของ picture ที่กำหนดให้ ยิ่งไปกว่านั้น, ถ้า PTYPE ซึ่งถูกแสดงไว้ใน picture header เป็นอันเดียวกับของ picture ที่ถูกส่งผ่านไปอันก่อน, GFID จะมีค่าเดียวกับใน picture ก่อนหน้านั้นด้วยม ส่งผลให้ไม่มี PLUSPTYPE ในกรณีนี้ อย่างไรก็ตาม, ถ้า PTYPE ใน certain picture header แตกต่างไปจาก PTYPE ใน picture header ที่ถูกส่งผ่านไปอันก่อนหน้านี้, ค่าของ GFID ในรูปนั้นจะต่างจากค่าที่อยู่ใน picture อันก่อนหน้านี้

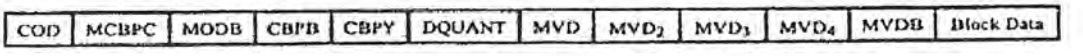
ถ้ามี PLUSPTYPE, ค่าของ GFID จะเท่ากับใน picture อันก่อน (ใน layer เดียวกัน) ถ้า the PTYPE, และ PLUSPTYPE, และทุก ๆ present fields ระหว่าง CPFMT, EPAR, CPCFC, SSS, ELNUM, RLNUM, UUI, RPSMF, และ RPRP จะยังคงมีผลดังเช่นใน picture อันก่อนหน้านี้; นอกเหนือจากเงื่อนไขดังกล่าว GFID จะแตกต่างจากใน picture อันก่อนหน้านี้

- Quantizer Information (GQUANT) (5 bits)

A fixed length codeword ของ 5 bits ที่แสดงถึงการใช้งาน การปรับค่าความละเอียดของข้อมูล QUANT สำหรับส่วนที่ยังคงเหลืออยู่ของ picture จนกระทั่งมีการ update โดย GQUANT หรือ DQUANT ใด ๆ ที่ตามหลังมา. The codewords มีลักษณะเป็น การแทนในหน่วยของเลขฐานสอง ของค่าของ QUANT ซึ่งจะมีค่าครึ่งหนึ่งของ step sizes, มีช่วงอยู่ในช่วง 1 ถึง 31

2.5.3. Macroblock layer

Data สำหรับแต่ละ macroblock ประกอบด้วย 1 macroblock header และตามด้วย data สำหรับ blocks. โครงสร้างถูกแสดงอยู่ในรูปที่ 2-10 COD จะมีอยู่เฉพาะใน pictures ที่ไม่ใช่แบบ "INTRA", สำหรับแต่ละ macroblock ใน pictures เหล่านี้. MCBPC จะมีอยู่เมื่อมีการระบุถึงโดย COD หรือเมื่อมีลักษณะของ picture เป็นแบบ "INTRA" MODB จะมีอยู่สำหรับ MB-type 0-4 ถ้า PTYPE ระบุถึง "PB-frame" CBPY, DQUANT, MVD และ MVD2-4 จะมีอยู่เมื่อมีการระบุถึงโดย MCBPC และ CBPY MVD2-4 จะมีอยู่เฉพาะในกรณีที่อยู่ใน Advanced Prediction mode (อ้างถึง Annex F) หรือ Deblocking Filter mode ,MODB, CBPB และ MVDB จะมีอยู่เฉพาะในกรณีที่อยู่ใน PB-frames mode



รูปที่ 2-13 แสดงถึงโครงสร้างของ macroblock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Coded macroblock indication (COD) (1 bit)

บิตที่เมื่อถูก set ให้มีค่าเป็น "0" หมายถึง macroblock ถูก coded. ถ้าถูก set ให้มีค่าเป็น "1" หมายถึง ไม่มีข้อมูลใด ๆ ถูกส่งผ่านสำหรับ macroblock นี้; ในกรณีนี้ decoder จะปฏิบัติกับ macroblock เสมือนเป็นแบบ INTER macroblock ที่มี motion vector สำหรับ block ทั้งหมดเท่ากับ 0 และไม่มีข้อมูลตัวเลขสัมประสิทธิ์. COD จะมีอยู่เฉพาะใน pictures ที่ไม่ได้เป็นแบบ "INTRA", สำหรับแต่ละ macroblock ใน pictures เหล่านี้

NOTE – ใน Advanced Prediction mode , overlapped block motion compensation จะถูกทำขึ้นมาถ้า COD เป็น "1"; และใน Deblocking Filter mode , deblocking filter สามารถมีผลกับค่าของ pixels บางอันของ macroblocks ที่ COD ถูก set เป็น "1".

- Macroblock type & Coded Block Pattern for Chrominance (MCBPC) (Variable length)

MCBPC เป็น variable length codeword ที่ให้ข้อมูลเกี่ยวกับ macroblock type และ the coded block pattern สำหรับ chrominance The codewords สำหรับ MCBPC ถูกกำหนดให้ในตาราง 7 และ 8. MCBPC จะถูกรวมอยู่ใน coded macroblocks เสมอ.

An extra codeword จะ available ในตารางสำหรับ bit stuffing. Codeword นี้ควรจะถูกละทิ้งโดย decoders ถ้า Improved PB-frame ถูกระบุโดย MPPTYPE bits 1-3 และ Custom Source Format ถูกระบุโดย OPPTYPE bits 1-3 แล้ว MBA จะไม่ stuffing ก่อน macroblock แรกของ picture (เพื่อป้องกันการซ้ำของ start code)

The macroblock type ให้ข้อมูลเกี่ยวกับ macroblock และ data elements Macroblock type และ elements ที่ถูกรวมถึงถูก listed ใน ตาราง 9 และ ตาราง10. Macroblock type 5 จะไม่ถูกแสดงนอกเสียจากว่า จะมี extended PTYPE (PLUSPTYPE) ใน picture header และมีการใช้ mode ต่อไปนี้ mode ใด mode หนึ่ง – Advanced Prediction mode หรือ Deblocking Filter mode และจะไม่ถูกแสดงสำหรับ macroblock แรกของ picture. Encoders จะไม่อนุญาตให้ MCBPC code สำหรับ macroblock type 5 ตาม seven consecutive zeros ใน bitstream ทันที (ซึ่งสามารถเกิดขึ้นโดย particular INTRADC codes ถูกตามด้วย COD=0), เพื่อป้องกันการซ้ำของ start code Codes สำหรับ macroblock type 5 สามารถถูกทำให้มาก่อนโดยการ stuffing เมื่อจำเป็นเพื่อรองรับความต้องการในส่วนนี้ (สำหรับ macroblocks อื่น ๆ ที่ไม่ใช่ macroblock แรกของ picture)

The coded block pattern สำหรับ chrominance บ่ง Cb และ/หรือ Cr blocks เมื่อ non-INTRADC transform coefficient อย่างน้อยที่สุด 1 อันถูกส่งผ่านไป (INTRADC เป็น dc-coefficient สำหรับ INTRA blocks), นอกเสียจากว่ามีการใช้ Advanced INTRA Coding mode. CBPCn = 1 ถ้ามี non-INTRADC coefficient ใด ๆ สำหรับ block N, หรือ = 0สำหรับ CBPC5 และ CBPC6 ใน coded block pattern. ถ้ามีการใช้ Advanced INTRA Coding, การใช้เหมือน ๆ กัน, แต่ INTRADC coefficient จะถูกแสดงในวิธีเดียวกันกับ other coefficient Block numbering ถูกกำหนดในรูปที่ 2-5 เมื่อ MCBPC = Stuffing, ส่วนที่ยังเหลืออยู่ของ macroblock layer จะถูกข้ามไป. ในกรณีนี้ COD = 0 ที่อยู่ก่อนหน้า จะไม่สัมพันธ์กับ coded หรือ not-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

coded macroblock ใด ๆ เลย ดังนั้น จะไม่มีการเพิ่มของ macroblock number. สำหรับ P-pictures, multiple stuffings ถูกทำได้โดย multiple sets ของ COD =0 และ MCBPC = Stuffing ดู ตารางที่ 2-7 และ 2-8

Index	MB type	CBPC (56)	Number of bits	Code
0	3	00	1	1
1	3	01	3	001
2	3	10	3	010
3	3	11	3	011
4	4	00	4	0001
5	4	01	6	0000 01
6	4	10	6	0000 10
7	4	11	6	0000 11
8	Stuffing	–	9	0000 0000 1

ตารางที่ 2-7 แสดงถึง ค่า variable length code สำหรับ MCBPC (สำหรับ I –picture)

Index	MB type	CBPC (56)	Number of bits	Code
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	Stuffing	–	9	0000 0000 1
21	5	00	11	0000 0000 010
22	5	01	13	0000 0000 0110 0
23	5	10	13	0000 0000 0111 0
24	5	11	13	0000 0000 0111 1

ตารางที่ 2-8 แสดงถึงค่า variable length code สำหรับ MCBPC (สำหรับ I –picture)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture type	MB type	Name	COD	MCBPC	CBPY	DQUANT	MVD	MVD_{2-4}
INTER	Not coded	–	X					
INTER	0	INTER	X	X	X		X	
INTER	1	INTER+Q	X	X	X	X	X	
INTER	2	INTER4V	X	X	X		X	X
INTER	3	INTRA	X	X	X			
INTER	4	INTRA+Q	X	X	X	X		
INTER	5	INTER4V+Q	X	X	X	X	X	X
INTER	Stuffing	–	X	X				
INTRA	3	INTRA		X	X			
INTRA	4	INTRA+Q		X	X	X		
INTRA	Stuffing	–		X				
NOTE – "X" means that the item is present in the macroblock.								

ตารางที่ 2-9 แสดงถึง macroblock data type และ องค์ประกอบของ normal picture

Picture type	MB type	Name	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD_{2-4}
INTER	Not coded	–	X								
INTER	0	INTER	X	X	X	X	(X)		X	(X)	
INTER	1	INTER+Q	X	X	X	X	(X)	X	X	(X)	
INTER	2	INTER4V	X	X	X	X	(X)		X	(X)	X
INTER	3	INTRA	X	X	X	X	(X)		X	(X)	
INTER	4	INTRA+Q	X	X	X	X	(X)	X	X	(X)	
INTER	5	INTER4V+Q	X	X	X	X	(X)	X	X	(X)	X
INTER	Stuffing	–	X	X							
NOTE 1 – "X" means that the item is present in the macroblock.											
NOTE 2 – CBPB and MVDB are only present if indicated by MODB.											
NOTE 3 – B-blocks are always coded in INTER mode, even if the macroblock type of the PB-macroblock indicates INTRA.											

ตารางที่ 2-10 แสดงถึง macroblock data type และ องค์ประกอบของ PB frame

- Macroblock mode for B-blocks (MODB) (Variable length)

MODB จะมีอยู่สำหรับ MB-type 0-4 ถ้า PTYPE ระบุถึง "PB-frame" และเป็น variable length codeword indicating ไม่ว่าจะมี CBPB (แสดงว่า B-coefficients ถูกส่งผ่านสำหรับ macroblock นี้) และ/หรือจะมี MVDB อยู่ Codeword สำหรับ MODB ถูกนิยามไว้ในตารางที่ 2-11 MODB ถูก coded ในลักษณะที่แตกต่างกันสำหรับ Improved PB-framesr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index	CBPB	MVDB	Number of bits	Code
0			1	0
1		X	2	10
2	X	X	2	11
NOTE – "X" means that the item is present in the macroblock.				

ตารางที่ 2-11 แสดงถึงค่า variable length code สำหรับ MODB

- Coded Block Pattern for B-blocks (CBPB) (6 bits)

CBPB จะมีอยู่ใน PB-frames mode เฉพาะในกรณีที่ถูกระบุถึงโดย MODB CBPBn = 1 ถ้ามี coefficient ใด ๆ สำหรับ B-block N, หรือ = 0 สำหรับแต่ละ bit CBPBn ใน coded block pattern. Block numbering ถูกกำหนดให้ในรูปที่ 2-5, บิตซ้ายสุดของ CBPB จะตรงกันกับ block number 1

- Coded Block Pattern for luminance (CBPY) (Variable length)

Variable length codeword ให้ a pattern number ที่บ่งว่า Y blocks ใน macroblock อย่างน้อยที่สุด 1 non-INTRADC transform coefficient ถูกส่งผ่าน (INTRADC เป็น dc-coefficient สำหรับ INTRA blocks, ดูหัวข้อ 4.4.1), นอกเสียจากว่ามีการใช้ Advanced INTRA Coding mode. ถ้ามีการใช้ Advanced INTRA Coding, INTRADC จะถูกแสดงในลักษณะที่เหมือนกันกับตัวเลขสัมประสิทธิ์อื่น ๆ

CBPYn = 1 ถ้ามี non-INTRADC coefficient ใด ๆ สำหรับ block N, หรือ = 0 สำหรับแต่ละ bit CBPYn ใน the coded block pattern. Block numbering ถูกกำหนดให้ในรูป ที่5, bit ซ้ายสุดของ CBPY จะตรงกันกับ block หมายเลข 1. สำหรับ pattern CBPYn ที่แน่นอน, codewords ที่แตกต่างกันจะถูกใช้สำหรับ INTER และ INTRA macroblocks ตามที่ถูกระบุไว้ในตาราง ที่ 2-13

- Quantizer Information (DQUANT) (2 bits/Variable Length)

ถ้าไม่มีการใช้ Modified Quantization mode, DQUANT จะเป็น สองbit code เพื่อกำหนดการเปลี่ยนแปลงใน DQUANT. ในตารางที่ 12 ค่าที่แตกต่างกันสำหรับ codewords ที่แตกต่างกันจะถูกกำหนดไว้ QUANT มี range ในช่วง 1 ถึง 31; ถ้าค่าสำหรับ QUANT หลังจากเพิ่ม differential value เข้าไปแล้วมีค่าน้อยกว่า 1 หรือมีค่ามากกว่า 31, มันจะถูกกำหนดค่าให้กลายเป็น 1 และ 31 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index	Differential value	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

