เครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน

นางสาวภัทธีรา อุทัยชนะ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2546
ISBN 974-17-4575-3
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A SIGN LANGUAGE VIDEO ENLARGER FOR HEARING-IMPAIRED

Miss Patheera Uthaichana

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2003
ISBN 974-17-4575-3

หัวข้อวิทยานิพนธ์	เครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน
โดย	นางสาวภัทธีรา อุทัยชนะ
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.เอกชัย ลีลารัศมี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(อาจารย์สุวิทย์ นาคพีระยุทธ)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.เอกชัย ลีลารัศมี)

.....กรรมการ
(อาจารย์ ดร. สุภาวดี อร่ามวิทย์)

.....กรรมการ
(อาจารย์ ดร. สุรีย์ พุ่มรินทร์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภัทธีรา อุทัยชนะ : เครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน. (A SIGN LANGUAGE VIDEO ENLARGER FOR HEARING-IMPAIRED) อ. ที่ปรึกษา : รศ.ดร.เอกชัย ลีลารัศมี, 71 หน้า. ISBN 974-17-4575-3.

วิทยานิพนธ์นี้นำเสนอการออกแบบสิ่งประดิษฐ์ต้นแบบสำหรับใช้งานร่วมกับเครื่องรับโทรทัศน์ที่มีชื่อว่าเครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางได้ยินซึ่งจะช่วยเหลือคนหูหนวกในการรับชมรายการโทรทัศน์ เครื่องนี้จะใช้การประมาณค่าแบบไบลิเนียร์ในการขยายหน้าต่างภาพที่มีขนาดอย่างน้อย 128x128 พิกเซลให้ใหญ่ขึ้นเป็น 4 เท่าหรือ 2.25 เท่าของขนาดเดิม ฮาร์ดแวร์ภายในเครื่องขยายวีดิทัศน์นี้ประกอบด้วยส่วนหลัก 3 ส่วน ส่วนแรกคือ วงจรถอดรหัสและเข้ารหัสสัญญาณวีดิทัศน์สำหรับแปลงสัญญาณวีดีโอรวมในระบบ PAL ไปเป็นสัญญาณแบบดิจิทัลในมาตรฐาน ITU-R BT.656 ที่ความถี่ 27 MByte/s และแปลงกลับมาเพื่อป้อนให้กับเครื่องรับโทรทัศน์ ส่วนที่สองเป็นกลุ่มของหน่วยความจำ 3 ตัวสำหรับเก็บข้อมูลภาพซึ่งมีทั้งภาพที่ขยายไปบางส่วนและขยายเสร็จแล้ว และส่วนที่สามคือ ชิพ FPGA ซึ่งถูกโปรแกรมให้ทำหน้าที่เป็นวงจรควบคุมการขยายภาพ ชิพ FPGA นี้จะนำภาพที่ถูกแปลงเป็นข้อมูลดิจิทัลแล้วมาขยายแบบไบลิเนียร์โดยแบ่งการขยายภาพออกเป็นการขยายภาพทางแนวนอนและทางแนวตั้งซึ่งจะทำงานพร้อมกันแบบไปป์ไลน์และเสร็จภายในเวลา 2 เฟรมภาพหรือ 80 ms นอกจากนี้วงจรนี้ยังทำหน้าที่ถอดรหัสสัญญาณอินฟราเรดของรีโมตคอนโทรลสำหรับให้ผู้ชมสามารถกำหนดตำแหน่งภาพภาษามือและอัตราขยายภาพที่ต้องการได้ ตัวควบคุมนี้ใช้ฟลิปฟลอปไปทั้งหมด 616 ตัว และ LUT 959 ตัวซึ่งได้รวมหน่วยความจำแบบสองพอร์ตขนาด 512 ไบต์จำนวน 6 ตัวแล้ว สิ่งประดิษฐ์นี้ให้ประโยชน์อย่างมากกับกลุ่มคนหูหนวกโดยไม่รบกวนการรับชมรายการของคนธรรมดา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2546

4570457221 : MAJOR ELECTRICAL ENGINEERING

KEY WORD : BILINEAR INTERPOLATION / DIGITAL VIDEO PROCESSING / FPGA / SIGN LANGUAGE IMAGE / VIDEO ENLARGER

PATHEERA UTHAICHANA : A SIGN LANGAUGE VIDEO ENLARGER FOR HEARING-IMPARIED. THESIS ADVISOR : ASSOC. PROF. EKACHAI LEELARASMEE, Ph.D., 71 pp. ISBN 974-17-4575-3.

A prototype TV set top box device, called sign language video enlarger for hearing-impaired, to enhance the TV viewing of deaf people is proposed in this thesis. It used bilinear interpolation to enlarge any screen window of size at least 128x128 pixels by 4 or 2.25 times its area. Its internal hardware consists of three main parts. The first part is a video decoder/encoder for converting a PAL composite video signl into a 27 MByte/s ITU-R BT.656 standard digital format and vice versa. The second part is a group of three memory chips for storing partially/fully expanded images. The third part is an FPGA programmed as an expansion controller. The FPGA implements the bilinear interpolation of the digitized image by splitting into horizontal and vertical expansions that can be pipelinely executed within a two video frame delay or 80 ms. It also functions as an infrared remote control decoder which allows the viewer to locate the sign language window and select its expansion ratio to suit his viewing need. This controller utilizes 616 flip-flops and 959 LUTs which includes six blocks of dual port 512 Byte RAM. It is expected that this device can be highly beneficial to the deaf community without disturbing the viewing of normal persons.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Electrical Engineering Student's signature .....

Field of study Electrical Engineering Advisor's signature .....

Academic year 2003 .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของรองศาสตราจารย์ ดร.เอกชัย ลีลารัศมี อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและให้การสนับสนุนการวิจัยด้วยดีตลอดมา

ขอบคุณจุฬาลงกรณ์มหาวิทยาลัยที่ให้ความรู้ ให้เพื่อนและประสบการณ์ที่ดีมาตลอด 6 ปี

ขอขอบคุณพี่ๆ และน้องๆ ในห้องปฏิบัติการวิจัยการออกแบบและประยุกต์วงจรรวมทุกๆ ท่านทั้งที่อยู่ในสายงานโทรทัศน์และอยู่ในสายงานอื่นๆ ซึ่งได้ให้ความช่วยเหลือ ช่วยรับฟังปัญหาและให้คำแนะนำต่างๆ ด้วยดีมาตลอด 3 ปี

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา ย่า อย่างสูงที่เลี้ยงดู สนับสนุนในด้านการเงิน ให้กำลังใจ ขอบคุณพี่ชาย น้องสาวที่ให้ความช่วยเหลือต่างๆ มาตลอด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
บทที่ 1 บทนำ 1
1.1 แนวเหตุผลในการทำวิทยานิพนธ์ 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 วิธีดำเนินการวิจัย 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2
1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย 3
บทที่ 2 ความรู้พื้นฐาน และปริทัศน์วรรณกรรม 4
2.1 โครงสร้างของสัญญาณวีดิทัศน์ในระบบ PAL 4
2.2 โครงสร้างของสัญญาณวีดิทัศน์แบบดิจิทัล 6
2.2.1 มาตรฐาน ITU-R BT.601 7
2.2.2 มาตรฐาน ITU-R BT.656 8
2.3 วิธีการขยายภาพ 10
2.3.1 วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood) 11
2.3.2 การประมาณค่าแบบไบลิเนียร์ (Bilinear interpolation) 13
2.3.3 การประมาณค่าแบบไบคิวบิก (Bicubic interpolation) 17
2.4 การขยายภาพบนสัญญาณวีดิทัศน์แบบดิจิทัล 20
2.5 รูปแบบข้อมูลอินฟราเรดของรีโมตคอนโทรลสำหรับโทรทัศน์ 22
2.5.1 การเข้ารหัสแบบสองเฟส 22
2.5.2 การมอดูเลชั่นโดยใช้ระยะทางระหว่างพัลส์ (Pulse-distance modulation) 22
2.5.3 การเข้ารหัสโดยใช้ความกว้างของพัลส์ (Pulse-length code) 23
2.6 การติดต่อแบบ I^2C 23

สารบัญตาราง

หน้า

บทที่ 3 การออกแบบเครื่องขยายวีดิทัศน์ 27
3.1 วงจรถอดรหัสสัญญาณวีดิทัศน์ (Video decoder) 29
3.2 วงจรเข้ารหัสสัญญาณวีดิทัศน์ (Video encoder) 31
3.3 วงจรรับสัญญาณอินฟราเรด (IR receiver module) 33
บทที่ 4 วงจรควบคุมการขยายภาพ 35
4.1 วงจรขยายภาพ 38
4.1.1 วงจรขยายภาพทางแนวนอน (Horizontal expander) 38
4.1.2 วงจรขยายภาพทางแนวตั้ง (Vertical expander) 41
4.2 หน่วยควบคุม (Control unit) 42
4.2.1 ตัวส่งสัญญาณผ่าน I^2C (I^2C transmitter) 42
4.2.2 วงจรประมวลผลสัญญาณรีโมต (Remote processor) 45
4.2.3 วงจรสร้างสัญญาณควบคุม (Control signal generator) 48
4.2.4 วงจรนับสำหรับสร้างค่าแอดเดรส (Address counter) 49
4.3 หน่วยความจำ 50
บทที่ 5 การทดสอบ และสรุปผล 52
5.1 การทดสอบการทำงานของเครื่องขยายวีดิทัศน์ 52
5.2 ปัญหาในการทำงาน 55
5.3 สรุป 55
5.4 ข้อเสนอแนะ 56
รายการอ้างอิง 57
ภาคผนวก 58
ภาคผนวก ก การออกแบบวงจรด้วยภาษาวีเอชดีแอล 59
ภาคผนวก ข บทความที่ได้รับการตีพิมพ์ใน ECTI-CON 2004 66
ประวัติผู้เขียนวิทยานิพนธ์ 71

สารบัญตาราง

หน้า

ตารางที่ 4-1 แสดงการทำงานของตัวสวิตช์ 37
ตารางที่ 4-2 แสดงลำดับการทำงานของหน่วยความจำ 51

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1-1 การใช้งานของเครื่องขยายวีดิทัศน์ 1
รูปที่ 2-1 การแสดงภาพบนจอโทรทัศน์ 4
รูปที่ 2-2 ส่วนประกอบของสัญญาณวีดิทัศน์ 1 เส้น 5
รูปที่ 2-3 สัญญาณวีดิทัศน์ช่วงเปลี่ยนฟิลด์ 6
รูปที่ 2-4 รูปแบบของสัญญาณองค์ประกอบต่างๆ ในแบบความถี่ 13.5 MHz 8
รูปที่ 2-5 รูปแบบการส่งสัญญาณวีดิทัศน์ตามมาตรฐาน ITU-R BT.656 9
รูปที่ 2-6 การขยายภาพขนาด kxk เท่า 10
รูปที่ 2-7 การชักตัวอย่างซ้ำเพื่อขยายภาพในแนวนอนขนาด k เท่า เมื่อ k = 4 10
รูปที่ 2-8 การหาค่าที่ตำแหน่งพิกเซล p(m+y,n+x) ซึ่งคำนวณจากพิกเซลที่อยู่ใกล้ที่สุด
เมื่อขยายภาพแบบประมาณค่าจากค่าที่อยู่ใกล้ที่สุด 11
รูปที่ 2-9 การขยายภาพขนาด 4 เท่าโดยประมาณค่าจากค่าที่อยู่ใกล้ที่สุด 12
รูปที่ 2-10 ภาพที่ขยาย 2X2 เท่าโดยวิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด 13
รูปที่ 2-11 การหาค่าพิกเซล p(m+y,n+x) จากพิกเซลรอบๆ 4 พิกเซล
ด้วยวิธีไบลิเนียร์ 13
รูปที่ 2-12 การขยายภาพขนาด 4 เท่าด้วยวิธีไบลิเนียร์ 14
รูปที่ 2-13 การขยายภาพขนาด 2.25 เท่าด้วยวิธีไบลิเนียร์ 16
รูปที่ 2-14 ภาพที่ได้จากการขยายโดยการประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (ภาพซ้ายมือ) เปรียบ
เทียบกับภาพที่ได้จากการขยายแบบไบลิเนียร์ (ภาพขวามือ) 17
รูปที่ 2-15 การหาค่า p(m+y,n+x) จากพิกเซลรอบๆ 16 พิกเซลด้วยวิธีไบคิวบิก 18
รูปที่ 2-16 ภาพจากการขยายด้วยวิธีไบคิวบิก (ภาพด้านขวา) เปรียบเทียบกับวิธีไบลิเนียร์
(ภาพด้านซ้าย) 19
รูปที่ 2-17 การสุ่มสัญญาณ Y, U, V เพื่อขยายข้อมูลทางแนวนอน 2 เท่า 20
รูปที่ 2-18 การสุ่มสัญญาณ Y, U, V เพื่อขยายข้อมูลในแนวนอน 1.5 เท่า 21
รูปที่ 2-19 รูปแบบของสัญญาณที่เข้ารหัสด้วยวิธีสองเฟส 22
รูปที่ 2-20 การส่งข้อมูลที่มอดูเลตโดยใช้ระยะทางระหว่างพัลส์ 22
รูปที่ 2-21 การเข้ารหัสโดยใช้ความกว้างของพัลส์ 23
รูปที่ 2-22 บิตเริ่มต้นและบิตจบการทำงาน ของ I^2C 24
รูปที่ 2-23 ลักษณะการส่งข้อมูลของ I^2C 24

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 2-24 รูปแบบของการส่งสัญญาณโดยวงจรแม่เป็นตัวส่ง 25
รูปที่ 2-25 รูปแบบการส่งข้อมูลโดยวงจรแม่เป็นตัวอ่าน 26
รูปที่ 2-26 รูปแบบการส่งข้อมูลแบบ combined format 26
รูปที่ 3-1 โครงสร้างของเครื่องขยายวีดิทัศน์ภาษามือ 27
รูปที่ 3-2 การขยายภาพซึ่งแบ่งเป็น 2 ขั้นตอน 28
รูปที่ 3-3 การใช้งานวงจรถอดรหัสสัญญาณวีดิทัศน์ 29
รูปที่ 3-4 โครงสร้างภายในวงจรถอดรหัสสัญญาณวีดิทัศน์ 30
รูปที่ 3-5 การใช้งานวงจรเข้ารหัสสัญญาณวีดิทัศน์ 31
รูปที่ 3-6 โครงสร้างภายในวงจรเข้ารหัสสัญญาณวีดิทัศน์ 32
รูปที่ 3-7 การใช้งานวงจรรับสัญญาณอินฟราเรด 33
รูปที่ 3-8 การทำงานภายในวงจรรับสัญญาณอินฟราเรด 34
รูปที่ 4-1 โครงสร้างวงจรภายในของวงจรควบคุมการขยายภาพ 35
รูปที่ 4- 2 การทำงานของตัวสวิตช์ในช่วง 3 เฟรมภาพติดต่อกัน V แทนภาพต้นฉบับทั้งภาพ S แทนส่วนที่เป็นภาพภาษามือ HV แทนการขยายภาพทางแนวนอนและทางแนวตั้งตามลำดับ 36
รูปที่ 4-3 วงจรขยายภาพทางแนวนอน 38
รูปที่ 4-4 โครงสร้างรีจิสเตอร์แบงค์ 1 39
รูปที่ 4-5 โครงสร้างรีจิสเตอร์แบงค์ 2 39
รูปที่ 4-6 วงจรประมาณค่า (Interpolator) 40
รูปที่ 4-7 วงจรขยายภาพทางแนวตั้ง 41
รูปที่ 4-8 หน่วยควบคุม 42
รูปที่ 4-9 ตัวส่งสัญญาณผ่าน I^2C 42
รูปที่ 4-10 แผนผังสถานะของหน่วยควบคุม I^2C 44
รูปที่ 4-11 วงจรประมวลผลสัญญาณรีโมต 45
รูปที่ 4-12 รูปแบบของสัญญาณรีโมต 45
รูปที่ 4-13 แผนผังสถานะของหน่วยควบคุมรีโมต 46
รูปที่ 4-14 วงจรสร้างสัญญาณควบคุม 48
รูปที่ 4-15 วงจรนับสำหรับสร้างค่าแอดเดรส 49
รูปที่ 4-16 การทำงานของหน่วยความจำ 50

สารบัญภาพ (ต่อ)

หน้า

รูปที่ 5-1 เครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน. 52
รูปที่ 5-2 ภาพรายการต้นฉบับที่ยังไม่ได้ขยาย 53
รูปที่ 5-3 ภาพรายการเมื่อทำงานขยายภาพขนาด 2.25 เท่า 54
รูปที่ 5-4 ภาพรายการเมื่อทำงานขยายภาพขนาด 4 เท่า 54
รูปที่ 5-5 ภาพรายการเมื่อเลือกเปลี่ยนตำแหน่งการแสดงภาพ 55

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

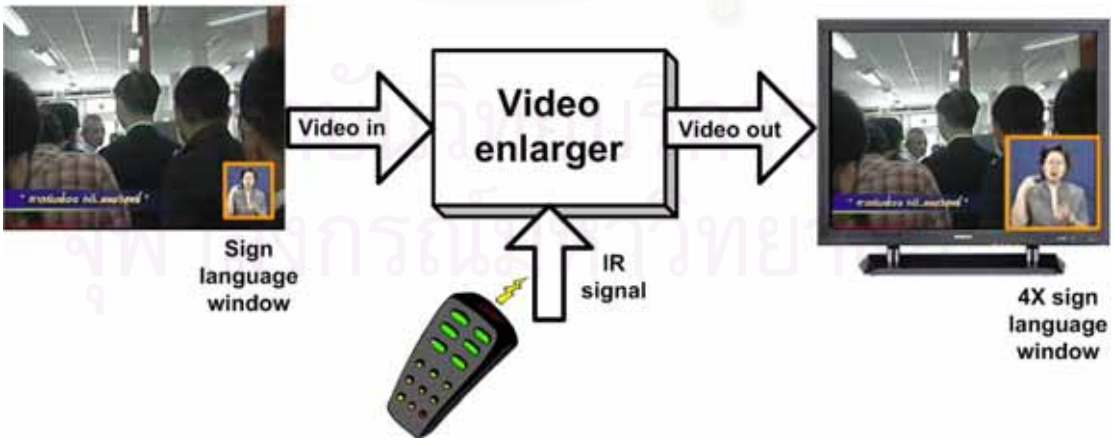
บทที่ 1

บทนำ

1.1 แนวเหตุผลในการทำวิทยานิพนธ์

สังคมไทยนี้มีจำนวนผู้พิการอยู่ไม่น้อยที่จะต้องดำเนินกิจกรรมต่างๆ ในชีวิตประจำวันอย่างลำบาก ในบางครั้งต้องใช้อุปกรณ์และเทคโนโลยีต่างๆ เพื่อช่วยเพิ่มความสะดวกมากขึ้น ซึ่งหนึ่งในตัวอย่างของสื่อที่ช่วยผู้พิการทางการได้ยิน ได้แก่ การแทรกภาพภาษามือมากับรายการโทรทัศน์เพื่อให้คนหูหนวกสามารถเข้าใจในเนื้อหาของรายการนั้นได้ เช่น รายการข่าวทางสถานีโทรทัศน์ช่อง 11 เป็นต้น แต่การแทรกภาพภาษามือมากับรายการโทรทัศน์จะทำให้ภาพหลักของรายการถูกรบกวนไปบางส่วนซึ่งจะสร้างความรำคาญให้กับคนปรกติได้ ดังนั้นขนาดของภาพภาษามือที่ส่งมากับรายการจึงถูกบีบให้มีขนาดเล็กมาก ทำให้คนหูหนวกไม่สามารถอ่านปากและดูภาษามือของผู้แปลได้ชัดเจน การเรียนรู้ต่างๆ จากรายการโทรทัศน์ก็จะเป็นไปอย่างลำบาก

งานวิจัยนี้เกิดจากแนวคิดเพื่อแก้ไขปัญหานี้ให้กับผู้พิการทางการได้ยิน โดยการสร้างเครื่องขยายวีดิทัศน์ภาษามือขึ้นซึ่งจะขยายหน้าต่างของภาพภาษามือเดิมที่มีขนาดเล็กดังรูปด้านซ้ายมือของรูปที่ 1-1 ให้มีขนาดใหญ่ขึ้นดังรูปด้านขวามือของรูปที่ 1-1 ดังนั้นผู้พิการทางการได้ยินก็จะสามารถชมภาพภาษามือได้ชัดเจนขึ้นและเข้าใจเนื้อหารายการโทรทัศน์ได้ดีขึ้น



รูปที่ 1-1 การใช้งานของเครื่องขยายวีดิทัศน์

เครื่องขยายวีดิทัศน์จะรับสัญญาณวีดิทัศน์จากภายนอกเข้ามา แล้วแยกเฉพาะข้อมูลส่วนที่เป็นภาพภาษามือออกมาเพื่อขยายให้มีขนาดใหญ่ขึ้น จากนั้นจึงนำข้อมูลที่ได้ขยายแล้วนี้ไปแทรกในข้อมูลภาพเดิมที่มีอยู่ แล้วส่งสัญญาณวีดิทัศน์ที่ได้นี้ไปแสดงภาพบนเครื่องรับโทรทัศน์ ซึ่งจะได้ภาพเป็นภาพรายการเดิมแต่มีภาพภาษามือขนาดใหญ่ขึ้นนั่นเอง

1.2 วัตถุประสงค์ของการวิจัย

1. ออกแบบและพัฒนาเครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน เพื่อช่วยขยายภาพภาษามือให้มีขนาดใหญ่ขึ้น

1.3 ขอบเขตของการวิจัย

1. สร้างต้นแบบของเครื่องขยายภาพวีดิทัศน์ภาษามือที่มีคุณสมบัติดังนี้
   a. ขยายภาพภาษามือให้มีขนาดใหญ่ขึ้นได้ โดยใช้วิธีขยายแบบไบลีเนียร์
   b. เลือกอัตราขยายได้ 2 แบบ คือ แบบ 2.25 เท่า หรือแบบ 4 เท่า
   c. เปลี่ยนตำแหน่งภาพที่จะขยายได้แม้จะเป็นตำแหน่งที่ไม่มีภาพภาษามือก็ตาม
   d. ควบคุมการทำงานของเครื่องผ่านรีโมตคอนโทรลได้

1.4 วิธีดำเนินการวิจัย

1. ศึกษากระบวนการขยายภาพแบบต่างๆ และรูปสัญญาณวีดิทัศน์แบบดิจิทัล
2. ออกแบบและพัฒนาเครื่องขยายวีดิทัศน์โดยใช้วิธีขยายภาพโดยประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood)
3. พัฒนาเครื่องขยายวีดิทัศน์ให้ขยายภาพได้ด้วยวิธีประมาณค่าแบบไบลิเนียร์
4. พัฒนาเครื่องขยายวีดิทัศน์ให้สามารถขยายภาพได้ 2 ขนาด คือ 2.25 เท่าและ 4 เท่า
5. ทดสอบการทำงาน และปรับปรุงอุปกรณ์ต้นแบบ
6. สรุปผลการทดลอง และเขียนวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยให้คนพิการทางการได้ยินสามารถเข้าใจเนื้อหาของรายการโทรทัศน์ได้ดีขึ้น
2. เกิดความรู้ความเข้าใจเกี่ยวกับระบบของสัญญาณวีดิทัศน์แบบดิจิทัลมากขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 5 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขต และวิธีดำเนินงานวิจัย รวมทั้งประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 จะสรุปแนวความคิด ทฤษฎีที่เกี่ยวข้องกับการวิจัย เช่น รูปแบบของสัญญาณโทรทัศน์และวิธีการขยายภาพ เป็นต้น ส่วนการออกแบบเครื่องขยายวีดิทัศน์จะกล่าวถึงในเนื้อหาบทที่ 3 ในขณะที่วงจรควบคุมการขยายภาพซึ่งใช้ควบคุมการทำงานของเครื่องขยายวีดิทัศน์จะกล่าวแยกไว้ในเนื้อหาบทที่ 4 ซึ่งจะมีรายละเอียดเกี่ยวกับโครงสร้างและการทำงานของวงจรนี้ ในบทสุดท้ายจะเสนอในเรื่องผลการทดสอบ รวมทั้งบทสรุปงานวิจัย และข้อเสนอแนะ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

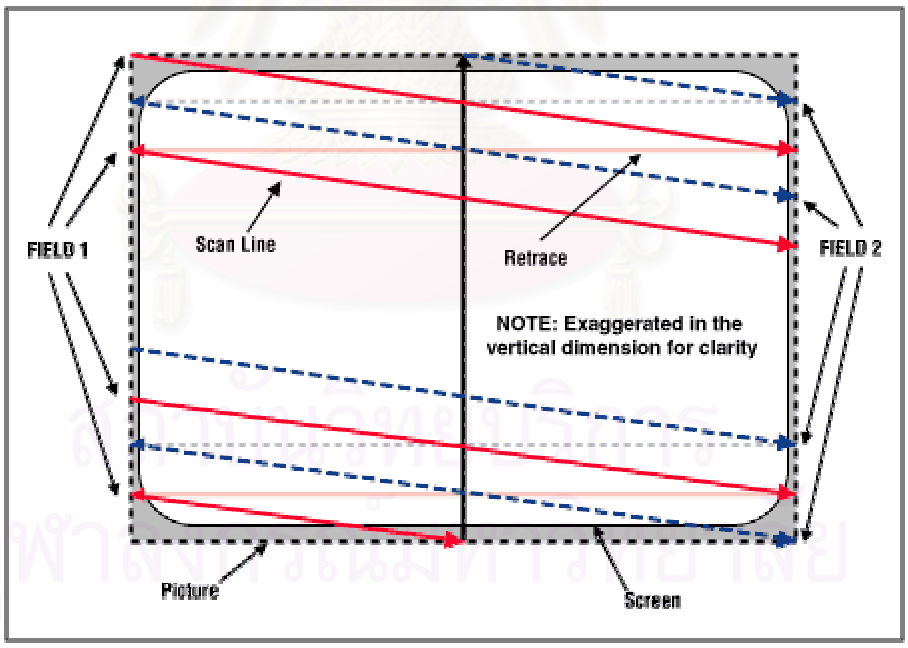
บทที่ 2

ความรู้พื้นฐาน และปริทัศน์วรรณกรรม

เครื่องขยายวีดิทัศน์ภาษามือนี้ใช้หลักการที่เกี่ยวข้องในหลายๆ เรื่อง เช่น รูปแบบของสัญญาณวีดิทัศน์ในระบบ PAL ซึ่งเป็นระบบที่วงการโทรทัศน์ในเมืองไทยเลือกใช้ ลักษณะและรูปแบบสัญญาณวีดิทัศน์แบบดิจิทัลตามมาตรฐาน ITU-R BT.601 และ ITU-R BT.656 โดยทั้งสองมาตรฐานนี้เป็นที่ยอมรับกันมากในปัจจุบัน และสมการและวิธีขยายภาพที่ได้รับความนิยมกันหลายๆ วิธี เป็นต้น เนื้อหาของหลักการทั้งหมดจะกล่าวถึงต่อไปในบทนี้

2.1 โครงสร้างของสัญญาณวีดิทัศน์ในระบบ PAL

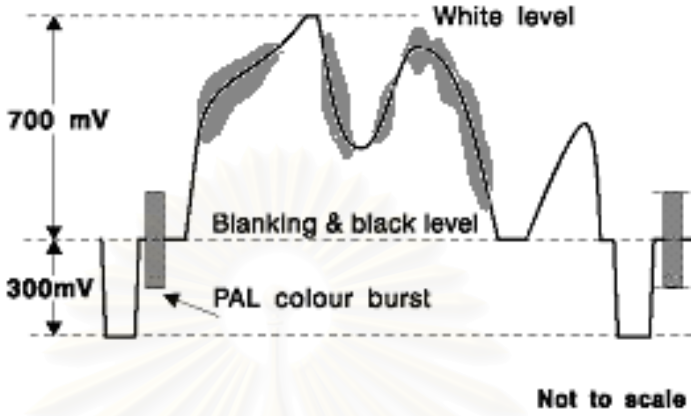
ภาพที่เคลื่อนไหวได้บนจอโทรทัศน์นั้นเกิดจากการแสดงภาพนิ่งทั้งหมด 25 ภาพใน 1 วินาที ซึ่งการแสดงภาพบนจอโทรทัศน์จะเกิดจากการกวาดอิเล็กตรอนจากมุมบนซ้ายมือมาด้านขวามือทีละเส้นๆ จนกระทั่งถึงมุมด้านล่างขวาดังแสดงในรูปที่ 2-1



รูปที่ 2-1 การแสดงภาพบนจอโทรทัศน์

ภาพที่แสดงบนจอโทรทัศน์ทั้งภาพจะเรียกแทนว่า เฟรม โดย 1 เฟรมภาพจะประกอบด้วยเส้นภาพทั้งหมด 625 เส้น การแสดงภาพบนโทรทัศน์จะเป็นแบบสอดประสาน (Interlace) คือ ภาพ 1 ภาพจะถูกแบ่งออกเป็น 2 ฟิลด์ คือ ฟิลด์คี่ (ฟิลด์ที่ 1) ซึ่งจะแสดงเส้นภาพเส้นทึบในรูปที่

2-1 ส่วนอีกฟิลด์หนึ่งจะเป็นฟิลด์คู่ซึ่งแสดงเส้นภาพที่เป็นเส้นประในรูปที่ 2-1 ทำให้เส้นภาพ 2 เส้นที่อยู่ติดกันจะถูกส่งออกมาคนละฟิลด์ภาพ ภาพทั้งหมด 1 ฟิลด์ภาพจึงประกอบด้วยเส้นภาพทั้งหมด 312.5 เส้น



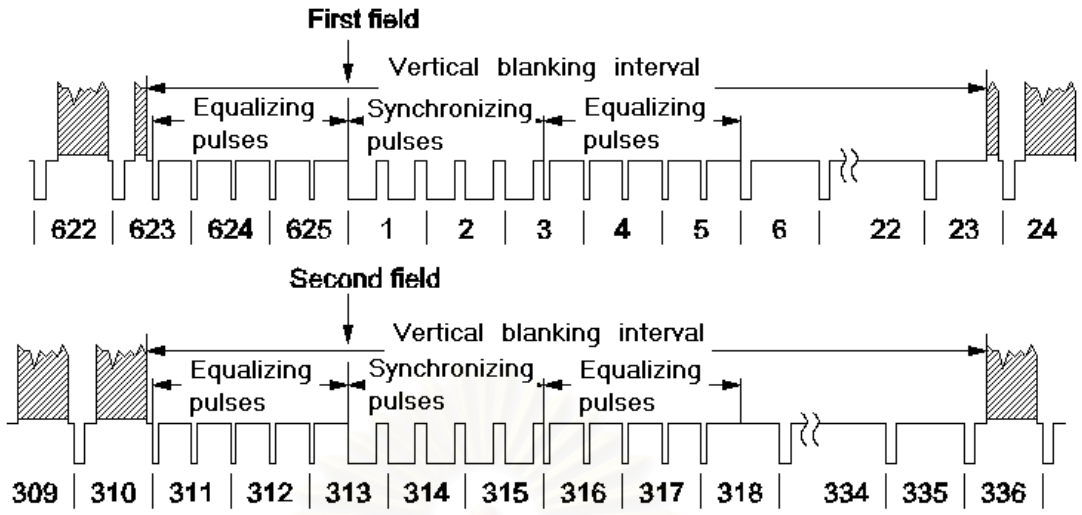
รูปที่ 2-2 ส่วนประกอบของสัญญาณวีดิทัศน์ 1 เส้น

รูปที่ 2-2 แสดงรายละเอียดของสัญญาณวีดิทัศน์ 1 เส้น ซึ่งประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