ตารางที่ 2-12 ค่า DQUANT และค่าความแตกต่างของ QUANT

Index	CBPY(INTRA) (12, 34)	CBPY(INTER) (12, 34)	Number of bits	Code
0	00 00	11 11	4	0011
1	00 01	11 10	5	0010 1
2	00 10	11 01	5	0010 0
3	00 11	11 00	4	1001
4	01 00	10 11	5	0001 1
5	01 01	10 10	4	0111
6	01 10	10 01	6	0000 10
7	01 11	10 00	4	1011
8	10 00	01 11	5	0001 0
9	10 01	01 10	6	0000 11
10	10 10	01 01	4	0101
11	10 11	01 00	4	1010
12	11 00	00 11	4	0100
13	11 01	00 10	4	1000
14	11 10	00 01	4	0110
15	11 11	00 00	2	11

ตารางที่ 2-13 แสดงถึงค่า variable length code ของ CBPY

- Motion Vector Data (MVD) (Variable length)

MVD ถูกรวมอยู่สำหรับ INTER macroblocks ทั้งหมด (เช่นเดียวกันกับใน PB-frames mode สำหรับ INTRA macroblocks) และประกอบด้วย variable length codeword สำหรับ horizontal component และตามด้วย variable length codeword สำหรับ vertical component Variable length codes ถูกกำหนดไว้ในตาราง ที่ 2-14.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Motion Vector Data (MVD2-4) (Variable length)

The three codewords MVD2-4 ถูกรวมอยู่ด้วยถ้ามีการระบุถึงโดย PTYPE และ MCBPC, และประกอบด้วยแต่ละ variable length codeword สำหรับ horizontal component ที่ถูกตามโดย variable length codeword สำหรับ vertical component ของแต่ละ vector Variable length codes ถูกกำหนดให้ในตาราง ที่ 2-14 MVD2-4 จะมีอยู่เฉพาะในกรณีที่อยู่ใน Advanced Prediction mode หรือ Deblocking Filter mode

- Motion Vector Data for B-macroblock (MVDB) (Variable length)

MVDB จะมีอยู่ใน PB-frames หรือ Improved PB-frames mode เฉพาะในกรณีที่ถูกระบุถึงโดย MODB, และประกอบด้วย variable length codeword สำหรับ horizontal component ที่ถูกตามโดย variable length codeword สำหรับ vertical component ของแต่ละ vector Variable length codes ถูกกำหนดให้ในตารางที่ 2-14 สำหรับการใช้ของ MVDB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index	Vector	Differences	Bit number	Codes
0	-16	16	13	0000 0000 0010 1
1	-15.5	16.5	13	0000 0000 0011 1
2	-15	17	12	0000 0000 0101
3	-14.5	17.5	12	0000 0000 0111
4	-14	18	12	0000 0000 1001
5	-13.5	18.5	12	0000 0000 1011
6	-13	19	12	0000 0000 1101
7	-12.5	19.5	12	0000 0000 1111
8	-12	20	11	0000 0001 001
9	-11.5	20.5	11	0000 0001 011
10	-11	21	11	0000 0001 101
11	-10.5	21.5	11	0000 0001 111
12	-10	22	11	0000 0010 001
13	-9.5	22.5	11	0000 0010 011
14	-9	23	11	0000 0010 101
15	-8.5	23.5	11	0000 0010 111
16	-8	24	11	0000 0011 001
17	-7.5	24.5	11	0000 0011 011
18	-7	25	11	0000 0011 101
19	-6.5	25.5	11	0000 0011 111
20	-6	26	11	0000 0100 001
21	-5.5	26.5	11	0000 0100 011
22	-5	27	10	0000 0100 11
23	-4.5	27.5	10	0000 0101 01
24	-4	28	10	0000 0101 11
25	-3.5	28.5	8	0000 0111
26	-3	29	8	0000 1001
27	-2.5	29.5	8	0000 1011
28	-2	30	7	0000 111
29	-1.5	30.5	5	0001 1
30	-1	31	4	0011
31	-0.5	31.5	3	011
32	0		1	1
33	0.5	-31.5	3	010
34	1	-31	4	0010
35	1.5	-30.5	5	0001 0
36	2	-30	7	0000 110
37	2.5	-29.5	8	0000 1010
38	3	-29	8	0000 1000
39	3.5	-28.5	8	0000 0110
40	4	-28	10	0000 0101 10
41	4.5	-27.5	10	0000 0101 00
42	5	-27	10	0000 0100 10
43	5.5	-26.5	11	0000 0100 010
44	6	-26	11	0000 0100 000
45	6.5	-25.5	11	0000 0011 110

ตารางที่ 2-14 แสดงถึงค่า variable length code สำหรับ MVD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index	Vector	Differences	Bit number	Codes
46	7	-25	11	0000 0011 100
47	7.5	-24.5	11	0000 0011 010
48	8	-24	11	0000 0011 000
49	8.5	-23.5	11	0000 0010 110
50	9	-23	11	0000 0010 100
51	9.5	-22.5	11	0000 0010 010
52	10	-22	11	0000 0010 000
53	10.5	-21.5	11	0000 0001 110
54	11	-21	11	0000 0001 100
55	11.5	-20.5	11	0000 0001 010
56	12	-20	11	0000 0001 000
57	12.5	-19.5	12	0000 0000 1110
58	13	-19	12	0000 0000 1100
59	13.5	-18.5	12	0000 0000 1010
60	14	-18	12	0000 0000 1000
61	14.5	-17.5	12	0000 0000 0110
62	15	-17	12	0000 0000 0100
63	15.5	-16.5	13	0000 0000 0011 0

ตารางที่ 2-14 (ต่อ) แสดงถึงค่า variable length code สำหรับ MVD

2.5.5. Block Layer

ถ้าไม่ได้อยู่ใน PB-frames mode, macroblock ประกอบด้วย 4 luminance blocks และ สอง chrominance block (ดูรูปที่ 2-5) โครงสร้างของ block layer ถูกแสดงในรูปที่ 2-11 INTRADC จะมีอยู่สำหรับทุก ๆ block ของ macroblock ถ้า MCBPC แสดงถึง macroblock type 3 หรือ 4 (ดูตารางที่ 2-7 และ 2-8) TCOEF จะมีอยู่ถ้ามีการระบุถึงโดย MCBPC หรือ CBPY

ใน PB-frames mode, a macroblock ประกอบด้วย 12 blocks. เริ่มแรก ข้อมูลสำหรับ 6 P-blocks จะถูกส่งผ่านใน H.263 mode แบบ default, แล้วจึงดำเนินการดังกล่าวกับ B-blocks INTRADC จะมีอยู่สำหรับทุก ๆ P-block ของ macroblock ถ้า MCBPC แสดงถึง macroblock type 3 หรือ 4 (ดูตารางที่ 2-7 หรือ 2-8) INTRADC ไม่มีสำหรับ B-blocks TCOEF จะมีอยู่สำหรับ P-blocks ถ้าถูกระบุถึงโดย MCBPC หรือ CBPY TCOEF จะมีอยู่สำหรับ B-blocks ถ้ามีการระบุถึงโดย CBPB

INTRADC	TCOEF

รูปที่ 2-14 แสดงถึงโครงสร้างของ block layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DC coefficient for INTRA blocks (INTRADC) (8 bits)

A codeword 8 bits ไม่มีการใช้ code 0000 0000 และ code 1000 0000, the reconstruction level ของ 1024 จะถูกทำการ coded เป็น 1111 1111 (ดูตารางที่ 2-15)

- Transform Coefficient (TCOEF) (Variable length)

EVENTs ที่เกิดขึ้นแบบทั่ว ๆ ไปและเกิดบ่อย ๆ จะถูก coded ด้วย variable length codes ที่กำหนดให้ในตาราง 16 Bit "s" สุดท้าย หมายถึง sign of the level, เป็น "0" สำหรับ positive, เป็น "1" สำหรับ negative

An EVENT เป็นการรวมกันของ a last non-zero coefficient indication (LAST; "0" หมายถึง มี non-zero coefficients ใน block นี้จำนวนมาก, "1" หมายถึง เป็น last non-zero coefficient ใน block นี้แล้ว), จำนวนของ successive zeros นำหน้า the coded coefficient (RUN), กับ the non-zero value ของ coded coefficient (LEVEL)

การรวมกันที่ยังเหลืออยู่ของ LAST, RUN, LEVEL ถูก coded ด้วย 22-bit word ที่ประกอบด้วย 7 bits ESCAPE, 1 bit LAST, 6 bits RUN และ 8 bits LEVEL. ไม่มีข้อห้ามในการใช้ 22-bit word นี้สำหรับ encoding the combinations ที่ถูก listed ในตารางที่ 2-16 สำหรับ 8-bit word สำหรับ LEVEL, the code 0000 0000 ถูกห้าม, และ the code 1000 0000 ถูกห้าม the codes สำหรับ RUN และสำหรับ LEVEL ถูกกำหนดให้ในตารางที่ 2-17

Index	FLC	Reconstruction level into inverse transform
0	0000 0001 (1)	8
1	0000 0010 (2)	16
2	0000 0011 (3)	24
.	. .	.
.	. .	.
126	0111 1111 (127)	1016
127	1111 1111 (255)	1024
128	1000 0001 (129)	1032
.	. .	.
.	. .	.
252	1111 1101 (253)	2024
253	1111 1110 (254)	2032

ตารางที่ 2-15 แสดงถึงระดับการ reconstruction สำหรับ INTRA DC Coefficient

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
0	0	0	1	3	10s
1	0	0	2	5	1111s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110s
19	0	2	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0100 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s
31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100s
45	0	14	1	9	0001 1011s

ตารางที่ 2-16 แสดงถึง variable length code ของ TCOEF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	0000 1111 0s
50	0	19	1	10	0000 1110 1s
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s
57	0	26	1	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 1s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 0s
64	1	3	1	7	0010 11s
65	1	4	1	7	0010 10s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	14	1	9	0001 0101s
76	1	15	1	9	0001 0100s
77	1	16	1	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s
89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s

ตารางที่ 2-16 (ต่อ) แสดงถึง variable length code ของ TCOEF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDEX	LAST	RUN	|LEVEL|	BITS	VLC CODE
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ESCAPE			7	0000 011

ตารางที่ 2-16 (ต่อ) แสดงถึง variable length code ของ TCOEF

Index	Run	Code
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Index	Level	Code
–	–128	see text
0	–127	1000 0001
.	.	.
125	–2	1111 1110
126	–1	1111 1111
–	0	FORBIDDEN
127	1	0000 0001
128	2	0000 0010
.	.	.
253	127	0111 1111

ตารางที่ 2-17 แสดงถึงรูปแบบการทำนาย motion vectors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6. Decoding process

2.6.1. Motion compensation

ใน Subclause นี้, อธิบายถึง the motion compensation สำหรับ default H.263 prediction mode

Differential motion vectors

Macroblock vector ได้มาจากการบวก predictors ไปกับ the vector differences ที่ถูกระบุถึงโดย MVD (ดูตารางที่ 2-14). ในกรณีของ 1 vector ต่อ macroblock, the candidate predictors สำหรับ differential coding จะถูกนำมาจาก 3 surrounding macroblocks ตามที่ระบุในรูปที่ 2-12 โดย predictors จะถูกคำนวณให้แยกจากกันสำหรับ horizontal และ vertical components.

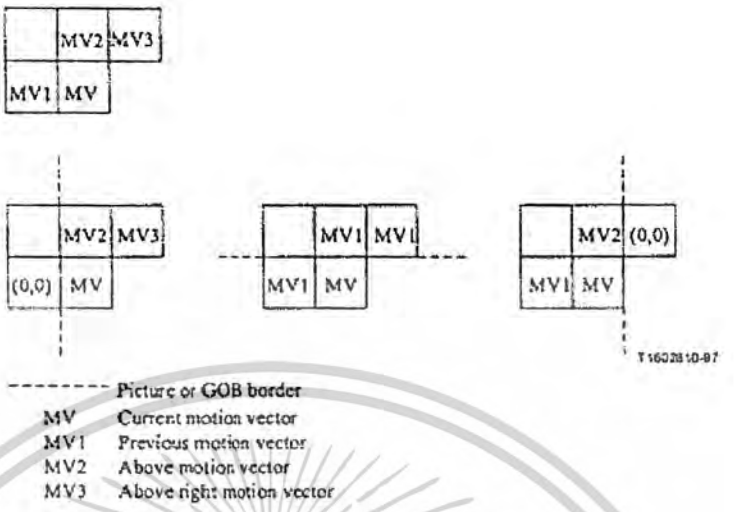
ในกรณีพิเศษ, ที่ ขอบเขต ของ current GOB, slice, หรือ picture, decision rules ต่อไปนี้จะมีการเพิ่ม order เข้าไป

1) เมื่อ Macroblock ถูก coded ใน INTRA mode (โดยไม่ได้อยู่ใน PB-frames mode ที่มี bidirectional prediction) หรือไม่ถูก coded (COD=1), the candidate predictor จะถูก set เป็น 0
2) The candidate predictor MV1 ถูก set เป็น 0 ถ้า corresponding macroblock อยู่ด้านนอก picture (ด้านซ้าย)
3) ต่อมา, candidate predictors MV2 และ MV3 จะถูก set เป็น MV1 ถ้า corresponding macroblocks อยู่ด้านนอก picture (ด้านบน) หรืออยู่ด้านนอก GOB (ด้านบน) ถ้า GOB header ของ GOB ปัจจุบันมีค่า (non-empty); หรืออยู่ด้านนอก slice เมื่ออยู่ใน Slice Structured mode
4) แล้ว Candidate predictor MV3 จะถูก set เป็น 0 ถ้า corresponding macroblock อยู่ด้านนอก picture (ด้านขวา)

สำหรับแต่ละ Component, the predictor เป็นค่ากลางของ 3 candidate predictors สำหรับ component นี้

ข้อดีที่เกิดขึ้นคือช่วงของ motion vector component values จะถูกจำกัดไว้ แต่ละ VLC word สำหรับ MVD จะแสดงเป็นคู่ของค่าที่แตกต่างกัน มีเพียงคู่เดียวที่จะมี macroblock vector component อยู่ในช่วงที่อนุญาต (ช่วง [-16, 15.5]) ค่าบวก ของ horizontal หรือ vertical component ของ motion vector บ่งว่าการ prediction ถูกกระทำขึ้นมาจาก pixels ใน picture อันก่อนหน้านี้ที่ซึ่งมีระยะว่างทางด้านขวาหรือทางด้านล่างของ pixels ที่ถูก predicted นั้น ๆ. ถ้ามีการใช้ the unrestricted motion vector mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-15 รูปแบบการ predict motion vector