1. สัญญาณซิงโครไนซ์ทางแนวราบ (horizontal sync) หรือสัญญาณซิงโครไนซ์ของเส้นภาพจะใช้บอกตำแหน่งเริ่มต้นของเส้นภาพและเป็นช่วงเวลาที่ลำอิเล็กตรอนในจอโทรทัศน์จะเคลื่อนย้ายจากด้านขวามือกลับมาที่ด้านซ้ายมือของจอใหม่เพื่อรอจังหวะการแสดงภาพในเส้นใหม่

2. เบิรสต์สี (color burst) ใช้บอกเฟสอ้างอิงของสัญญาณสี (chrominance) ที่ถูกมอดูเลตเชิงเฟสด้วยพาหะรองความถี่ 4.43 MHz สัญญาณเบิรสต์สีที่ส่งมานี้จะถูกใช้ในเครื่องรับโทรทัศน์สำหรับสร้างสัญญาณความถี่ 4.43 MHz คืนมาและใช้เทียบเป็นเฟสอ้างอิงในการดีมอดูเลตของวงจรแยกองค์ประกอบสัญญาณสี

3. สัญญาณภาพจริง ประกอบด้วย 2 ส่วนคือ สัญญาณความสว่าง (luminance) และสัญญาณสี (chrominance) สัญญาณสีมีลักษณะเป็นสัญญาณความถี่สูง (4.43 MHz) ประกอบอยู่บนสัญญาณความสว่างอีกทีหนึ่งดังที่เห็นเป็นแถบหนาบนสัญญาณความสว่างดังในรูปที่ 2-2



รูปที่ 2-3 สัญญาณวีดิทัศน์ช่วงเปลี่ยนฟิลด์

เมื่อจบการแสดงภาพ 1 ฟิลด์ก็จะมีสัญญาณช่วงไร้ภาพทางแนวตั้งเกิดขึ้นซึ่งมีรูปแบบดังรูปที่ 2-3 เพื่อให้อิเล็กตรอนสามารถวิ่งจากตำแหน่งมุมล่างขวามือกลับไปด้านซ้ายบนของจอภาพได้ทัน สัญญาณวีดิทัศน์ในช่วงไร้ภาพทางแนวตั้งนี้จะแบ่งออกเป็นช่วงต่างๆ ดังนี้

1. *สัญญาณอีควอไลซิ่งพัลส์ (Equalizing pulses)* สัญญาณนี้จะมีอยู่ 2 ช่วง คือ ก่อนและหลังสัญญาณซิงโครไนซ์พัลส์ โดยจะถูกใช้ร่วมกับสัญญาณซิงโครไนซ์พัลส์เพื่อให้วงจรอินทิเกรตภายในเครื่องรับโทรทัศน์สามารถนำไปสร้างเป็นสัญญาณซิงค์แนวตั้งได้ เนื่องจากสัญญาณจาก 3 ช่วงนี้จะมีระดับไฟตรงเฉลี่ยต่างกัน โดยช่วงสัญญาณอีควอไลซิ่งนี้จะทำให้สัญญาณขาออกที่ได้จากวงจรอินทิเกรตมีระดับไฟตรงเฉลี่ยสูง

2. *สัญญาณซิงโครไนซ์พัลส์ (Synchronizing pulses)* สัญญาณช่วงนี้จะมีค่าไฟตรงเฉลี่ยต่ำทำให้สัญญาณออกจากวงจรอินทิเกรตมีค่าต่ำลงจนถึงระดับที่วงจรตรวจสอบซิงค์แนวตั้งตรวจพบได้และสร้างเป็นสัญญาณซิงค์ส่งไปควบคุมวงจรอื่นๆ ต่อไป

3. *สัญญาณวีดิทัศน์ช่วงไร้ภาพทางแนวตั้ง* ช่วงรอยต่อระหว่างสัญญาณอีควอไลซิ่งจนถึงสัญญาณภาพจริงๆ จะมีสัญญาณวีดิทัศน์ว่างซึ่งไม่มีข้อมูลภาพอยู่เลยอีกประมาณ 13 เส้นต่อฟิลด์ ช่วงสัญญาณนี้ในบางประเทศจะแทรกข้อมูลพิเศษต่างๆ ลงไป เช่น คำบรรยายภาพแบบซ่อนได้ (closed caption) สัญญาณภาพทดสอบ เป็นต้น

2.2 โครงสร้างของสัญญาณวีดิทัศน์แบบดิจิทัล

มาตรฐานที่เกี่ยวกับสัญญาณวีดิทัศน์แบบดิจิทัลที่ได้รับความนิยมกันจะมีอยู่ 2 มาตรฐานหลัก คือ ITU-R BT.601 [5] และ ITU-R BT.656 [6] ซึ่งจะกล่าวถึงรูปแบบในการแปลงสัญญาณ

จากแบบแอนะล็อกมาเป็นแบบดิจิทัลและรูปแบบในการส่งสัญญาณเพื่อให้ผู้รับและผู้ส่งสามารถเข้าใจได้ตรงกัน รายละเอียดของทั้งสองมาตรฐานนี้มีดังต่อไปนี้

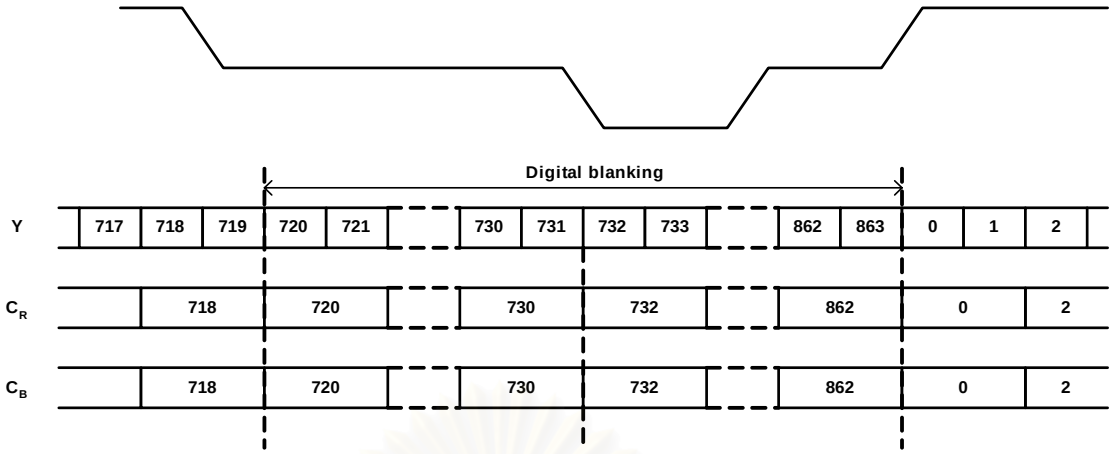
2.2.1 **มาตรฐาน** ITU-R BT.601

มาตรฐานของ ITU-R BT.601 จะกล่าวถึงรูปแบบของสัญญาณวีดิทัศน์แบบดิจิทัลซึ่งแปลงมาจากสัญญาณแอนะล็อก เนื้อหาของมาตรฐานนี้จะแยกออกเป็น 2 ส่วนตามความถี่ในการชักตัวอย่างของสัญญาณวีดิทัศน์ คือ ความถี่ 13.5 MHz และความถี่ 18 MHz โดยความถี่แรกจะใช้กับจอโทรทัศน์แบบทั่วๆ ไปซึ่งมีสัดส่วนความกว้างต่อความยาวเท่ากับ 4:3 ส่วนจอแบบกว้าง (wide screen) ที่มีสัดส่วนจอเท่ากับ 16:9 จะใช้ที่ความถี่สูงกว่า คือ 18 MHz ซึ่งคำนวณได้จาก 13.5Mx(16/9)x(3/4) จอแบบนี้จะใช้ข้อมูลในแต่ละเส้นมากกว่าจึงต้องส่งข้อมูลที่ความถี่สูงกว่า แต่ในงานวิจัยนี้จะประยุกต์ใช้งานกับจอทั่วๆ ไป คือ ความถี่ 13.5 MHz และใช้งานกับระบบโทรทัศน์แบบ PAL เท่านั้น จึงขอสรุปเนื้อหาเฉพาะส่วนที่เกี่ยวข้องดังต่อไปนี้

สัญญาณวีดิทัศน์จะถูกแบ่งออกเป็น 3 องค์ประกอบ คือ สัญญาณความสว่าง (Y) และสัญญาณสี 2 สัญญาณ (C_R, C_B) โดยข้อมูลของสัญญาณแต่ละสัญญาณจะถูกแปลงให้มีขนาด 8 บิต หรือสามารถแทนค่าได้ทั้งหมด 256 ระดับ คือตั้งแต่ค่า 0 จนถึงค่า 255 ในเลขฐานสิบ แต่ในการส่งจริงๆ แล้วจะสงวนค่าที่ระดับ 0 และ 255 ไว้เพื่อใช้กับข้อมูลสัญญาณซิงโครไนซ์โดยเฉพาะ ส่วนสัญญาณวีดิทัศน์จะแทนด้วยค่า 1 ถึง 254 เท่านั้น

มาตรฐานนี้กำหนดให้สัญญาณความสว่างแทนค่าได้ทั้งหมดเพียง 220 ระดับ โดยค่าต่ำสุดที่เป็นไปได้หรือสีดำจะแทนด้วย 16 ส่วนค่าสูงสุดหรือสีขาวแทนด้วยค่า 235 ส่วนสัญญาณสีจะใช้ค่าได้ทั้งหมด 225 ระดับ คือ ตั้งแต่ค่า 16 จนถึง 240 ซึ่งจะใช้แทนค่าทั้งค่าบวกและลบ โดยระดับสัญญาณที่ 0 จะแทนด้วยค่า 128

สัญญาณในแต่ละเส้นจะประกอบด้วยส่วนที่เป็นสัญญาณซิงโครไนซ์ และส่วนที่เป็นสัญญาณจริง (active line) โดยรูปแบบของสัญญาณในองค์ประกอบต่างๆ แสดงได้ดังรูปที่ 2-4 โดยตัวเลขในกรอบสี่เหลี่ยมแทนลำดับตัวอย่างที่ชักในแต่ละเส้น



รูปที่ 2-4 รูปแบบของสัญญาณองค์ประกอบต่างๆ ในแบบความถี่ 13.5 MHz

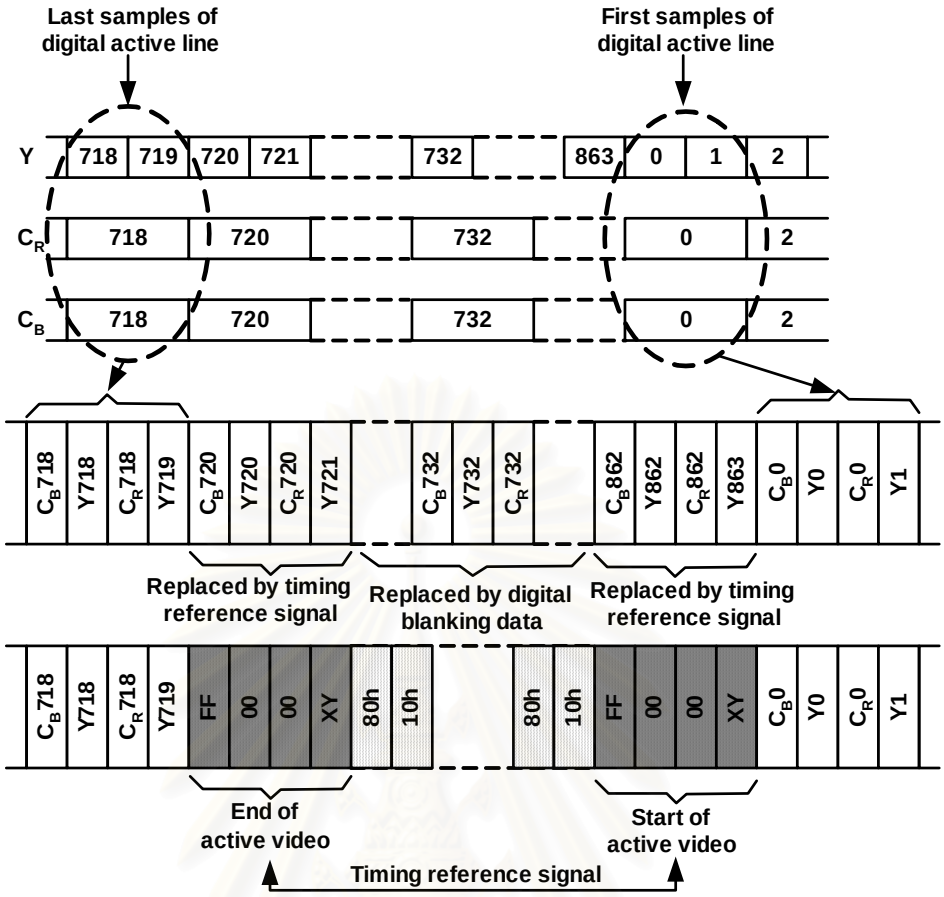
- ใน 1 เส้นภาพจะประกอบด้วย สัญญาณความสว่าง 864 ตัวอย่างและสัญญาณสีอย่างละ 432 ตัวอย่าง แต่จำนวนตัวอย่างที่อยู่ในสัญญาณจริงของสัญญาณความสว่างนั้นจะเหลือ 720 ตัวอย่าง และของสัญญาณโครมิแนนซ์มีอย่างละ 360 ตัวอย่าง

- ความถี่ในการชักตัวอย่างของสัญญาณความสว่างเท่ากับ 13.5 MHz ส่วนสัญญาณสีจะถูกชักตัวอย่างอย่างละ 6.75 MHz

จากเนื้อหาดังที่กล่าวมาจะเห็นว่ามาตรฐาน ITU-R BT.601 นี้จะกล่าวเฉพาะองค์ประกอบของสัญญาณวีดิทัศน์แบบดิจิทัลซึ่งแปลงมาจากสัญญาณวีดิทัศน์แบบเดิมที่เป็นสัญญาณแอนะล็อก แต่จะไม่กล่าวถึงการจัดรูปแบบของสัญญาณที่ได้ออกมาว่าจะมีลำดับการส่งอย่างไรบ้าง เนื้อหาในส่วนนี้ได้ถูกแยกไปเป็นมาตรฐานอีกอันหนึ่งเลย คือ ITU-R BT.656 ซึ่งจะมีรายละเอียดดังต่อไปนี้

2.2.2 **มาตรฐาน** ITU-R BT.656

มาตรฐาน ITU-R BT.656 จะกล่าวถึงรูปแบบการส่งสัญญาณวีดิทัศน์ที่แปลงมาเป็นสัญญาณดิจิทัลตามมาตรฐานของ ITU-R BT.601 ซึ่งได้ถูกแยกออกมาเป็นสัญญาณความสว่างและสัญญาณสีแล้ว ลำดับในการส่งสัญญาณทั้งสามองค์ประกอบสามารถแสดงได้ดังรูปที่ 2-5



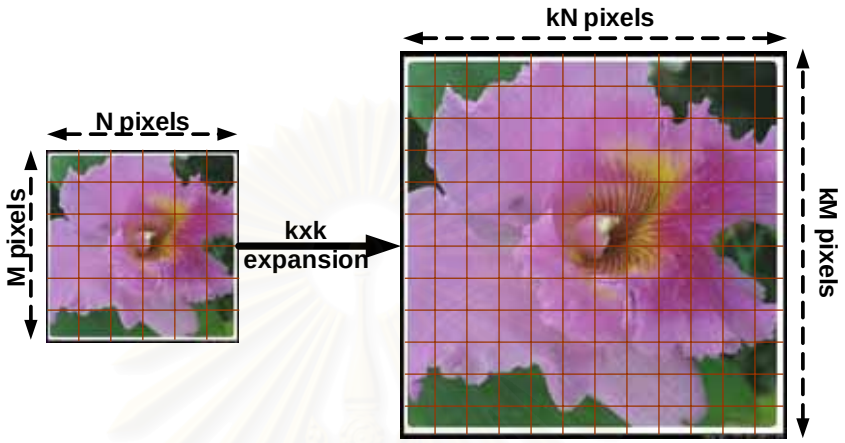
รูปที่ 2-5 รูปแบบการส่งสัญญาณวีดิทัศน์ตามมาตรฐาน ITU-R BT.656

รูปแบบการส่งสัญญาณวีดิทัศน์ที่แสดงนี้เป็นแบบขนานและมีขนาดของข้อมูลแต่ละตัวเท่ากับ 8 บิต การส่งสัญญาณทั้งสามองค์ประกอบนี้จะส่งมาทีละองค์ประกอบสลับกันไปด้วยสัดส่วน 4:2:2 ซึ่งเป็นสัดส่วนของจำนวนข้อมูลสัญญาณ $Y:C_B:C_R$ โดยข้อมูลสัญญาณวีดิทัศน์ทั้งหมดจะถูกส่งมาที่ความถี่ 27 MHz (สัญญาณความสว่าง 13.5 MHz และสัญญาณสีอย่างละ 6.75 MHz)

สัญญาณภาพในแต่ละเส้นประกอบด้วยสัญญาณจริงและช่วงสัญญาณไร้ภาพทางแนวนอน ในช่วงสัญญาณจริงนั้นข้อมูลจะทยอยส่งออกมาทีละประเภทด้วยลำดับดังนี้ คือ C_B, Y, C_R, Y, C_B, Y, C_R ส่วนสัญญาณไร้ภาพทางแนวนอนจะมีช่วงที่เรียกว่าสัญญาณรหัสเวลาอ้างอิง (timing reference code) อยู่ 2 ช่วง คือ ช่วงเริ่มต้นสัญญาณภาพ (SAV) และช่วงที่จบสัญญาณภาพ (EAV) ซึ่งจะประกอบด้วยข้อมูลทั้งหมด 4 ไบต์ คือ FF 00 00 XY ตามลำดับ โดย XY จะเป็นรหัสที่บอกรายละเอียดว่าเส้นภาพนี้อยู่ในฟิลด์คู่หรือคี่ อยู่ในช่วงซิงค์แนวตั้งหรือไม่ เป็นสัญญาณ SAV หรือ EAV และมีบิตสำหรับตรวจสอบความผิดพลาดของข้อมูลที่ส่งมา ส่วนข้อมูลที่เหลือในช่วงสัญญาณไร้ภาพจะถูกเติมด้วยค่า 80h, 10h, 80h, 10h สลับกันไปเรื่อยๆ จนจบ

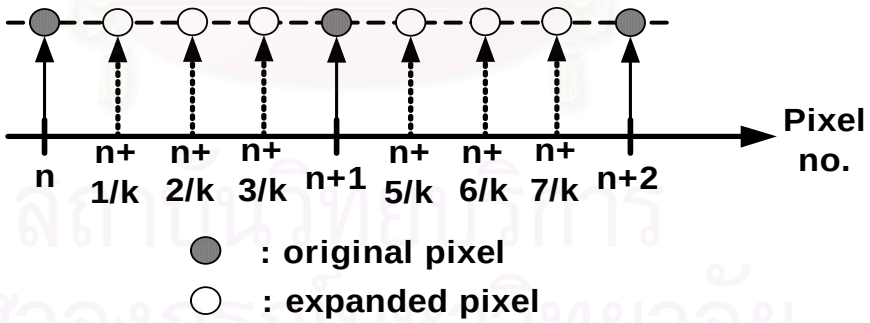
จากเนื้อหาที่กล่าวมาจะเห็นว่ารูปแบบของสัญญาณวีดิทัศน์ในระบบดิจิทัลจะมีช่วงเวลาต่างๆ ตรงกับสัญญาณในแบบแอนะล็อกมากซึ่งจะง่ายต่อวงจรแปลงสัญญาณและทำให้ผู้ใช้ซึ่งเคยชินกับรูปแบบสัญญาณเก่าสามารถเข้าใจสัญญาณแบบดิจิทัลได้ไม่ยาก

2.3 วิธีการขยายภาพ



รูปที่ 2-6 การขยายภาพขนาด kxk เท่า

การขยายภาพนั้นจะขยายขนาดภาพต้นฉบับจากเดิมขนาด MxN พิกเซลเป็นขนาด kMxkN พิกเซล โดยที่ k เป็นค่าคงที่ใดๆ ที่มากกว่า 1 ดังแสดงในรูปที่ 2-6 ดังนั้นภาพใหม่ที่ได้ก็จะมีขนาดเพิ่มขึ้นเป็น kxk เท่าของภาพเดิม ในเครื่องขยายวีดิทัศน์ภาษามือนี้จะใช้ค่า k = 2 และ 1.5 เพื่อขยายภาพใหม่ให้มีขนาดเป็น 2Mx2N หรือ 1.5Mx1.5N พิกเซลตามลำดับ



รูปที่ 2-7 การชักตัวอย่างซ้ำเพื่อขยายภาพในแนวนอนขนาด k เท่า เมื่อ k = 4

การขยายภาพคือการนำค่าในแต่ละพิกเซลในภาพต้นฉบับมาชักตัวอย่างซ้ำ (resampling) ด้วยความถี่ที่สูงขึ้นกว่าความถี่ของข้อมูลขาเข้าดังแสดงในรูปที่ 2-7 ซึ่งเป็นตัวอย่างการขยายภาพในแนวนอนขนาด 4 เท่า (k = 4) หากช่วงเวลาที่ชักตัวอย่างตรงกับค่าของพิกเซลเดิม (แทนด้วยวงกลมที่แรเงา) ค่าที่ได้ออกมาก็จะไม่เปลี่ยนแปลง เช่น ตำแหน่งที่ n+1 ซึ่งตรงกับตำแหน่งที่

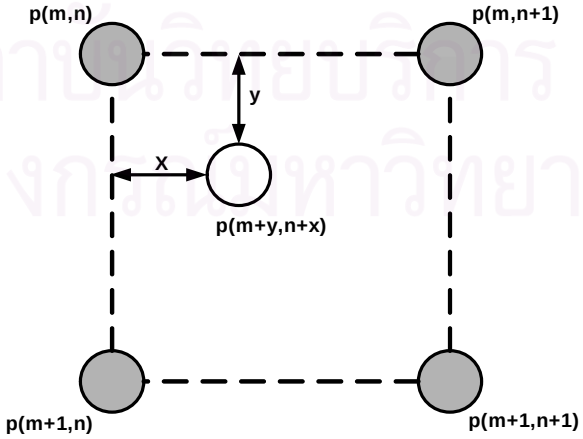
n+4/k พอดี แต่หากไม่ตรงกันพิกเซลนั้นจะสามารถประมาณค่าได้จากพิกเซลที่อยู่ใกล้เคียง โดยรูปแบบของสมการที่ใช้ในการประมาณค่าพิกเซลที่สร้างใหม่ขึ้นอยู่กับวิธีการขยายภาพที่เลือกใช้

วิธีขยายภาพที่ได้รับความนิยมกันในปัจจุบันมีอยู่หลายวิธี ซึ่งจะแบ่งออกเป็น 2 ประเภทคือ แบบที่เป็นเชิงเส้นและแบบที่ไม่เป็นเชิงเส้น แบบที่เป็นเชิงเส้นจะมีกระบวนการในการคำนวณไม่ซับซ้อน ทำให้สามารถนำไปประยุกต์ใช้ในงานจริงได้ดีและประหยัดทรัพยากรในวงจรคำนวณ แต่ข้อเสียของวิธีขยายภาพแบบเชิงเส้น คือ จะทำให้ขอบของภาพไม่คม แตกและไม่ชัดได้ ในขณะที่แบบที่ไม่เป็นเชิงเส้นซึ่งรวมถึงวิธีการขยายภาพแบบปรับตัวได้ (Adaptive expansion technique) [4] จะสามารถรักษาความคมของขอบวัตถุต่างๆ ในภาพได้ดีกว่า แต่สมการที่ใช้ในการประมาณค่าก็จะซับซ้อนขึ้นมากเมื่อเทียบกับแบบเชิงเส้น

เนื่องจากในงานวิจัยชิ้นนี้จะประยุกต์ใช้งานกับสัญญาณวีดิทัศน์ซึ่งต้องทำงานเป็นระบบเวลาจริง (real-time) ทำให้อัลกอริทึมที่เลือกใช้นี้ต้องไม่ซับซ้อนมากและสามารถออกแบบบนชิพได้จริง เมื่อพิจารณาจากวิธีขยายภาพทั้งหมดแล้วพบว่ามีวิธีที่น่าสนใจและเป็นวิธีที่ได้รับความนิยมในการไปใช้งานจริงกันมาก ได้แก่ วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood) [1] วิธีประมาณค่าแบบไบลิเนียร์ (bilinear interpolation) [2] และวิธีประมาณค่าแบบไบคิวบิก (bicubic interpolation) [3] ซึ่งรายละเอียดของแต่ละวิธีนั้นมีดังต่อไปนี้

2.3.1 **วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด** (Nearest neighborhood)

การขยายภาพแบบประมาณค่าจากค่าที่อยู่ใกล้ที่สุดจะสร้างพิกเซลใหม่โดยเลือกใช้ค่าเดียวกับพิกเซลของภาพเดิมที่อยู่ใกล้ที่สุดในบรรดาพิกเซลรอบๆ 4 พิกเซลดังแสดงเป็นตัวอย่างในรูปที่ 2-8



รูปที่ 2-8 การหาค่าที่ตำแหน่งพิกเซล p(m+y,n+x) ซึ่งคำนวณจากพิกเซลที่อยู่ใกล้ที่สุดเมื่อขยายภาพแบบประมาณค่าจากค่าที่อยู่ใกล้ที่สุด

ให้ p(m,n) แทนค่าความเข้ม (intensity) ของพิกเซลในภาพเดิมขนาด MxN พิกเซล โดยค่า m = 0, 1, 2, …, M-1 และ n = 0, 1, 2, …, N-1 การหาค่าพิกเซลที่ชักตัวอย่าง ณ ตำแหน่งสัมพัทธ์ (y,x) จะคำนวณได้จากสมการต่อไปนี้

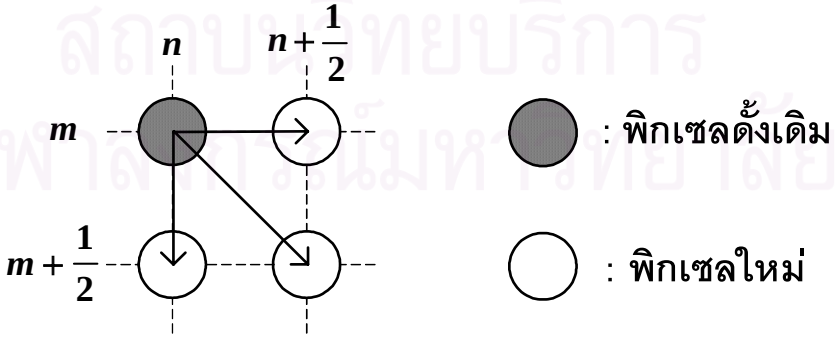
$$p(m+y,n+x)=\begin{cases} p(m,n) & ; \quad 0\le x\le 0.5, \quad 0\le y\le 0.5 \\ p(m,n+1) & ; \quad 0\le x\le 0.5, \quad 0.5< y<1 \\ p(m+1,n) & ; \quad 0.5< x<1, \quad 0\le y\le 0.5 \\ p(m+1,n+1) & ; \quad 0.5< x<1, \quad 0.5< y<1 \end{cases}$$

สมการที่(2.1)

จากสมการจะพบว่า พิกเซลใหม่ก็จะเลือกใช้ค่าความเข้มของพิกเซลที่มีระยะทางถึงพิกเซลใหม่น้อยที่สุด ดังนั้นการขยายภาพแบบนี้จะใช้แค่ตัวตรวจสอบระยะทางเท่านั้นโดยไม่ต้องใช้วงจรคำนวณสำหรับการประมาณค่าจากพิกเซลหลายๆ พิกเซลที่อยู่รอบๆ ดังนั้นการขยายภาพแบบนี้จะสามารถทำได้เร็วมาก

หากระยะทางระหว่างสัญญาณที่ถูกชักออกมามีค่าคงที่เท่ากับ 1/k พิกเซล ทำให้ได้ค่า x และ y = 0, 1/k, 2/k, …, k/k ภาพก็จะถูกขยายในแนวนอนและแนวตั้งอย่างละ k เท่า หรือขยายเพิ่มขึ้นเป็น k^2 เท่าของภาพเดิมนั่นเอง

ในกรณีขยายภาพขนาด 4 เท่า (k = 2) จะชักตัวอย่างในแนวนอนด้วยความถี่เพิ่มขึ้นเป็น 2 เท่าเทียบกับสัญญาณเดิมหรือทุกๆ 1/2 พิกเซล และภาพใหม่ที่ได้จะมีขนาด 2Mx2N พิกเซล ให้ p(m,n) แทนค่าความเข้มในแต่ละพิกเซลของภาพที่ผ่านการขยายแล้ว โดย m = 0, 0.5, 1, 1.5, …, M-0.5 และ n = 0, 0.5, 1, 1.5, N-0.5



รูปที่ 2-9 การขยายภาพขนาด 4 เท่าโดยประมาณค่าจากค่าที่อยู่ใกล้ที่สุด

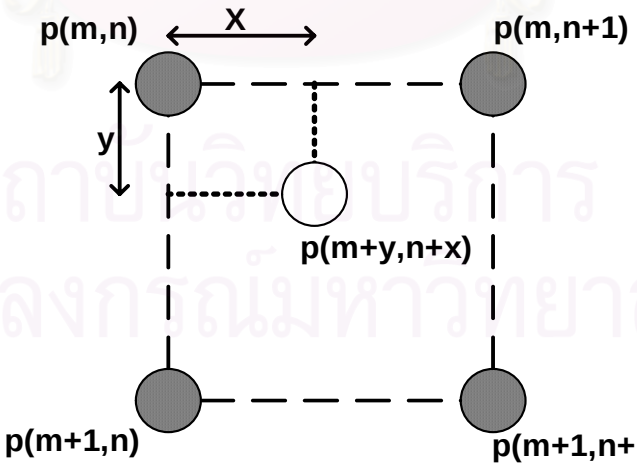
จากรูปที่ 2-9 ค่าความเข้มในพิกเซลแต่ละพิกเซลจากภาพเดิมจะถูกนำมาซ้ำค่าเพื่อสร้างพิกเซลใหม่ขึ้น 3 พิกเซล โดยสามารถเขียนได้เป็นสมการดังนี้

$$p(m+\frac{1}{2},n) = p(m+n+\frac{1}{2}) = p(m+\frac{1}{2},n+\frac{1}{2}) = p(m,n) \qquad \text{สมการที่ (2.2)}$$

ดังที่กล่าวมาจะพบว่าการขยายภาพโดยประมาณค่าจากค่าที่ใกล้ที่สุดนี้สามารถทำได้ง่ายและรวดเร็ว ซึ่งเป็นข้อดีที่สำคัญของการขยายภาพแบบนี้นั่นเอง แต่ข้อเสียของการขยายภาพโดยใช้การซ้ำพิกเซลนี้ คือ คุณภาพของภาพที่ได้จะไม่ดีโดยเฉพาะที่ขอบของวัตถุจะเกิดปรากฏการณ์ซิกแซกหรือแตกเป็นเหลี่ยมๆ ออกมา ซึ่งหากนำมาใช้ขยายภาพภาษามือก็จะทำให้ผู้ชมรู้สึกรำคาญตาได้ดังแสดงในรูปที่ 2-10

รูปที่ 2-10 ภาพที่ขยาย 2X2 เท่าโดยวิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด

2.3.2 การประมาณค่าแบบไบลิเนียร์ (Bilinear interpolation)



รูปที่ 2-11 การหาค่าพิกเซล p(m+y,n+x) จากพิกเซลรอบๆ 4 พิกเซล ด้วยวิธีไบลิเนียร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การขยายภาพโดยวิธีแบบไบลิเนียร์จะคำนวณพิกเซลที่ชักตัวอย่างเพิ่มขึ้นมาโดยใช้พิกเซลที่อยู่รอบๆ 4 พิกเซลมาประมาณค่าด้วยวิธีประมาณแบบเชิงเส้นระหว่างจุดทั้ง 4 ดังสมการต่อไปนี้

$$p(m+y,n+x)=\left[(1-x)p(m,n)+xp(m,n+1)\right](1-y)+\left[(1-x)p(m+1,n)+xp(m+1,n+1)\right]y \quad \text{สมการที่ (2.3)}$$

เนื่องจากข้อมูลของสัญญาณวีดิทัศน์ที่ส่งเข้ามาเพื่อขยายภาพนี้จะส่งเป็นลำดับจากด้านซ้ายมือมาขวามือและจากด้านบนสุดลงมาถึงล่างสุด ดังนั้นขั้นตอนการขยายภาพจึงถูกแบ่งออกเป็น 2 ขั้นตอน คือ การขยายภาพทางแนวนอน (horizontal expansion) และการขยายภาพทางแนวตั้ง (vertical expansion) ซึ่งในแต่ละขั้นตอนจะประกอบด้วยสมการดังต่อไปนี้

การขยายภาพทางแนวนอน

$$p(m,n+x)=(1-x)p(m,n)+(x)p(m,n+1)$$
$$p(m+1,n+x)=(1-x)p(m+1,n)+(x)p(m+1,n+1) \quad \text{สมการที่ (2.4)}$$

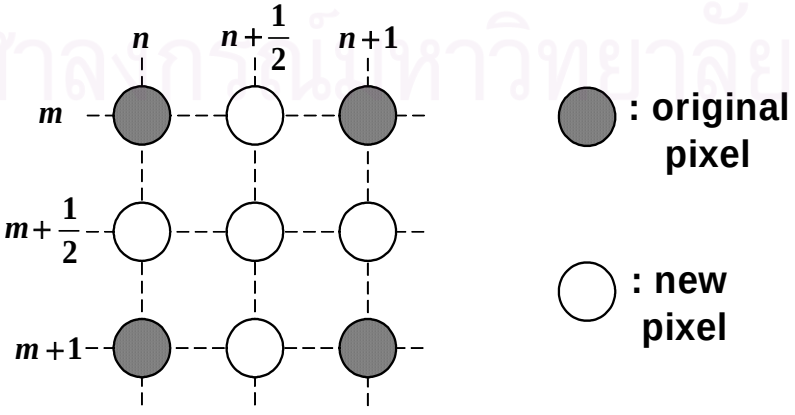
การขยายภาพทางแนวตั้ง

$$p(m+y,n+x)=(1-y)p(m,n+x)+(y)p(m+1,n+x) \quad \text{สมการที่ (2.5)}$$

การขยายภาพทางแนวตั้งจะนำพิกเซลที่คำนวณได้จากการขยายทางแนวนอนในสมการที่ 2.4 มาใช้ ซึ่งจะทำให้สมการที่ 2.5 นี้ดูง่ายขึ้นเมื่อเทียบกับสมการที่ 2.3 ในเครื่องขยายวีดิทัศน์นี้จะขยายภาพด้วยกัน 2 ขนาดคือ 4 เท่าและ 2.25 เท่า ซึ่งจะมีระยะทางระหว่างการชักตัวอย่าง (k) เท่ากับ 1/2 และ 2/3 นั่นเอง

1) <u>การขยายภาพขนาด 4 เท่า</u>

การขยายภาพแบบ 4 เท่าด้วยวิธีประมาณค่าแบบไบลิเนียร์สามารถแสดงได้ดังรูปที่ 2-12



รูปที่ 2-12 การขยายภาพขนาด 4 เท่าด้วยวิธีไบลิเนียร์

เมื่อแทนค่า x = y = 0.5 ในสมการที่ 2.4 และ 2.5 จะได้ผลดังนี้

การขยายภาพทางแนวนอน

ให้ n = 0, 1, 2, …, N-1 (N คือ จำนวนคอลัมน์ของพิกเซลทั้งหมดในภาพ)

$$p(m,n+\frac{1}{2}) = \frac{1}{2}p(m,n) + \frac{1}{2}p(m,n+1) \qquad \text{สมการที่ (2.6)}$$

การขยายภาพทางแนวตั้ง

ให้ m = 0, 1, 2, …, M-1 (M คือ จำนวนแถวของพิกเซลทั้งหมดในภาพ)

$$p(m+\frac{1}{2},n) = \frac{1}{2}p(m,n) + \frac{1}{2}p(m+1,n) \qquad \text{สมการที่ (2.7)}$$

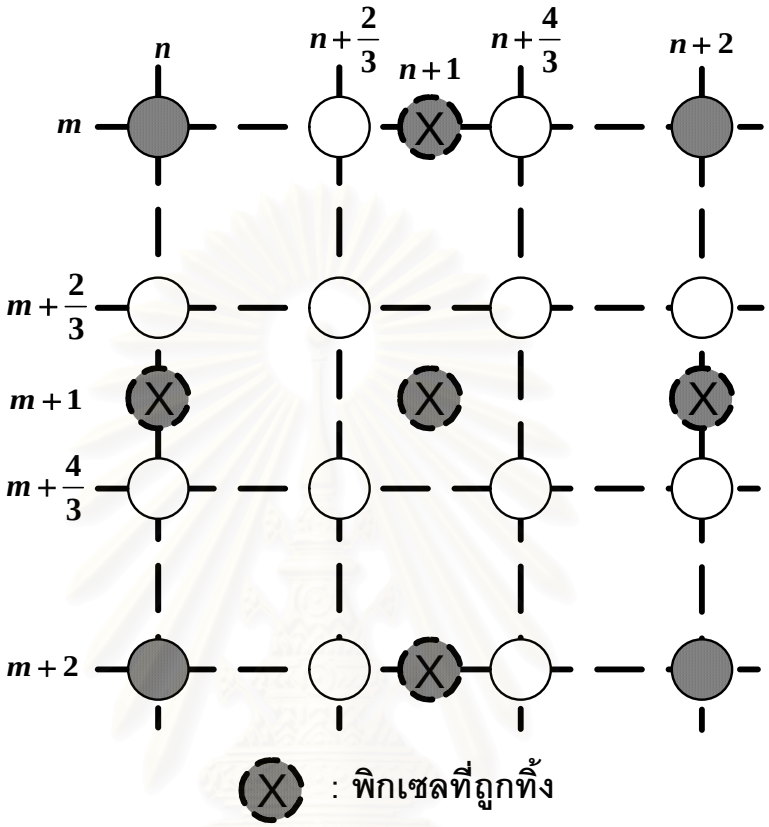
$$p(m+\frac{1}{2},n+\frac{1}{2}) = \frac{1}{2}p(m,n+\frac{1}{2}) + \frac{1}{2}p(m+1,n+\frac{1}{2}) \qquad \text{สมการที่ (2.8)}$$

การขยายภาพทางแนวนอนจะใช้ข้อมูลที่อยู่ติดกันในด้านซ้ายมือและขวามือที่อยู่ในเส้นภาพเดียวกันมาคำนวณ ซึ่งตรงกับรูปแบบการส่งข้อมูลของสัญญาณวีดิทัศน์ที่ส่งเรียงมาทีละพิกเซลจากด้านซ้ายถึงขวามือ ดังนั้นกระบวนการขยายภาพทางแนวนอนนั้นจะรับข้อมูลไปพร้อมกับนำข้อมูลมาคำนวณเพื่อขยายภาพได้ในเวลาเดียวกัน ส่วนการขยายภาพทางแนวตั้งจะต่างกันเพราะสัญญาณวีดิทัศน์นั้นเป็นระบบแบบสอดประสาน คือข้อมูล 2 เส้นที่อยู่ติดกันจะส่งแยกออกจากกันในแต่ละฟิลด์ ดังนั้นการคำนวณเพื่อขยายภาพทางแนวตั้งจะเกิดขึ้นได้เมื่อมีข้อมูลภาพครบทั้ง 2 ฟิลด์ก่อน เหตุผลนี้เองทำให้วงจรประมวลผลสำหรับการขยายภาพต้องมีที่พักข้อมูลเพื่อเก็บภาพในฟิลด์แรกไว้รอการประมวลผลกับข้อมูลในฟิลด์ถัดไป

ตามเหตุผลที่กล่าวมาทำให้การขยายภาพจากสัญญาณวีดิทัศน์นี้จะมีความล่าช้าอยู่ในช่วงการขยายภาพทางแนวตั้ง และจำเป็นต้องมีที่พักข้อมูลขนาดใหญ่ที่สามารถเก็บภาพได้อย่างน้อย 1 ฟิลด์ในวงจรขยายภาพอันเป็นข้อเสียอย่างหนึ่งในการขยายข้อมูลภาพที่ส่งมาด้วยระบบสอดประสาน

2) *<u>การขยายภาพขนาด 2.25 เท่า</u>*

กระบวนการขยายภาพขนาด 2.25 เท่าสามารถแสดงได้ดังรูปที่ 2-13



รูปที่ 2-13 การขยายภาพขนาด 2.25 เท่าด้วยวิธีไบลิเนียร์

เช่นเดียวกับการขยายขนาด 4 เท่า สมการที่ใช้ในการคำนวณจะถูกแบ่งออกเป็น 2 ส่วน คือ การขยายทางแนวนอนและการขยายทางแนวตั้ง

การขยายภาพทางแนวนอน

ให้ n = 0, 2, 4, …, N-1

$$P(m,n+\frac{2}{3}) = \frac{1}{3}p(m,n) + \frac{2}{3}p(m,n+1) \qquad \text{สมการที่ (2.9)}$$

$$P(m,n+\frac{4}{3}) = \frac{2}{3}p(m,n+1) + \frac{1}{3}p(m,n+2) \qquad \text{สมการที่ (2.10)}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การขยายภาพทางแนวตั้ง

ให้ M = 0, 2, 4, …, M-1 และ i = 0, 2, 4

$$P(m+\frac{2}{3},n+\frac{i}{3})=\frac{1}{3}P(m,n+\frac{i}{3})+\frac{2}{3}P(m+1,n+\frac{i}{3})$$ สมการที่ (2.11)

$$P(m+\frac{4}{3},n+\frac{i}{3})=\frac{2}{3}P(m+1,n+\frac{i}{3})+\frac{1}{3}P(m+2,n+\frac{i}{3})$$ สมการที่ (2.12)

การขยายภาพขนาด 2.25 เท่านี้จะมีข้อมูลภาพเดิมบางส่วนที่อยู่ในแถว m+1 และคอลัมน์ n+1 หายไปในภาพที่ขยายออกมา เนื่องจากช่วงที่ชักตัวอย่างของสัญญาณนั้นไม่ตรงกับข้อมูลเหล่านี้

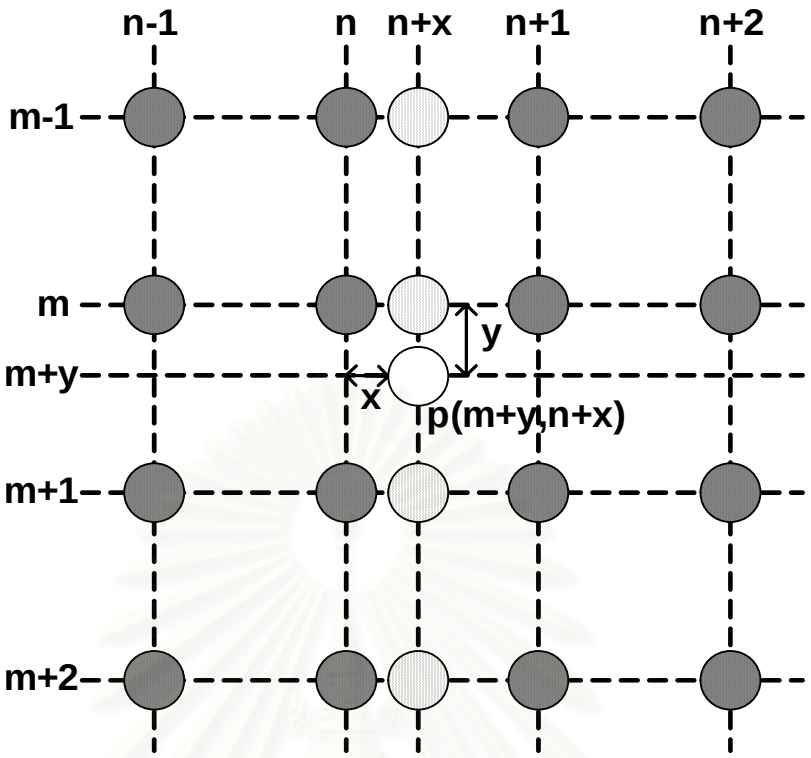
เมื่อนำภาพภาษามือมาทดสอบการขยายภาพโดยใช้วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุดเปรียบเทียบกับวิธีประมาณค่าแบบไบลิเนียร์โดยใช้โปรแกรม Photoshop จะได้ผลดังนี้

รูปที่ 2-14 ภาพที่ได้จากการขยายโดยการประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (ภาพซ้ายมือ) เปรียบเทียบกับภาพที่ได้จากการขยายแบบไบลิเนียร์ (ภาพขวามือ)

จากรูปที่ 2-14 จะพบว่าการขยายภาพแบบประมาณค่าแบบไบลิเนียร์นี้จะให้คุณภาพของภาพดีกว่าแบบประมาณค่าจากค่าที่อยู่ใกล้ที่สุด กล่าวคือ ขอบของวัตถุจะเรียบขึ้น ไม่แตกชัดเจน ซึ่งมาจากกระบวนการคำนวณที่ใช้การประมาณแบบเชิงเส้น ข้อเสียของการขยายภาพแบบไบลีเนียร์ คือ ขอบของภาพที่ขยายออกมาจะเบลอ ไม่คมชัด หรือข้อมูลภาพย่านความถี่สูงถูกทำลายไปนั่นเอง ทำให้ไม่เหมาะกับการใช้งานที่ต้องการความคมของขอบวัตถุมาก

2.3.3 **การประมาณค่าแบบไบคิวบิก** (Bicubic interpolation)

การขยายภาพแบบไบคิวบิกนี้จะคำนวณพิกเซลที่ชักตัวอย่างเพิ่มมาใหม่โดยใช้พิกเซลที่อยู่รอบๆ ทั้งหมด 16 พิกเซล ขั้นตอนการขยายภาพแบบไบคิวบิกจะคล้ายกับแบบไบลิเนียร์คือจะแยกการคำนวณออกเป็น 2 ส่วนคือ การคำนวณทางแนวนอนและการคำนวณทางแนวตั้ง



รูปที่ 2-15 การหาค่า p(m+y,n+x) จากพิกเซลรอบๆ 16 พิกเซลด้วยวิธีไบคิวบิก

รูปที่ 2-15 แสดงการประมาณค่าความเข้มของพิกเซลใหม่จากค่าความเข้มของพิกเซลรอบๆ 16 พิกเซล ในการประมาณค่าแบบไบคิวบิกจะต้องใช้ข้อมูลทั้งหมด 16 พิกเซลมาคำนวณด้วยสมการดังนี้

การคำนวณทางแนวนอน

ให้ i = -1, 0, 1, 2 และให้

$$do = p(m+i,n-1) - p(m+i,n)$$

$$d1 = p(m+i,n+1) - p(m+i,n)$$

$$d2 = p(m+i,n+2) - p(m+i,n)$$

$$a0 = p(m+i,n) \quad ; \quad a1 = -\frac{1}{3}d0 + d1 - \frac{1}{6}d2$$

$$a2 = \frac{1}{2}do + \frac{1}{2}d1 \quad ; \quad a3 = -\frac{1}{6}do - \frac{1}{2}d1 + \frac{1}{6}d2$$

จะได้ว่า

$$p(m+i,n+x) = a0 + a1(x) + a2(x)^2 + a3(x)^3 \qquad \text{สมการที่ (2.13)}$$

การคำนวณทางแนวตั้ง

$$d3 = p(m-1,n+x) - p(m,n+x)$$
$$d4 = p(m+1,n+x) - p(m,n+x)$$
$$d5 = p(m+2,n+x) - p(m,n+x)$$

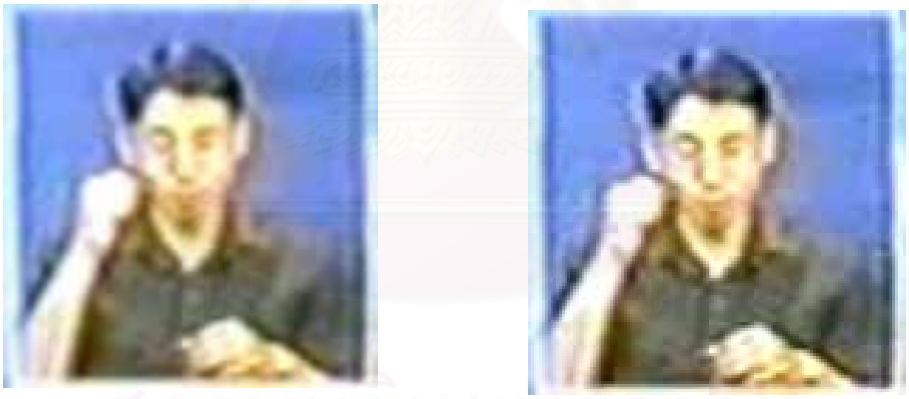
$$a4 = p(m,n+x) \quad ; \quad a5 = -\frac{1}{3}d3 + d4 - \frac{1}{6}d5$$

$$a6 = \frac{1}{2}d3 + \frac{1}{2}d4 \quad ; \quad a7 = -\frac{1}{6}d3 - \frac{1}{2}d4 + \frac{1}{6}d5$$

จะได้ว่า

$$p(m+y,n+x) = a4 + a5(y) + a6(y)^2 + a7(y)^3 \qquad \text{สมการที่ (2.14)}$$

จากสมการข้างต้นจะเห็นได้ว่าการประมาณค่าแบบไบคิวบิกจะต้องใช้สมการคำนวณอันดับสามและมีขั้นตอนการคำนวณมาก ดังนั้นวงจรในการประมวลผลจะต้องมีความซับซ้อนสูงและใช้เวลาในการประมวลผลมาก คุณภาพของภาพที่ได้จากวิธีไบคิวบิกนี้เมื่อนำมาเปรียบเทียบกับวิธีประมาณค่าแบบไบลิเนียร์จะได้ผลดังแสดงในรูปที่ 2-16



รูปที่ 2-16 ภาพจากการขยายด้วยวิธีไบคิวบิก (ภาพด้านขวา) เปรียบเทียบกับวิธีไบลิเนียร์ (ภาพด้านซ้าย)

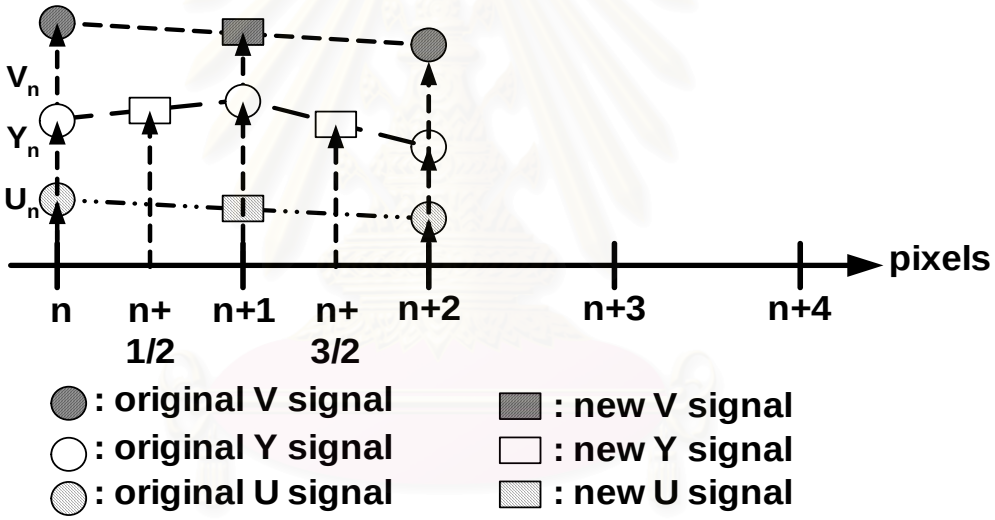
ภาพที่ได้จากการขยายภาพแบบไบคิวบิกจะรักษาความคมของขอบภาพได้ดีกว่าแบบไบลิเนียร์ และภาพที่ได้ไม่แตกเป็นพิกเซลเหมือนแบบประมาณค่าจากค่าที่ใกล้ที่สุด เมื่อเปรียบเทียบข้อดี ข้อเสียของวิธีขยายภาพทั้ง 3 วิธีแล้วพบว่า วิธีที่เหมาะสมกับเครื่องขยายวีดิทัศน์ที่สุด คือ วิธีประมาณค่าแบบไบลิเนียร์ซึ่งจะให้คุณภาพของภาพค่อนข้างดีและไม่สิ้นเปลืองทรัพยากรของวงจรประมวลผลมากเมื่อเปรียบเทียบกับวิธีไบคิวบิก

2.4 การขยายภาพบนสัญญาณวีดิทัศน์แบบดิจิทัล

เนื่องจากข้อมูลของสัญญาณวีดิทัศน์ตามมาตรฐาน ITU-R BT.656 จะมีการบีบอัดจำนวนข้อมูลสัญญาณสีให้เหลือเพียงครึ่งหนึ่งของข้อมูลสัญญาณความสว่าง ดังนั้นสมการที่ใช้ในการประมาณค่าเพื่อสร้างข้อมูลทั้งสามเพิ่มจึงไม่เหมือนกัน แต่โครงสร้างของสมการจะเหมือนกับที่กล่าวไว้ในหัวข้อที่แล้ว ดังนั้นในเนื้อหาส่วนต่อไปจะกล่าวถึงการขยายภาพทางแนวนอนด้วยวิธีไบลิเนียร์ขนาด 2 เท่าและขนาด 1.5 เท่าบนสัญญาณวีดิทัศน์แบบดิจิทัล ซึ่งมีรายละเอียดต่างๆ ดังต่อไปนี้

1) <u>การขยายภาพขนาด 2 เท่า</u>

ให้ U แทนสัญญาณสี C_B และ V แทนสัญญาณสี C_R การขยายภาพบนสัญญาณวีดิทัศน์ขนาด 2 เท่าทางแนวนอนสามารถแสดงได้ดังรูปที่ 2-17



รูปที่ 2-17 การสุ่มสัญญาณ Y, U, V เพื่อขยายข้อมูลทางแนวนอน 2 เท่า

สมการที่ใช้ในการคำนวณเพื่อขยายสัญญาณความสว่าง (Y) และสัญญาณสี (U, V) ขนาด 2 เท่าทางแนวนอนมีดังต่อไปนี้

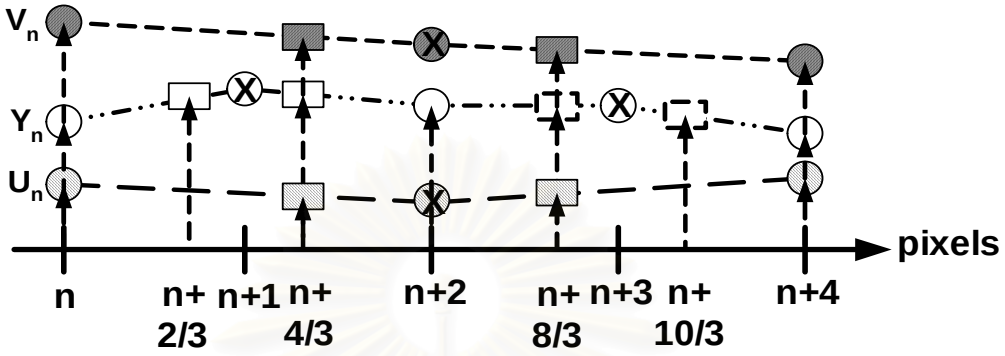
$$Y_{n+\frac{1}{2}} = \frac{1}{2}Y_n + \frac{1}{2}Y_{n+1} \quad ; n = 0, 1, 2, \ldots$$

$$U_{n+1} = \frac{1}{2}U_n + \frac{1}{2}U_{n+2}$$

$$V_{n+1} = \frac{1}{2}V_n + \frac{1}{2}V_{n+2} \quad ; n = 0, 2, 4, \ldots \qquad \text{สมการที่ (2.15)}$$

การสร้างสัญญาณความสว่างจะใช้ข้อมูลของพิกเซลที่อยู่ติดกันมาหาค่าเฉลี่ย ส่วนสัญญาณสีจะเกิดที่ความถี่ต่ำกว่า เพราะข้อมูลที่อยู่ติดกันถูกส่งมาทุกๆ 2 พิกเซล

2) *<u>การขยายภาพขนาด 1.5 เท่า</u>*



รูปที่ 2-18 การสุ่มสัญญาณ Y, U, V เพื่อขยายข้อมูลในแนวนอน 1.5 เท่า

การขยายข้อมูลสัญญาณวีดิทัศน์ดังแสดงในรูปที่ 2-18 นี้จะใช้สมการในการคำนวณดังนี้

$$Y_{n+\frac{2}{3}} = \frac{1}{3}Y_n + \frac{2}{3}Y_{n+1} \; ; \; Y_{n+\frac{4}{3}} = \frac{2}{3}Y_{n+1} + \frac{1}{3}Y_{n+2} \; ; \; n = 0, 2, 4, \ldots$$

$$U_{n+\frac{4}{3}} = \frac{1}{3}U_n + \frac{2}{3}U_{n+2} \; ; \; U_{n+\frac{8}{3}} = \frac{2}{3}U_{n+2} + \frac{1}{3}U_{n+4} \; ; \; n = 0, 4, 8, \ldots$$

$$V_{n+\frac{4}{3}} = \frac{1}{3}V_n + \frac{2}{3}V_{n+2} \; ; \; V_{n+\frac{8}{3}} = \frac{2}{3}V_{n+2} + \frac{1}{3}V_{n+4} \; ; \; n = 0, 4, 8, \ldots \text{ สมการที่ (2.16)}$$

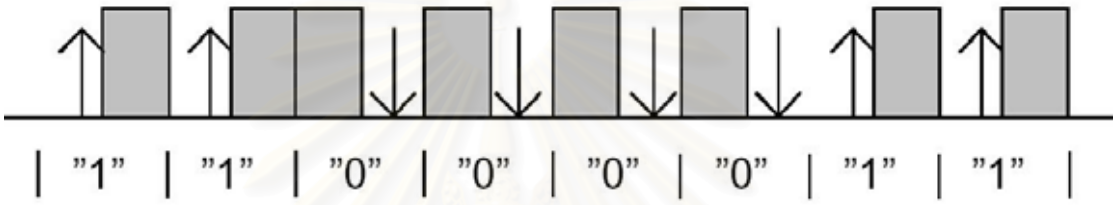
การขยายภาพขนาด 1.5 เท่านี้จะมีความถี่ในการสร้างข้อมูลทั้งสามประเภทเหมือนกับขนาด 2 เท่า แต่ข้อมูลของภาพเดิมบางส่วนจะถูกละทิ้งไปซึ่งในรูปที่ 2-18 แทนด้วยรูปวงกลมที่มีสัญลักษณ์กากบาทอยู่

จากสมการที่ (2.15) – (2.16) จะพบว่าสมการที่ใช้ในการขยายข้อมูลทั้ง 3 ชนิด มีรูปแบบเหมือนกัน คือ ประกอบด้วยการบวก 1 ครั้งและการหาร (2 หรือ 3) 1 ครั้ง และเนื่องจากช่วงเวลาที่ข้อมูลทั้งสามนี้เข้ามาและส่งออกไปไม่ตรงกัน คือ สลับกันเป็นลำดับ C_B, Y, C_R, Y ดังนั้นวงจรที่ใช้ในการประมาณค่าของสัญญาณทั้งสามนี้สามารถรวมกันเป็นวงจรเดียวกันได้โดยเพิ่มวงจรส่วนที่ใช้จัดเวลาการทำงานของสัญญาณทั้งสามประเภทลงไป รายละเอียดในการออกแบบวงจรจะกล่าวถึงในบทถัดไป

2.5 รูปแบบข้อมูลอินฟราเรดของรีโมตคอนโทรลสำหรับโทรทัศน์

สัญญาณอินฟราเรดจากรีโมตคอนโทรลที่ใช้กับเครื่องรับโทรทัศน์จะเข้ารหัสและส่งมาในช่วงย่านความถี่ตั้งแต่ 30 kHz จนถึง 60 kHz โดยรูปแบบของสัญญาณที่ใช้แทนข้อมูลแต่ละบิตที่ได้รับความนิยมกันจะมีอยู่ 3 รูปแบบ คือ แบบสองเฟส (Bi-phase) แบบระยะทางระหว่างพัลส์ (pulse-distance) และแบบความกว้างของพัลส์ (pulse-length) ซึ่งรายละเอียดของการส่งสัญญาณแต่ละแบบมีดังต่อไปนี้

2.5.1 การเข้ารหัสแบบสองเฟส

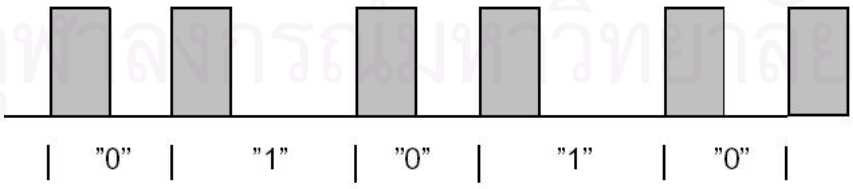


รูปที่ 2-19 รูปแบบของสัญญาณที่เข้ารหัสด้วยวิธีสองเฟส

การเข้ารหัสข้อมูลแบบสองเฟส จะใช้ขอบของสัญญาณแทนค่าของข้อมูลแต่ละบิต โดยถ้าสัญญาณเปลี่ยนจาก ‘0’ เป็น ‘1’ หรือเป็นขอบขาขึ้นจะหมายถึงข้อมูลค่า ‘1’ แต่หากสัญญาณที่รับไปเปลี่ยนจาก ‘1’ เป็น ‘0’ หรือเป็นขอบขาลงก็จะหมายถึงข้อมูลค่า ‘0’

ข้อดีของการส่งแบบสองเฟส คือ เวลาที่ใช้ส่งข้อมูลบิต 1 หรือ 0 นานเท่ากัน อัตราการส่งข้อมูลจะคงที่ไม่ขึ้นกับค่าของข้อมูล ซึ่งจะต่างกับวิธีเข้ารหัสอีก 2 แบบที่จะกล่าวต่อไปที่จะใช้เวลาในการส่งข้อมูลเปลี่ยนแปลงตามค่าของข้อมูลที่จะส่ง

2.5.2 การมอดูเลชั่นโดยใช้ระยะทางระหว่างพัลส์ (Pulse-distance modulation)

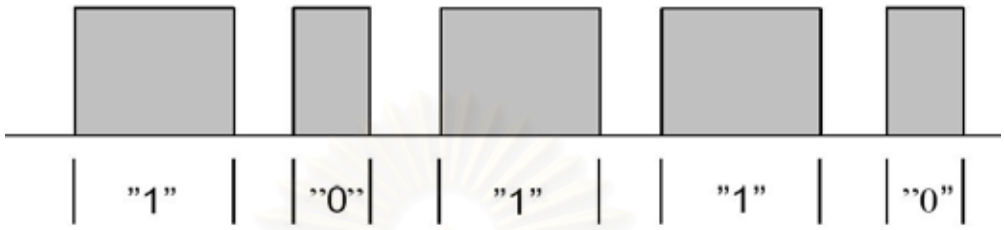


รูปที่ 2-20 การส่งข้อมูลที่มอดูเลตโดยใช้ระยะทางระหว่างพัลส์

การตรวจสอบข้อมูลที่ส่งด้วยการมอดูเลชั่นโดยใช้ระยะทางระหว่างพัลส์จะใช้การตรวจสอบช่วงเวลาระหว่างสัญญาณเบิรตซ์ (สัญญาณพัลส์ที่มีค่าเป็น ‘1’) ถ้าหากสัญญาณเบิรตซ์ห่างกัน