The motion vector ถูกใช้สำหรับทุก ๆ pixels ใน 4 luminance blocks ทั้งหมดใน macroblock. Motion vectors สำหรับ chrominance blocks ทั้ง 2 จะถูก derived โดยการแบ่ง component values ของ macroblock vector ด้วย 2, ตาม the lower chrominance format. The component values ของ the resulting quarter pixel resolution vectors จะถูกเปลี่ยนแปลงไปทางตำแหน่งที่ใกล้ half pixel ที่สุด ซึ่งถูกแสดงอยู่ในตารางที่ 2-18

Quarter-pixel position	0	1/4	1/2	3/4	1
Resulting position	0	1/2	1/2	1/2	1

ตารางที่ 2-18 แสดงถึงการแก้ไข ขนาดของ chrominance vectors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2. Coefficients decoding

- Inverse quantization

กระบวนการ inverse quantization ถูกอธิบายไว้ใน subclause นี้, ยกเว้นสำหรับกรณีที่มีการใช้ optional Advanced INTRA Coding mode. ถ้า LEVEL = "0", the reconstruction level REC จะ = 0. The reconstruction level ของ INTRADC ถูกกำหนดโดยตาราง Table 15. The reconstruction levels ของ non-zero coefficients ทั้งหมดที่นอกเหนือจากของ INTRADC ถูกกำหนดโดยสูตรต่อไปนี้

$$|REC| = QUANT * (2 * |LEVEL| + 1) \quad \text{ถ้า QUANT = "odd"}$$

$$|REC| = QUANT * (2 * |LEVEL| + 1) - 1 \quad \text{ถ้า QUANT = "even"}$$

NOTE – กระบวนการนี้จะไม่ยอมให้จำนวนที่ถูก คำณวนออกมาได้เป็นเลขคู่ข้อจำกัดนี้ถูกตั้งขึ้นมาเพื่อป้องกันการสะสมของ IDCT mismatch errors.หลังการคำนวณของ |REC|, the sign จะถูกเพิ่มเข้าไป ค่า REC จะกลายเป็น REC: REC = sign (LEVEL) * |REC|

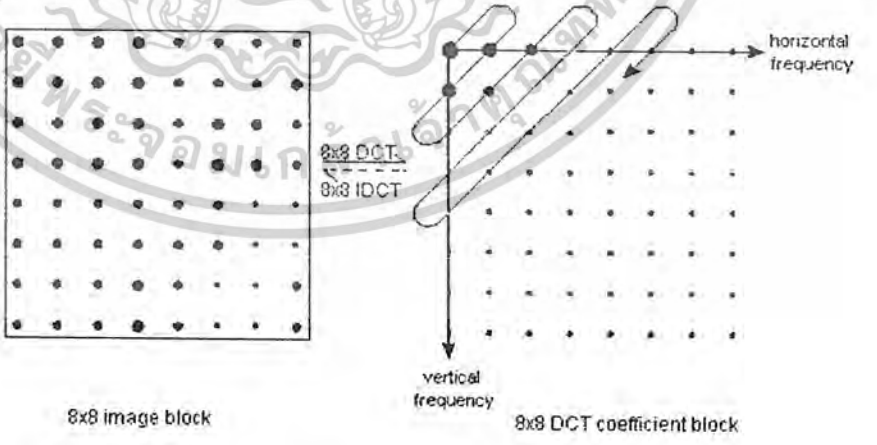
Sign (LEVEL) ถูกกำหนดโดย bit สุดท้ายของ TCOEF code (ดูตารางที่ 2-16) หรือถูกกำหนดโดยตารางที่ 2-17

- Clipping of reconstruction levels

หลังจาก ที่ทำ inverse quantization, ค่า reconstruction levels ของตัวเลขสัมประสิทธิ์ทั้งหมดนอกเหนือจากของ INTRADC จะถูก clipped อยู่ในช่วง -2048 ถึง 2047

- Zigzag positioning

ค่า Quantized coefficientจะ ถูกจัดวางในรูปแบบ 8 x 8 block ตามลำดับ (sequence) ที่กำหนดให้ในรูปที่ 2-14,. Coefficient 1 เป็น dc-coefficients



รูปที่ 2-16 แสดงถึงการจัดเรียงค่า quantized coefficient ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Inverse transform

หลังการ inverse quantization และการ zigzag ของ coefficients, ผลลัพธ์ 8 x 8 blocks จะถูกดำเนินการโดย inverse discrete cosine transform แบบ สองมิติของ size 8 by 8. The output จาก inverse transform ที่มี range ตั้งแต่ -256 ถึง +255 หลังจากผ่านการ clipping แล้วจะถูกแสดงเป็น 9 บิต the transfer function ของ the inverse transform จะถูกกำหนดโดยสมการ

$$C(u),C(v)=\begin{cases}\frac{1}{\sqrt{2}} & u,v=0\\ 1 & otherwise\end{cases} \tag{2.1}$$

$$f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{2.2}$$

Note – ภายใน block ที่จะถูก transformed, x = 0 หมายถึง pixel อยู่ใกล้มุมซ้าย (left edge) ของ picture ที่สุด และ y = 0 หมายถึง pixel อยู่ใกล้มุมบน (top edge) ของรูปมากที่สุด

2.6.3. Reconstruction of blocks

- Summation

หลังทำการ motion compensation และการ coefficients decoding (รวมถึงการ inverse transform), reconstruction จะถูกสร้างขึ้นมาสำหรับแต่ละ luminance และ chrominance block สำหรับ INTRA blocks, the reconstruction จะเท่ากับผลลัพธ์ของการ inverse transformation สำหรับ INTER blocks, the reconstruction จะถูกสร้างขึ้นโดยการ summing the prediction และผลลัพธ์ของการ inverse transformation การ summation ถูกกระทำบนพิกเซลพื้นฐาน

- Clipping

Clipping functions ถูกแทรกเข้ามาโดยมีจุดประสงค์เพื่อป้องกัน quantization distortion ของ transform coefficient amplitudes ซึ่งเป็นสาเหตุของการเกิด arithmetic overflow ใน encoder และ decoder loops. The clipper จะดำเนินการหลังจากการ summation ของ prediction และ reconstructed prediction error, สำหรับ pixel ลัพธ์ ที่มีค่าน้อยกว่า 0 จะถูกเปลี่ยนค่าเป็น 0 และสำหรับ pixel ลัพธ์ ที่มีค่ามากกว่า 255 จะถูกเปลี่ยนค่าเป็น 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

Interface Algorithm (IALG)

3.1. บทนำ

ความก้าวหน้าในเทคโนโลยี Digital Signal Processor ในการประยุกต์ใช้กับ ระบบโทรศัพท์, รูปภาพ, วิดีโอ และ ระบบเสียง มีผลการวิจัยและพัฒนาที่ก้าวหน้ามากยิ่งขึ้น ตัวอย่างเช่น อัลกอริทึมที่ใช้ในระบบโทรศัพท์ที่ได้พัฒนามาหลายปี การ Implement ของ DSP มีความแตกต่างมากๆจากระบบอื่นๆ เพราะว่าระบบมี วิธีในการจัดการหน่วยความจำ และระบบในการจัดการ อินพุต/เอาท์พุต ที่แตกต่างกันออกไป เนื่องจากการขาดแคลนวิธีในการรวมระบบที่สอดคล้องกัน หรือ มาตรฐานของโปรแกรม มันจึงไม่จำเป็นที่วิธีการของ DSP จะถูกใช้มากกว่า 1 วิธี หรือการใช้งานโดยปราศจากการปรับปรุงระบบ, การทดสอบ และการ Integrate

เป้าหมายของบทนี้คือ สนับสนุนให้ผู้อ่านสร้างอัลกอริทึมที่ใช้กับ TMS320 DSP Algorithm Standard ซึ่งเป็นส่วนหนึ่งของ TI's eXpressDSP ในการริเริ่มเทคนิคของมาตรฐานอัลกอริทึมขึ้นมา

eXpressDSP Algorithm Standard (XDAIS) ถูกแบ่งได้เป็น 3 ส่วนคือ Clients (Users) of algorithms, Producers of algorithms และ Creators of algorithm-specific interfaces การใช้งานนี้จะเน้นไปที่ Producers of algorithms และแสดงส่วนเชื่อมต่อ (Interface) ที่กำหนดขึ้น

เพื่อให้ Algorithm เป็นที่ยอมรับและเป็นมาตรฐาน อัลกอริทึมต้องกำหนด IALG Interface มาตรฐาน ขึ้นมา

IALG Interface คือส่วนเชื่อมต่อย่อย หรือ Service Provider Interface (SPI) และถูกกำหนดขึ้นใน header ไฟล์ ialg.h รวมไปถึงวิธีการเชื่อมต่อ IALG algorithm ต้องเป็นไปตามกฎข้อบังคับ ยกตัวอย่าง การประกาศตัวแปลต้องมีชื่อเป็นไปตามกฎการตั้งชื่อ, อัลกอริทึมต้องไม่เข้าถึงอุปกรณ์ที่นำข้อมูลเข้า/ออกใดๆโดยตรง

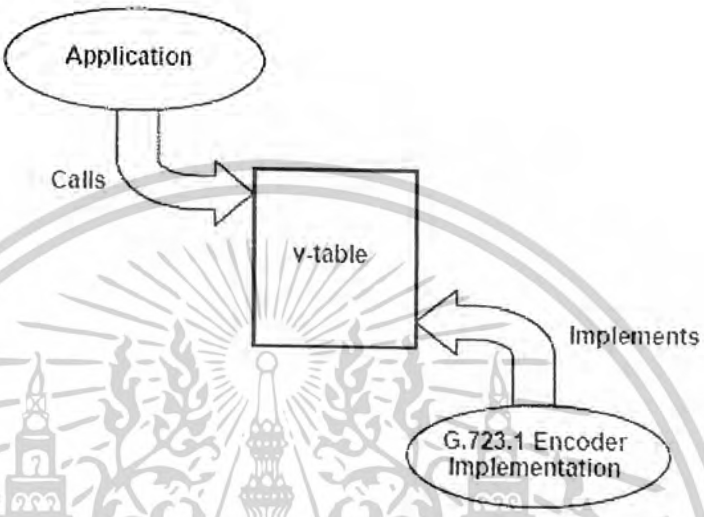
ผู้ใช้อัลกอริทึมสามารถจัดการอัลกอริทึมได้โดยใช้ Table of function pointers (V-table) ทุกๆ อัลกอริทึมต้องสร้างจาก V-table เพื่อจะได้เป็นมาตรฐาน V-table สามารถจัดการ algorithm instance ได้ ตัวอย่างเช่น สร้างและลบ instance object ของอัลกอริทึม เช่นเดียวกับการ run อัลกอริทึมดังรูปที่ 3-1

3.2. The IALG Interface

IALG interface คือ ส่วนเชื่อมต่อหลัก (Core interface) ที่กำหนดเป็นมาตรฐานที่อัลกอริทึมต้องการ มันถูกกำหนดรูปแบบและค่าคงที่ไว้ใน v-table, IALG_Fxns จะถูกอัลกอริทึมทุกๆอัลกอริทึมกำหนดขึ้นมา และตั้งค่า (initialize) ค่าตัวแปรของรูปแบบ IALG_Fxns เพื่อที่จะประกาศ v-table ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางอันใน v-table สามารถเลือกที่จะกำหนดหรือไม่กำหนดได้ ขณะที่ algAlloc(), algInit() และ algFree() ต้องถูกกำหนดไว้



รูปที่ 3-1 Application, implementation และ the V-table

3.2.1. algAlloc()

อัลกอริทึมต้องใช้ฟังก์ชัน algAlloc() เพื่อประกาศความต้องการหน่วยความจำ ยกตัวอย่างเช่น ต้องการหน่วยความจำถาวรจำนวน 1 block ขนาดของ block ต้องใหญ่พอที่จะเก็บ ข้อมูลของ algorithm instance ได้ เช่นเดียวกับการทำงานของบัฟเฟอร์ต่างๆที่ต้องการระหว่างการทำงาน แสดงได้ดังรูปที่ 3-2

```
Int G723ENC_TI_algAlloc (const IALG_Params *algParams, IALG_Fxns **pf,
IALG_MemRec memTab[])
{
        /* Request memory for G723ENC instance object */
        memTab[0].size = sizeof(G723ENC_TI_Obj);
        memTab[0].alignment = 0;
        memTab[0].space = IALG_DARAM0;
        memTab[0].attrs = IALG_PERSIST;

        return (1); /* return number of memory blocks requested */
}
```

รูปที่ 3-2 แสดงตัวอย่าง algAlloc() ที่ใช้จองหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ algAlloc() ในการบอกความต้องการหน่วยความจำโดยการเพิ่ม memTab ลงไป การใช้งานจะถูกกำหนดความต้องการหน่วยความจำก่อนที่จะมีการเรียกใช้ฟังก์ชันในการเริ่มต้นการทำงาน

```
typedef struct G723ENC_TI_Obj {
        IALG_Obj ialg;              /* Points to the v-table */
        IG723_Rate workingRate;     /* 5.3 or 6.3 kbps */
        XDAS_Bool hPFilter;         /* High Pass filter on/off */
        XDAS_Bool VAD;              /* Voice activity detection on/off */
        ……                          /* specifics to  the implementation */
} G723ENC_TI_Obj;
```

รูปที่ 3-3 แสดงส่วนของ Object ที่ implement v-table ไว้ภายใน

จากรูปที่ 3-3 สังเกตุว่า IALG_Obj คือตัวชี้ไปที่ v-table ซึ่งเราจะสร้างขึ้นมาทีหลัง พารามิเตอร์นี้ต้องเป็นบรรทัดแรกในการกำหนดโครงสร้างของวัตถุใดๆ และเป็นหน้าที่ความรับผิดชอบของผู้ใช้ที่ต้องกำหนดให้ตัวชี้นี้ ชี้ไปที่ v-table ในตอนสร้าง instance ของอัลกอริทึมนั้นๆ ดังนั้น G723ENC_TI_Obj จะจองหน่วยความจำ block แรก (memTab[0]) เพื่อใช้เก็บข้อมูล