มากก็จะได้ค่าข้อมูลเป็น ‘1’ แต่หากว่าสัญญาณห่างกันน้อยแสดงว่าข้อมูลมีค่าเป็น ‘0’ ดังนั้นการตรวจสอบค่าของข้อมูลประเภทนี้จะใช้การตรวจสอบช่วงเวลาระหว่างที่ข้อมูลมีค่าเป็น ‘0’ นั่นเอง

2.5.3 **การเข้ารหัสโดยใช้ความกว้างของพัลส์** (Pulse-length code)



รูปที่ 2-21 การเข้ารหัสโดยใช้ความกว้างของพัลส์

การส่งข้อมูลซึ่งเข้ารหัสโดยใช้ความกว้างของพัลส์จะใช้สัญญาณเบิรตซ์ 2 แบบในการส่งเพื่อใช้แทนข้อมูลบิต ‘1’ และบิต ‘0’ โดยถ้าหากข้อมูลเป็น ‘1’ สัญญาณเบิรตซ์จะมีความกว้างมากกว่าสัญญาณเบิรตซ์ที่ส่งเมื่อข้อมูลเป็น ‘0’ ดังนั้นการตรวจสอบข้อมูลโดยใช้ความกว้างของพัลส์จะใช้การตรวจสอบความกว้างของสัญญาณช่วงที่มีค่าเป็น ‘1’

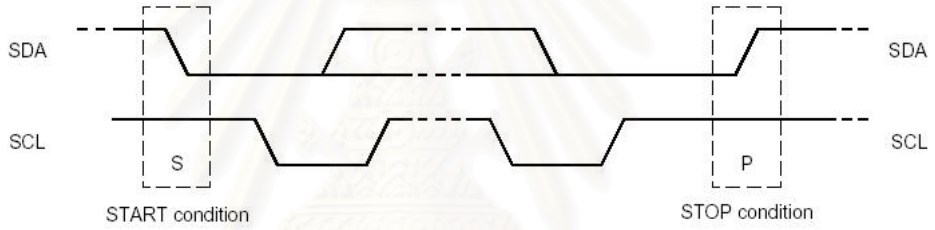
2.6 **การติดต่อแบบ** I^2C

เนื่องจากเครื่องมือที่เราออกแบบไว้นั้นประกอบขึ้นด้วยวงจรย่อยๆ หลายวงจร ดังนั้นการรับส่งข้อมูลระหว่างชิพเหล่านี้จะต้องมีวงจรแม่ซึ่งทำหน้าที่ควบคุมไม่ให้วงจรลูกอื่นๆ ส่งข้อมูลมาชนกันในบัส การส่งข้อมูลระหว่างชิพเหล่านี้ส่วนใหญ่จะใช้การส่งแบบอนุกรมซึ่งในปัจจุบันมีมาตรฐานที่นิยมใช้งานอยู่หลายๆ แบบ แต่ที่จะพบกันมากในชิพด้านวีดีโอคือ มาตรฐาน I^2C [8] ซึ่งในเครื่องขยายวีดิทัศน์เองก็ใช้มาตรฐานนี้เช่นกัน ดังนั้นในเนื้อหาส่วนต่อไปจะกล่าวถึงรายละเอียดของการรับส่งข้อมูลแบบ I^2C

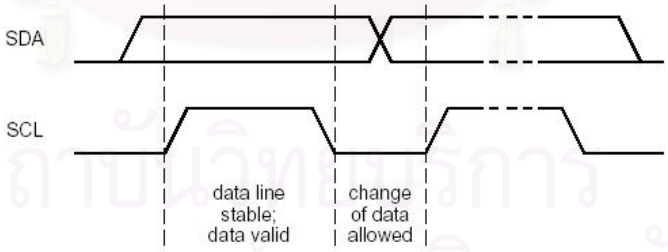
การรับส่งข้อมูลแบบ I^2C เป็นการส่งข้อมูลแบบอนุกรมและสามารถส่งข้อมูลได้ทั้ง 2 ทิศทาง โดยใช้สายสัญญาณในการเชื่อมต่อกันเพียง 2 สายเท่านั้น คือ สัญญาณ SCL (Serial clock line) ซึ่งใช้ส่งสัญญาณนาฬิกาเพื่อบอกจังหวะการอ่านและส่งข้อมูลให้กับตัวรับ และ SDA (Serial data line) ที่ใช้ส่งข้อมูลกันระหว่างชิพแต่ละตัว ในการติดต่อสื่อสารกันระหว่างชิพหลายๆ ตัวนี้จะต้องมีชิพอย่างน้อย 1 ตัวที่ทำหน้าที่เป็นตัวแม่ (master) เพื่อเป็นตัวควบคุมหลัก ส่วนชิพที่เหลือนั้นจะทำหน้าที่เป็นตัวลูก (slave)

วงจรที่ทำหน้าที่เป็นตัวแม่นี้จะเป็นตัวกำหนดจังหวะในการรับส่งข้อมูลโดยการสร้างสัญญาณนาฬิกาออกผ่านในบัส SCL ตลอดช่วงการรับส่งข้อมูล ส่วนตัวลูกจะแค่รับสัญญาณนาฬิกามาแล้วทำงานตามจังหวะที่ถูกกำหนด นอกจากนี้ตัวแม่ยังทำหน้าที่สร้างสัญญาณควบคุมอื่นๆ เช่น บิตเริ่มต้น (start bit) บิตจบการทำงาน (stop bit) สัญญาณแอดเดรสของตัวลูกที่ต้องการติดต่อในปัจจุบัน สัญญาณแอดเดรสนี้มีขนาด 7 บิตเท่านั้น ส่วนบิตที่ 8 ซึ่งเป็นบิต LSB เป็นตัวกำหนดทิศทางการไหลของข้อมูลระหว่างชิพทั้งสอง ในการรับส่งข้อมูลผ่าน I^2C จะมีขั้นตอนดังนี้

เมื่อจะเริ่มต้นรับส่งข้อมูลผ่านบัส ตัวแม่จะต้องส่งบิตเริ่มต้นออกมาก่อนเพื่อให้ชิพที่เหลือในบัสรับรู้ว่าจะมีการส่งข้อมูล และเตรียมพร้อมรับค่าเริ่มต้นอื่นๆ ต่อไป เมื่อจะจบการติดต่อนั้นๆ ตัวแม่ก็จะส่งบิตจบการทำงานออกมาเพื่อทำให้บัสนี้กลับไปอยู่ในสภาวะว่างอีกหน รูปแบบของบิตเริ่มต้นและบิตจบการทำงานสามารถแสดงได้รูปที่ 2-22



รูปที่ 2-22 บิตเริ่มต้นและบิตจบการทำงาน ของ I^2C



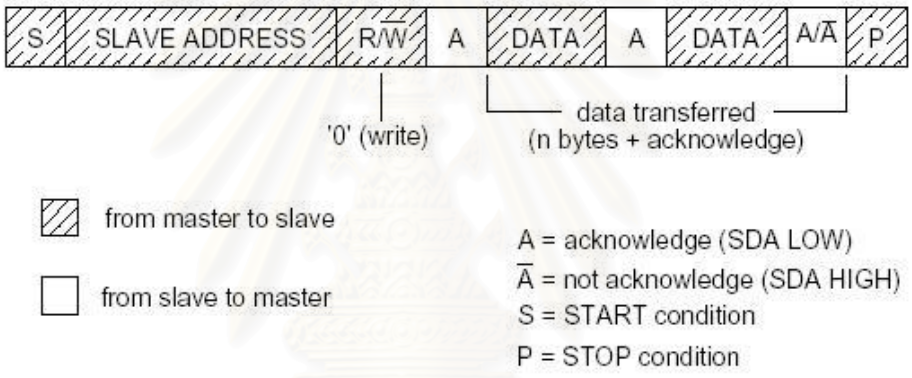
รูปที่ 2-23 ลักษณะการส่งข้อมูลของ I^2C

ในช่วงที่บัสว่างอยู่สัญญาณ SDA และ SCL จะมีค่าเท่ากับ '1' ทั้งคู่ บิตเริ่มต้นจะเกิดขึ้นเมื่อเกิดขอบขาลงของขา SDA ในขณะที่ SCL = '1' ในทางตรงข้ามถ้าพบขอบขาขึ้นของ SDA เมื่อ SCL = '1' วงจรก็จะรับรู้ว่าเป็นบิตจบการทำงาน ดังนั้นการส่งข้อมูลต่างๆ นั้น เมื่อจะเปลี่ยนค่าของข้อมูลในสาย SDA จะต้องทำช่วงที่ SCL = '0' เท่านั้นดังแสดงในรูปที่ 2-23 และการรับส่ง

ข้อมูลนั้นจะส่งออกทีละบิต เมื่อส่งข้อมูลครบ 8 บิตตัวรับก็จะส่งสัญญาณตอบรับออกมาเพื่อบอกให้ตัวส่งรู้ว่าได้รับข้อมูลครบแล้ว

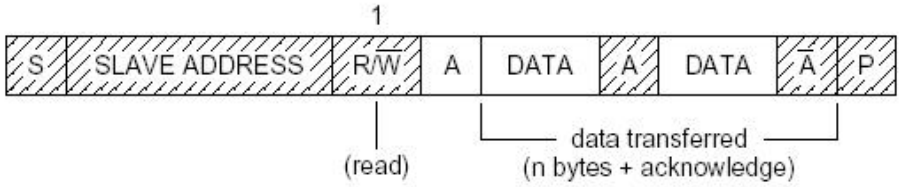
การติดต่อสื่อสารผ่าน I^2C จะมีอยู่ทั้งหมด 3 รูปแบบ คือ

1) ตัวแม่เป็นตัวส่ง (Master-transmitter) ในรูปแบบนี้ทิศทางของข้อมูลจะไหลจากตัวแม่ไปที่ตัวลูก โดยการส่งจะเริ่มจากตัวแม่สร้างบิตเริ่มต้นและส่งแอดเดรสของวงจรลูกออกไปซึ่งต้องลงท้ายด้วยค่า ‘0’ เท่านั้น ถ้าแอดเดรสตรงกับชิพใดชิพนั้นก็จะส่งสัญญาณตอบรับ (acknowledge) มาเพื่อเริ่มการรับส่งข้อมูลกัน เมื่อส่งข้อมูลครบ 1 ไบต์ ตัวลูกจะสร้างสัญญาณตอบรับมา 1 ครั้งจนกระทั่งส่งข้อมูลหมดแล้ว ตัวแม่ก็จะส่งบิตจบการทำงานออกมาที่บัส



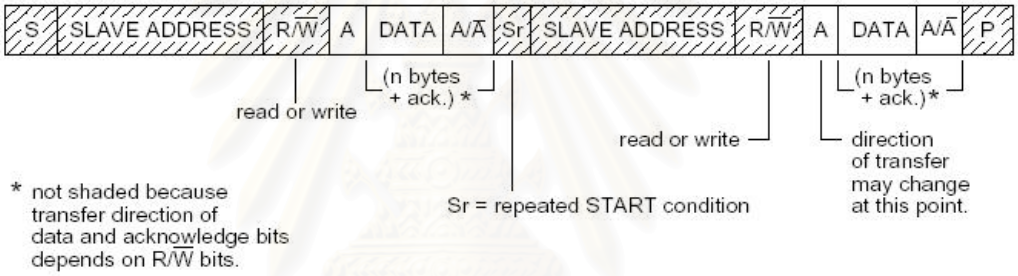
รูปที่ 2-24 รูปแบบของการส่งสัญญาณโดยวงจรแม่เป็นตัวส่ง

2) ตัวแม่เป็นตัวอ่าน (Master reading) ซึ่งทิศทางของข้อมูลจะกลับกันกับแบบแรก คือ จะไหลจากตัวลูกมาที่ตัวแม่ รูปแบบนี้ค่าแอดเดรสของตัวลูกต้องมีค่าบิตสุดท้ายเป็น ‘1’ หลังจากตัวแม่ส่งค่าแอดเดรสเสร็จแล้วทิศทางของข้อมูลจะเปลี่ยนไป คือ ข้อมูลจะส่งมาจากตัวลูกแทน ด้านตัวแม่ก็จะอ่านข้อมูลทีละไบต์และเมื่อจบ 1 ไบต์ก็จะส่งสัญญาณตอบรับ (acknowledge) ซึ่งมีค่าเป็น ‘0’ ออกมาจนกระทั่งข้อมูลไบต์สุดท้าย ค่าสัญญาณตอบรับที่ตัวแม่ส่งต้องเป็นค่า ‘1’ เพื่อหยุดการส่งข้อมูลจากตัวลูก แล้วตัวแม่ก็จะส่งบิตจบการทำงานเพื่อเลิกการติดต่อ



รูปที่ 2-25 รูปแบบการส่งข้อมูลโดยวงจรแม่เป็นตัวอ่าน

3) แบบรวม (combined format) คือ ตัวแม่จะทั้งส่งและรับข้อมูลจากตัวลูกภายในการติดต่อเพียงครั้งเดียว ลำดับการส่งสัญญาณจะคล้าย 2 แบบแรก เพียงแต่เมื่อต้องการจะเปลี่ยนทิศทางของข้อมูลก็จะใช้การส่งบิตเริ่มต้นใหม่แล้วกำหนดบิตสุดท้ายของแอดเดรสให้ถูกต้องว่าต้องการอ่านหรือเขียนข้อมูลไปให้ตัวลูกจนกระทั่งเมื่อต้องการจบการติดต่อกันแล้วตัวแม่จึงค่อยส่งบิตจบการทำงานมา

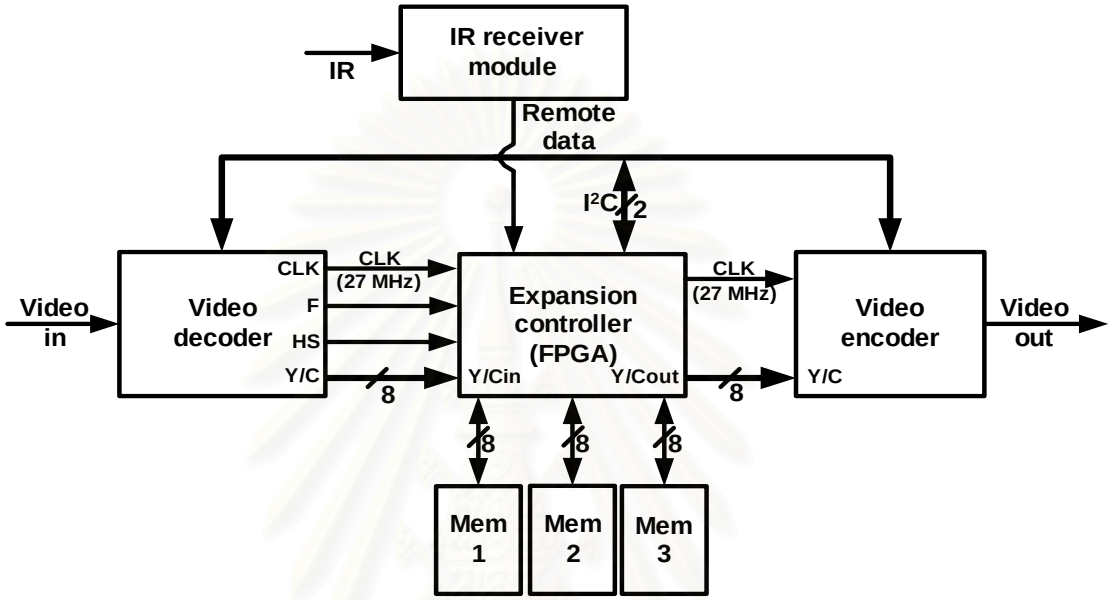


รูปที่ 2-26 รูปแบบการส่งข้อมูลแบบ combined format

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบเครื่องขยายวีดิทัศน์



รูปที่ 3-1 โครงสร้างของเครื่องขยายวีดิทัศน์ภาษามือ

เครื่องขยายวีดิทัศน์ประกอบด้วยชิพต่างๆ ทั้งหมด 7 ตัวดังแสดงในรูปที่ 3-1 ซึ่งเราสามารถแบ่งกลุ่มตามการทำงานได้เป็น 2 กลุ่มหลักๆ ดังนี้

1) วงจรแปลงสัญญาณวีดิทัศน์และวงจรรับสัญญาณอินฟราเรด

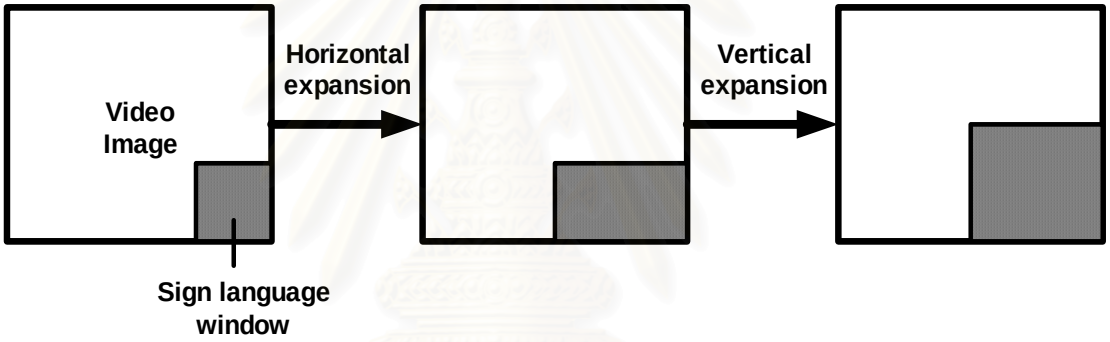
วงจรส่วนนี้ประกอบด้วย วงจรถอดรหัสสัญญาณวีดิทัศน์ (Video decoder) วงจรเข้ารหัสสัญญาณวีดิทัศน์ (Video encoder) และวงจรรับสัญญาณอินฟราเรด (IR receiver module) ซึ่งเป็นวงจรสำหรับรับและส่งสัญญาณต่างๆ กับอุปกรณ์อื่นภายนอกเครื่องขยายวีดิทัศน์

สัญญาณวีดิทัศน์จากภายนอก (Video in) จะถูกแปลงให้เป็นข้อมูลสัญญาณวีดิทัศน์แบบดิจิทัลตามมาตรฐาน ITU-R BT.601 และ ITU-R BT.656 ในระบบ PAL ด้วยวงจรถอดรหัสสัญญาณวีดิทัศน์ ทำให้ได้ข้อมูลภาพขาเข้า (Y/C in) ออกมาด้วยความถี่ 13.5 Mpixels/s หรือได้ความละเอียดของภาพ 720 พิกเซลต่อ 1 เส้นภาพ นอกจากข้อมูลภาพแล้ววงจรยังแยกสัญญาณซิงโครไนซ์ที่สำคัญต่างๆ ออกมาด้วย ได้แก่ สัญญาณซิงค์แนวนอน (HS) สัญญาณบอกเฟรมภาพ (F) และสัญญาณนาฬิกาความถี่ 27 MHz ซึ่งจะถูกนำไปใช้ในวงจรควบคุมการขยายภาพต่อไป

ข้อมูลภาพขาออกจากวงจรขยายภาพ (Y/C out) ซึ่งมีข้อมูลภาพภาษามือที่ขยายแล้วรวมอยู่ด้วย จะส่งออกมาเพื่อแปลงข้อมูลทั้งหมดให้กลับเป็นสัญญาณวีดิทัศน์ขาออก (Video out) โดยวงจรเข้ารหัสสัญญาณวีดิทัศน์เพื่อจะนำไปแสดงภาพบนเครื่องรับโทรทัศน์ต่อไป ส่วนวงจรรับสัญญาณรีโมตจะเป็นตัวดีมอดูเลตสัญญาณอินฟราเรดจากรีโมตคอนโทรล แล้วส่งสัญญาณเบสแบนด์ที่ได้ไปถอดรหัสคำสั่งต่อไปในวงจรควบคุมการขยายภาพ

2) วงจรควบคุมการขยายภาพและหน่วยความจำ

วงจรควบคุมการขยายภาพ (Expansion controller) นี้ถูกออกแบบอยู่บน FPGA ซึ่งจะใช้ประมวลผลเพื่อขยายภาพภาษามือ ขั้นตอนในการขยายภาพจะแบ่งออกได้เป็น 2 ขั้นตอน คือ การขยายภาพทางแนวนอน (Horizontal expansion) และการขยายภาพทางแนวตั้ง (Vertical expansion) โดยกระบวนการทั้งสองจะทำงานกันแบบไปป์ไลน์เป็นลำดับดังแสดงในรูปที่ 3-2

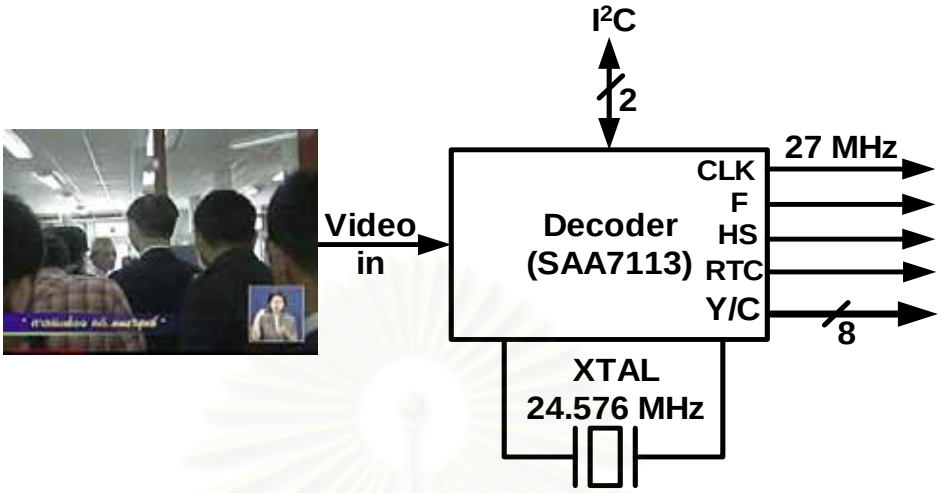


รูปที่ 3-2 การขยายภาพซึ่งแบ่งเป็น 2 ขั้นตอน

การขยายภาพในแต่ละขั้นตอนจะใช้เวลาในการประมวลผล 1 เฟรมภาพหรือ 40 ms รวม 2 ขั้นตอนแล้วภาพภาษามือ 1 เฟรมจะใช้เวลาขยายภาพทั้งหมด 2 เฟรมภาพ ดังนั้นภาพขยายที่นำไปแสดงบนจอจะเป็นภาพที่ถูกประวิงเวลาไปประมาณ 80 ms ซึ่งถือว่ามีค่าน้อยมากเพราะภาพภาษามือจะมีการเปลี่ยนแปลงช้ากว่ามาก การที่วงจรขยายภาพต้องใช้เวลาในการประมวลผล 2 เฟรมภาพนี้ทำให้ในเครื่องขยายวีดิทัศน์ต้องมีหน่วยความจำซึ่งใช้เป็นที่พักข้อมูลภาพภาษามือทั้งหมด 3 ตัว ในแต่ละตัวจะใช้เก็บภาพภาษามือ 1 เฟรม นอกจากการขยายข้อมูลภาพภาษามือแล้ววงจรควบคุมการขยายภาพยังทำหน้าที่โปรแกรมการทำงานให้กับวงจรถอดรหัสและเข้ารหัสสัญญาณวีดิทัศน์ผ่านบัส I^2C รวมถึงการถอดรหัสคำสั่งจากรีโมตคอนโทรลเพื่อปรับเปลี่ยนการทำงานของเครื่องใหม่ตามที่ผู้ใช้ต้องการ

ในเนื้อหาบทนี้จะกล่าวถึงเฉพาะวงจรส่วนแรกเท่านั้น คือ วงจรถอดรหัสสัญญาณวีดิทัศน์ วงจรเข้ารหัสสัญญาณวีดิทัศน์และวงจรรับสัญญาณอินฟราเรด ส่วนวงจรควบคุมการขยายภาพและหน่วยความจำจะกล่าวถึงในเนื้อหาของบทต่อไป

3.1 วงจรถอดรหัสสัญญาณวีดิทัศน์ (Video decoder)



รูปที่ 3-3 การใช้งานวงจรถอดรหัสสัญญาณวีดิทัศน์

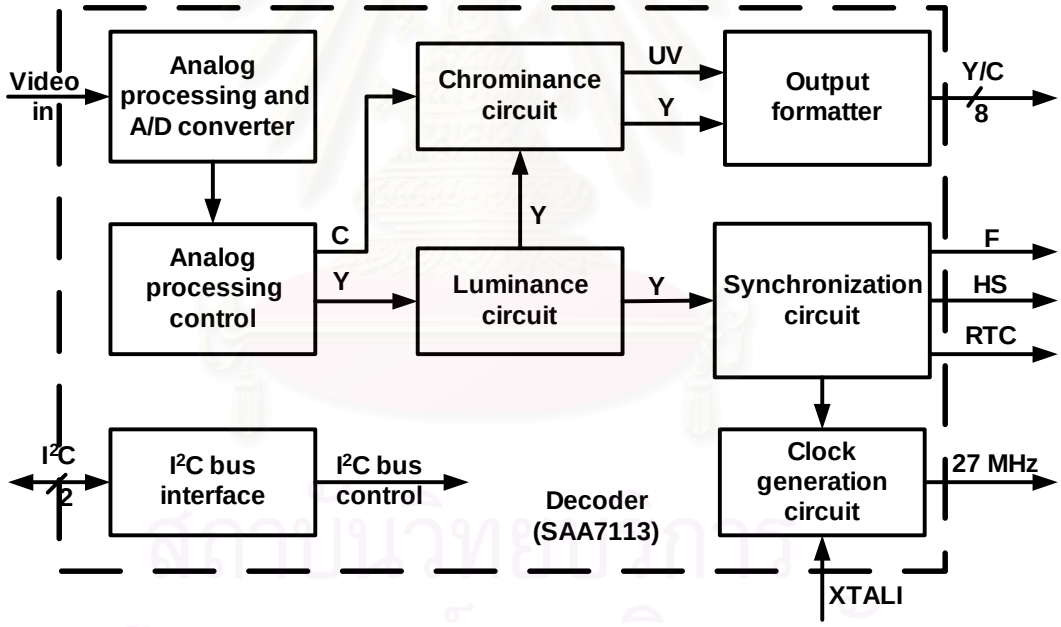
วงจรถอดรหัสสัญญาณวีดิทัศน์นี้ใช้ชิพเบอร์ SAA7113 ของบริษัทฟิลิปส์เซมิคอนดักเตอร์ ในการทำงานเพื่อแปลงสัญญาณวีดิทัศน์จากแบบแอนะล็อกให้เป็นข้อมูลแบบดิจิทัล โดยข้อมูลสัญญาณวีดิทัศน์ที่ได้ออกมานี้จะถูกแยกองค์ประกอบออกเป็น 3 สัญญาณ คือ สัญญาณความสว่าง (Y) และสัญญาณสี 2 สัญญาณ (C) สัญญาณทั้งสามจะส่งสลับกันออกมาผ่านทางสัญญาณ Y/C ตามมาตรฐาน ITU-R BT.656 ด้วยความถี่ 27 MHz ทำให้วงจรต้องสังเคราะห์สัญญาณนาฬิกาความถี่ 27 MHz เพิ่มออกมาด้วยเพื่อให้ตัวรับข้อมูลนำไปใช้ประสานเวลา (synchronize) กับข้อมูลวีดิทัศน์ที่รับไป

นอกจากสร้างสัญญาณ Y/C แล้วชิพนี้ยังแยกสัญญาณประสานเวลาอื่นๆ ที่จำเป็นออกมาด้วย ได้แก่ สัญญาณซิงค์แนวนอน (HS) เพื่อบอกจังหวะการขึ้นเส้นภาพใหม่ สัญญาณบอกฟิลด์ของภาพ (F) ซึ่งใช้ตรวจสอบจังหวะการเปลี่ยนฟิลด์ของภาพ สัญญาณทั้งสองนี้จะถูกส่งไปให้กับวงจรควบคุมการขยายภาพเพื่อใช้ในกระบวนการขยายภาพต่อไป และยังสร้างสัญญาณควบคุมเวลาจริง (RTC: Real-time control) ให้กับวงจรเข้ารหัสสัญญาณวีดิทัศน์ด้วย เพื่อบอกสภาวะการทำงานของวงจรเฟสล็อกลูปภายในชิพ ได้แก่ ความถี่และเฟสจริงๆ ของสัญญาณพาหะรอง 4.43 MHz ความถี่จริงของสัญญาณซิงโครไนซ์ต่างๆ รวมถึงความถี่ของสัญญาณนาฬิกาของระบบที่ใช้ในวงจรถอดรหัส ข้อมูลเหล่านี้จะทำให้วงจรเข้ารหัสสามารถนำไปประมวลผลเพื่อสร้างสัญญาณวีดิทัศน์กลับมาให้ใกล้เคียงกับสัญญาณวีดิทัศน์ขาเข้าเดิมมากที่สุด

การทำงานของวงจรถอดรหัสสัญญาณวีดิทัศน์นี้จะสามารถกำหนดได้โดยการโปรแกรมค่าคงที่ไปยังรีจิสเตอร์ต่างๆ ที่อยู่ภายในชิพผ่านบัส I^2C การปรับเปลี่ยนค่าในรีจิสเตอร์เหล่านี้ทำให้เราสามารถปรับการทำงานของชิพให้เหมาะสมกับงานของเราได้ เช่น

- กำหนดรูปแบบสัญญาณขาเข้า ได้แก่ สัญญาณวีดีโอรวม (composite video) หรือ สัญญาณความสว่าง/สัญญาณสี (S-Video)
- กำหนดช่องสัญญาณวีดิทัศน์ที่จะทำงาน ซึ่งจะมีอยู่ทั้งหมด 4 ช่องสัญญาณ
- กำหนดระบบสัญญาณวีดิทัศน์ ได้แก่ ระบบ PAL หรือ NTSC
- กำหนดตำแหน่งและความกว้างของสัญญาณซิงโครไนซ์
- กำหนดคุณสมบัติต่างๆ ให้กับวงจรขยายและวงจรกรองภายในชิพเพื่อปรับคุณสมบัติให้เหมาะสมกับลักษณะของสัญญาณขาเข้าซึ่งเปลี่ยนแปลงตามการใช้งาน

การโปรแกรมค่าในรีจิสเตอร์เหล่านี้เป็นหน้าที่ของวงจรควบคุมการขยายภาพซึ่งจะกล่าวในรายละเอียดในเนื้อหาของบทต่อไป