3.2.2. algInit()

อัลกอริทึมจะใช้ algInit() ในการกำหนดค่าเริ่มต้นหลังจากมีการจองหน่วยความจำจาก algAlloc() หลังจากที่เรียก algInit() instance ของอัลกอริทึมที่ชี้จะพร้อมถูกใช้งาน ดูตัวอย่างได้ดังรูปที่ 3-4

```
Int G723ENC_TI_algInit (IALG_Handle handle, const IALG IALG_MemRec
memTab[], IALG_Handle p, const IALG_Params *algParams)
{
        G723ENC_TI_Obj *enc = (Void *)handle;
        Const IG723ENC_Params *param = (Void *)algParams;
        if(params == NULL) {
                params = &IG723ENC_PARAMS;   /* set default parameters */
        }
        /* copy creation params into the object */
        enc->workingRate = params->rate;
        enc->hPFilter = params->hpfEnable;
        enc->VAD = params->vadEnable;

        g723EncInit(enc);       /* Initalize all other instance variables */

        return (IALG_EOK);
}
```

รูปที่ 3-4 แสดงการ initialize ตัวแปร ของ algInit()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ค่า NULL สำหรับตัวพารามิเตอร์อัลกอริทึมจะใช้ค่าพารามิเตอร์พื้นฐาน (default parameters) ตั้งค่าให้กับ instance นั้นๆ ค่าพารามิเตอร์พื้นฐานถูกกำหนดโดยผู้สร้างอัลกอริทึมนี้ขึ้นมา

3.2.3. algFree()

ฟังก์ชันสุดท้ายที่อัลกอริทึมต้องการคือ algFree() ที่อัลกอริทึมรับผิดชอบในการตั้งค่าที่อยู่ (address) และขนาดของหน่วยความจำใน algAlloc() เพื่อลบ instance ของอัลกอริทึมออกโดยปราศจากการค้างพื้นที่หน่วยความจำที่จองไว้ปล่อยทิ้งไว้ ดูตัวอย่างได้จากรูปที่ 3-5

```
Int G723ENC_TI_algFree(IALG_Handle handle, IALG_MemRec memTab[])
{
        G723ENC_TI_Obj *enc = (Void *)handle;
        algAlloc(NULL, NULL, memTab);   /* Fill the memTab struct */
        memTab[0].base = (Void *)&enc;
        return (1);
}
```

รูปที่ 3-5 แสดงส่วนฟังก์ชัน algFree()

เมื่อเรียก algAlloc() จะตั้งค่าพารามิเตอร์ทุกตัวใน โครงสร้างของ memTab ยกเว้น base ซึ่งคือที่อยู่ของ block หน่วยความจำที่ใช้งาน

3.3. Module-Specific Interface

ก่อนหน้านี้ เราได้ใช้ 3 ฟังก์ชัน เพื่อสร้าง, กำหนดค่าเริ่มต้น และลบ instance ของอัลกอริทึม เรายังมีความต้องการใช้ IG723ENC interface ที่เป็น SPI สำหรับตัวอย่างเวอร์ชัน TI ของ ITU G.723.1 annex A encoder IG723ENC interface ได้ถูกกำหนดไว้ใน header ไฟล์ ig723enc.h โดย IG723ENC_Fxns ได้ขยายความสามารถ IALG_Fxns ดังในรูปที่ 3-6

```
typedef struct IG723ENC_Fxns {
        IALG_Fxns ialg;       /* IG723ENC extends IALG */
        XDAS_Bool   (*control)     (IG723ENC_Handle handle, IG723_Cmd cmd,
                                    IG723ENC_Status *status);
        XDAS_Bool   (*encode)      (IG723ENC_Handle handle, XDAS_UInt16 *in,
                                    XDAS_UInt16 *out);

}IG723ENC_Fxns;
```

รูปที่ 3-6 แสดงโครงสร้างของ SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวมาตรฐานแนะนำรูปแบบข้อมูลมาตรฐาน, เช่น XDAS_UInt16 เพื่อรองรับรูปแบบที่สอดคล้องสามารถดูได้จาก xdas.h เพื่อทำให้การกำหนดรูปแบบข้อมูลเป็นมาตรฐานยิ่งขึ้น

เราต้องการสร้างฟังก์ชัน control() และ encode() เพื่อที่จะทำให้อัลกอริทึมสมบูรณ์สำหรับ IG723ENC interface ฟังก์ชันเหล่านี้จะถูกคลุมด้วยวิธีการเดิม ดูได้จากตัวอย่างรูปที่ 3-7 ข้างล่าง

```
XDAS_Bool G723ENC_TI_encode (IG723ENC_Handle handle, XDAS_UInt16 *in,
XDAS_UInt16 *out)
{
        G723ENC_TI_Obj *enc = (Void *)handle;
        if(encoder(enc, in, out))               /* do the processing */
                return (XDAS_TRUE);
        return (XDAS_FALSE);
}
```

รูปที่ 3-7 แสดงฟังก์ชัน G723ENC_TI_encode

IG723ENC interface ได้กำหนดการสร้างพารามิเตอร์ไว้ใน ig723enc.c ถ้าต้องการใช้การสร้างพารามิเตอร์อื่นนอกเหนือจากที่กำหนดไว้ ตัวโปรแกรมจะสำเนา (copy) พารามิเตอร์ของอัลกอริทึมและแก้ไขตามความต้องการก่อนที่จะสร้าง instance โดยต้องเรียกผ่านฟังก์ชัน algAlloc() และ algInit() เพื่อสร้าง instance ที่ได้แก้ไขพารามิเตอร์แล้ว

3.3.1 V-table

ในตอนนี้เราต้องการที่จะกำหนดและตั้งค่าเริ่มต้น IG723ENC v-table นั่นคือ IG723_Fxns ที่ได้ขยายมาจาก IALG_Fxns ตัว IG723ENC v-table ได้รวบรวมเอาฟังก์ชันใน IALG v-table เข้าไปด้วยดังรูปที่ 3-8

บรรทัดแรกใน V-table คือตำแหน่งของคาราง ส่วนนี้ถูกใช้เป็นค่าเอกลักษณ์ของแต่ละอัลกอริทึม สังเกตุว่ามี 3 ฟังก์ชันเท่านั้นที่ IALG ต้องประกาศไว้ และฟังก์ชันอื่นที่ไม่กำหนดใน v-table ทั้งหมดจะถูกตั้งค่าชี้ไปที่ NULL

ฟังก์ชัน asm() ใช้กำหนดให้ชื่อ G723ENC_TI_IALG ให้มีค่าเหมือนกับ G723ENC_TI_IG723ENC ซึ่งหมายความว่า IALG v-table และ IG723ENC v-table จะถูกแชร์ร่วมกัน

การตั้งชื่อใน V-table

1. G723ENC_TI_IALG อ่านว่า "TI's implementation of the IALG interface for the G723ENC module"
2. G723ENC_TI_IG723ENC อ่านว่า "TI's implementation of the IG723ENC interface for the G723ENC module"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

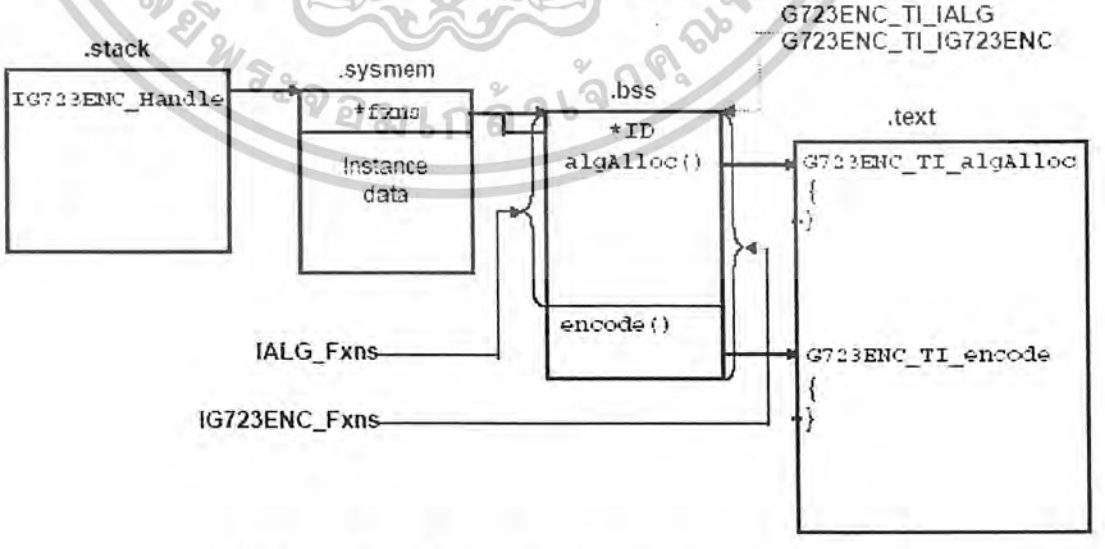
```
#define IALGFXNS \
&G723ENC_TI_IALG,              /* module ID */          \
NULL,                          /* activate */           \
G723ENC_TI_algAlloc            /* alloc */              \
NULL,                          /* control */            \
NULL,                          /* deactivate */         \
G723ENC_TI_algFree,            /* free */               \
G723ENC_TI_algInit,            /* init */               \
NULL,                          /* moved */              \
NULL                           /* numAlloc */
IG723ENC_Fxns G723ENC_TI_IG723ENC = {
IALGFXNS,                      /* IALG functions *.
G723ENC_TI_control,
G723ENC_TI_encode
} G723ENC_TI_IG723ENC;
asm (" _G723ENC_TI_IALG .set _G723ENC_TI_IG723ENC");
```

รูปที่ 3-8 แสดง code ที่ใช้ในการประกาศ v-table

ถ้ามีการเรียกเข้าไปใน V-table ของ IALG_Handle จะสามารถเข้าได้เพียง IALG_Fxns และมีความจำเป็นต้องใช้ IG723ENC_Handle เพื่อเข้าถึงฟังก์ชัน encode () และ control ()

3.3.2 โครงสร้างอัลกอริทึม

รูปที่ 3-9 แสดงให้เห็นถึงวัตถุ (Object) ของ v-table และการใช้งานตามส่วนจริงที่อยู่ในหน่วยความจำ การจัดการ instance ของวัตถุอยู่ในรูปแบบของ stack ดังรูปที่ 3-9



รูปที่ 3-9 แสดง โครงสร้างของ TMS320 DSP Algorithm Standard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4. คำนิยามของ Dynamic System

Dynamic System เป็นระบบซึ่งสามารถนำหน่วยความจำมาใช้งานใหม่ได้ แม้ในขณะที่โปรแกรมนั้นๆทำงานอยู่ การพิจารณาระบบ Dynamic นั้นจะยุ่งยากมากๆ หากเรานำเอาการใช้งาน Real-time algorithm และการใช้งาน Multi-tasking มาร่วมพิจารณาด้วย ดังนั้นเราจะมองเฉพาะระบบที่มีการทำงานเดี่ยว (single-tasking) เท่านั้น ซึ่งจะมีเฉพาะส่วนของหน่วยความจำข้อมูลเท่านั้นที่ถูกนำมาใช้งานใหม่

เราจะอธิบายการใช้งานหน่วยความจำของ Dynamic System ด้วยระบบโทรศัพท์แบบ Secure-voice โดยโทรศัพท์แบบนี้จะมีการติดตั้งระบบ DSP ซึ่งสามารถเข้าและถอดรหัสเสียงพูดได้ และเมื่อไม่ได้ใช้งานโทรศัพท์ ตัว DSP ก็จะมีอัลกอริทึมซึ่งสามารถจดจำและจำแนกเสียงต่างๆเพื่อสื่อสารกับเครื่องคอมพิวเตอร์ได้ ซึ่งทั้ง 2 โปรแกรมนี้จะมีการจัดการการใช้งานหน่วยความจำที่แตกต่างกันขึ้นอยู่กับว่าโปรแกรมไหนถูกใช้งานอยู่ในขณะนั้น ก่อนที่จะพูดถึงเรื่องโทรศัพท์ เรามาดูว่ามาตรฐานของ TMS320 DSP Algorithm Standard เป็นเช่นไร ซึ่งได้อธิบายมาบ้างแล้วจากข้างต้น โดยในส่วนนี้จะเป็นส่วนของฟังก์ชันอื่นๆที่เพิ่มเข้ามาจาก 3 ฟังก์ชันหลักที่ต้อง implement ได้แก่ algActivate(), algControl(), algDeactivate(), algMoved() และ algNumAlloc()
และจะกล่าวถึงการจัดการหน่วยความจำด้วย

3.4.1. The Create Function

ฟังก์ชันนี้ใช้ทำการจัดสรรหน่วยความจำสำหรับอัลกอริทึมต่างๆดังที่ได้กล่าวมาแล้ว โดยในที่นี้จะใช้ XMP ('Example algorithm') เป็นตัวอธิบาย

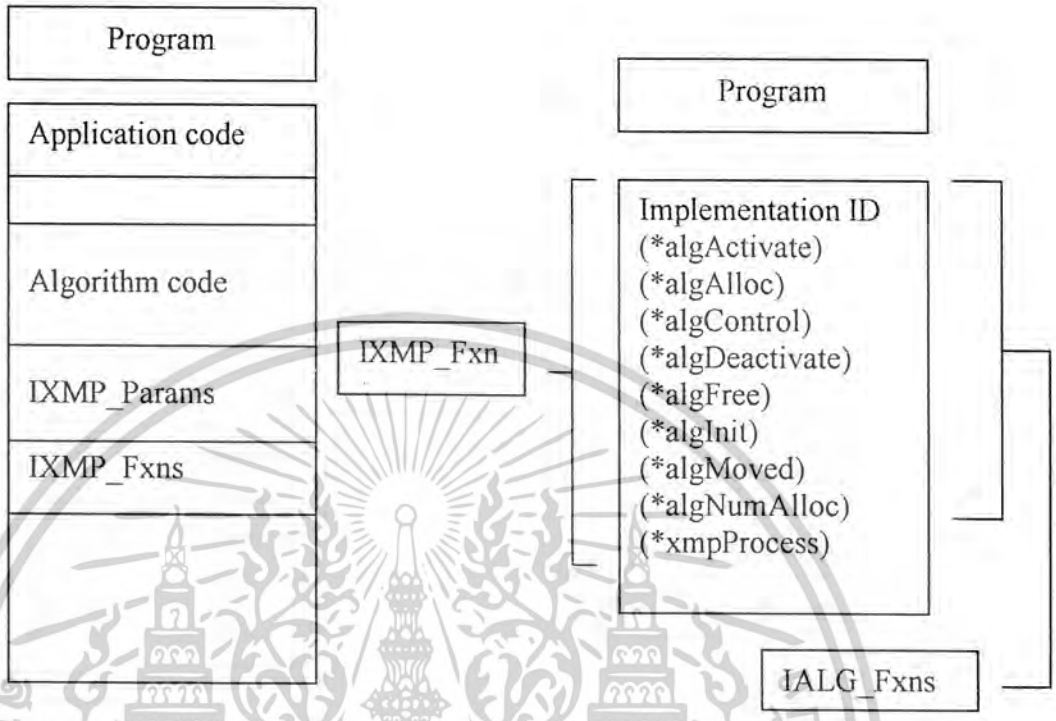
ตัวตารางฟังก์ชันซึ่งเป็นจุดเริ่มต้นของกระบวนการต่างๆที่เกี่ยวข้องกับ Standard Algorithm คือโครงสร้าง IALG_Fxns ได้ถูกประกาศไว้ดังรูปที่ 3-10

```
IXMP_Fxns XMP_IXMP = {
        &XMP_IXMP,      /* implementation ID */         \
        xmpActivate,    /* algActivate */               \
        xmpAlloc,       /* algalloc */                  \
        xmpControl,     /* algControl */                \
        xmpDeactivate,  /* algDeactivate */             \
        xmpFree,        /* algFree */                   \
        xmpInit,        /* algInit */                   \
        xmpMoved,       /* algMoved */                  \
        xmpNumAlloc,            /* algNumAlloc */           \
        xmpprocess      /* algorithm specific processing */
};
```

รูปที่ 3-10 แสดงตัวอย่าง v-table ของ XMP Algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3-11 ใช้ในการอธิบายตำแหน่งของหน่วยความจำ สำหรับตัวอย่าง XMP นี้



รูปที่ 3-11 แสดงหน่วยความจำของ XMP Algorithm

3.4.2. The Algorithm-Specific Create Function

ฟังก์ชัน XMP_create() เป็นฟังก์ชันระดับสูง ซึ่งใช้สร้างอัลกอริทึมขึ้นมาโดยคืนค่าเป็นตำแหน่งของวัตถุนั้นๆ

```
XMP_Handle handle;

Handle = XMP_create(&XMP_IXMP, NULL);
```

รูปที่ 3-12 แสดง High-level function ที่ชื่อว่า XMP_create ()

XMP_create() จะรับค่าพอยต์เตอร์ที่ชี้ไปยัง Fn table และพอยต์เตอร์ที่ชี้ไปยัง พารามิเตอร์ที่ใช้ในการสร้าง instance โดยค่านี้อาจใส่เป็น NULL ถ้าหากเราต้องการใช้ค่าพารามิเตอร์พื้นฐาน

ถ้าใช้ค่า NULL จะได้ดังรูปที่ 3-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(prms == NULL) {
        prms = &XMP_PARAMS;
}
```

รูปที่ 3-13 แสดงส่วนของ Code เมื่อผ่านค่าพารามิเตอร์เข้ามาเป็น NULL

หลังจากค่าพารามิเตอร์ที่ใช้ในการสร้าง Instance ได้ถูกกำหนดแล้ว ฟังก์ชัน ALG_create() จะถูกเรียกขึ้นมาใช้งาน เพื่อใช้กำหนดหน่วยความจำที่ต้องการ, จัดสรรหน่วยความจำ และเริ่มต้นการทำงานของอัลกอริทึมของ instance นั้นๆ ฟังก์ชันทั่วไปของ ALG_create() แสดงดังรูปที่ 3-14