รูปที่ 3-4 โครงสร้างภายในวงจรถอดรหัสสัญญาณวีดิทัศน์

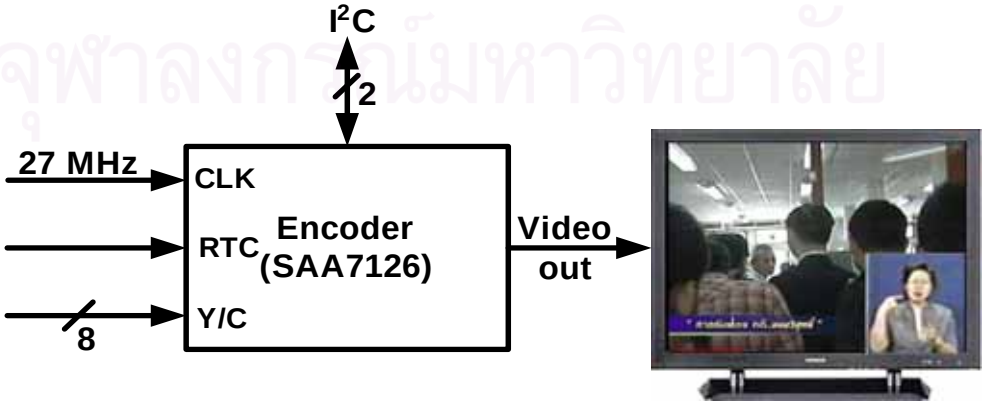
การทำงานภายในวงจรถอดรหัสสัญญาณวีดิทัศน์สามารถแสดงได้ดังรูปที่ 3-4 สัญญาณวีดิทัศน์จากทั้งหมด 4 ช่องจะถูกเลือกมา 1 ช่องตามที่เลือกไว้ แล้วสัญญาณวีดิทัศน์ขาเข้า (Video in) จะถูกส่งมาตั้งระดับไฟตรงด้วยวงจรแคลมป์ ถูกปรับขนาดด้วยวงจรปรับอัตราขยายอัตโนมัติ (AGC: Automatic gain control) และกรองความถี่ให้เหมาะสมโดยวงจรวงจรกรองสัญญาณก่อนการสุ่ม (Anti-aliasing Filter) ที่อยู่ภายในวงจรประมวลผลสัญญาณแอนะล็อกขาเข้า (Analog

input processing) หลังจากนั้นจึงนำสัญญาณไปแปลงรูปแบบให้เป็นสัญญาณแบบดิจิตัลตามมาตรฐาน ITU-R BT.601 ที่ตัวแปลงแอนะล็อกเป็นดิจิทัล (A/D converter) ข้อมูลภาพขนาด 9 บิตที่ได้ออกมาจะถูกนำมาแยกองค์ประกอบเป็นสัญญาณความสว่างและสัญญาณสีที่วงจรควบคุมการประมวลผลสัญญาณแอนะล็อก (Analog processing control)

สัญญาณสีจะถูกดีมอดูเลตและแยกออกเป็น 2 องค์ประกอบ คือ R-Y (C_R หรือ V) และ B-Y (C_B หรือ U) ที่วงจรสัญญาณสี (Chrominance circuit) โดยสัญญาณทั้งสองนี้จะถูกบีบอัดข้อมูลให้ส่งออกมาด้วยความถี่ 6.75 MHz ส่วนสัญญาณความสว่างจะถูกส่งไปประมวลผลที่วงจรสัญญาณความสว่าง (Luminance circuit) ซึ่งจะสร้างข้อมูลออกมาด้วยความถี่ที่สูงเป็น 2 เท่าของสัญญาณสี คือ 13.5 MHz ทั้งนี้เนื่องจากตาของคนเราจะไวต่อความสว่างของภาพมากกว่าสีของภาพ ข้อมูลสัญญาณความสว่างจะถูกส่งมาปรับความสว่าง (brightness) และค่าความต่าง (contrast) ที่วงจรสีซึ่งมีทำหน้าที่ปรับความอิ่มตัว (saturation) ของสัญญาณสีด้วย จากนั้นตัวจัดรูปแบบสัญญาณออก (output formatter) จะทำหน้าที่จัดลำดับการส่งสัญญาณให้เป็นลำดับ C_B, Y, C_R, Y ด้วยรูปแบบ 4:2:2 (Y:C_B:C_R)

ส่วนสัญญาณประสานเวลาต่างๆ เช่น สัญญาณซิงโครไนซ์ทางแนวนอน (HS) สัญญาณบอกฟิลด์ (F) สัญญาณควบคุมเวลาจริง (RTC) จะถูกสร้างที่วงจรซิงโครไนซ์ (synchronization circuit) ส่วนวงจรเฟสล็อกลูปที่มีอยู่ภายในวงจรสร้างสัญญาณนาฬิกา (Clock generation circuit) ซึ่งต่ออยู่กับคริสตอล จะสร้างสัญญาณนาฬิกา 27 MHz ออกมาเพื่อป้อนให้วงจรอื่นๆ ต่อไป ในขณะที่วงจรต่อประสานบัส I^2C (I^2C bus interface) จะอ่านข้อมูลสำหรับโปรแกรมค่าในรีจิสเตอร์ต่างๆ จากบัสมาประมวลผล แล้วส่งสัญญาณไปควบคุมการทำงานของวงจรส่วนต่างๆ ตามที่ผู้ใช้กำหนด

3.2 **วงจรเข้ารหัสสัญญาณวีดิทัศน์** (Video encoder)

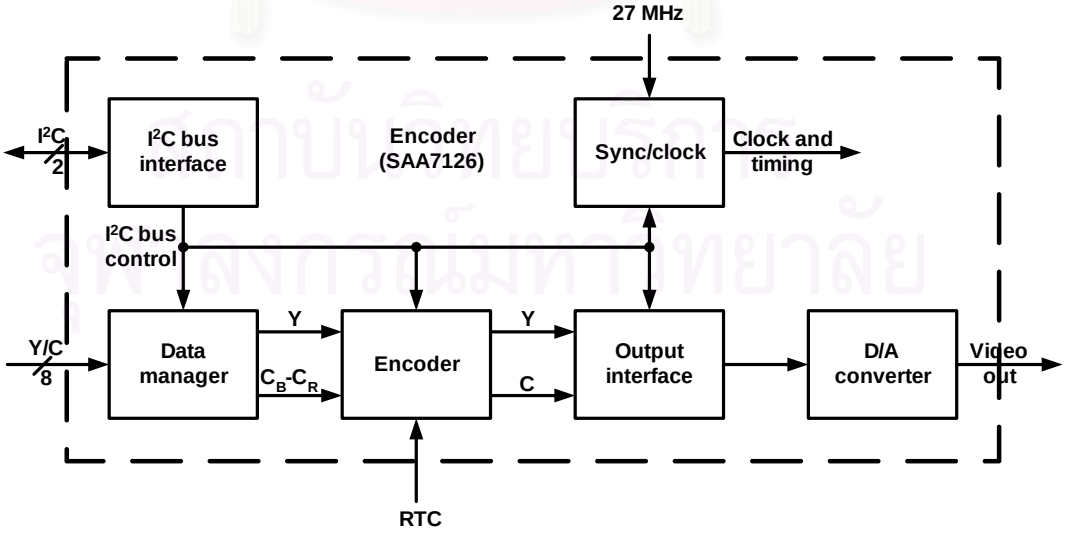


รูปที่ 3-5 การใช้งานวงจรเข้ารหัสสัญญาณวีดิทัศน์

วงจรเข้ารหัสสัญญาณวีดิทัศน์นี้เลือกใช้ชิพเบอร์ SAA7126 ของบริษัทฟิลิปส์เซมิคอนดักเตอร์ในการทำงาน โดยชิพจะรับข้อมูลสัญญาณวีดิทัศน์ขนาด 8 บิต (Y/C) มาจากวงจรควบคุมการขยายภาพซึ่งเป็นข้อมูลภาพรายการเดิมแต่มีภาพภาษามือที่ผ่านการขยายจากวงจรขยายภาพแล้วแทรกมาด้วยพร้อมทั้งรับสัญญาณนาฬิกามาด้วยเพื่อบอกจังหวะการส่งข้อมูล นอกจากนี้วงจรจะอ่านข้อมูล RTC จากวงจรถอดรหัสสัญญาณวีดิทัศน์มาด้วยเพื่อนำไปช่วยในการสร้างสัญญาณวีดิทัศน์แอนะล็อกให้ได้ใกล้เคียงกับสัญญาณเดิมมากที่สุด สัญญาณวีดิทัศน์ที่ได้ออกมาจากวงจรนี้จะป้อนให้กับเครื่องรับโทรทัศน์ต่อไป ซึ่งจะได้รูปภาพที่มีภาพภาษามือขนาดใหญ่ออกมาดังแสดงในรูปที่ 3-5

การควบคุมการทำงานของวงจรเข้ารหัสสัญญาณวีดิทัศน์นี้จะเหมือนกับวงจรถอดรหัสสัญญาณวีดิทัศน์คือ ภายในชิพจะมีรีจิสเตอร์ภายในซึ่งสามารถโปรแกรมค่าผ่านบัส I^2C เพื่อปรับเปลี่ยนการทำงานของชิพได้ เช่น

- กำหนดระบบสัญญาณวีดิทัศน์ที่รับมาว่าเป็นระบบ PAL หรือ NTSC
- กำหนดรูปแบบของสัญญาณขาออกว่าจะเป็นสัญญาณวีดีโอรวมหรือเป็นสัญญาณแม่สี 3 สี คือ สีแดง สีเขียวและสีน้ำเงิน
- กำหนดคุณสมบัติของวงจรขยายภายในชิพนี้
- กำหนดให้สัญญาณขาออกเป็นภาพที่เราป้อนไปหรือเป็นสัญญาณภาพทดสอบรูปแท่งสี 8 แท่งซึ่งจะทำงานได้โดยไม่ต้องป้อนข้อมูลภาพเข้าไปเลยสำหรับใช้ในการตรวจสอบการต่อวงจรภายนอกได้ว่ามีข้อผิดพลาดเกิดขึ้นหรือไม่

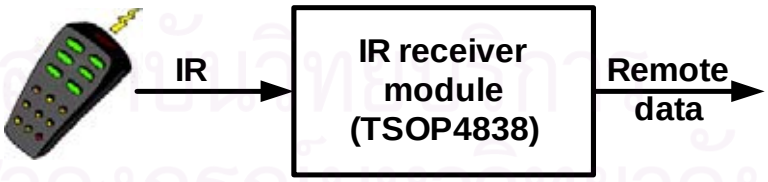


รูปที่ 3-6 โครงสร้างภายในวงจรเข้ารหัสสัญญาณวีดิทัศน์

กระบวนการสร้างสัญญาณวีดิทัศน์แบบแอนะล็อกของวงจรเข้ารหัสสัญญาณวีดิทัศน์นี้สามารถแสดงได้ดังรูปที่ 3-6 ข้อมูลภาพ (Y/C) ที่รับมาจากวงจรควบคุมการขยายภาพจะถูกแยกประเภทเป็นสัญญาณความสว่าง (Y) และสัญญาณสี (C_B, C_R) ที่ตัวจัดการข้อมูล (data manager) ซึ่งนอกจากจะแยกสัญญาณแล้วภายในวงจรนี้ยังมีตารางสำหรับเก็บข้อมูลซึ่งเป็นค่าคงที่ชุดหนึ่งไว้เพื่อใช้ในการสร้างสัญญาณภาพทดสอบรูปแท่งสี 8 แท่ง ตัวจัดการข้อมูลจะเลือกข้อมูลภาพอย่างใดอย่างหนึ่งส่งออกมาให้วงจรเข้ารหัส (encoder) เพื่อปรับคุณสมบัติของสัญญาณ เช่น ปรับอัตราขยายด้วยวงจร AGC ปรับความถี่เพื่อลดสัญญาณรบกวนด้วยวงจรกรอง ชักตัวอย่างสัญญาณเพิ่มให้สัญญาณความสว่างและสัญญาณสีมีความถี่เป็น 27 Msample/s เท่ากันหมด เป็นต้น สัญญาณความสว่างและสีจะถูกส่งต่อมาให้วงจรต่อประสานภายนอก (output interface) ซึ่งทำงานร่วมกับตัวแปลงดิจิทัลเป็นแอนะล็อก (D/A converter) ความละเอียด 10 บิตเพื่อรวมสัญญาณความสว่างและสัญญาณสีเข้าด้วยกันแล้วสร้างเป็นสัญญาณวีดิทัศน์แอนะล็อกออกมา

วงจรต่อประสานบัส I^2C (I^2C bus interface) จะทำหน้าที่เหมือนกับวงจรนี้ในชิพวงจรถอดรหัสสัญญาณวีดิทัศน์คือ ใช้อ่านค่าที่ส่งมาโปรแกรมรีจิสเตอร์ทางบัส I^2C แล้วส่งสัญญาณมาควบคุมการทำงานวงจรให้ถูกต้อง ส่วนวงจรซิงค์และวงจรสัญญาณนาฬิกา (Sync/clock) จะใช้สร้างสัญญาณซิงโครไนซ์ต่างๆ ในสัญญาณวีดิทัศน์เพื่อนำไปผสมกับสัญญาณความสว่างและสัญญาณสีแล้วสร้างเป็นสัญญาณวีดิทัศน์ที่สมบูรณ์ โดยข้อมูลของสัญญาณซิงโครไนซ์จะถูกถอดรหัสมาจากจากข้อมูลภาพขาเข้า (Y/C) ซึ่งจะมีการส่งสัญญาณซิงค์ต่างๆ มาด้วย

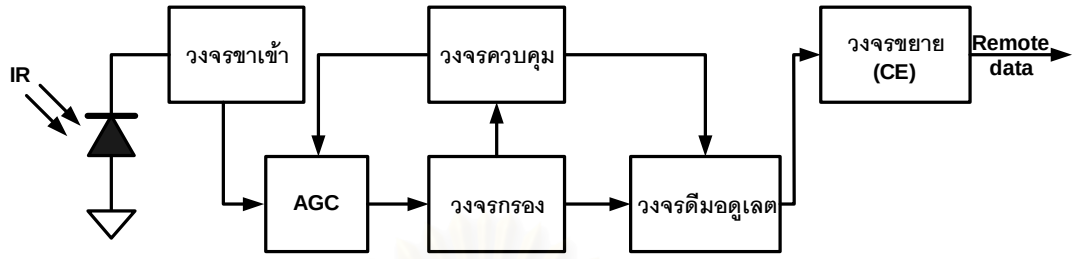
3.3 **วงจรรับสัญญาณอินฟราเรด** (IR receiver module)



รูปที่ 3-7 การใช้งานวงจรรับสัญญาณอินฟราเรด

วงจรรับสัญญาณอินฟราเรดซึ่งเลือกใช้ชิพเบอร์ TSOP4838 ทำงานนี้จะรับสัญญาณอินฟราเรดที่ส่งมาจากรีโมตคอนโทรลเพื่อดีมอดูเลตความถี่พาหะทิ้งไปให้เหลือแต่สัญญาณเบสแบนด์ โดยชิพนี้จะใช้กับความถี่พาหะ 38 kHz ส่วนข้อมูลขาออกนี้จะมีความเร็วสูงสุดได้ 800 kbit/s สัญญาณเบสแบนด์ที่ได้ออกมาเป็นข้อมูลแบบดิจิทัลซึ่งจะส่งออกมาแบบอนุกรมผ่านทาง

สัญญาณข้อมูลรีโมต (Remote data) ข้อมูลที่ได้ออกมานี้จะถูกป้อนให้กับวงจรควบคุมการขยายภาพเพื่อจะนำไปตรวจสอบปุ่มของรีโมตที่ถูกกดหรือคำสั่งของผู้ใช้ต่อไป



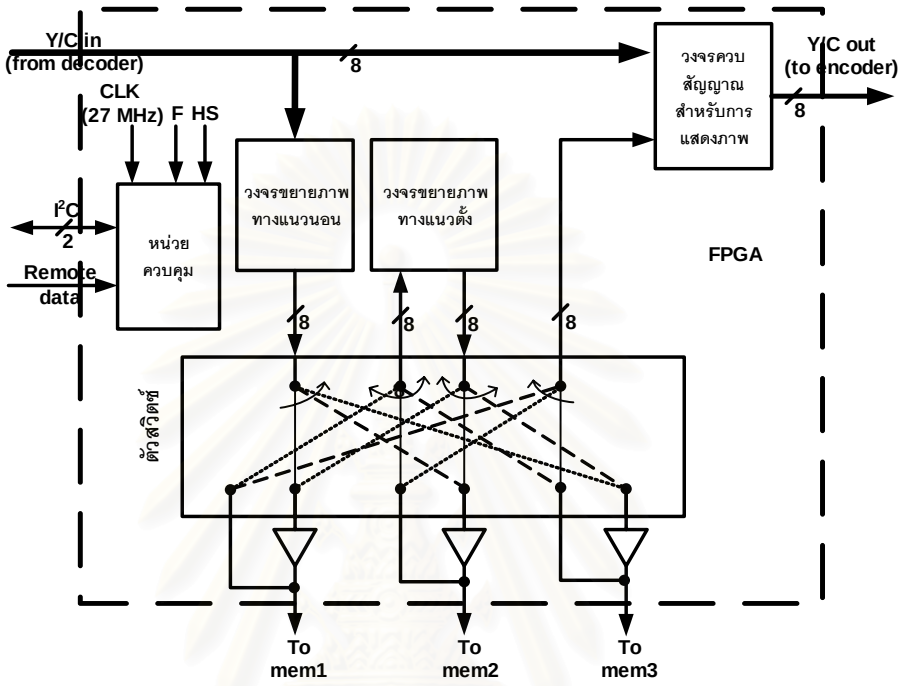
รูปที่ 3-8 การทำงานภายในวงจรรับสัญญาณอินฟราเรด

การทำงานถอดข้อมูลของวงจรรับสัญญาณอินฟราเรดแสดงดังรูปที่ 3-8 ตัวตรวจรับแสงซึ่งจะมีค่ากระแสเปลี่ยนแปลงตามความเข้มของแสงอินฟราเรดนั้นจะทำงานร่วมกับวงจรขาเข้า (input) เพื่อตัดสินว่ามีข้อมูลส่งมาหรือยัง หากพบข้อมูลจริงแล้วสัญญาณที่รับได้จะถูกส่งไปยังวงจร AGC เพื่อขยายขนาดของสัญญาณให้เหมาะสม ผ่านวงจรกรองเพื่อลดสัญญาณรบกวนที่เกิดขึ้น และผ่านวงจรสำหรับดีมอดูเลตเพื่อแยกสัญญาณพาหะทิ้งไปให้เหลือแต่ส่วนที่เป็นข้อมูลจริงส่งออกไปยังวงจรขยายเพื่อปรับให้ข้อมูลเป็นสัญญาณแบบดิจิทัลโดยแทนค่า ‘1’ ด้วยระดับสัญญาณขนาด 5 V และ ‘0’ ด้วยขนาดสัญญาณ 0 V การทำงานของวงจรย่อยๆ ในชิพนี้จะถูกควบคุมด้วยวงจรควบคุมภายในอีกที

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

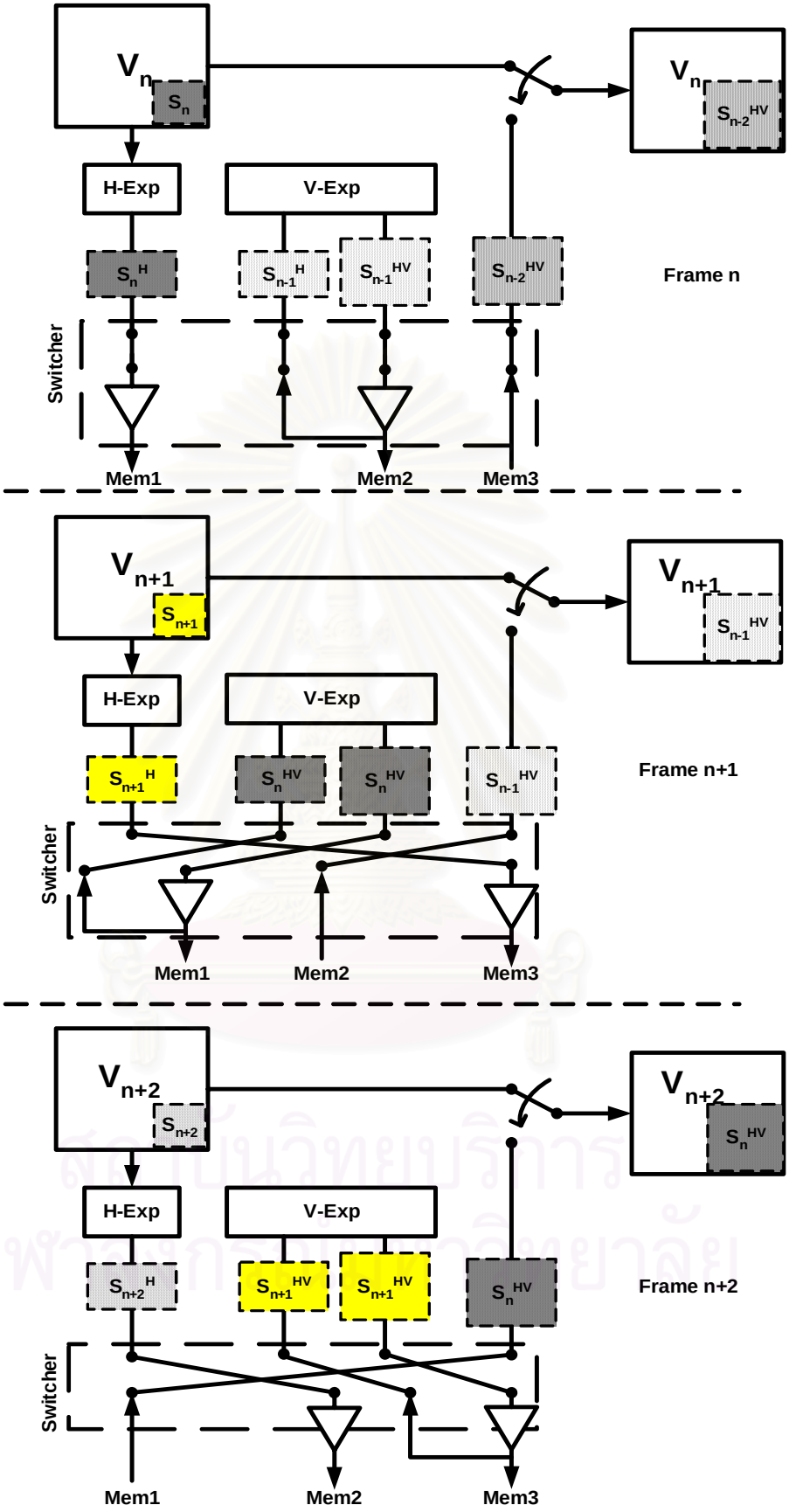
วงจรควบคุมการขยายภาพ



รูปที่ 4-1 โครงสร้างวงจรภายในของวงจรควบคุมการขยายภาพ

วงจรควบคุมการขยายภาพ (Expansion controller) ถูกออกแบบอยู่บนชิพ FPGA โดยมีโครงสร้างภายในดังแสดงในรูปที่ 4-1 ซึ่งจะมีหน่วยควบคุม (control unit) สำหรับสร้างสัญญาณควบคุมวงจรย่อยอื่นๆ ซึ่งประกอบด้วย วงจรขยายภาพ 2 วงจร คือ วงจรขยายภาพทางแนวนอน (Horizontal expander) และวงจรขยายภาพทางแนวตั้ง (Vertical expander) ตัวสวิตช์ (Switcher) สำหรับควบคุมเส้นทางข้อมูลของหน่วยความจำทั้งสาม และวงจรควบสัญญาณสำหรับการแสดงภาพ (Display multiplexer) ซึ่งใช้แทรกข้อมูลภาพที่ขยายแล้วลงไปในข้อมูลภาพเดิมเพื่อส่งไปให้วงจรเข้ารหัสสัญญาณวีดิทัศน์ต่อไป

วงจรขยายภาพทางแนวนอนและทางแนวตั้งนี้จะทำงานพร้อมกันแบบไปป์ไลน์ โดยวงจรขยายภาพทางแนวนอนจะขยายข้อมูลภาพภาษามือในเฟรมปัจจุบัน ส่วนวงจรขยายภาพทางแนวตั้งจะขยายข้อมูลภาพภาษามือในเฟรมที่แล้วที่ผ่านการขยายแนวนอนมาก่อนแล้ว วงจรทั้งสองนี้จะรับส่งข้อมูลภาพกับหน่วยความจำผ่านทางตัวสวิตช์ซึ่งมีการทำงานเปลี่ยนแปลงตามเฟรมภาพดังแสดงในรูปที่ 4-2



รูปที่ 4-2 การทำงานของตัวสวิตช์ในช่วง 3 เฟรมภาพติดต่อกัน V แทนภาพต้นฉบับทั้งภาพ S แทนส่วนที่เป็นภาพภาษามือ HV แทนการขยายภาพทางแนวนอนและทางแนวตั้งตามลำดับ

ข้อมูลจากวงจรขยายภาพจะถูกส่งผ่านตัวสวิตช์ซึ่งมีวงจรขาออกเป็นตัวบัฟเฟอร์ 3 สถานะ (tri-state buffer) สำหรับกันไม่ให้ข้อมูลจากปลาย 2 ด้านเกิดชนกัน ตัวสวิตช์นี้จะเป็นตัวเลือกเส้นทางข้อมูลจากวงจรขยายทั้งสองและวงจรควบสัญญาณสำหรับแสดงภาพเพื่อส่งไปยังหน่วยความจำทั้งสาม และจะมีการสลับเส้นทางกันเมื่อข้อมูลขาเข้าส่งมาครบ 1 เฟรมภาพ ข้อมูลจากวงจรขยายภาพทางแนวนอนจะส่งข้อมูลภาพเฟรมล่าสุดซึ่งผ่านการขยายทางแนวนอนแล้วให้กับหน่วยความจำที่ว่างอยู่ เช่น หน่วยความจำที่ 1 ในเฟรมที่ n ส่วนวงจรขยายภาพทางแนวตั้งจะรับข้อมูลภาพในเฟรมก่อนหน้านี้ 1 เฟรมจากหน่วยความจำที่มีข้อมูลภาพซึ่งผ่านการขยายทางแนวนอนมาแล้ว เช่น หน่วยความจำที่ 2 ในเฟรมที่ n มาประมวลผลเพื่อขยายภาพทางแนวตั้งจนครบทั้งภาพแล้วส่งไปเก็บในหน่วยความจำนั้นอีกครั้ง ในขณะเดียวกันภาพภาษามือจากหน่วยความจำที่เก็บภาพที่ขยายเสร็จแล้วใน 2 เฟรมก่อนหน้านี้ เช่น หน่วยความจำที่ 3 ในเฟรมที่ n จะถูกส่งผ่านตัวสวิตช์มาให้กับวงจรควบสัญญาณสำหรับการแสดงภาพเพื่อแทรกข้อมูลภาพภาษามือที่ขยายแล้วลงไปในภาพรายการเดิม

ลำดับการทำงานของตัวสวิตช์จะมีอยู่ 3 รูปแบบสลับเวียนกันไปเรื่อยๆ สามารถสรุปได้ดังตารางที่ 4-1

ตารางที่ 4-1 แสดงการทำงานของตัวสวิตช์

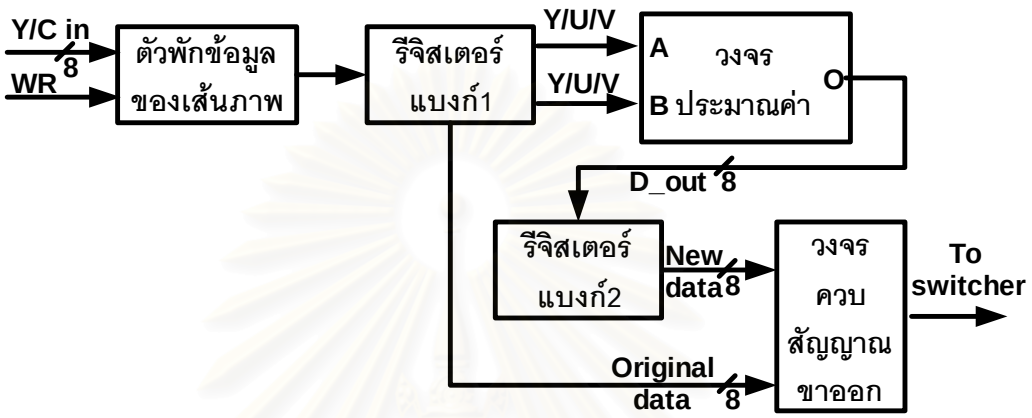
	เฟรมที่ n	เฟรมที่ n+1	เฟรมที่ n+2
วงจรขยายภาพทางแนวนอน	หน่วยความจำที่ 1	หน่วยความจำที่ 3	หน่วยความจำที่ 2
วงจรขยายภาพทางแนวตั้ง	หน่วยความจำที่ 2	หน่วยความจำที่ 1	หน่วยความจำที่ 3
วงจรแสดงภาพ	หน่วยความจำที่ 3	หน่วยความจำที่ 2	หน่วยความจำที่ 1

ด้านหน่วยควบคุม (control unit) จะทำหน้าที่ในการสร้างสัญญาณควบคุมต่างๆ ให้กับวงจรขยายภาพ ตัวสวิตช์ และสร้างสัญญาณแอดเดรสให้กับหน่วยความจำทั้งสามตัว โดยการทำงานของวงจรทั้งหมดนี้จะทำงานที่ความถี่ 27 MHz ซึ่งเป็นความถี่ของสัญญาณนาฬิกาที่รับมาจากวงจรถอดรหัสสัญญาณวีดิทัศน์ นอกจากนี้หน่วยควบคุมยังทำหน้าที่ส่งค่าสำหรับโปรแกรมรีจิสเตอร์ภายในวงจรถอดรหัสและวงจรเข้ารหัสสัญญาณวีดิทัศน์ผ่านบัส I^2C รวมถึงการถอดรหัสคำสั่งจากรีโมตคอนโทรล แล้วปรับสัญญาณควบคุมที่ส่งให้วงจรอื่นๆ ใหม่เพื่อเปลี่ยนการทำงานตามที่ผู้ใช้ต้องการ

4.1 วงจรขยายภาพ

วงจรขยายภาพจะมีอยู่ 2 ส่วนคือ วงจรขยายภาพทางแนวนอนและวงจรขยายภาพทางแนวตั้ง ซึ่งการออกแบบวงจรและการทำงานของวงจรทั้งสองนี้มีรายละเอียดดังต่อไปนี้

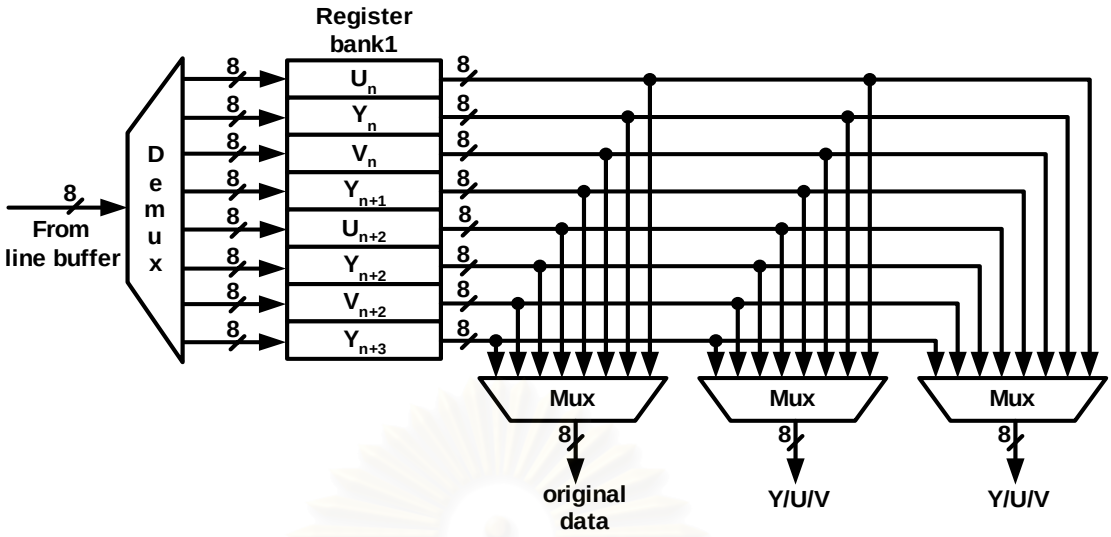
4.1.1 วงจรขยายภาพทางแนวนอน (Horizontal expander)