```
handle = (XMP_Handle)ALG_create(
        (IALG_Fxns *) fxns,         /* function table */
        (IALG_Handle) NULL,         /* parent object or NULL */
        (IALG_Param *) prms         /*creation parameters */
);
```

รูปที่ 3-14 แสดงส่วนการเรียก ALG_create () ทั่วไปเพื่อใช้งาน

3.4.3. ความต้องการหน่วยความจำ (Memory Requirements)

ขั้นต่อมาของการเรียกใช้ อัลกอริทึม ก็คือการจัดสรรค์หน่วยความจำตามที่อัลกอริทึมต้องการ โดยที่อัลกอริทึมจะแสดงค่าของ Block หน่วยความจำที่ต้องการใส่ลงไปในตารางหน่วยความจำ

ขั้นตอนการสร้างตารางหน่วยความจำ ฟังก์ชัน algNumAlloc () จะเป็นตัวบอกเราถึงจำนวนของ block ของหน่วยความจำที่สูงที่สุดที่อัลกอริทึมนั้นๆต้องการ ถ้าเราไม่ได้ทำการเรียกใช้ฟังก์ชันนี้ ค่าพื้นฐานก็จะถูกนำมาใช้งาน โดยค่าพื้นฐานนี้จะมีค่าเท่ากับ 4

3.4.4. ฟังก์ชัน Activate

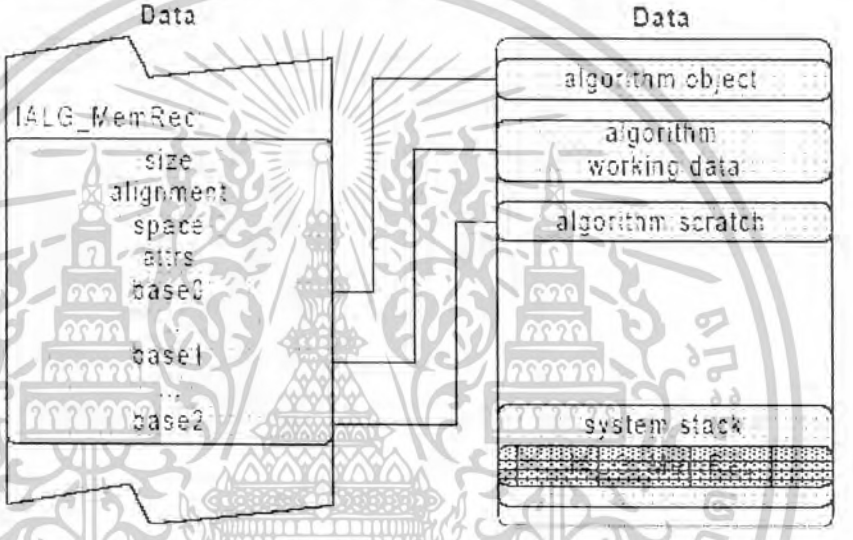
ในส่วนนี้เป็นส่วนทางเลือกที่จะทำการ implement หรือไม่ก็ได้ โดยในส่วนของฟังก์ชันนี้เพื่อใช้จัดการกับหน่วยความจำชั่วคราว (scratch memory) และหน่วยความจำแชร์ (shared memory) ส่วนของฟังก์ชัน algActivate() ใช้เมื่อต้องการนำข้อมูลจากส่วนของหน่วยความจำถาวร ไปไว้ที่หน่วยความจำชั่วคราวซึ่งมีความเร็วมากกว่า ส่วน algDeactivate() ใช้เมื่อต้องการนำข้อมูลจากหน่วยความจำชั่วคราวมาไว้ที่หน่วยความจำถาวร เพื่อป้องกันการสูญหายของข้อมูลเมื่ออัลกอริทึมอื่นมาใช้งานในส่วนของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

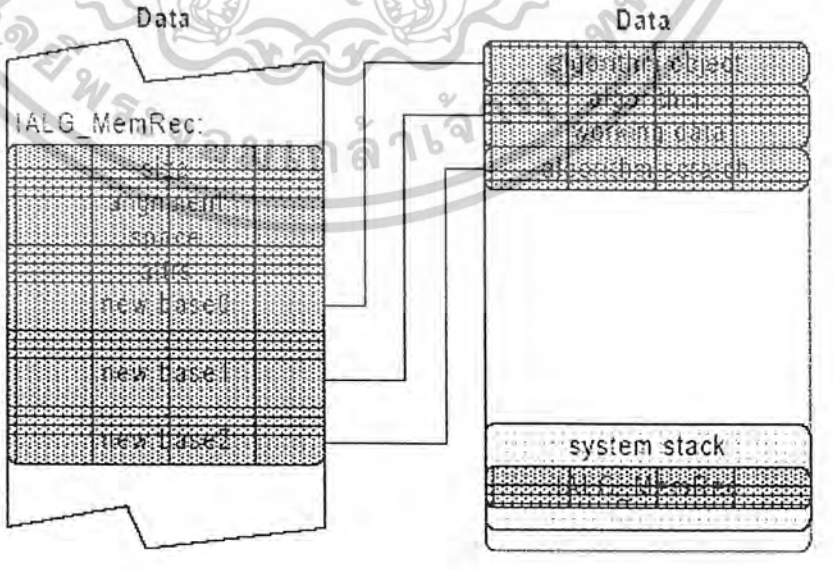
หน่วยความจำชั่วคราวและทำการเขียนข้อมูลอื่นที่ไม่เกี่ยวข้องกับอัลกอริทึมนี้ลงไป ทำให้อัลกอริทึมเมื่อกลับมาทำงานใหม่เกิดการผิดพลาดที่ข้อมูลไม่ตรงได้

3.4.5. ฟังก์ชัน Move

เมื่อมีความจำเป็นที่จะต้องจัดตำแหน่งของข้อมูลใหม่ในอัลกอริทึมใดๆนั้น โปรแกรมจะทำการแจ้งไปยังอัลกอริทึมด้วยการเรียกใช้ฟังก์ชัน algMoved () ค่าตัวแปรต่างๆที่ถูกส่งไปยัง algMoved() นั้นจะถูกส่งต่อไปยัง algInit() ด้วย จะแตกต่างกันเพียงแค่ค่าพอยต์เตอร์เท่านั้น ตัวอัลกอริทึมนั้นจะต้องเปลี่ยนค่าที่ใช้ในการอ้างอิงข้อมูลให้ตรงกับค่าพอยต์เตอร์ตัวใหม่ ฟังก์ชันนี้เป็นเพียงทางเลือกเท่านั้น ถ้าฟังก์ชันไม่ต้องการใช้ให้ตั้งค่าเป็น NULL ไว้ สามารถอธิบายการจัดเรียงข้อมูลได้ดังรูปที่ 3-15A และ รูปที่ 3-15B



รูปที่ 3-15A แสดงข้อมูลในหน่วยความจำก่อนจัดการเรียงข้อมูล



รูปที่ 3-15B แสดงข้อมูลในหน่วยความจำหลังจากจัดเรียงข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5. ทำไมต้องเป็น Dynamic System

ในระบบ DSP ทุกๆระบบจะต้องมีหน่วยความจำ 2 แบบ ทั้งแบบ On-chip และแบบ external การที่ระบบ DSP จะใช้งานอย่างมีประสิทธิภาพสูงสุดออกมา อัลกอริทึมต้องเก็บข้อมูลที่ใช้งานไว้ในหน่วยความจำ on-chip ซึ่งมีความเร็วสูงกว่าหน่วยความจำ external โดยไม่สามารถหลีกเลี่ยงได้เมื่อปริมาณของกระบวนการมากขึ้นจนเกินปริมาณของหน่วยความจำ on-chip เมื่อเกิดเหตุการณ์เช่นนี้ขึ้นถ้าเราสามารถสลับการทำงานฟังก์ชันได้อย่างรวดเร็ว และสามารถโยกย้ายข้อมูลไปมาระหว่าง on-chip และ off-chip ได้เร็วแล้ว เราก็จะได้ระบบที่ทำงานแบบขนานได้

3.6. Dynamic System ทำงานอย่างไร

จากที่ได้กล่าวไว้นั้น เราจะใช้ระบบโทรศัพท์ที่มีความสามารถในการใช้ DSP มาใช้ในการอธิบาย โดยที่ตัวโทรศัพท์นั้นสามารถทำงานได้อย่างปกติ และมันยังมีระบบจดจำเสียง และการเข้ารหัสเสียงอีกด้วย โดยระบบนี้มีคุณสมบัติดังนี้

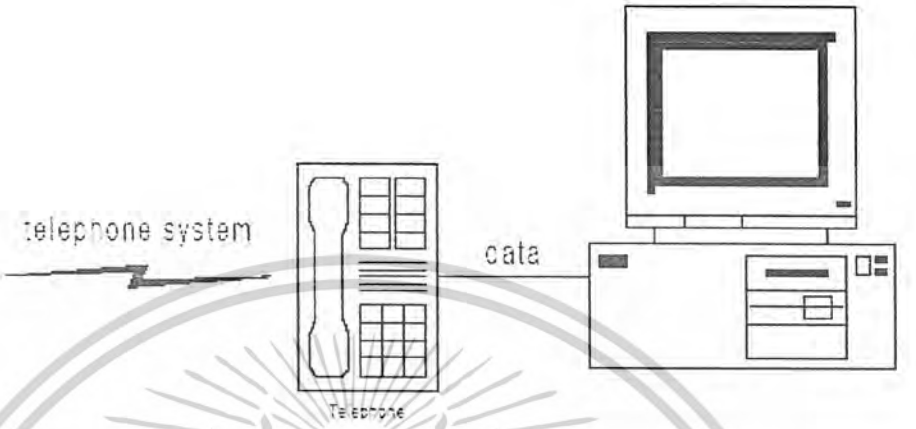
1. ระบบเป็นแบบ non-preemption
   - การสลับการทำงานของอัลกอริทึมจะเกิดขึ้นภายใต้การควบคุมของโปรแกรม
2. การใช้งานหน่วยความจำของโปรแกรมจะเป็นแบบ stack
   - จะมีหน่วยความจำโปรแกรมที่พอเพียง ที่โปรแกรมจะถูกบรรจุไว้ในหน่วยความจำ
3. การใช้งานหน่วยความจำข้อมูลจะเป็นแบบ Dynamic
   - หน่วยความจำแบบ on-chip จะมีอยู่อย่างจำกัดและถูกแชร์ไว้ โดยจะมีหน่วยความจำที่พอเพียงอยู่ใน off-chip

เราจะมาดูกันที่เรื่องต่อไปนี้เพื่อดูว่าส่วนของหน่วยความจำข้อมูลนั้นถูกจัดการอย่างไร

1. Power on โปรแกรมจดจำเสียงพูด (ผ่านทาง speaker phone) จะเป็นตัวป้อนข้อมูลเข้าและออก ให้กับคอมพิวเตอร์
2. เริ่มต้นการโทรออก ตัวสร้างสัญญาณ DTMF และโปรแกรมตัดสัญญาณเสียงสะท้อน (Echo) จะถูกใช้ในการโทรออก
3. การเข้ารหัสสัญญาณเสียง ตัวอัลกอริทึมจะทำหน้าที่เข้ารหัสเสียง เพื่อส่งต่อไปยังโทรศัพท์ที่สามารถรับได้ หรืออุปกรณ์อื่น
4. การยกเลิกการโทร โทรศัพท์จะกลับเข้าสู่ระบบจดจำเสียงสำหรับการเชื่อมต่อกับเครื่องคอมพิวเตอร์อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์ตัวอย่างที่ใช้ในการอธิบายเป็นไปตามรูปที่ 3-16

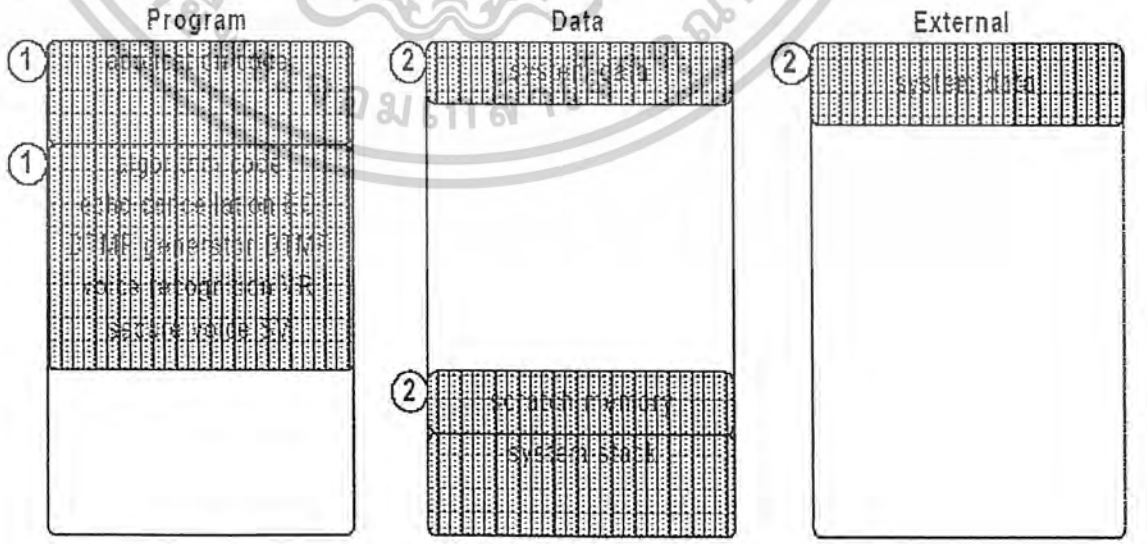


รูปที่ 3-16 แสดงระบบตัวอย่างที่ใช้ในการอธิบาย Dynamic System

3.6.1. สถานะตอน Power-On

เมื่อเปิดเครื่องโทรศัพท์ขึ้นมา Code สำหรับการใช้งานโทรศัพท์จะถูกเก็บไว้ใน ROM ที่ความเร็วต่ำและราคาถูก โดยหน่วยความจำโปรแกรมแบบ on-chip, หน่วยความจำข้อมูลแบบ on-chip และหน่วยความจำภายนอก ซึ่งก็คือ RAM นั้นจะยังไม่ถูกกำหนดหรือถูกใช้งาน

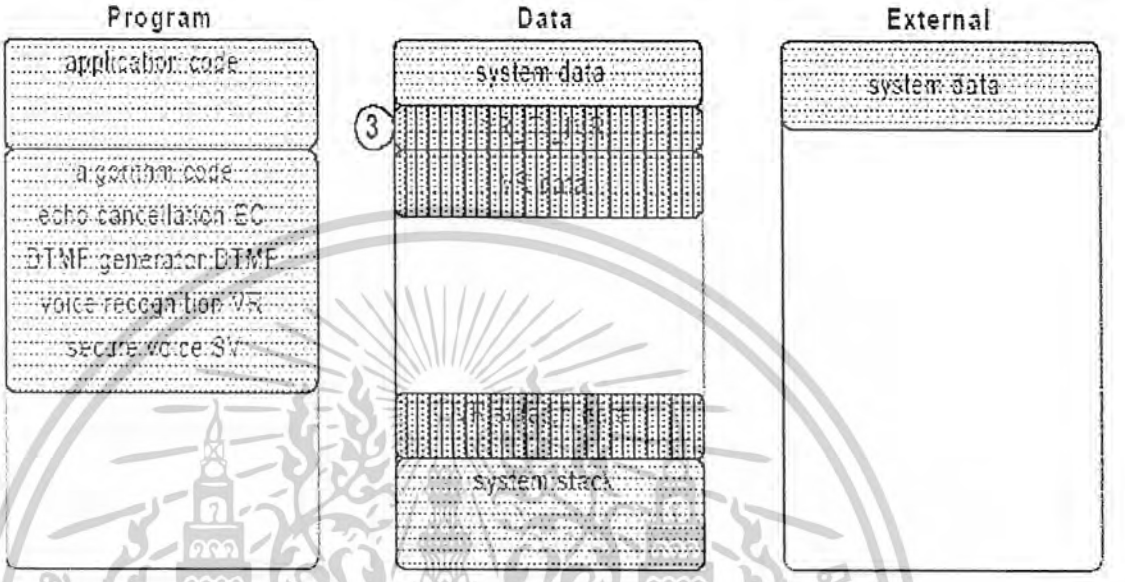
โปรแกรม Boot ระบบจะทำการเรียกโปรแกรมเข้าสู่หน่วยความจำโปรแกรมแบบ on-chip จาก ROM และจะสร้าง stack ของระบบไว้บน Data RAM และส่วนที่เหลือที่ต้องใช้งานจะถูกเก็บไว้ใน Data RAM หรือ external RAM แสดงได้ดังรูปที่ 3-17



รูปที่ 3-17 แสดงหน่วยความจำของระบบตัวอย่างเมื่อตอน Power-on

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

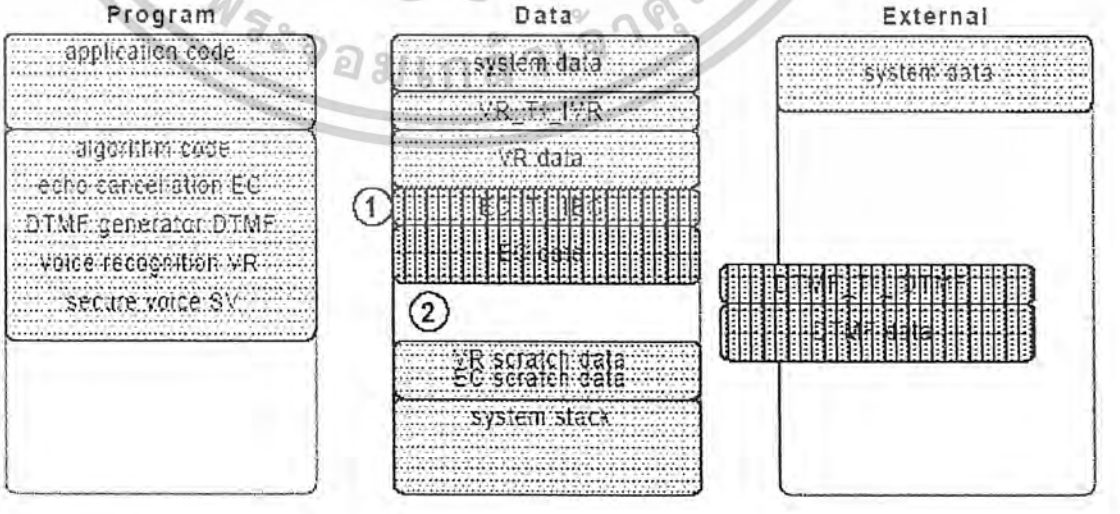
ขณะนี้โปรแกรมนั้นทำงานอยู่โดยไม่มีการใช้งานอัลกอริทึมอื่นใดๆ เข้ามาใช้งาน โปรแกรมจะทำการเรียก Voice-recognition ขึ้นมาแล้วเริ่มต้นใช้งานตามรูปที่ 3-18 ซึ่งโปรแกรมนี้จะทำให้เราสามารถสั่งงานเครื่องคอมพิวเตอร์ทางเสียงผ่านทาง speaker phone ได้



รูปที่ 3-18 แสดงหน่วยความจำเมื่อ Voice-recognition ขึ้นมาทำงาน

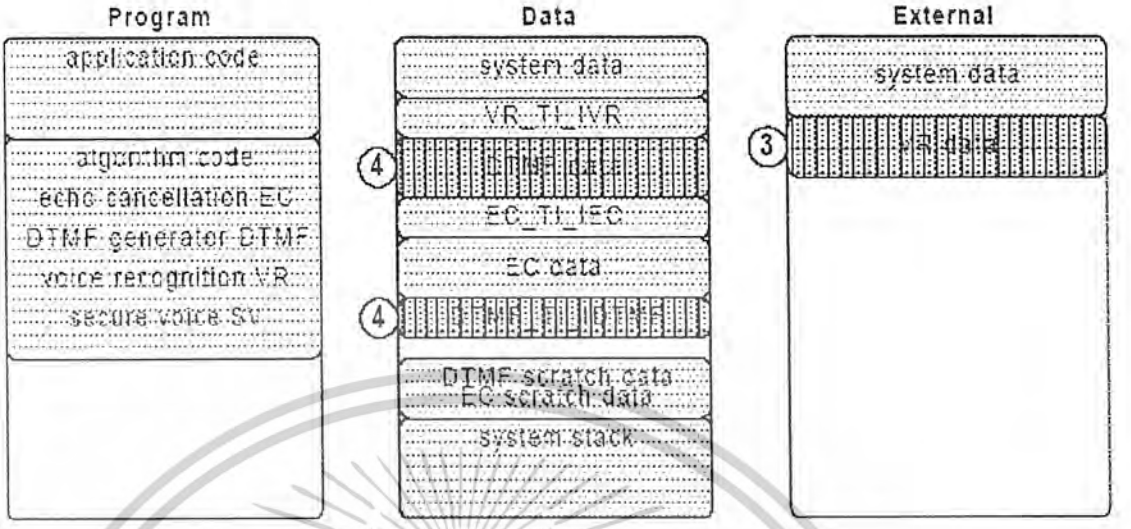
3.6.2. การเริ่มต้นการโทรออก

ในการที่จะเริ่มต้นโทรออก ส่วนของ Echo cancellation และส่วนการสร้าง DTMF จะต้องถูกสร้างขึ้นและถูกเรียกใช้งาน จากแผนผังการใช้งานหน่วยความจำ ส่วนของการตัดเสียงรบกวนสามารถใช้งานได้ตามรูปที่ 3-19(1) แต่ไม่มีหน่วยความจำมากพอสำหรับเรียกใช้ DTMF ตามรูปที่ 3-19(2)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-19 แสดงหน่วยความจำที่ไม่เพียงพอสำหรับ DTMF อัลกอริทึม



รูปที่ 3-20 แสดงการย้ายข้อมูลของ Voice-recognition ไปไว้ที่หน่วยความจำภายนอก

ทางแก้ก็คือสลับข้อมูลจากหน่วยความจำข้อมูลที่ถูกใช้งานโดย Voice-recognition ตามรูปที่ 3-20 โดยมีขั้นตอนดังนี้

1. สำเนาข้อมูล Voice-recognition จากหน่วยความจำข้อมูลไปยังตำแหน่งใน หน่วยความจำภายนอก
2. เรียกใช้งานฟังก์ชัน algMoved() เพื่อบอกอัลกอริทึมว่าข้อมูลได้ย้ายไปอยู่ที่ใด
3. เรียก VR_deActivate() เพื่อให้คงสถานะของอัลกอริทึมไว้เช่นนั้น

เมื่อทำตามกระบวนการนี้เสร็จ เราก็จะมีที่ว่างสำหรับสร้าง DTMF

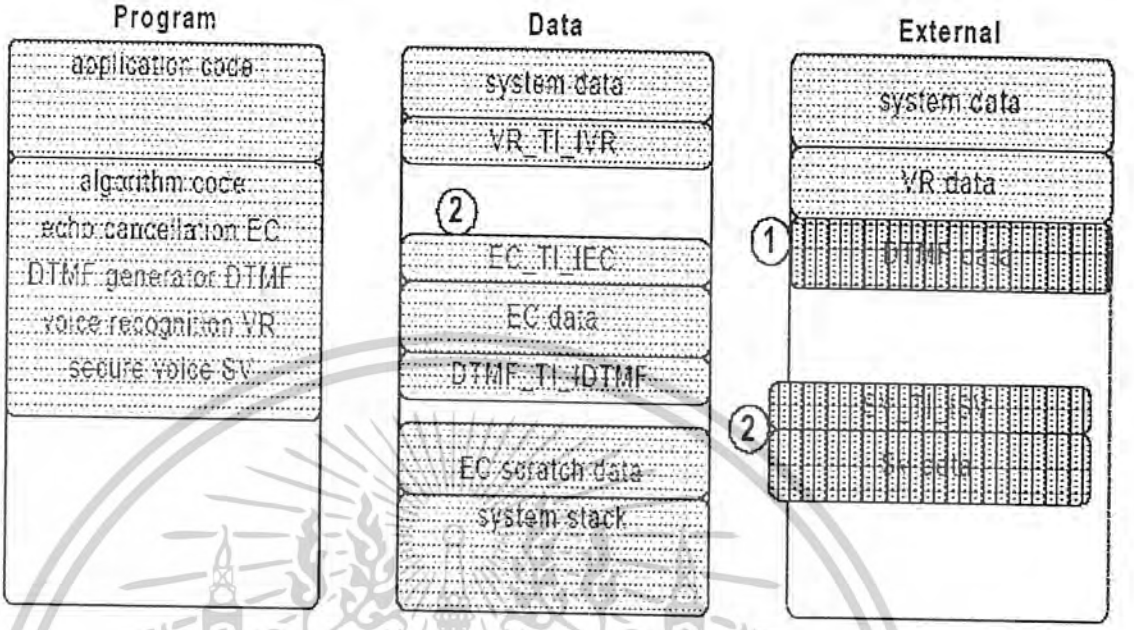
3.6.3. การสลับไปใช้งานการเข้ารหัสเสียง

การประกอบกันระหว่าง การบีบอัดเสียง, การเข้ารหัสข้อมูล และฟังก์ชันต่างๆของโมเด็ม ทำให้เราสามารถสร้างช่องทางสัญญาณที่มีความปลอดภัยขึ้นมา เพื่อสื่อสารกับโทรศัพท์ปลายทางที่สามารถรองรับการทำงานนี้ได้

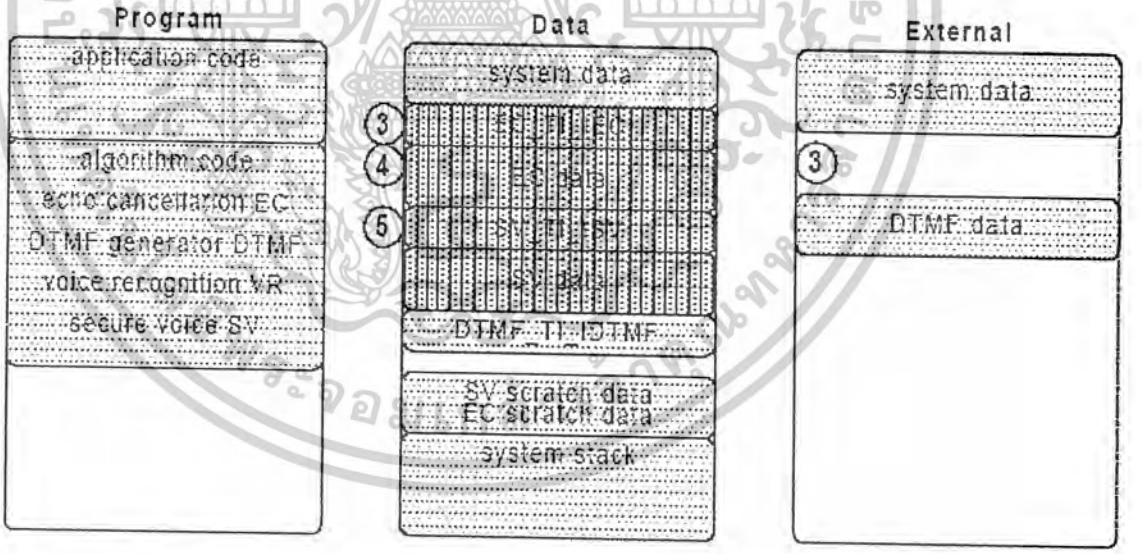
จากขั้นตอนที่แล้ว เราจะเห็นได้ว่ามีหน่วยความจำเหลืออยู่ไม่มาก การจะเริ่มใช้งานอัลกอริทึมนี้เราจำเป็นต้องหาที่ว่างในหน่วยความจำก่อน เนื่องจาก Tone generator จะไม่ได้ถูกใช้งานในขณะที่เราใช้งานในโหมด Secure-voice ดังนั้นเริ่มต้น เราต้องย้ายหน่วยความจำข้อมูล และปิดการทำงานของมันก่อน ตามรูปที่ 3-21(1)

หากเราต้องการหน่วยความจำเพิ่มอีก ตามรูปที่ 3-21(2) เราสามารถลบอัลกอริทึมของ Voice-recognition ออกได้ตามรูปที่ 3-22(3) และเราจะได้หน่วยความจำที่เพียงพอแล้วแต่ไม่ได้อยู่รวมกันเป็น block เราจะสามารถแก้ปัญหานี้ได้โดยการจัดเรียงข้อมูลในหน่วยความจำของตัวกำจัดเสียงก้องใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-21 แสดงหน่วยความจำไม่เพียงพออีกครั้ง



รูปที่ 3-22 แสดงการจัดเรียงข้อมูลใหม่ในระบบตัวอย่าง

เราจะต้องสร้างตารางหน่วยความจำขึ้นมาใหม่ เพื่อเก็บตำแหน่งของข้อมูลอัลกอริทึมใหม่ แล้วส่งค่าของตารางนั้นไปยังอัลกอริทึม โดยใช้ฟังก์ชัน algMoved() โดยทำตามขั้นตอนดังนี้

1. สร้างตารางจัดสรรข้อมูลในหน่วยความจำขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้ algAlloc() เพื่อให้ค่าในตารางเป็นค่าเดียวกับตอนที่อัลกอริทึมได้ถูกสร้างขึ้น
3. ใส่ค่าตำแหน่งลงไปในตาราง
4. ย้ายข้อมูลส่วนต่างของหน่วยความจำไปไว้ที่ตำแหน่งใหม่
5. เรียกใช้ algMoved() ด้วยตารางที่ถูกสร้างขึ้นมาใหม่

ด้วยการย้ายข้อมูลของ Echo-cancellation ทำให้เรามีที่ว่างในการเรียกใช้งาน secure-voice อัลกอริทึมได้ ทำให้เราสามารถสื่อสารด้วยช่องสัญญาณที่ปลอดภัยได้ โดยใช้สายโทรศัพท์เส้นเดิม

3.6.4. การวางสายโทรศัพท์

การวางสายโทรศัพท์ออก จะทำให้เราต้องเรียกใช้งาน Voice-recognition อีกครั้ง โดยทำตามขั้นตอนดังนี้

1. ส่วนของ secure-voice อัลกอริทึม และข้อมูลในหน่วยความจำจะถูกย้ายออกไปยังหน่วยความจำภายนอกตามรูปที่ 3-23(1) แล้วอัลกอริทึมนั้นก็จะถูกปิดลง ทำให้หน่วยความจำถูกปล่อยออก
2. Voice-recognition ได้ถูกสร้างขึ้นมาใหม่ตามรูปที่ 3-23(2) และถูกเรียกใช้งาน
3. DTMF ถูกลบทิ้งออกจากหน่วยความจำ

เครื่องคอมพิวเตอร์จะสามารถรับคำสั่งจากเสียงได้จนถึงการโทรครั้งต่อไป

3.7. บทสรุปท้ายบท

TMS320 DSP Algorithm standard จะทำงานโดยสามารถแบ่งเป็น 3 ส่วน คือ Client (Users) of algorithms, Producers of algorithms และ Creators of algorithm specific interface การใช้งานนี้จะเน้นไปที่ producers of algorithms มันถูกจัดการจากผู้สร้าง XDAS Software กับเทคนิคของส่วนประกอบพื้นฐานที่เป็นไปตามกฎของ eXpressDSP-compliant และมีการจัดการที่เป็นตัวเสริมกรทำงานบางอย่างในการปรับปรุงให้ดีขึ้นในอนาคต

กล่าวอ้างอิงถึง TMS320 DSP Algorithm Standard Development Guild (SPRU424) สำหรับการดำเนินการเพื่อสร้างอัลกอริทึม eXpressDSP-compliant ที่สมบูรณ์ มีขั้นตอนดังนี้

1. Implement algAlloc(), algInit() และ algFree()
2. Implement module-specific interface
3. สร้าง v-table
4. มั่นใจว่าทำตามข้อกำหนดหรือกฎข้อบังคับ
5. สร้าง library และ header ไฟล์

เมื่อถึงจุดนี้อัลกอริทึมควรจะเป็นไปตาม eXpressDSP-compliant ส่วนขั้นตอนอื่นๆนอกจากนี้อาจพิจารณาถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

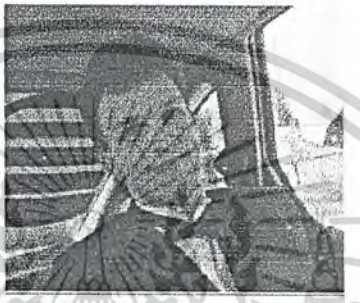
1. การแบ่งหน่วยความจำที่ต้องการออกเป็นหลายๆ block
2. การใช้งานฟังก์ชันใน IALG interface (algActivate(), algDeactivate() และฟังก์ชันอื่นๆ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบและผลการทดสอบ

4.1 ไฟล์วิดีโอที่ใช้ในการทดสอบการทำงานของโปรแกรม



รูปที่ 4-1 แสดงรูปภาพตัวอย่างจากสัญญาณภาพวิดีโอที่ใช้ในการทดสอบ

4.2 การทดสอบการทำงานในส่วนเข้ารหัส

4.2.1 การทดสอบที่ 1.

ทดสอบการเข้ารหัสสัญญาณวิดีโอบน Board TMS320C6713 DSP Starter Kit โดยใช้งานเพียง option เดียวในการทดสอบ โดยมีขั้นตอนการทดสอบดังนี้

1. ตั้งค่าการบีบอัดตาม option ที่ได้ implement ไว้
    a. Unrestricted Motion Vector Mode
    b. Syntax-base Arithmetic Coding mode
    c. Advance Prediction mode
    d. PB-frame mode
    e. Insert GOB
    f. ไม่มีการกำหนดใช้งาน option ใดๆ
2. หาค่า PSNR จากการประมวลผลจาก option ต่างๆ
3. วัดขนาดของไฟล์ที่ได้จากการ Encode ที่ได้ใช้งานแต่ละ option ที่เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

Setting	PSNR	File Size[kb]
Not uses option	13.138892	6.57
Unrestricted Motion Vector Mode	13.130404	6.49
Uses Group Of Bloc	13.138892	6.87
Syntax-base Arithmetic Coding mode	13.138892	6.07
Advance Prediction mode	13.099500	6.41
PB-frame mode	13.061009	5.57

ตารางที่ 4-1 ตารางแสดงผลการทดลองจากการทดสอบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดสอบที่ 2.

ทดสอบการทำงานของโปรแกรมโดยใช้งาน Option ที่ใช้ในการเข้ารหัสหลายๆ option และทดลองสั่งให้โปรแกรมประมวลในการเข้ารหัสสัญญาณวิดีโอ

ผลการทดลอง

Setting	PSNR	File Size[kb]
Uses 2 Options		
UM & SAC	13.167684	5.6
UM & ADV	13.156008	5.46
UM & PB	13.058220	5.49
SAC & ADV	13.156008	5.46
SAC & PB	13.061009	5.08
ADV & PB	13.050612	5.35
Uses 3 Options		
UM & SAC & ADV	13.148530	5.48
UM & SAC & PB	13.058220	5.02
UM & ADV & PB	13.047533	5.35
SAC & ADV & PB	13.050612	4.89
Uses 4 Options		
UM & SAC & ADV & PB	13.047533	4.91

Note –

- UM : Unrestricted Motion Vector Mode
- SAC: Syntax-base Arithmetic Coding mode
- ADV: Advance Prediction mode
- PB : PB-frame mode

ตารางที่ 4-2 แสดงผลการทดลองจากการทดสอบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการทำงานในส่วนถอดรหัส

4.3.1 การทดสอบที่ 3

ทดสอบการถอดรหัสสัญญาณวิดีโอบน Board TMS320C6713 DSP Starter Kit โดยใช้งานเพียง option เดียวในการทดสอบ โดยมีขั้นตอนการทดสอบดังนี้

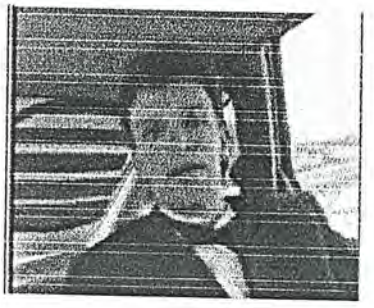
1) ตั้งค่าการบีบอัดตาม Option ที่ได้ implement ไว้
   a. Unrestricted Motion Vector Mode
   b. Syntax-base Arithmetic Coding mode
   c. Advance Prediction mode
   d. PB-frame mode
   e. ไม่มีการกำหนดใช้งาน option ใดๆ
2) หาค่า อัตราความเร็วในการ Decode ในหน่วย frame /sec

โหมดการทำงาน	เริ่มการ decode ที่เวลา	เสร็จสิ้นการ decode ที่เวลา	อัตราความเร็วในการ Decode/frame (sec/frame)	ดูผลการทดลองได้ที่
ไม่มีการเซตโหมดใดๆ	10.24.32	10.54.15	32.8	รูปที่ 4-2
Advanced prediction mode	10.38.39	10.44.02	32.3	รูปที่ 4-3
Syntax arithmetic coding	3.16.17	3.21.55	33.8	รูปที่ 4-4
Unrestricted motion vector	4.09.13	4.14.19	30.6	รูปที่ 4-5
PB-frame	11.28.44	11.34.09	32.5	รูปที่ 4-6

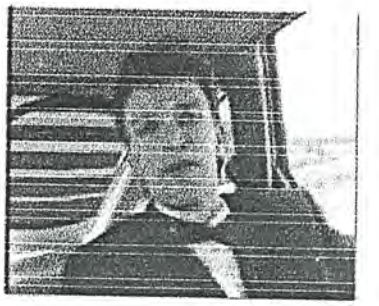
ตารางที่ 4-3 แสดงผลการทดลองจากการทดสอบ decode ที่ 3

Note เมื่อการทดลองนี้ได้ทดลองการทำงานที่ 10 เฟรม ของ รูปที่ได้นำมาทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 ไม่มีการเซตโหมดใดๆ



รูปที่ 4-3 เซต advanced prediction mode

รูปที่ 4-4 เซต syntax-base arithmetic coding

รูปที่ 4-5 เซต unrestricted motion vector

รูปที่ 4-6 เซต pb-frames

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดสอบที่ 4.

ทดสอบการทำงานของโปรแกรมโดยใช้งาน Option ที่ใช้ในการเข้ารหัสหลายๆ option และทดลองสั่งให้โปรแกรมประมวลในการถอดรหัสสัญญาณวิดีโอ

ผลการทดลอง

โหมดการทำงาน	เริ่มการ decode ที่เวลา	สิ้นสุดการ decode ที่เวลา	อัตราความเร็วในการ Decode/frame (sec/frame)	ดูผลการทดลองได้ที่
ADV & PB	3.28.17	3.33.14	29.7	รูปที่ 4-7
SAC & ADV	3.51.42	3.57.24	34.2	รูปที่ 4-8
SAC & PB	4.14.26	4.19.52	31.6	รูปที่ 4-9
UM & ADV	12.31.14	12.36.51	33.7	รูปที่ 4-10
UM & PB	4.23.16	4.28.15	29.9	รูปที่ 4-11
UM & SAC	4.30.33	4.36.27	35.4	รูปที่ 4-12
UM & SAC & ADV	4.48.16	4.53.59	34.3	รูปที่ 4-13
UM & SAC & PB	4.56.22	5.01.26	30.4	รูปที่ 4-14
SAC & ADV & PB	5.11.35	5.16.46	31.1	รูปที่ 4-15
UM & ADV & PB	5.24.55	5.30.02	30.7	รูปที่ 4-16
UM & SAC & ADV & PB	5.44.34	5.49.44	31.0	รูปที่ 4-17

ตารางที่ 4-4 แสดงผลการทดลองจากการทดสอบที่ 4

รูปที่ 4-7 เซต advanced prediction mode และ Pb-frame mode

รูปที่ 4-8 เซต advanced prediction mode และ syntax-base arithmetic coding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-9 เซต syntax-base arithmetic coding และ PB frame mode

รูปที่ 4-10 เซต unrestricted motion vector mode และ advanced prediction mode

รูปที่ 4-11 เซต unrestricted motion vector mode และ PB frame mode

รูปที่ 4-12 เซต unrestricted motion vector mode และ syntax-base arithmetic coding mode

รูปที่ 4-13 เซต unrestricted motion vector mode และ Syntax-base arithmetic coding mode และ Advanced prediction mode

รูปที่ 4-14 เซต unrestricted motion vector mode และ syntax-base arithmetic coding mode และ PB frame mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-15 เซต *syntax-base arithmetic coding*
และ *Advanced prediction mode*