รูปที่ 4-3 วงจรขยายภาพทางแนวนอน

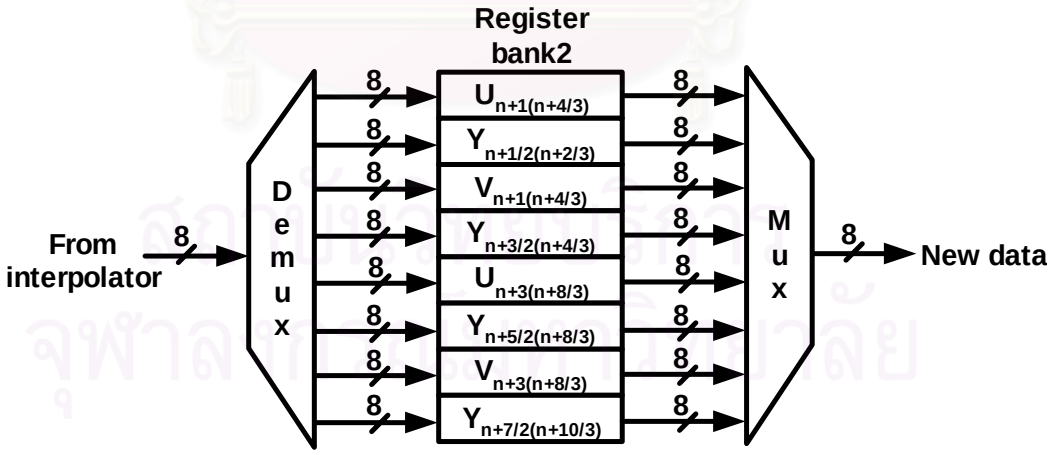
เนื่องจากข้อมูลขาเข้าและข้อมูลขาออกนั้นส่งด้วยความถี่ 27 MHz เท่ากัน แต่จำนวนข้อมูลขาออกนั้นจะมากกว่าเป็น 2 เท่าหรือ 1.5 เท่าตามอัตราขยายที่ผู้ใช้เลือก ดังนั้นภายในวงจรขยายทางแนวนอนจึงต้องมีตัวพักข้อมูลของเส้นภาพ (Line buffer) ที่ส่งเข้ามาในขณะที่วงจรด้านหลังยังทำงานขยายข้อมูลก่อนหน้านี้ไม่เสร็จ ตัวพักนี้ใช้แรมที่มีทางเข้าออก 2 ทาง (dual port RAM) ขนาด 512 ไบต์ที่มีอยู่แล้วในชิพ FPGA ของบริษัท Xilinx ดังนั้นตัวพักนี้จะสามารถเก็บข้อมูลภาพภาษามือขาเข้าได้ 2 เส้นภาพ (1 เส้นใช้เนื้อที่ 256 ไบต์) ในขณะที่ด้านหนึ่งของแรมกำลังเก็บข้อมูลภาพเส้นปัจจุบันอยู่ ข้อมูลของเส้นที่แล้วก็จะถูกส่งออกมาไปคำนวณเพื่อขยายภาพที่วงจรส่วนหลัง

ข้อมูลภาพที่ส่งมาจากแรมจะมีข้อมูลสัญญาณสีและสัญญาณความสว่างส่งรวมกันมาเป็นลำดับ C_B, Y, C_R, Y การขยายภาพแบบไบลิเนียร์จะนำสัญญาณประเภทเดียวกัน 2 ตัวที่อยู่ติดกันมาคำนวณที่วงจรประมาณค่า (Interpolator) ดังนั้นข้อมูลภาพที่ส่งมาจะถูกเก็บไว้ในรีจิสเตอร์แบงค์ 1 ก่อนเพื่อรอข้อมูลประเภทเดียวกันให้ครบ 2 ตัวก่อน เช่นเดียวกับผลลัพธ์ที่ได้จากวงจรประมาณค่าก็จะถูกนำไปเก็บไว้ชั่วคราวที่รีจิสเตอร์แบงค์ 2 ก่อนเพื่อให้วงจรควบสัญญาณขาออก (output mux) สามารถจัดลำดับการส่งสัญญาณขาออกได้ โครงสร้างภายในรีจิสเตอร์แบงค์ทั้งสองนี้แสดงดังรูปที่ 4-4 และรูปที่ 4-5



รูปที่ 4-4 โครงสร้างรีจิสเตอร์แบงค์ 1

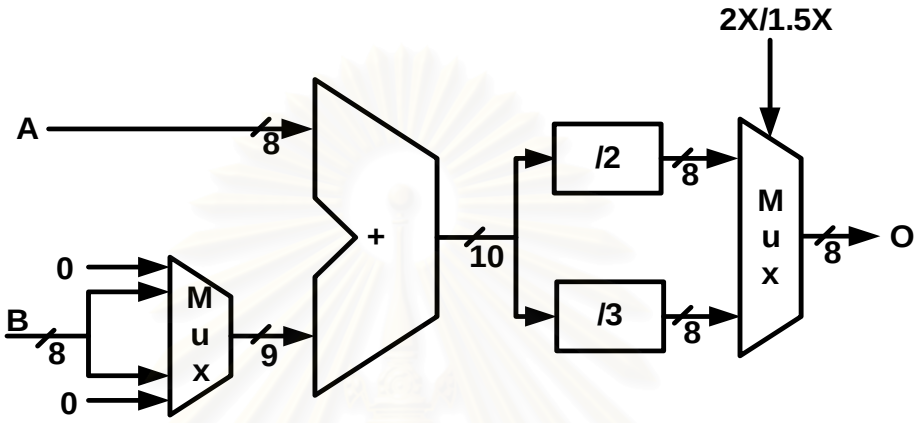
รีจิสเตอร์แบงค์ 1 นั้นใช้เก็บข้อมูลขาเข้าชั่วคราวเพื่อรอการคำนวณที่วงจรประมาณค่า ซึ่งการคำนวณข้อมูลแต่ละประเภทจะต้องรอข้อมูลให้ครบ 2 ชุด แต่ข้อมูลสัญญาณสีนั้นจะส่งมาเพียง 2 พิกเซลต่อ 1 สัญญาณ ดังนั้นการประมาณค่าของสัญญาณสีจะเกิดขึ้นได้เมื่อมีข้อมูลเข้ามาครบ 4 พิกเซลหรือ 8 ไบต์นั่นเอง ทำให้รีจิสเตอร์ในแบงค์ 1 จึงต้องมีทั้งหมด 8 ตัว ข้อมูลที่ส่งมาให้รีจิสเตอร์ทั้ง 8 นี้จะมาจากตัวพักข้อมูลผ่านการเลือกเส้นทางด้วยตัวดีมักซ์ ส่วนข้อมูลขาออกของรีจิสเตอร์แบงค์นี้มีทั้งหมด 3 สัญญาณ คือ ส่งให้วงจรประมาณค่า 2 สัญญาณ และวงจรควบสัญญาณขาออก 1 สัญญาณ ทำให้ต้องมีมักซ์ที่ด้านออกทั้งหมด 3 ตัวในวงจรนี้



รูปที่ 4-5 โครงสร้างรีจิสเตอร์แบงค์ 2

รีจิสเตอร์ในแบงค์ 2 ตัวเลขที่ห้อยอยู่นอกวงเล็บเป็นค่าเมื่อวงจรขยายภาพทำงานแบบขยาย 2 เท่า ส่วนตัวเลขในวงเล็บจะเป็นค่าเมื่อทำงานแบบขยาย 1.5 เท่า รีจิสเตอร์ในแบงค์ 2 จะใช้เก็บข้อมูลที่ถูกขยายโดยวงจรประมาณค่า การขยายภาพขนาด 2 เท่าและ 1.5 เท่าจะสร้างข้อมูลใหม่

มาจำนวนเท่ากับข้อมูลขาเข้า เช่น ขนาด 2 เท่าจะขยายข้อมูลจาก 4 พิกเซลเป็น 8 พิกเซล ดังนั้น 4 พิกเซลที่เกิดขึ้นใหม่ซึ่งสร้างจากวงจรประมาณค่านี้จะถูกเก็บไว้ในรีจิสเตอร์แบงค์ 2 ก่อนเพื่อรอให้วงจรควบสัญญาณขาออกสามารถนำข้อมูลเก่าและใหม่มาเรียงใหม่ให้ถูกต้อง ทำให้รีจิสเตอร์แบงค์ 2 ต้องมีขนาด 8 ไบต์เท่ากับแบงค์ 1 ด้านข้อมูลที่จะไหลเข้าและออกจากรีจิสเตอร์แบงค์นี้มีอย่างละ 1 เส้น ดังนั้นจึงใช้มักซ์และดีมักซ์เพียงตัวเดียวเท่านั้น

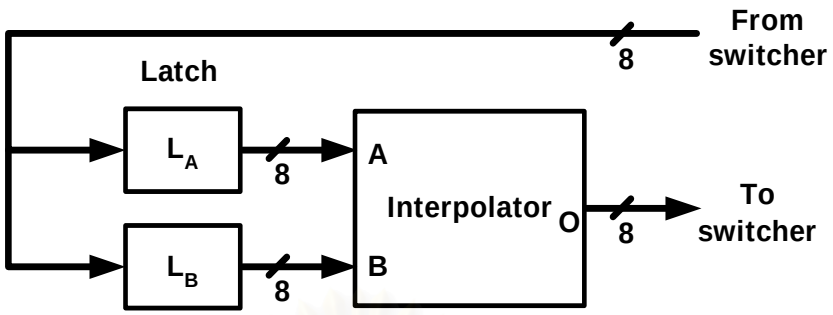


รูปที่ 4-6 วงจรประมาณค่า (Interpolator)

วงจรประมาณค่ามีการคำนวณอยู่ 2 แบบ คือ คำนวณเพื่อขยาย 2 เท่า (ตามสมการที่ 2.15) ซึ่งจะใช้สูตร (A+B)/2 และแบบ 1.5 เท่า (ตามสมการที่ 2.16) ด้วยสูตร (A+2B)/3 ดังนั้นข้อมูลขาเข้าของสัญญาณ B จะมีทั้งแบบคูณ 2 คือ การเติม ‘0’ ไว้ที่บิต LSB เมื่อทำงานแบบขยาย 1.5 เท่า และการปล่อยผ่านข้อมูลเดิมโดยการเติม ‘0’ ไว้ที่ MSB เมื่อทำงานแบบขยาย 2 เท่า ข้อมูลที่ได้จากวงจรบวกจะส่งไปยังวงจรหาร 2 และหาร 3 โดยวงจรหาร 2 คือการเลื่อนบิตข้อมูลมาทางขวา 1 ครั้ง ส่วนการหาร 3 จะใช้การเปิดตารางซึ่งสร้างจากหน่วยความจำภายในขนาด 1024 ไบต์ เนื่องจากวิธีนี้จะสามารถส่งผลลัพธ์ออกมาได้ภายใน 1 สัญญาณนาฬิกาพร้อมกับผลลัพธ์จากวงจรหาร 2 พอดี วงจรควบสัญญาณจะส่งข้อมูลจากวงจรหาร 2 ออกไปเมื่อขยายแบบ 2 เท่า ในทางกลับกันก็จะปล่อยสัญญาณจากวงจรหาร 3 ออกไปเมื่อขยายภาพแบบ 1.5 เท่า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.1.2 **วงจรขยายภาพทางแนวตั้ง** (Vertical expander)



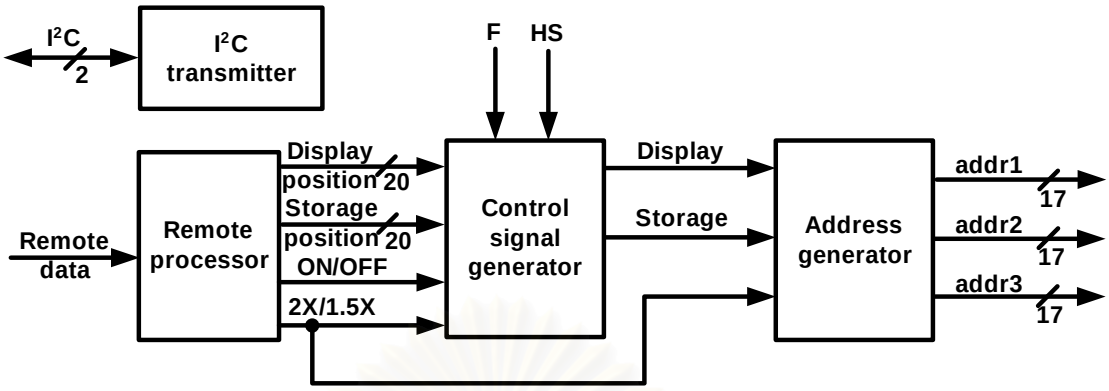
รูปที่ 4-7 วงจรขยายภาพทางแนวตั้ง

รูปที่ 4-7 เป็นโครงสร้างภายในวงจรขยายภาพทางแนวตั้งซึ่งจะนำข้อมูลภาพที่ผ่านการขยายทางแนวนอนมาแล้วมาขยายทางแนวตั้งต่อเพื่อให้ได้ภาพขยายที่สมบูรณ์ การทำงานของวงจรจะแบ่งได้เป็น 4 ขั้นตอน คือ อ่านข้อมูลชุดที่ 1 จากตัวสวิตช์มาเก็บไว้ในวงจรแลตช์ ขั้นตอนที่ 2 คือ อ่านข้อมูลชุดที่ 2 จากตัวสวิตช์ซึ่งเป็นข้อมูลที่อยู่ในตำแหน่งที่ติดกับข้อมูลแรกทางแนวตั้งมาเก็บไว้ในวงจรแลตช์อีกตัว แล้วนำมาประมาณค่าตามสมการขยายภาพในขั้นตอนที่ 3 สุดท้ายคือเขียนข้อมูลที่คำนวณได้ไปให้หน่วยความจำผ่านตัวสวิตช์

วงจรประมาณค่าในวงจรนี้จะเหมือนกับในวงจรขยายภาพทางแนวนอน คือ มีโครงสร้างดังในรูปที่ 4-6 เพราะสมการการขยายภาพทางแนวตั้งมีรูปแบบเหมือนกับการขยายทางแนวนอน คือประกอบด้วยวงจรบวกและวงจรหาร

กระบวนการขยายภาพที่ออกแบบนี้จะต้องทำงานถึง 4 ขั้นตอนตามลำดับและไม่สามารถทำงานพร้อมกันได้หมดเหมือนการขยายภาพแนวนอน เพราะบัสข้อมูลสำหรับอ่านและส่งข้อมูลเป็นบัสของหน่วยความจำตัวเดียวกัน ในขณะที่บัสข้อมูลขาเข้าและออกของการขยายภาพแนวนอนจะแยกกันเพราะจะอ่านข้อมูลจากวงจรถอดรหัสและส่งข้อมูลออกไปให้หน่วยความจำ ดังนั้นการขยายภาพทางแนวตั้งจะเสียเวลามากจากขั้นตอนที่มากกว่า นอกจากนี้จำนวนข้อมูลที่ต้องขยายของทางแนวตั้งก็มากกว่าแนวนอนเป็น 2 เท่าหรือ 1.5 เท่าเพราะต้องขยายภาพที่ถูกขยายทางแนวนอนมาแล้ว

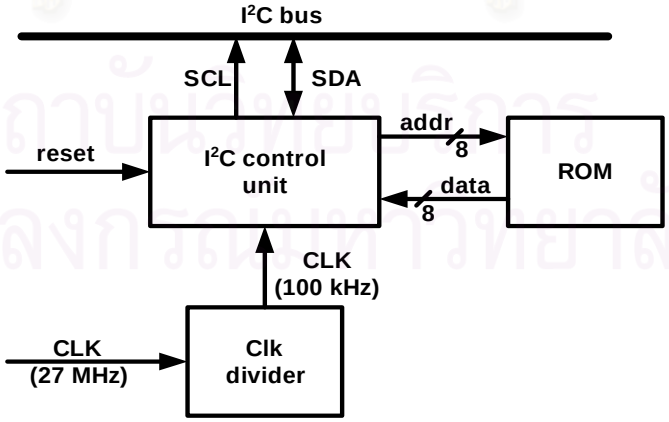
4.2 **หน่วยควบคุม** (Control unit)



รูปที่ 4-8 หน่วยควบคุม

หน่วยควบคุมประกอบด้วยวงจรย่อยๆ 4 วงจร ได้แก่ ตัวส่งสัญญาณผ่าน I^2C (I^2C transmitter) ทำหน้าที่ส่งค่าเริ่มต้นของวงจรถอดรหัสและวงจรเข้ารหัสสัญญาณวีดิทัศน์ผ่านบัส I^2C ตัวประมวลผลสัญญาณรีโมต (Remote processor) ซึ่งจะถอดรหัสสัญญาณจากรีโมตคอนโทรลว่าปุ่มที่ถูกกดนี้คือคำสั่งใดแล้วปรับเปลี่ยนสัญญาณออกใหม่ เช่น ปรับตำแหน่งที่จะเก็บภาพ ตำแหน่งที่จะแสดงภาพภาษามือ เป็นต้น ตัวสร้างสัญญาณควบคุม (Control signal generator) เพื่อควบคุมจังหวะการเก็บภาพและการแสดงภาพของวงจรต่างๆ และวงจรสุดท้ายคือ ตัวสร้างสัญญาณแอดเดรส (Address generator) ซึ่งจะสร้างค่าแอดเดรสให้กับหน่วยความจำทั้ง 3 ตัว

4.2.1 **ตัวส่งสัญญาณผ่าน** I^2C (I^2C transmitter)



รูปที่ 4-9 ตัวส่งสัญญาณผ่าน I^2C

ตัวส่งสัญญาณผ่าน I^2C จะทำหน้าที่เป็นตัวแม่ในบัส I^2C ซึ่งมีหน้าที่สร้างสัญญาณนาฬิกาผ่าน SCL และสร้างสัญญาณควบคุมต่างๆ ในบัส SDA ด้วย การทำงานเป็นแบบตัวแม่เป็นตัวส่ง

(Master-transmitter) เพราะการติดต่อที่เกิดขึ้นนี้เพื่อโปรแกรมค่ารีจิสเตอร์ภายในชิพถอดรหัสและเข้ารหัส

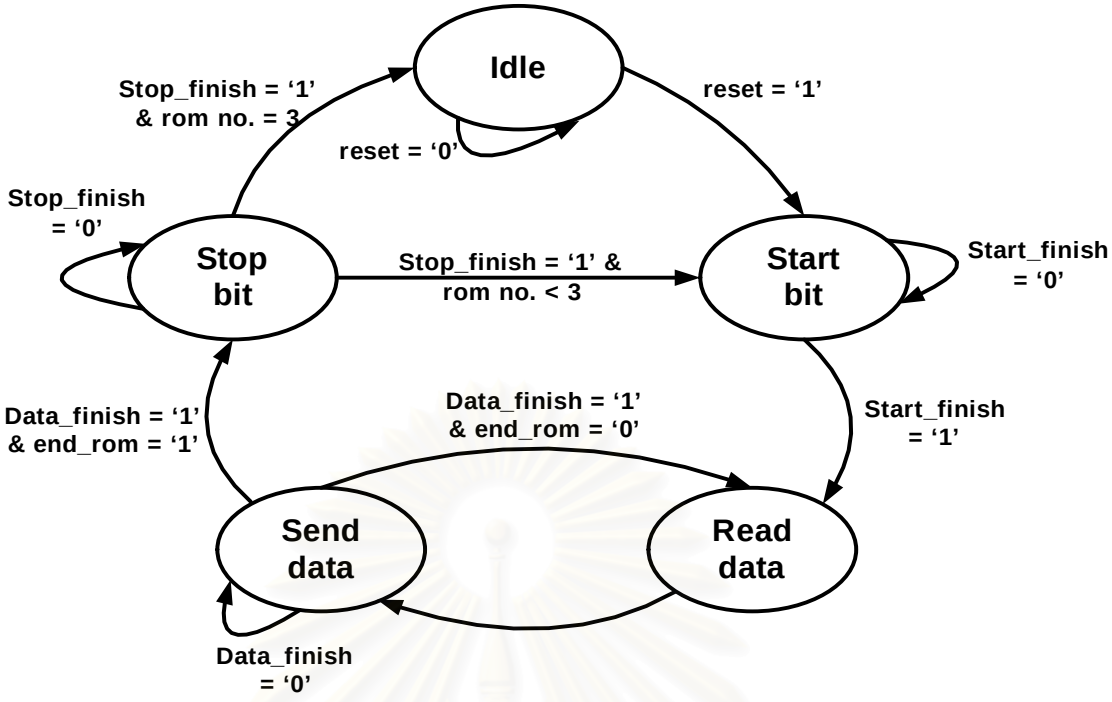
วงจรภายในตัวส่งสัญญาณ I^2C จะแบ่งเป็น 3 ส่วน คือ หน่วยควบคุม I^2C (I^2C Control unit) ซึ่งจะเป็นตัวสร้างสัญญาณที่จะส่งออกไปที่บัส I^2C ทั้งหมด วงจรหารความถี่ (Clk divider) สำหรับสร้างสัญญาณนาฬิกาความถี่ 100 kHz ซึ่งจะนำมาใช้งานภายในตัวส่ง I^2C นี้ และหน่วยความจำแบบรอมซึ่งใช้เก็บข้อมูลที่จะส่งไปโปรแกรมให้กับรีจิสเตอร์ทั้งหมด

รอมในตัวส่ง I^2C นี้มีขนาด 256 ไบต์ ถูกออกแบบโดยใช้แรมที่มีอยู่ภายในชิพ FPGA ข้อมูลที่เก็บในรอมนี้จะแบ่งออกเป็น 4 ส่วนเนื่องจากรีจิสเตอร์ในชิพถอดและเข้ารหัสมีตำแหน่งแอดเดรสสำหรับเขียนค่าเพื่อโปรแกรมการทำงานของชิพไม่เรียงกันต่อเนื่อง กล่าวคือ ในชิพแต่ละตัวจะมีแอดเดรสของรีจิสเตอร์สำหรับการอ่านค่าแทรกอยู่ตรงกลาง ทำให้แอดเดรสของรีจิสเตอร์สำหรับการเขียนค่าถูกแยกออกจากกันเป็น 2 ส่วน การโปรแกรมค่าไปที่รีจิสเตอร์แต่ละชิพจึงต้องถูกแบ่งเป็น 2 ครั้งด้วย เพราะต้องส่งค่าแอดเดรสเริ่มต้นของรีจิสเตอร์ใหม่ รวม 2 ชิพแล้วข้อมูลจะถูกแบ่งเป็น 4 ส่วน ข้อมูลที่เก็บในแต่ละส่วนจะเรียงดังนี้ คือ ค่าแอดเดรสของชิพซึ่งจะเป็นของวงจรถอดรหัสหรือเข้ารหัส ค่าแอดเดรสเริ่มต้นของรีจิสเตอร์ที่จะเขียนค่าไป และข้อมูลสำหรับโปรแกรมค่าในรีจิสเตอร์แต่ละตัวเรียงกันไปตามลำดับแอดเดรสรีจิสเตอร์

วงจรหารความถี่จะสร้างสัญญาณความถี่ 100 kHz ซึ่งจะใช้ในทำงานภายในวงจรนี้และในตัวประมวลผลสัญญาณรีโมตคอนโทรลซึ่งจะกล่าวในรายละเอียดต่อไป โดยจะรับสัญญาณนาฬิกาของระบบซึ่งมีค่าเป็น 27 MHz มาหารด้วย 256 เพื่อให้ได้ความถี่สัญญาณออกมาประมาณ 100 kHz

หน่วยควบคุม I^2C มีแผนผังสถานะแสดงการทำงานดังแสดงในรูปที่ 4-10

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-10 แผนผังสถานะของหน่วยควบคุม I^2C

หน่วยควบคุม I^2C จะเริ่มทำงานในสภาวะว่างงาน (Idle) ซึ่งในสถานะนี้สัญญาณควบคุมที่ใช้ในหน่วยควบคุมนี้จะถูกกำหนดค่าเริ่มต้นไว้ ได้แก่ สัญญาณที่ใช้ตรวจสอบว่าส่งข้อมูลในรอมครบ 1 ส่วนหรือยัง (end_rom) สัญญาณบอกหมายเลขรอมว่าอยู่ในส่วนใดจากทั้งหมด 4 ส่วน (rom no.) สัญญาณตรวจสอบการทำงานของแต่ละสถานะ เช่น ตรวจสอบว่าส่งบิตเริ่มต้นเสร็จแล้ว (start_finish) ตรวจสอบว่าส่งบิตข้อมูลครบแล้ว (data_finish) และตรวจสอบว่าส่งบิตจบการทำงานเสร็จแล้ว (stop_finish) เป็นต้น ค่าเหล่านี้จะถูกรีเซตให้เป็น 0 ทั้งหมด

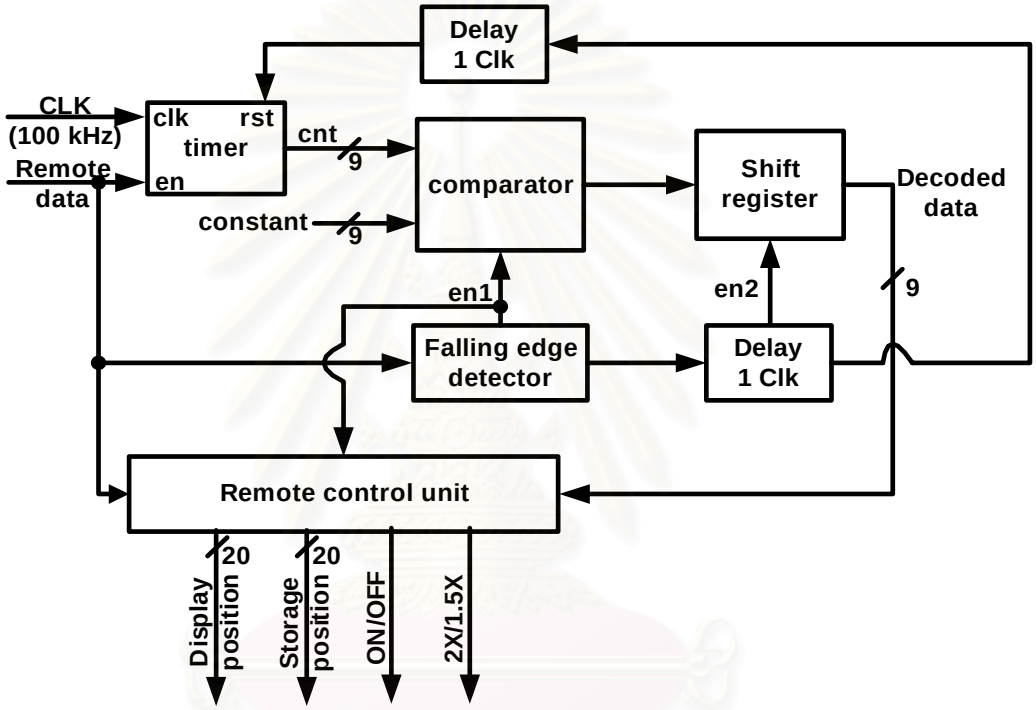
เมื่อมีสัญญาณรีเซตเข้ามาซึ่งจะเกิดขึ้นเมื่อเครื่องขยายวีดิทัศน์เริ่มทำงานหรือเมื่อปุ่มรีเซตภายนอกถูกกด หน่วยควบคุมก็จะเข้าสู่สถานะใหม่และสร้างบิตเริ่มต้น (Start bit) ออกมาทางสัญญาณ SDA พร้อมสัญญาณนาฬิกาทาง SCL แล้วจึงเซตค่า start_finish = '1' ออกไปเมื่อสร้างเสร็จ เพื่อจะเปลี่ยนการทำงานไปที่สถานะถัดไปคือ อ่านข้อมูล (Read data)

ข้อมูลที่จะส่งออกไปทาง SDA ถูกอ่านมาจากรอมทีละไบต์ เมื่ออ่านเสร็จก็จะกระโดดไปที่สถานะส่งข้อมูล (Send data state) ซึ่งจะทยอยส่งออกมาทีละบิตจนครบ 1 ไบต์ ก็จะสร้างสัญญาณ data_finish = '1' ออกมาบอกด้วย จากนั้นจะตรวจสอบว่าวงจรภาครับ เช่น วงจรถอดรหัส ว่ามีการส่งสัญญาณตอบรับ (ACK) ออกมาหรือไม่ ถ้ามีก็จะเลื่อนค่าแอดเดรสในการอ่านข้อมูลจากรอมไปอ่านไบต์ถัดไปที่สถานะอ่านข้อมูล แต่ถ้าไม่มีก็จะคงค่าแอดเดรสเดิมไว้เพื่อ

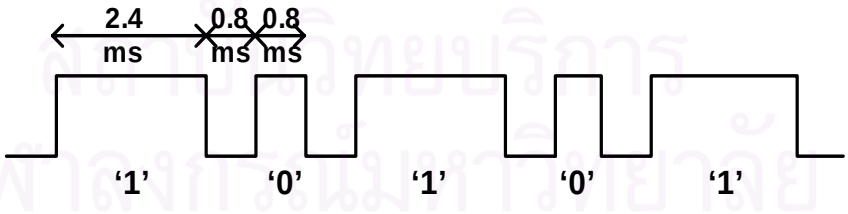
ส่งข้อมูลเดิมซ้ำอีกรอบ วงจรจะหลุดไปทำงานที่สถานะส่งบิตจบการทำงาน (Stop bit) เมื่อส่งข้อมูลในรอมครบ 1 ส่วน

เมื่อส่งบิตจบการทำงานเสร็จ rom no. จะถูกเพิ่มค่าไป 1 เพื่ออ่านข้อมูลในส่วนถัดไปของรอมจนกระทั่งครบ 4 ส่วน หน่วยควบคุมก็จะกลับไปสู่สภาวะว่างอีกครั้ง แต่หากไม่ครบ 4 ส่วน วงจรก็จะเริ่มกลับไปส่งบิตเริ่มต้นใหม่

4.2.2 **วงจรประมวลผลสัญญาณรีโมต** (Remote processor)



รูปที่ 4-11 วงจรประมวลผลสัญญาณรีโมต



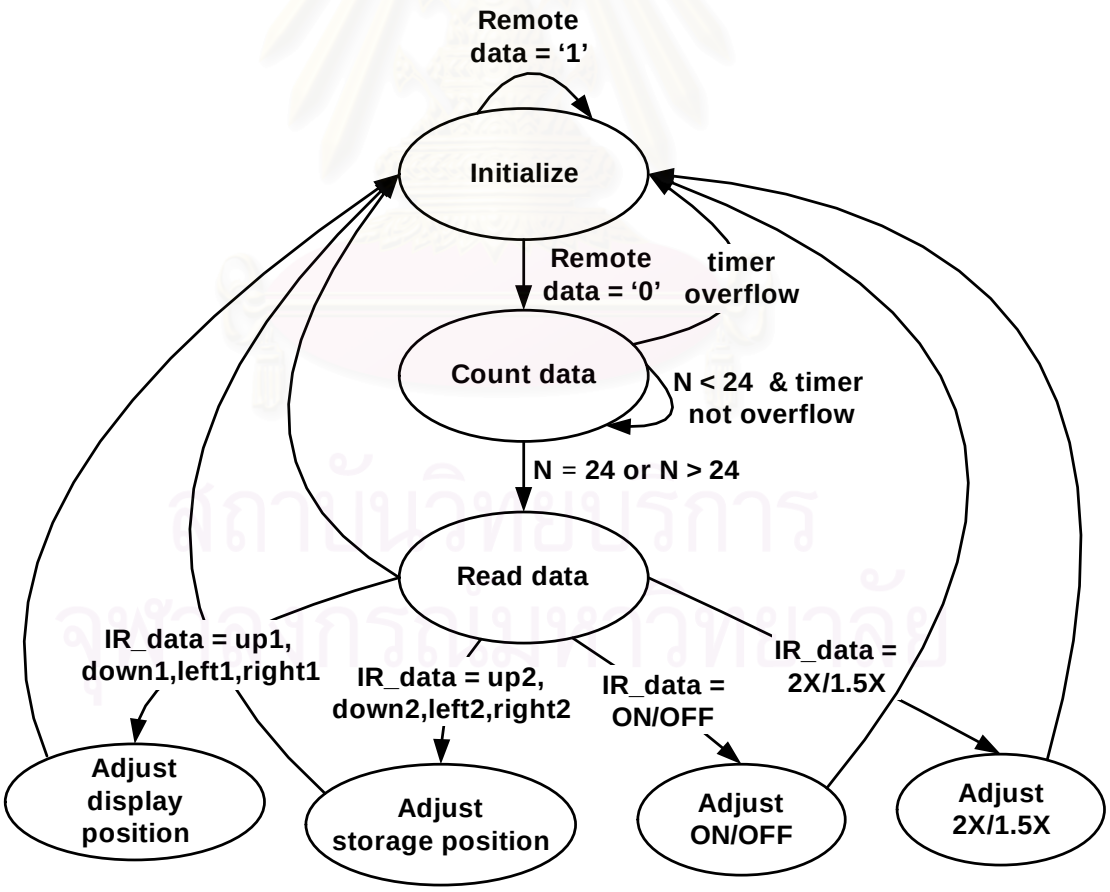
รูปที่ 4-12 รูปแบบของสัญญาณรีโมต

สัญญาณจากรีโมตคอนโทรลที่ใช้นี้ถูกเข้ารหัสโดยใช้ความกว้างของพัลส์ดังแสดงในรูปที่ 4-12 ซึ่งจะสามารถตรวจสอบบิตข้อมูลได้โดยการวัดความกว้างช่วงที่สัญญาณรีโมตเป็น '1' จากการตรวจสอบพบว่า ช่วงที่บิตข้อมูลเป็น '1' สัญญาณจะมีความกว้างเท่ากับ 2.4 ms และเมื่อเป็น

‘0’ จะมีความกว้าง 0.8 ms ดังนั้นในวงจรประมวลผลสัญญาณรีโมตจะตั้งระดับตัดสินบิตไว้ที่ 1.6 ms

การถอดรหัสจะเริ่มจากการให้ตัวจับเวลา (timer) วัดช่วงเวลาที่เป็น ‘1’ ของสัญญาณรีโมต (remote data) ว่ามีความกว้างเท่าไหร่โดยใช้สัญญาณนาฬิกาความถี่ 100 kHz ที่สร้างจากตัวส่งสัญญาณ I^2C ในการนับ เมื่อสัญญาณรีโมตกลับมาเป็น ‘0’ วงจรเปรียบเทียบ (comparator) จึงอ่านค่าความกว้างที่วัดได้จากตัวจับเวลาว่ามีค่ามากกว่าหรือน้อยกว่า 1.6 ms แล้วส่งผลลัพธ์ที่ได้ไปเก็บไว้ในวงจรเลื่อนบิตข้อมูล (Shift register) ในสัญญาณนาฬิกาถัดไป หลังจากนั้นอีก 1 สัญญาณนาฬิกาตัวจับเวลาจะถูกรีเซตกลับเป็น 0 เพื่อรอการนับสัญญาณครั้งต่อไป

หน่วยควบคุมรีโมตจะตรวจสอบสัญญาณรีโมตว่าส่งมาครบ 24 บิตแล้ว (การกดปุ่ม 1 ปุ่มจะได้ข้อมูลออกมาทั้งหมด 24 บิต) จึงอ่านข้อมูลจากวงจรเลื่อนบิตข้อมูลมา โดยอ่านเพียง 9 บิตบนมาเปรียบเทียบว่าตรงกับคำสั่งใด แล้วจะปรับค่าสัญญาณขาออกใหม่ตามคำสั่งที่ได้รับมา การทำงานของหน่วยควบคุมรีโมตสามารถสรุปเป็นแผนผังสถานะได้ดังรูปที่ 4-13



รูปที่ 4-13 แผนผังสถานะของหน่วยควบคุมรีโมต

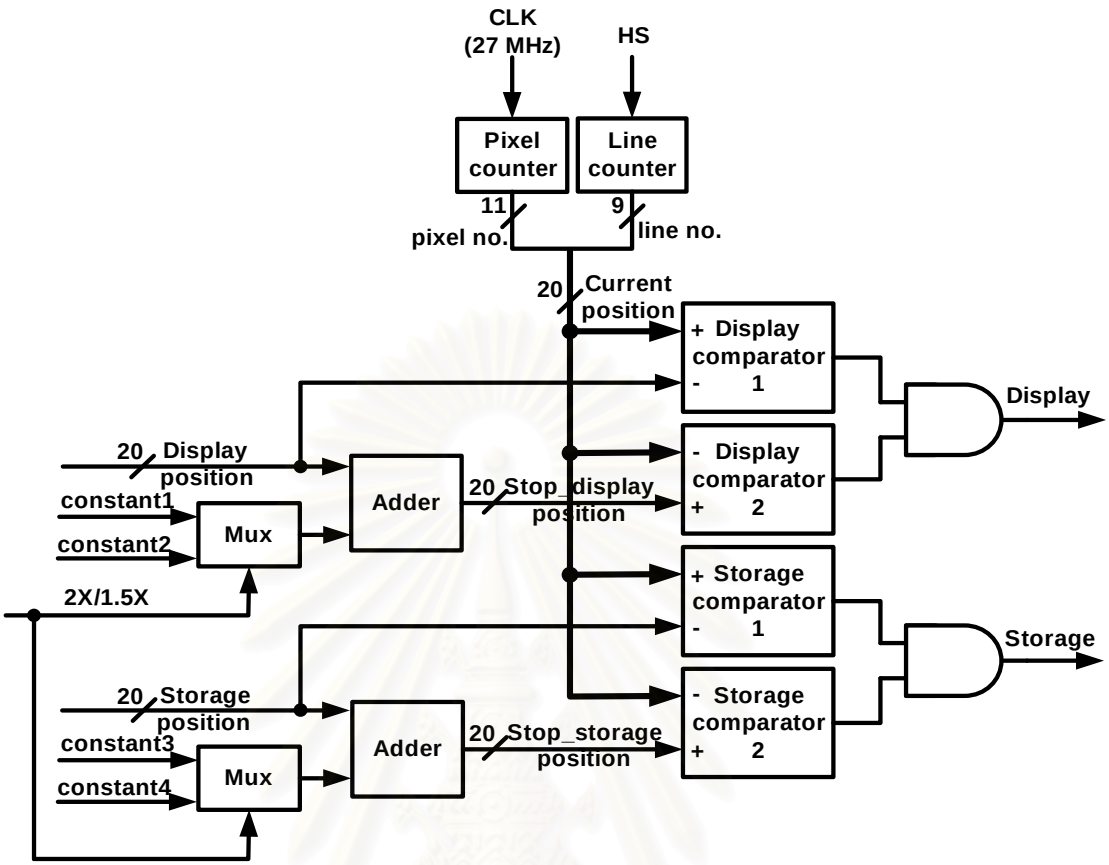
การทำงานของหน่วยควบคุมจะเริ่มจากการกำหนดค่าเริ่มต้นต่างๆ ให้มีค่าเป็น 0 ได้แก่ สัญญาณสำหรับนับบิตข้อมูลของสัญญาณรีโมต (N) สัญญาณสำหรับเก็บข้อมูลที่ถูกถอดรหัสแล้วจากวงจรเลื่อนบิต (IR_data) และสัญญาณสำหรับจับเวลาเพื่อใช้ในการเช็คว่าบัสข้อมูลรีโมตอยู่ในสภาวะว่างหรือไม่ (timer)

ในสภาวะว่างข้อมูลรีโมตที่ส่งมาจะมีค่าเป็น '1' ตลอด แต่เมื่อมีสัญญาณจากรีโมตส่งมาสัญญาณจะถูกดึงมาที่ '0' ชั่วเวลาหนึ่งเป็นบิตเริ่มต้นแล้วจึงเริ่มส่งบิตข้อมูลในเวลาต่อมา เมื่อมีขอบขาลงเกิดขึ้น 1 ครั้งวงจรนับก็จะเพิ่มค่าสัญญาณ N ไปทีละ 1 พร้อมกันนี้สัญญาณ timer ก็จะโดนรีเซตเป็น 0 สัญญาณ timer นี้ใช้เพื่อกันความผิดพลาดในกรณีที่มีสัญญาณรบกวนหรือข้อมูลที่รับมาได้ไม่ครบ 24 บิตทำให้วงจรยังทำงานคาอยู่สถานะการนับอยู่ทั้งๆ ที่ข้อมูลจากรีโมตส่งมาครบ 24 บิตแล้ว ซึ่งจะส่งผลกระทบต่อการรับข้อมูลใหม่เพราะยังมีข้อมูลเก่าเหลืออยู่ ดังนั้นหาก timer ตรวจสอบสัญญาณรีโมตว่ามีค่า '1' นานเกินไปก็จะกระโดดกลับไปทำงานที่สภาวะเริ่มต้นเพื่อล้างข้อมูลเก่าไป และรอรับข้อมูลเข้ามาใหม่

เมื่อข้อมูลส่งมาจนครบ 24 บิต หน่วยควบคุมรีโมตก็จะอ่านค่าที่ถอดรหัสได้มาเก็บไว้แล้วเปรียบเทียบว่าตรงกับรหัสของคำสั่งที่กำหนดไว้หรือไม่ โดยมีคำสั่งทั้งหมด 10 คำสั่ง คือ คำสั่งสำหรับเปลี่ยนตำแหน่งในการแสดงภาพภาษามือ 4 คำสั่ง (up1, down1, left1, right1) ซึ่งจะทำให้สัญญาณออก Display position มีค่าเปลี่ยนไป คำสั่งสำหรับเปลี่ยนตำแหน่งในการเก็บภาพภาษามือ 4 คำสั่ง (up2, down2, left2, right2) ซึ่งจะมีผลทำให้สัญญาณ Storage position เปลี่ยนค่า คำสั่งเปิดปิดภาพ (ON/OFF) 1 คำสั่ง และคำสั่งเปลี่ยนขนาดของภาพ (2X/1.5X) 1 คำสั่ง ถ้าหากไม่ตรงกับคำสั่งใดเลยตัวควบคุมจะกลับไปอยู่ในสภาวะเริ่มต้นใหม่ แต่ถ้าตรงก็จะปรับสัญญาณขาออกใหม่แล้วจึงกลับไปสภาวะว่าง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.3 **วงจรสร้างสัญญาณควบคุม** (Control signal generator)



รูปที่ 4-14 วงจรสร้างสัญญาณควบคุม

วงจรสร้างสัญญาณควบคุมทำหน้าที่สร้างสัญญาณควบคุมการทำงานต่างๆ ของวงจรอื่น ได้แก่ สัญญาณที่บอกว่าเป็นช่วงข้อมูลที่จะต้องเก็บภาพภาษามือแล้ว (Storage) และสัญญาณที่บ่งบอกว่าถึงช่วงเวลาที่ต้องแสดงภาพแล้ว (Display)

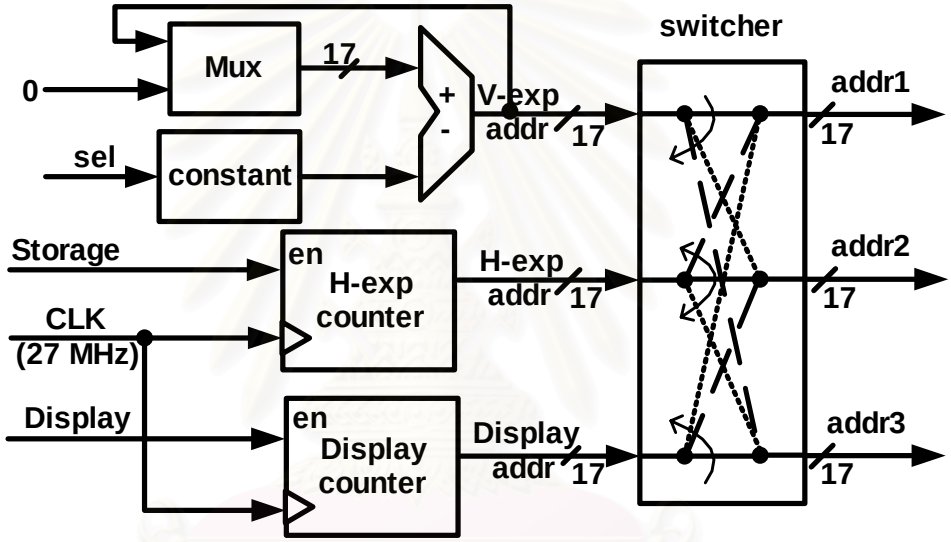
การทำงานจะเริ่มจากการคำนวณหาค่าตำแหน่งภาพปัจจุบันโดยใช้วงจรนับทั้งหมด 2 วงจร คือ วงจรนับพิกเซล (Pixel counter) ขนาด 11 บิตซึ่งจะใช้บอกตำแหน่งพิกเซลภาพปัจจุบัน และวงจรนับเส้น (Line counter) ขนาด 9 บิตสำหรับบอกตำแหน่งเส้นภาพปัจจุบัน ค่าทั้งสองนี้จะถูกรวมกันแล้วส่งไปที่วงจรเปรียบเทียบซึ่งมีอยู่ 4 วงจร

วงจรเปรียบเทียบเพื่อแสดงภาพ 1 (Display comparator1) จะส่งสัญญาณออกเป็น ‘1’ เมื่อตำแหน่งภาพในปัจจุบันมีค่ามากกว่าตำแหน่งเริ่มต้นการแสดงภาพที่ตั้งไว้แล้ว ส่วนวงจรเปรียบเทียบเพื่อแสดงภาพ 2 จะส่งสัญญาณออกเป็น ‘1’ เมื่อตำแหน่งภาพปัจจุบันมีค่าน้อยกว่าตำแหน่งสิ้นสุดการแสดงภาพ (Stop_display position) ซึ่งคำนวณจากการบวกตำแหน่งเริ่มต้นการแสดงภาพกับค่าคงที่ ค่าคงที่นี้จะมีอยู่ 2 ค่าเพื่อเลือกความยาวสัญญาณให้ถูกต้องตาม

อัตราขยายภาพ สัญญาณขาออกจากวงจรเปรียบเทียบทั้งสองนี้จะรวมกันสร้างเป็นสัญญาณแสดงภาพออกไปเพื่อใช้ในวงจรควบสัญญาณสำหรับการแสดงภาพ

ส่วนวงจรเปรียบเทียบเพื่อเก็บภาพ (Storage comparator) 1 และ 2 ก็เช่นกันจะใช้เปรียบเทียบตำแหน่งภาพปัจจุบันว่ามีค่ามากกว่าตำแหน่งเริ่มต้นการเก็บภาพ (Storage position) และมีค่าน้อยกว่าตำแหน่งสิ้นสุดการเก็บภาพ (Stop_storage position) ตามลำดับ ผลลัพธ์จากวงจรทั้งสองจะถูกนำมาสร้างเป็นสัญญาณเก็บภาพ (Storage) เพื่อไปควบคุมการทำงานของการขยายภาพทางแนวนอนต่อไป

4.2.4 **วงจรนับสำหรับสร้างค่าแอดเดรส** (Address counter)



รูปที่ 4-15 วงจรนับสำหรับสร้างค่าแอดเดรส

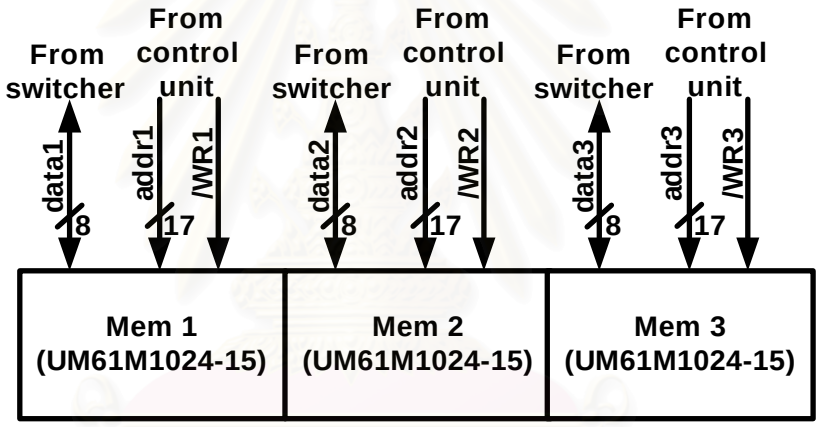
รูปที่ 4-15 แสดงการทำงานของวงจรนับสำหรับสร้างค่าแอดเดรสซึ่งจะถูกแบ่งออกเป็น 3 ส่วนตามจำนวนแอดเดรสที่ต้องสร้าง คือ วงจรสร้างค่าแอดเดรสสำหรับการขยายภาพในแนวตั้ง (V-exp addr) สำหรับการขยายภาพในแนวนอน (H-exp addr) และสำหรับการแสดงภาพ (Display addr) ซึ่งจะถูกส่งไปให้ตัวสวิตช์เพื่อเลือกเส้นทางไปป้อนหน่วยความจำทั้งสามให้ตรงกับการทำงานในแต่ละเฟรม รูปแบบของเส้นทางจะเหมือนกับตัวสวิตช์สำหรับเลือกข้อมูลส่งออกไปที่หน่วยความจำ

วงจรนับสำหรับสร้างสัญญาณแอดเดรสของการแสดงภาพและการขยายภาพทางแนวนอนจะใช้เป็นวงจรนับแบบเลขฐานสอง (Binary counter) ขนาด 17 บิต โดยนับด้วยความถี่ 27 MHz ส่วนวงจรนับสำหรับการขยายภาพทางแนวตั้งจะไม่สามารถสร้างจากวงจรนับปรกติธรรมดาได้

เนื่องจากค่าแอดเดรสที่สร้างจะมีทั้งใช้สำหรับการอ่านค่าและเขียนค่าซึ่งไม่มีความสัมพันธ์แบบเพิ่มขึ้นทีละ 1 เหมือนกันสัญญาณแอดเดรส 2 แบบแรก

ดังที่กล่าวไว้ในวงจรขยายภาพแนวตั้ง การขยายภาพจะแบ่งออกเป็น 4 ช่วงคือ อ่านข้อมูลไบต์ที่ 1 อ่านข้อมูลไบต์ที่ 2 ซึ่งติดกันในแนวตั้ง คำนวณค่า และส่งข้อมูลไปเก็บ ดังนั้นใน 4 สัญญาณนาฬิกาที่ต่อเนื่องกัน ค่าแอดเดรสที่สร้างออกไปจะมีทั้งหมด 3 ค่า คือ สำหรับอ่านค่าข้อมูลที่ติดกันในแนวตั้ง 2 ค่าซึ่งข้อมูลทั้ง 2 นี้มีค่าแอดเดรสต่างกัน 512 ไบต์หรือ 1 เส้นภาพที่ผ่านการขยายแนวนอนแล้ว และอีก 1 ค่าสำหรับการเขียนข้อมูลซึ่งจะอยู่ห่างจากตำแหน่งข้อมูลที่อ่าน 64 กิโลไบต์ ดังนั้นการสร้างแอดเดรสจึงใช้วงจรบวกลบเพื่อเพิ่มความยืดหยุ่นให้แอดเดรสสามารถเปลี่ยนแปลงได้มากกว่าการเพิ่มขึ้นทีละ 1 ตามที่ต้องการ

4.3 หน่วยความจำ



รูปที่ 4-16 การทำงานของหน่วยความจำ

หน่วยความจำภายนอกที่เลือกใช้ในเครื่องขยายวีดิทัศน์นี้ คือ ชิพเบอร์ UM61M1024S-15 ทั้งหมด 3 ตัว ซึ่งเป็นหน่วยความจำแบบอะซิงโครนัสเอสแรมที่ใช้เวลาในการเข้าถึงไม่เกิน 15 ns การติดต่อกับหน่วยความจำแบบนี้จะมีอยู่ 2 แบบ คือ การอ่านค่าและการเขียนค่า การอ่านค่าจะทำได้โดยการให้สัญญาณ /WR มีค่าเป็น ‘1’ พร้อมกับส่งค่าแอดเดรสขนาด 17 บิตมาที่หน่วยความจำด้วย หลังจากนั้นภายในเวลา 15 ns ข้อมูลที่อยู่ในแอดเดรสนั้นของหน่วยความจำจะถูกส่งออกมาทางขาข้อมูล ในทางกลับกันการเขียนค่าจะต้องส่งสัญญาณ /WR = ‘0’ พร้อมทั้งส่งแอดเดรสและข้อมูลที่ต้องการเขียนลงไปให้กับหน่วยความจำ โดยสัญญาณทั้งสามจะต้องถูกประวิงเวลาไว้อย่างน้อย 15 ns เพื่อรอให้หน่วยความจำเขียนค่าที่เซลในแอดเดรสนั้นเสร็จ

ข้อมูลที่ส่งมาให้หน่วยความจำทั้งสามนี้จะส่งมาจากตัวสวิตช์ ส่วนสัญญาณแอดเดรสและสัญญาณ /WR จะส่งมาจากหน่วยควบคุมเพื่อควบคุมจังหวะการอ่านและเขียนค่า หน่วยความจำ

จะใช้สำหรับเก็บภาพภาษามือได้ทั้งหมด 3 เฟรม คือ เฟรมปัจจุบันซึ่งเป็นภาพที่กำลังขยายทางแนวนอน เฟรมที่แล้วซึ่งเป็นภาพที่กำลังขยายทางแนวตั้ง และ 2 เฟรมก่อนหน้านี้ซึ่งเป็นภาพที่ขยายเสร็จแล้วรอการแสดงภาพบนเครื่องรับโทรทัศน์ หน่วยความจำตัวหนึ่งจะเก็บข้อมูลภาพในเฟรมเดิมไว้นาน 3 เฟรมเพื่อผ่านกระบวนการขยายภาพให้ครบตั้งแต่การขยายภาพทางแนวนอนจนกระทั่งถึงการแสดงภาพแล้วจึงเริ่มเก็บภาพใหม่ การทำงานของหน่วยความจำทั้งสามสามารถสรุปเป็นตารางได้ดังตารางที่ 4-2

ตารางที่ 4-2 แสดงลำดับการทำงานของหน่วยความจำ

	เฟรมที่ n	เฟรมที่ n+1	เฟรมที่ n+2
หน่วยความจำที่ 1	วงจรขยายภาพทางแนวนอน	วงจรขยายภาพทางแนวตั้ง	วงจรแสดงภาพ
หน่วยความจำที่ 2	วงจรขยายภาพทางแนวตั้ง	วงจรแสดงภาพ	วงจรขยายภาพทางแนวนอน
หน่วยความจำที่ 3	วงจรแสดงภาพ	วงจรขยายภาพทางแนวนอน	วงจรขยายภาพทางแนวตั้ง

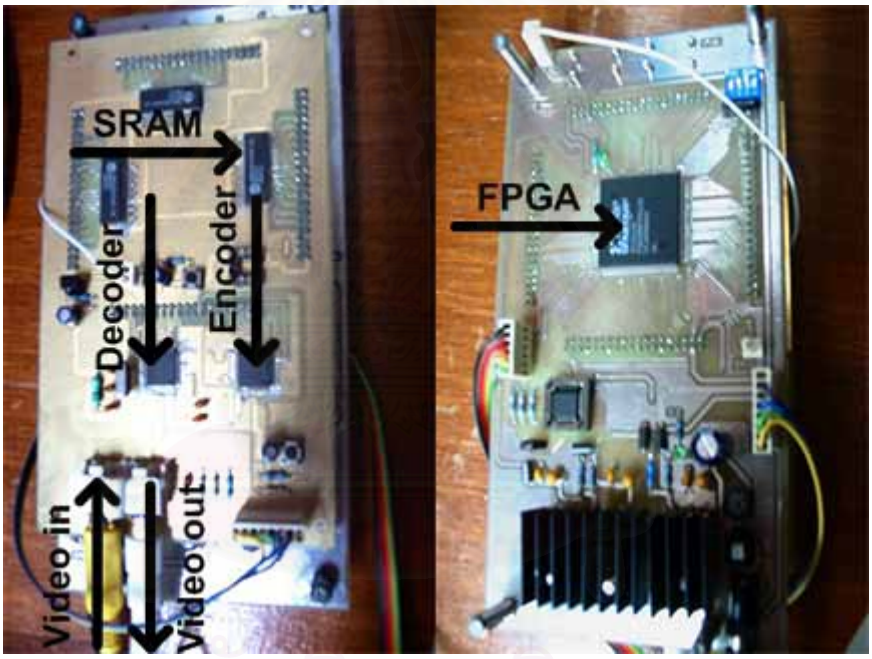
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบ และสรุปผล

5.1 การทดสอบการทำงานของเครื่องขยายวีดิทัศน์

ต้นแบบของเครื่องขยายภาพวีดิทัศน์ภาษามือแบ่งออกเป็น 2 บอร์ด คือ บอร์ดของชิพ FPGA ซึ่งทำหน้าที่เป็นวงจรควบคุมการขยายภาพ (รูปด้านขวามือของรูปที่ 5-1) และบอร์ดที่ 2 เป็นวงจรที่ต่อประสานกับ FPGA (รูปด้านซ้ายมือของรูปที่ 5-1) ซึ่งประกอบด้วยวงจรถอดรหัสและเข้ารหัสสัญญาณวีดิทัศน์ หน่วยความจำและวงจรรับสัญญาณอินฟราเรด



รูปที่ 5-1 เครื่องขยายวีดิทัศน์ภาษามือสำหรับผู้พิการทางการได้ยิน.

วงจรควบคุมการขยายภาพซึ่งถูกออกแบบบน FPGA (XC2S150 ของบริษัท Xilinx) ตามรายงานจากโปรแกรม webpack 6.1 พบว่าได้ใช้ทรัพยากรของ FPGA ไปดังนี้

- Slice Flip Flops: 616 จาก 3,456 (17%)
- 4 input LUTs: 959 จาก 3,456 (27%)

รายงานการใช้ทรัพยากรนี้ได้รวมถึงหน่วยความจำภายในแบบสองพอร์ตทั้งหมด 6 ตัวแล้ว (สำหรับวงจรหาร 3 ที่มีอยู่ 2 วงจร รวม 4 ตัว สำหรับตัวพักข้อมูลเส้นภาพ 1 ตัว และสำหรับเป็นรอมเก็บค่าโปรแกรมผ่าน I^2C 1 ตัว)

วงจรควบคุมการขยายภาพนี้ใช้ฟลิปฟลอปไปประมาณ 27% หรือ 1/3 เท่านั้น ดังนั้นการออกแบบเพื่อใช้งานจริงนี้สามารถเลือกใช้ FPGA ที่มีขนาดเล็กกว่านี้ เช่น XC2S50 ได้ซึ่งจะมีราคาที่ถูกลง

ในการทดสอบเครื่องขยายวีดิทัศน์นี้จะโปรแกรมวงจรควบคุมการขยายภาพลงบน FPGA และนำมาต่อกับวงจรต่อประสานอื่นๆ โดยใช้สัญญาณวีดิทัศน์จากเครื่องเล่นวีดิโอเทปซึ่งอัดรายการที่มีภาพภาษามือไว้มาป้อนเป็นสัญญาณขาเข้า และสัญญาณขาออกจากเครื่องขยายวีดิทัศน์จะต่อเข้ากับเครื่องรับโทรทัศน์เพื่อแสดงภาพที่ได้จากวงจรออกมา ซึ่งได้ผลดังต่อไปนี้

รูปที่ 5-2 ภาพรายการต้นฉบับที่ยังไม่ได้ขยาย

รูปที่ 5-2 เป็นภาพภาษามือต้นฉบับที่ส่งมาจากสถานีโทรทัศน์ ซึ่งจะเห็นว่าถูกบีบอัดจนมีขนาดเล็กมาก และเป็นอุปสรรคในการอ่านภาษามือมาก

เมื่อนำข้อมูลภาพมาขยายด้วยเครื่องขยายวีดิทัศน์ด้วยขนาด 2.25 เท่าจะได้ผลดังรูปที่ 5-3 ในขณะที่รูปที่ 5-4 จะเป็นภาพจากการขยาย 4 เท่า

รูปที่ 5-3 ภาพรายการเมื่อทำงานขยายภาพขนาด 2.25 เท่า

รูปที่ 5-4 ภาพรายการเมื่อทำงานขยายภาพขนาด 4 เท่า

ภาพที่ได้จากการขยายโดยใช้เครื่องขยายวีดิทัศน์นี้มีคุณภาพค่อนข้างดีทั้งในแบบ 2.25 เท่า และแบบ 4 เท่า ขนาดที่ใหญ่ขึ้นนี้จะช่วยให้ผู้ชมสามารถเข้าใจภาพภาษามือได้ง่ายขึ้นมากซึ่งจะเป็นประโยชน์อย่างมากกับกลุ่มคนหูหนวก โดยที่ไม่มีผลกระทบต่อการรับชมรายการของผู้ชมอื่นๆ ส่วนอัตราขยายภาพที่เหมาะสมนี้จะขึ้นอยู่กับความต้องการของผู้ชม เพราะขนาดภาพที่ใหญ่ขึ้นนี้แม้ว่าจะทำให้อ่านภาษามือได้ง่ายขึ้นแต่ภาพรายการเดิมเองก็จะถูกรบกวนมากขึ้นด้วย นอกจากนี้ยังมีปัจจัยอื่นเกี่ยวข้อง เช่น ขนาดจอของเครื่องรับโทรทัศน์ ซึ่งหากมีขนาดจอเล็กมากเช่น ขนาด 14 นิ้ว ภาพภาษามือที่ออกมาจะมีขนาดเล็กตามไปด้วย ดังนั้นการใช้อัตราขยายขนาด 4 เท่าก็จะเหมาะสมกว่า เป็นต้น

รูปที่ 5-5 ภาพรายการเมื่อเลือกเปลี่ยนตำแหน่งการแสดงภาพ

รูปที่ 5-5 เป็นภาพตัวอย่างของการเปลี่ยนตำแหน่งการแสดงภาพซึ่งผู้ชมจะสามารถกำหนดให้อยู่ ณ ตำแหน่งใดของจอก็ได้เพื่อให้อยู่ในตำแหน่งที่รบกวนภาพที่สำคัญน้อยที่สุด นอกจากนี้แล้วผู้ชมยังสามารถปรับตำแหน่งหน้าต่างภาพที่จะขยายได้ด้วยเนื่องจากรายการโทรทัศน์แต่ละรายการจะแทรกภาพภาษามือมาในตำแหน่งที่ต่างกัน ทำให้การปรับเปลี่ยนตำแหน่งภาพภาษามือได้จึงเป็นสิ่งที่จำเป็น

5.2 ปัญหาในการทำงาน

เมื่อนำเครื่องขยายวีดิทัศน์ไปประยุกต์ใช้กับภาพอื่นๆ ที่มีความซับซ้อนมากจะพบว่า ที่ขอบของวัตถุต่างๆ จะไม่เรียบและไม่คมชัด ซึ่งเป็นข้อเสียจากการออกแบบวงจรขยายภาพด้วยวิธีไบลิเนียร์