และ *PB frame mode*

รูปที่ 4-16 เซต *unrestricted motion vector mode*
และ *advanced prediction mode*
และ *advanced prediction mode*

รูปที่ 4-17 เซต *unrestricted motion vector mode*
และ *Syntax-base arithmetic coding mode*
และ *Advanced prediction mode*
และ *PB frame mode*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4. การจัดการหน่วยความจำ

4.4.1 ทดสอบการจัดการหน่วยความจำในส่วน Encoding

Instance Parameters	
File Format	2
Note - ● 2 ในตารางหมายถึง Picture Size 176 x 144	

ตารางที่ 4-5 แสดงการกำหนดค่าพารามิตอร์ที่ใช้ในการประมวลผล

Instance Memory	DARAM		SARAM		External	
	Size	Align	Size	Align	Size [bytes]	Align
Scratch	0	0	0	0	0	0
persistent	0	0	0	0	193001	0

ตารางที่ 4-6 แสดงการใช้งานหน่วยความจำของแต่ละ Instance object ที่ถูกสร้างส่วนเข้ารหัส

Module Memory	Code	
	Size [kb]	Align
h263enc_kmitl.l67	123	0
h263enc.l67	120	0

ตารางที่ 4-7 แสดงขนาดของหน่วยความจำของ Library ที่ใช้งาน

Table Memory	Code	
	Size [bytes]	Align
Constance Table	12590	0

ตารางที่ 4-8 แสดงขนาดหน่วยความจำของตาราง Code Book ที่ใช้งานในการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 ทดสอบการจัดการหน่วยความจำในส่วน Decoding

Instance Memory	DARAM		SARAM		External	
	Size	Align	Size	Align	Size [bytes]	Align
Scratch	0	0	0	0	0	0
persistent	0	0	0	0	295969	0

ตารางที่ 4-9 แสดงการใช้งานหน่วยความจำของแต่ละ Instance object ที่ถูกสร้างส่วนถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิจารณ์และบทสรุป

จากการทำงานและทดลองระบบสามารถสรุปผลที่ได้ดังนี้

1. โปรแกรมสามารถใช้งาน Option ต่างๆ ได้ถูกต้องดังข้างล่าง
   1.1. Unrestricted Motion Vector Mode
   1.2. Syntax-base Arithmetic Coding mode
   1.3. Advance Prediction mode
   1.4. PB-frame mode
2. โปรแกรมที่ได้เป็นไปตามมาตรฐานอัลกอริทึมที่ทาง TI ได้กำหนดไว้ ในส่วนของการเข้ารหัสสัญญาณวิดีโอ
3. โปรแกรมสามารถเข้า/ถอดรหัสสัญญาณวิดีโอมาตรฐาน H.263 ได้
4. โปรแกรมสามารถทำงานรองรับการทำงานแบบ Multi-channel ได้ ในส่วนของการเข้ารหัสสัญญาณวิดีโอ
5. ข้อดีของการใช้งานในแต่ละ option เป็นดังนี้
   5.1. Unrestricted Motion Vector mode จะช่วยลดขนาดความแตกต่างของภาพให้น้อยลงทำให้ได้ file bit stream ที่น้อยลง เมื่อรูปภาพของสัญญาณที่เข้ามาเป็นรูปของสิ่งของที่เคลื่อนไหวด้วยความเร็วสูง ซึ่ง Option นี้สามารถหา Motion vectors ของการเคลื่อนที่นั้นได้
   5.2. Syntax-base Arithmetic Coding mode จะเป็นส่วนที่ช่วยเพิ่มความสามารถของการ encode และ decode เพื่อใช้งานแทนส่วนของการเปิดตารางซึ่งมีค่ารูปแบบที่ตายตัว ซึ่งในส่วน Option นี้ลดจะผลกระทบจากความหลากหลายของข้อมูลรูปภาพทำให้ได้ข้อมูล Stream ที่เหมาะกับรูปแบบของข้อมูลที่ได้เข้ารหัสนั้นๆมากกว่า โดยขนาดข้อมูลที่ได้จากการเข้ารหัสจะเล็ก
   5.3. Advanced Prediction mode สามารถหา block ข้อมูลรูปภาพที่ Overlapped กันได้
   5.4. PB-frame จะช่วยเพิ่มประสิทธิภาพของการเข้ารหัสสัญญาณวิดีโอทำให้สามารถเพิ่ม Frame rate โดยที่บิตเรตของข้อมูลจากการเข้ารหัสไม่มากนัก จากการที่ใช้งาน Header ของข้อมูลที่ส่งไปอันเดียวกัน
6. ข้อเสียของการใช้งาน Option นั่นคือเรื่องของการใช้เวลาในการประมวลผลและค่าความผิดเพี้ยนของรูปภาพที่ได้ ซึ่งในบาง Option ค่าความผิดเพี้ยนอาจมากกว่า หรือบาง Option อาจมีค่าผิดเพี้ยนน้อยกว่า ซึ่งเป็นผลมาจากบาง Option เหมาะกับรูปภาพที่แตกต่างกันไปแล้วแต่สัญญาณที่เข้ามา
7. คุณสมบัติที่ระบบต้องการเป็นไปดังนี้
   7.1. Board ที่ใช้ในการทำงานจะต้องสามารถส่งข้อมูลจากเครื่องคอมพิวเตอร์มาให้กับตัว Board ได้โดยมี Transfer rate ในการส่งข้อมูลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1.1. สัญญาณวิดีโอที่มีขนาดรูปภาพ Sub-QCIF ช่องทางที่ใช้ส่งข้อมูลต้องสามารถส่งได้ 460800 ไบต์ต่อวินาที โดยคิดสัญญาณวิดีโอที่ 25 เฟรมต่อวินาที

7.1.2. สัญญาณวิดีโอที่มีขนาดรูปภาพ QCIF ช่องทางที่ใช้ส่งข้อมูลต้องสามารถส่งได้ 950400 ไบต์ต่อวินาที โดยคิดสัญญาณวิดีโอที่ 25 เฟรมต่อวินาที

7.1.3. สัญญาณวิดีโอที่มีขนาดรูปภาพ CIF ช่องทางที่ใช้ส่งข้อมูลต้องสามารถส่งได้ 3801600 ไบต์ต่อวินาที โดยคิดสัญญาณวิดีโอที่ 25 เฟรมต่อวินาที

7.1.4. สัญญาณวิดีโอที่มีขนาดรูปภาพ 4CIF ช่องทางที่ใช้ส่งข้อมูลต้องสามารถส่งได้ 15206400 ไบต์ต่อวินาที โดยคิดสัญญาณวิดีโอที่ 25 เฟรมต่อวินาที

7.1.5. สัญญาณวิดีโอที่มีขนาดรูปภาพ 16CIF ช่องทางที่ใช้ส่งข้อมูลต้องสามารถส่งได้ 60825600 ไบต์ต่อวินาที โดยคิดสัญญาณวิดีโอที่ 25 เฟรมต่อวินาที

7.2. Board ต้องมีความสามารถในการประมวลผลข้อมูลที่มีขนาดใหญ่ๆได้ (ต้องสามารถดึงข้อมูลจากหน่วยความจำมาให้ ตัวประมวลผลได้อย่างรวดเร็ว)

8. ข้อจำกัดของ Board ที่ใช้ในการทดลองครั้งนี้คือ อัตราในการส่งผ่านข้อมูลมีอัตราที่ค่อนข้างต่ำนั่นคือ สามารถส่งข้อมูลจากเครื่องคอมพิวเตอร์ไปสู่ Board ที่ใช้ในการทดสอบอัลกอริทึมได้ที่ 39 Kbytes/sec ซึ่งน้อยมากที่จะใช้งานกับระบบที่ต้องการส่งข้อมูลมากๆดังเช่นระบบนี้

9. ในส่วนของการถอดรหัสสัญญาณวิดีโอไม่เป็นไปตามมาตรฐานอัลกอริทึมของทาง TI เนื่องจากข้อจำกัดของ Board ที่ใช้งาน ใช้เวลาในการส่งข้อมูลช้า ทำให้ต้องใช้เวลาในการพัฒนาเพิ่มมากขึ้น

ข้อเสนอแนะในการพัฒนาโครงงานนี้ต่อ สำหรับผู้ที่สนใจและนำไปใช้งาน

1) เนื่องจากโครงงานนี้มีการรับส่งข้อมูลขนาดใหญ่ทำให้ประสิทธิภาพในการประมวลผลของ Board ทดลองมีประสิทธิภาพลดลงเป็นอันมาก แนวทางในการแก้ไขปัญหาข้อนี้ อาจแก้ไขได้โดยทำการ Implement ให้มีระบบ DMA ขึ้นมา

2) ตัวโปรแกรมยังไม่มีระบบควบคุมบิตเรตในตอนประมวลผลของส่วนเข้ารหัสสัญญาณข้อมูลเพื่อให้อัตราการส่งข้อมูลในสื่อกลางเป็นไปอย่างมีประสิทธิภาพ

3) โปรแกรมได้ Implement การใช้งาน Option เพียงบางส่วนของมาตรฐาน H.263 ของทาง ITU-T เพื่อให้โปรแกรมสามารถทำงานได้ครบตามมาตรฐาน สำหรับผู้พัฒนาต่อสามารถนำโปรแกรมไปพัฒนาเพิ่ม Option ต่างๆเข้าไปให้สามารถรองรับกับการทำงานตามมาตรฐานได้ทั้งหมด

4) Board ที่ใช้ในการพัฒนานี้ยังไม่เหมาะกับระบบ H.263 โดยทางผู้จัดทำโครงงานนี้ข้อเสนอแนวทางให้กับผู้พัฒนาต่อ หากเป็นไปได้ควรใช้งาน DSP เบอร์ TMS320DM642 ในการทำงานเนื่องจาก DSP ตัวนี้มีพอร์ตที่สามารถต่อเข้ากับกล้องวิดีโอได้โดยตรง

5) ในส่วนของการถอดรหัสสัญญาณให้เป็นไปตามมาตรฐานอัลกอริทึมของทาง TI ซึ่งในส่วนนี้ยังไม่ได้ Implement ในโครงงานนี้

แต่อย่างไรก็ตามโครงงานนี้ สามารถสำเร็จลุล่วงเป็นไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Abhinaba Basu:" Using External References in Algorithms Compliant with the TMS320 DSP Algorithm Standard", Application Report SPRA929A – August 2003

[2] Alan Campbell:" Achieving Zero Overhead With the TMS320 DSP Algorithm Standard IALG Interface"

[3] Carl Bergman:" Using the TMS320 DSP Algorithm Standard in a Dynamic DSP System", Application report SPRA580B

[4] Carl Bergman:" Using the TMS320 DSP Algorithm Standard in a Static DSP System", Application Report SPRA577B

[5] International Telecommunication Union:" Video coding for low bit rate communication", ITU-T Recommendation H.263

[6] Murat Karaoman:" Techniques or Implementing Shared Relocatable Buffers Using the TMS320 DSP Algorithm Standard", Application Report SPRA790 – October 2002

[7] Steve Blonstein และ Maher Katorgi. 2546:"eXpressDSP FOR DUMMIES"

[8] Stig Torud:"A Technic Overview of eXpressDSP-Compliant Algorithms for DSP Software Producers", Application Report SPRA579C – September 2002

[9] TEXAS INSTRUMENTS: "TMS320 DSP Algorithm Standard API Reference", Literature Number SPRU360C September 2002

[10] TEXAS INSTRUMENTS: "TMS320 DSP/BIOS User's Guide", Literature SPRU423B November 2002

[11] www.ti.com

[12] Xiangdong Fu และ Zhaohong Zhang:"A Multichannel/Algorithm Implementation on the TMS320C6000 DSP", Application Report SPRA556B – February 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้