5.3 สรุป

เครื่องขยายวีดิทัศน์ที่ออกแบบนี้จะใช้ขยายหน้าต่างภาพขนาด 128x128 พิกเซลหรือประมาณ 1/24 ของภาพ ซึ่งเป็นขนาดที่เหมาะสมกับการประยุกต์ใช้งานกับภาพภาษามือที่แทรกมาในรายการโทรทัศน์ ภายในเครื่องขยายวีดิทัศน์จะมีวงจรควบคุมการขยายภาพซึ่งสังเคราะห์บน FPGA ทำหน้าที่ขยายข้อมูลให้มีขนาดใหญ่ขึ้นด้วยวิธีประมาณค่าแบบไบลีเนียร์ ซึ่งเป็นวิธีที่ให้คุณภาพค่อนข้างดีและมีการคำนวณที่ไม่ซับซ้อน จากผลการทดสอบพบว่าวงจรควบคุมนี้ใช้ทรัพยากรบน FPGA เบอร์ XC2S150 ไปเพียง 27 % ทรัพยากรที่ใช้ไปนี้นอกจากจะใช้ขยายภาพแล้วยังรวมถึงวงจรถอดรหัสคำสั่งจากรีโมตคอนโทรลเพื่อให้ผู้ใช้สามารถกำหนดตำแหน่งภาพ

ภาษามือและอัตราขยายภาพได้ และมีวงจรส่งข้อมูลผ่านบัส I^2C สำหรับกำหนดการทำงานของวงจรถอดรหัสและเข้ารหัสสัญญาณวีดิทัศน์เพื่อแปลงสัญญาณแบบแอนะล็อกไปเป็นสัญญาณดิจิทัลและแปลงกลับมาตามลำดับ เมื่อทดสอบการทำงานเครื่องขยายวีดิทัศน์พบว่า เครื่องสามารถขยายภาพภาษามือให้ใหญ่ขึ้น 4 เท่าและ 2.25 เท่า ได้โดยให้คุณภาพของภาพที่ค่อนข้างดี

งานวิจัยนี้จะสามารถช่วยเหลือคนหูหนวกให้สามารถเข้าใจเนื้อหารายการโทรทัศน์ได้ดีขึ้น โดยไม่ส่งผลกระทบใดต่อสถานีโทรทัศน์หรือผู้ชมทั่วๆ ไป และเครื่องขยายวีดิทัศน์นี้ได้ผ่านการประเมินจากตัวแทนของสมาคมคนหูหนวกแล้วซึ่งค่อนข้างจะพอใจกับสิ่งประดิษฐ์นี้

5.4 ข้อเสนอแนะ

1) เลือกใช้วิธีการขยายภาพแบบปรับตัวได้ในส่วนของการขยายข้อมูลสัญญาณความสว่างซึ่งเป็นสัญญาณที่ไวต่อดวงตาของคนมากกว่าสัญญาณสี เพื่อเพิ่มคุณภาพของขอบวัตถุ
2) เพิ่มการทำงานให้สามารถปรับขนาดหน้าต่างของภาพเนื่องจากรายการโทรทัศน์แต่ละช่องมีขนาดของภาพภาษามือต่างกัน
3) ออกแบบวงจรขยายให้สามารถปรับอัตราขยายของภาพได้มากขึ้น
4) ใช้ FPGA เบอร์ที่มีขนาดหน่วยความจำภายในมาก เพื่อลดการใช้หน่วยความจำภายนอกซึ่งจะมีความเร็วช้าและมีสัญญาณรบกวนจากภายนอกได้ง่าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Jain, Anil K. Fundamentals of digital image processing. New Jersey : Prentice-Hall, 1989.
2. Tagliarini, Gene A. Enlarging or Contracting a Digital Image [online]. Available from: http://people.uncw.edu/tagliarinig/Courses/475-592/Lectures/Enlarging or Contracting a Digital Image.ppt [Mar,2004]
3. Hoffmann Gernot. Interpolations for Image Warping [online]. Available from: http://www.fhoemden.de/~hoffmann/bicubic03042002.pdf [Mar,2004]
4. Raghupathy, A.; Chandrachoodan. N and Liu K.J. Ray. Algorithm and VLSI architecture for high performance adaptive video scaling. IEEE Transaction on multimedia. (December 2003) : 489-502
5. ITU-R BT.601-5 standard [online]. Available from: http://www/itu.int/itudoc/itu-r/rec/bt/ [Mar,2004]
6. ITU-R BT.656-4 standard [online]. Available from: http://www/itu.int/itudoc/itu-r/rec/bt/ [Mar,2004]
7. Vishay Telefunken. Data Formats for IR Remote Control.
8. Philips semiconductor. The I^2C – bus Specification Version 2.1. January, 2000.
9. Philips semiconductor. SAA7113: 9-bit video input processor. 1999.
10. Philips semiconductor. SAA7126; 7127 Digital video encoder. 2000.
11. Xilinx Incorporation. The Programmable Logic Data Book 2000. United State of America, 2000.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
การออกแบบวงจรด้วยภาษาวีเอชดีแอล

vpu.vhd

```
##############################################################
#                Main program                #
##############################################################

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
library unisim;
use unisim.vcomponents.all;

entity vpu is
   Port ( data_in : in std_logic_vector(7 downto 0);
          clk : in std_logic;
          hs : in std_logic;
          odd : in std_logic;
          IR : in std_logic;
          reset_n : in std_logic;
          dclk : out std_logic;
          data_out : out std_logic_vector(7 downto 0);
          addr1 : out std_logic_vector(16 downto 0);
          addr2 : out std_logic_vector(16 downto 0);
          addr3 : out std_logic_vector(16 downto 0);
          data1 : inout std_logic_vector(7 downto 0);
          data2 : inout std_logic_vector(7 downto 0);
          data3 : inout std_logic_vector(7 downto 0);
          scl : out std_logic;
          sda : inout std_logic;
          nwr1 : out std_logic;
          nwr2 : out std_logic;
          nwr3 : out std_logic;
          nen1 : out std_logic;
          nen2 : out std_logic;
          nen3 : out std_logic);
end vpu;

architecture Behavioral of vpu is
component clk_divider is
   Port (  clk : in std_logic;
          clk1 : out std_logic;
          clk3 : out std_logic;
          clk90 : out std_logic;
          rs : out std_logic );
end component;
signal clk1,clk3,clk90,reset : std_logic;

component i2c is
   Port (  clk3      : in std_logic;
          reset      : in std_logic;
          reset_n    : in std_logic;
          scl        : out std_logic;
          sda        : inout std_logic );
end component;
```

```
component sw is
   Port (   clk3,reset : in std_logic;
            key : in std_logic_vector(4 downto 0);
            pic_on : out std_logic;
            start_line : out std_logic_vector(8 downto 0);
            start_pixel : out std_logic_vector(10 downto 0));
end component;

component remote2 is
   Port (   clk3 : in std_logic;
            reset : in std_logic;
            IR : in std_logic;
            pic_on : out std_logic;
            mode : out std_logic;
            start_pixel : out std_logic_vector(10 downto 0);
            start_line : out std_logic_vector(8 downto 0);
            start_pixel2 : out std_logic_vector(10 downto 0);
            start_line2 : out std_logic_vector(8 downto 0));
end component;
signal pic_on,mode : std_logic;
signal start_line,start_line2    : std_logic_vector (8 downto 0);
signal start_pixel,start_pixel2 : std_logic_vector (10 downto 0);

component horinter2 is
   Port (   clk : in std_logic;
            clk90 : in std_logic;
            f  : in std_logic;
            reset : in std_logic;
            datai : in std_logic_vector(7 downto 0);
            str : in std_logic;
            wr : in std_logic;
            mode : in std_logic;
            do : out std_logic_vector(7 downto 0);
            en_out : out std_logic;
            addro : out std_logic_vector(16 downto 0) );
end component;
signal data_h_out : std_logic_vector(7 downto 0);
signal h_en  : std_logic;
signal addr_str_h : std_logic_vector(16 downto 0);

component ver_add is
   Port (   datain : in std_logic_vector(7 downto 0);
            dataout : out std_logic_vector(7 downto 0);
            clk1,clk90 : in std_logic;
            en,reset  : in std_logic;
            mode : in std_logic;
            a : out std_logic_vector(16 downto 0);
            nwr : out std_logic );
end component;
signal v_en : std_logic;
signal data_v_in,data_v_out : std_logic_vector(7 downto 0);
signal addr_str_v : std_logic_vector(16 downto 0);
signal nwr_v2,nwr_v3 : std_logic;

component compare_block is
   Port (   clk1 : in std_logic;
            mode : in std_logic;
            start_line : in std_logic_vector(8 downto 0);
            start_pixel : in std_logic_vector(10 downto 0);
            start_line2 : in std_logic_vector(8 downto 0);
            start_pixel2 : in std_logic_vector(10 downto 0);
            li_cnt : in std_logic_vector(8 downto 0);
            pi_cnt : in std_logic_vector(10 downto 0);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
            HS_fall : in std_logic;
            str_en : out std_logic;
            shw_en : out std_logic;
            li_en : out std_logic;
            blk_li_en : out std_logic );
end component;
signal str,shw,li_en,blk_li_en : std_logic;

component line_counter is
   Port (   cnt_en : in std_logic;
            reset : in std_logic;
            clk : in std_logic;
            q : out std_logic_vector(8 downto 0));
end component;
signal li_cnt : std_logic_vector(8 downto 0);

component pixel_counter is
   Port (   clk : in std_logic;
            reset : in std_logic;
            qo : out std_logic_vector(10 downto 0));
end component;
signal pi_cnt : std_logic_vector(10 downto 0);

component shw_ad_cnt is
   Port (   clk : in std_logic;
            en : in std_logic;
            reset : in std_logic;
            f : in std_logic;
            mode : in std_logic;
            qo : out std_logic_vector(16 downto 0));
end component;
signal addr_shw : std_logic_vector(16 downto 0);

-- VPU parameter
-- counter
signal f : std_logic;
signal edge_f_1,edge_f_2,edge_HS_1,edge_HS_2  : std_logic;
signal HS_fall  : std_logic;
signal fedge,f_fall : std_logic;
signal frame : std_logic_vector(1 downto 0);

-- enable
signal shw1,shw2,shw3,shw4,shw5,shw6,shw7,shw8,shw9,shw10,shw_en,blk_en : std_logic;
signal li_shw1,li_shw2,li_shw3,li_shw4,li_shw5,li_shw6 : std_logic;
signal a_blk : std_logic;
signal data_blk : std_logic_vector(7 downto 0);
--signal data1_tmp_reg,data2_tmp_reg,data3_tmp_reg : std_logic_vector(7 downto 0);
signal data1_reg,data2_reg,data3_reg : std_logic_vector(7 downto 0);
signal data_in_reg,data_out_reg : std_logic_vector(7 downto 0);
signal addr1_reg,addr2_reg,addr3_reg : std_logic_vector(16 downto 0);
--signal o1_en,o2_en,o3_en : std_logic;
signal nwr1_reg,nwr2_reg,nwr3_reg : std_logic;

begin
f <= not odd;
dclk <= clk1;
```

```
-- F edge detection
process(clk1)
begin
	if rising_edge(clk1) then
		edge_f_2 <= edge_f_1;
		edge_f_1 <= f;
	end if;
end process;

f_fall <= (not edge_f_1) and edge_f_2;
fedge <= edge_f_1 xor edge_f_2;

-- HS edge detection
process(clk1)
begin
	if rising_edge(clk1) then
		edge_HS_2 <= edge_HS_1;
		edge_HS_1 <= HS;
	end if;
end process;

HS_fall <= (not edge_HS_1) and edge_HS_2;

-- frame counter
process (reset,clk1)
begin
	if reset = '1' then
		frame <= "10";
	elsif rising_edge(clk1) then
		if f_fall = '1' then
			case frame is
				when "10" => frame <= "00";
				when "00" => frame <= "01";
				when others => frame <= "10";
			end case;
		end if;
	end if;
end process;

-- shw signal delay
process(clk1,reset)
begin
	if reset = '1' then
		shw1 <= '0';
		shw2 <= '0';
		shw3 <= '0';
		shw4 <= '0';
		shw5 <= '0';
		shw6 <= '0';
		shw7 <= '0';
		shw8 <= '0';
		shw9 <= '0';
		shw10 <= '0';
	elsif rising_edge(clk1) then
		shw10 <= shw9;
		shw9 <= shw8;
		shw8 <= shw7;
		shw7 <= shw6;
		shw6 <= shw5;
		shw5 <= shw4;
		shw4 <= shw3;
		shw3 <= shw2;
		shw2 <= shw1;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
                shw1 <= shw;
        end if;
end process;

process(reset,clk1)
begin
        if reset = '1' then
                li_shw1 <= '0';
                li_shw2 <= '0';
                li_shw3 <= '0';
                li_shw4 <= '0';
                li_shw5 <= '0';
                li_shw6 <= '0';
        elsif rising_edge(clk1) then
                li_shw6 <= li_shw5;
                li_shw5 <= li_shw4;
                li_shw4 <= li_shw3;
                li_shw3 <= li_shw2;
                li_shw2 <= li_shw1;
                li_shw1 <= blk_li_en;
        end if;
end process;

shw_en <= shw2 and shw6;
--blk_en <= shw2 xor shw10;
blk_en <= (shw2 xor shw6) or li_shw6;
--
-- start component
-- clk_divider

clk_cir: clk_divider port map (clk,clk1,clk3,clk90,reset);

-- i2c
i2c_cir: i2c port map (clk3,reset,reset_n,scl,sda);

-- remote
remote_cir: remote2 port map (clk3,reset,IR,pic_on,mode,start_pixel,start_line,start_pixel2,start_line2);

-- horinter
hor_int_cir: horinter2 port map(clk1,clk90,f,f_fall,data_in_reg,li_en,str,mode,data_h_out,h_en,addr_str_h);

-- ver_add
v_en <= '1';
ver_add_cir: ver_add port map (data_v_in,data_v_out,clk1,clk90,v_en,reset,mode,addr_str_v,nwr_v2);

-- line counter
licnt: line_counter PORT MAP (HS_fall,fedge,clk1,li_cnt);

-- pixel counter
picnt: pixel_counter PORT MAP (clk1,HS,pi_cnt);

-- compare data part1
shw_str: compare_block port map
(clk1,mode,start_line,start_pixel,start_line2,start_pixel2,li_cnt,pi_cnt,HS_fall,str,shw,li_en,blk_li_en);

--create addr_shw
ad_shw: shw_ad_cnt port map (clk1,shw1,fedge,f,mode,addr_shw);

-- end component
--
```

```
process(clk1)
begin
	if rising_edge(clk1) then
		if blk_en <= '0' then
			a_blk <= '1';
		else
			a_blk <= not a_blk;
		end if;
	end if;
end process;

data_blk <=	"00010000" when a_blk = '0' else
			"10000000";

process(clk1)
begin
	if rising_edge(clk1) then
		nwr_v3 <= nwr_v2;
	end if;
end process;

data1 <= data_h_out when (h_en = '1' and frame = "00") else
		data_v_out when (nwr_v3 = '0' and frame = "01") else
		(others =>'Z');
data2 <= data_h_out when (h_en = '1' and frame = "01") else
		data_v_out when (nwr_v3 = '0' and frame = "10") else
		(others =>'Z');
data3 <= data_h_out when (h_en = '1' and frame = "10") else
		data_v_out when (nwr_v3 = '0' and frame = "00") else
		(others =>'Z');

-- data1&2 reg
process(clk1)
begin
	if rising_edge(clk1) then
		data1_reg <= data1;
		data2_reg <= data2;
		data3_reg <= data3;
		data_in_reg <= data_in;

		data_out <= data_out_reg;
		addr1 <= addr1_reg;
		addr2 <= addr2_reg;
		addr3 <= addr3_reg;
		nwr1 <= nwr1_reg;
		nwr2 <= nwr2_reg;
		nwr3 <= nwr3_reg;

	end if;
end process;

-- data_v_in mux clk2
data_v_in <=	data1_reg when frame = "01" else
				data2_reg when frame = "10" else
				data3_reg;

data_out_reg <=	data_blk when (pic_on = '1' and blk_en = '1') else
		data1_reg when (pic_on = '1' and blk_en = '0' and shw_en = '1' and frame = "10") else
		data2_reg when (pic_on = '1' and blk_en = '0' and shw_en = '1' and frame = "00") else
		data3_reg when (pic_on = '1' and blk_en = '0' and shw_en = '1' and frame = "01") else
		data_in_reg;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
-- addr1
addr1_reg <=        addr_str_h when frame = "00" else
                    addr_str_v when frame = "01" else
                    addr_shw;

-- addr2
addr2_reg <=        addr_str_h when frame = "01" else
                    addr_str_v when frame = "10" else
                    addr_shw;

-- addr2
addr3_reg <=        addr_str_h when frame = "10" else
                    addr_str_v when frame = "00" else
                    addr_shw;

-- nen1
nen1 <= '0';
nen2 <= '0';
nen3 <= '0';

-- nwr1
nwr1_reg <=         not h_en when frame = "00" else
                    nwr_v2 when frame = "01" else
                    '1';

nwr2_reg <=         not h_en when frame = "01" else
                    nwr_v2 when frame = "10" else
                    '1';

nwr3_reg <=         not h_en when frame = "10" else
                    nwr_v2 when frame = "00" else
                    '1';
end Behavioral;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
บทความที่ได้รับการตีพิมพ์ใน
ECTI-CON 2004

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A 4x Video Image Magnifier Using Pixel Replication Technique

Patheera Uthaichana and Ekachai Leelarasmee

Patheera@chula.com
Ekachai.l@chula.ac.th

ABSTRACT

A 4x video image magnifier enlarges a sign language window, inserted at the corner of main screen of TV program, by 4 times using pixel replication technique. The circuit has a video decoder, encoder and two 64KByte memory chips to digitize, store and display the window image, delayed by one video frame. Its control unit is implemented with a small complex programmable logic device. A microcontroller plus an infrared remote control are also provided to facilitate viewers in locating the stored and displayed windows.

Keywords: Video magnifier, pixel replication

1. INTRODUCTION

To assist viewers who are deaf, some TV programs insert a sign language window at the lower right corner of main screen as shown in the left picture of Fig.1. Nevertheless, the sign language window is typically too small for the viewer to recognize the movement of the mouth or hands of the interpreter. We therefore purpose a 4x video image magnifier solves this problem by enlarging the sign language window as shown in the right picture of Fig.1. It comes with an infrared remote controller for selecting the locations of the input and its expanded windows. The enlargement is carried out digitally and, hence, a video signal digitizer is used within the magnifier. To ease the hardware implementation, an enlarging mechanism based on the pixel replication algorithm is used.

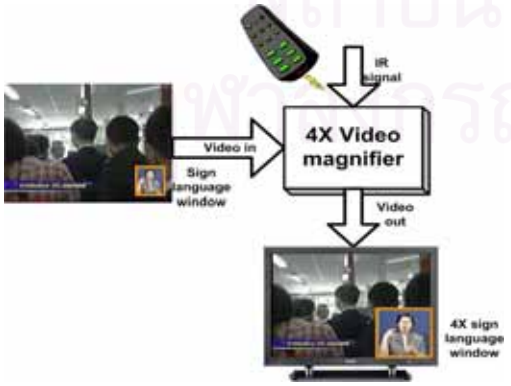


Fig.1: *The function of a 4x video image magnifier*

This algorithm copies the luminance and chrominance of a pixel to its neighbours. For the special case of a 4x magnification, this corresponds to duplicating the pixel to its down-right borders, creating 4 identical pixels as shown in Fig.2.

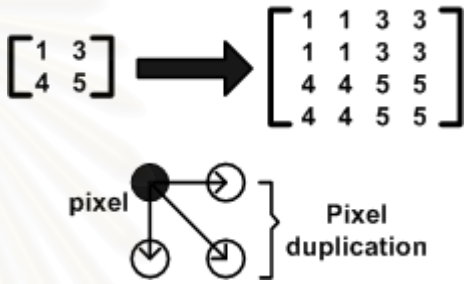


Fig.2: *Example of 4x enlarging process by pixel replication algorithm*

2. IMPLEMENTATION

We choose to implement the pixel replication algorithm to enlarge the sign language window as illustrated in the data flow diagrams of Fig.3a and 3b.

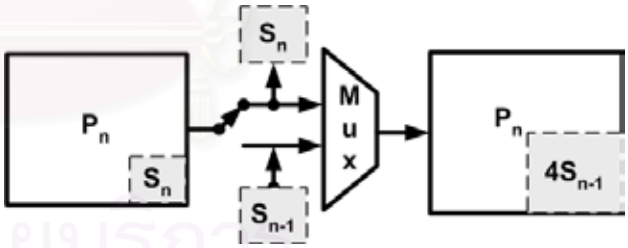


Fig.3a: *Data flow of digital video image during the n-th frame*

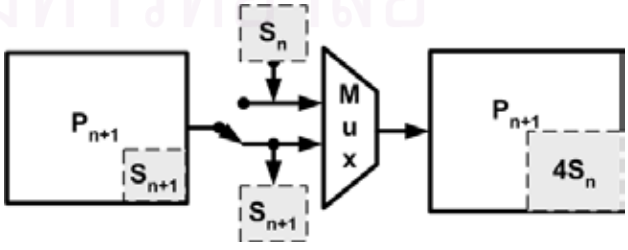


Fig.3b: *Data flow of digital video image during the (n+1)-th frame*

Here P_n denotes the digitized video picture at the n-th frame and S_n, its sign window. Two memory storages are used to store the last two consecutive sign language window, i.e. (S_n, S_{n-1}) at the n-th frame and (S_{n+1}, S_n) at the (n+1)-th frame. A data multiplexer (Mux) is used to generate the picture output by selecting either the current picture input P_n or its previous sign language window S_{n-1}. By repeat inserting S_{n-1} 4 times, a 4 times enlargement of the sign language window in P_n is achieved. Note that there is only a frame delay of the enlarged window. This effect should not be noticeable to the viewer.

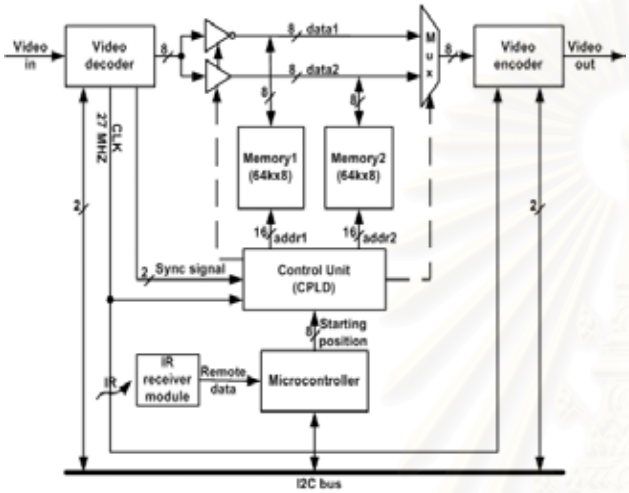


Fig.4: *The architecture of video image magnifier*

Fig.4 shows the architecture of a video image magnifier that implements the above data flow diagrams. The analog video input must be digitized by a video decoder to obtain P_n. Its output is an 8 bit data stream which can be routed to either one of the two data buses through tri-state buffers. Each bus has a 64KByte SRAM memory connected to it to store the sign language window S_n or S_{n-1}. These two buses also feed a two-to-one 8 bit multiplexers for generating the digital video picture Pn that has $4S_{n-1}$ inserted in it. This picture is the converted into an analog output by the video encoder. Other blocks in the architecture are the control unit and microcontroller. They are used to generate appropriate controlling signals and facilitate user programming through the infrared remote controller.

3. HARDWARE CIRCUITS AND THEIR OPERATIONS

Fig.5 shows the video decoding circuit using SAA7113 decoder chip. It digitizes the analog video input and generates an 8 bit data stream output at 27 MHz in accordance with ITU-R BT.601 [1] and ITU-R BT.656 standards [2]. The chip also extracts the field (F) and horizontal (HS) signals from the video input as well as providing a 27 MHz clock for synchronization with other blocks. An I^2C bus interface is also used to program its function such as selecting PAL/NTSC system and etc. Its output data stream consists of sequences of luminance (Y) and chrominance (Cr,Cb) in the following format Cb_n Y_n Cr_n Y_{n+1}. Hence a 64KByte memory can store up to 32 K pixels while the whole screen size is 720x572 pixels.

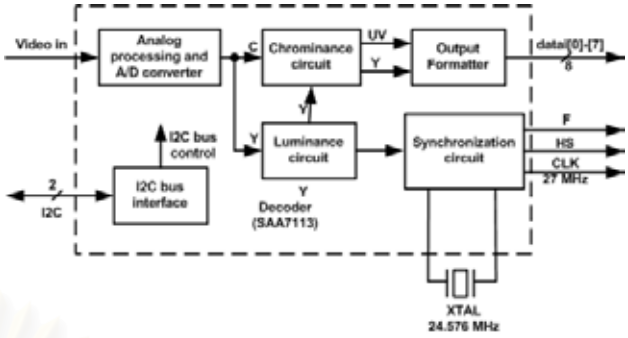


Fig.5: *Video decoder*

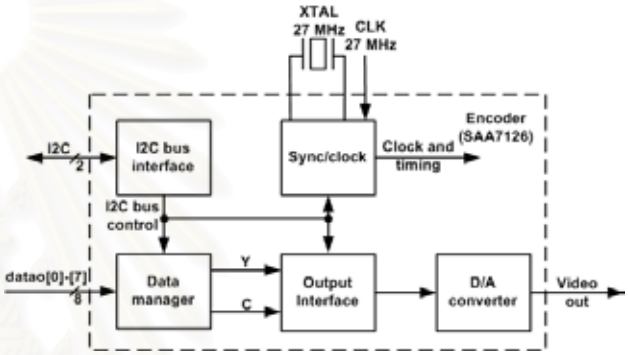


Fig.6: *Video encoder*

Fig.6 shows the encoding circuit which uses the SAA7126 chip operating at 27 MHz, e.g. PAL/NTSC, for receiving the video data stream and generating its analog composite video signal. Its function must also be programmed through the I^2C bus.

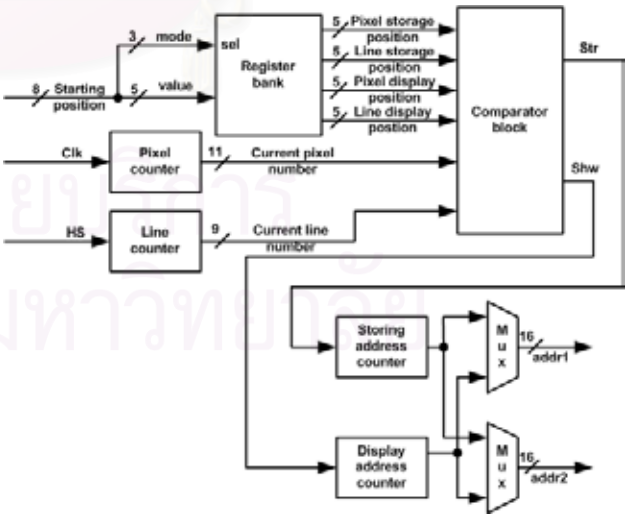


Fig.7: *Control Unit block diagram*

Fig.7 shows the block diagram of the control unit implemented in an Altera EPM7128LC84 CPLD chip. Its main use is to synthesize two 16 bit counters to generate two sets of memory addresses. One is for storing sign language window, i.e. S_n of Fig.3a. The other is for

displaying another sign language window, i.e. S_{n-1} of Fig.3a, which must be read out 4 times in order to produce a 4X magnification on the output screen.

Pixel and line counter generate current position and fed to comparator block for comparing with the set-point window positions. The set-point positions are extracted from starting position by decoder within register banks. These data consist of the storage and display window position. The comparator compares the position and generates enable signals to start and stop the address generation so that the locations of window storage and display can be adjusted by changing these data.

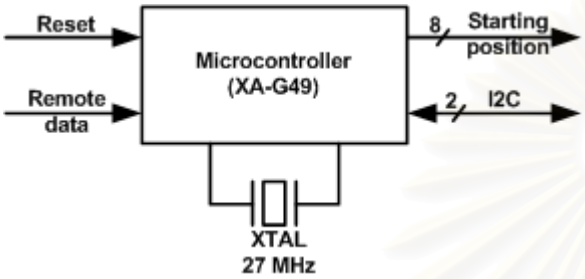


Fig.8: *Microcontroller*

Fig.8. shows the details of the microcontroller unit together with the infrared demodulator. The MCU is XA-G49, a 16 bit microcontroller chip using the extended architecture of the well known MCS-51 instruction set. It is used to decode the remote data obtained from the infrared demodulator. Then data are user commands for selecting the locations of the sign language window to be stored or displayed as well as turning off the magnification mode. The MCU then generates these locations in terms of pixels and TV scan lines and write them in the control registers within the comparator block of the control unit.

During initialization, the MCU also programs the function of the video decoder and encoder through its I^2C bus at 400 kHz frequency [9]. The MCU controls the bus by generating the clock signal and sends command to read or write register within decoder and encoder.

4. RESULTS

The prototype of a 4X video image magnifier is shown in Fig. 9.

In testing condition, the input source is obtained from video tape player which records TV program and the output is connected to a TV receiver for displaying the image.

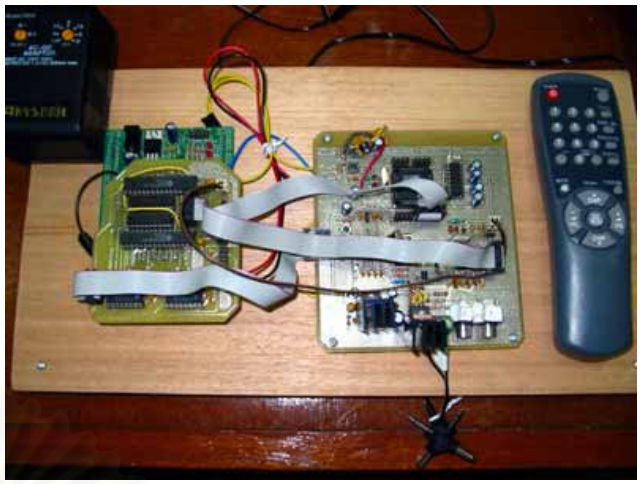


Fig.9: *A prototype of a 4x video image magnifier*

Fig.10*: The original image from a video tape player*

Fig. 10 shows the output image in normal operation with no magnification. Notice the display of the sign language at the lower right corner. This is too small for the deaf viewers. However, this window already obscures part of the TV screen and would annoy normal viewers, had it been bigger.

Fig.11: *The 4X enlarged image from the video image magnifier*

Fig.11 shows the output image when a 4X magnification is activated. The sign language is now easily seen.

Fig.12: *The enlarged image which is changed the position by remote controller*

Fig.12 shows the output image when the enlarged window is moved to the top right corner by using a remote controller. The viewer can adjust this location to suit his viewing purpose.

5. CONCLUSION

A simple hardware for enlarging an image window within a video picture frame is proposed, constructed and tested. The main idea is to store the digitized original image and replay it back using pixel replication algorithm. This allows the use of a small CPLD chip for implementing the algorithm. The addition of an infrared remote controller and microcontroller chip enable the viewer to turn on/off the magnification and set the locations of the windows. This 4X video image magnifier will be highly beneficial to deaf viewers since the original sign language window is too small to see by them.

6. REFERENCES

[1] *ITU-R BT.601-5 standard*, http://www/itu.int/itudoc/itu-r/rec/bt/

[2] *ITU-R BT.656-4 standard*, http://www/itu.int/itudoc/itu-r/rec/bt/

[3] Gene A. Tagliarini “Enlarging or Contracting a Digital Image,”

[4] Joseph Y. Pain ad Lori Lucke, “*Viewing Effect Algorithm for image expansion*”

[5] Anil K. Jain, *Fundamentals of digital image processing*, New Jersey, Prentice-Hall, 1989

[6] “*SAA7113H: 9-bit video input processor*”. Philips semiconductor, 1999

[7] “*SAA7126H: 7127H Digital video encoder*”. Philips semiconductor, 2000

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวภัทธีรา อุทัยชนะ เกิดวันที่ 24 มิถุนายน พ.ศ. 2524 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2544 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมไฟฟ้าอิเล็กทรอนิกส์เชิงเลข ที่คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย