

การออกแบบวงจรถยายภาพภาษามือแบบ 2x2 เท่า สำหรับโทรทัศน์ความละเอียดสูงเต็มรูปแบบโดย
ใช้วิธีการประมาณค่าในช่วงแบบไบควบิก



บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A Design of a 2x2 Sign Language Image Enlarger for Full HD Television Using Bicubic
Interpolation

Mr. Auangkun Rangsikunpum



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

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การออกแบบวงจรถยายภาพภาษามือแบบ 2x2 เท่า สำหรับโทรทัศน์ความละเอียดสูงเต็มรูปแบบโดยใช้วิธีการ ประมาณค่าในช่วงแบบไบคิวบิก
โดย	นายเอื้ออังกูร รังสิกรรพุม
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. สุรีย์ พุ่มรินทร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รองศาสตราจารย์ ดร. เอกชัย ลีลารัมย์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัย
หนึ่งของการศึกษาตามหลักสูตรปริญญาโทบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. วันเฉลิม โปรา)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร. สุรีย์ พุ่มรินทร์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(รองศาสตราจารย์ ดร. เอกชัย ลีลารัมย์)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร. วีทิต วรรณเลิศลักษณ์)

เอื้ออังกูร รังสิกรรพุม : การออกแบบวงจรขยายภาพภาษามือแบบ 2x2 เท่า สำหรับ
 โทรทัศน์ความละเอียดสูงเต็มรูปแบบโดยใช้วิธีการประมาณค่าในช่วงแบบไบคิวบิก (A
 Design of a 2x2 Sign Language Image Enlarger for Full HD Television Using
 Bicubic Interpolation) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. สุรีย์ พุ่มรินทร์, อ.ที่ปรึกษา
 วิทยานิพนธ์ร่วม: รศ. ดร. เอกชัย ลีลารัมย์, 58 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอการออกแบบวงจรสำหรับการขยายภาพภาษามือที่ถูกรบกวนไว้
 ในรายการโทรทัศน์บางรายการเพื่อช่วยให้ผู้พิการทางการได้ยินในการเข้าถึงข้อมูลได้ดีขึ้นโดยไม่ไป
 รบกวนผู้ชมทั่วไป วงจรนี้ใช้วิธีการประมาณค่าในช่วงแบบไบคิวบิกในการขยายภาพภาษามือขึ้นเป็น
 2x2 เท่าจากเดิม การขยายภาพ 2x2 เท่านี้จะทำให้สมการคำนวณของวิธีการประมาณค่าในช่วง
 แบบไบคิวบิกลดความซับซ้อนเหลือเพียงแค่การหารด้วย 2 หรือการเลื่อนขวาในระบบเลขฐานสอง ซึ่ง
 ทำให้การนำไปใช้งานกับฮาร์ดแวร์สามารถทำได้อย่างมีประสิทธิภาพ สำหรับวงจรต้นแบบโครงสร้าง
 ของวงจรถูกนำไปใช้กับบอร์ดทดลอง Zedboard ที่มีชิพที่สามารถโปรแกรมได้ Zynq-7000 SoC ที่
 ประกอบไปด้วย หน่วยประมวลผล dual-core ARM Cortex-A9 และ FPGA หน่วยประมวลผลมี
 หน้าที่เพียงตั้งค่าเริ่มต้นให้กับวงจรส่วน FPGA ต้องทำการขยายภาพให้ได้ตามเวลาจริง ตำแหน่งและ
 ขนาดของกรอบภาพการขยายจะสามารถปรับได้โดยใช้ปุ่มกดบนบอร์ดเพื่อทำการจับภาพภาษามือ ใน
 รุ่นต้นแบบขนาดของกรอบการขยายสามารถปรับได้ตั้งแต่ 115x115 จุดภาพถึง 371x371 จุดภาพ
 ผลการทดลองแสดงให้เห็นว่าวงจรขยายภาพภาษามือสามารถทำงานได้ดีกับระบบโทรทัศน์ที่มี
 สัญญาณภาพความละเอียดสูงเต็มรูปแบบหรือ 1920x1080 จุดภาพ ที่ความถี่ภาพ 60 Hz ซึ่ง
 หมายความว่าวงจรสามารถขยายภาพได้ทันการแสดงผลภาพ 1 เฟรมหรือ 16.7 ms ที่ความถี่นาฬิกา
 148.5 MHz

ภาควิชา วิศวกรรมไฟฟ้า

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2559

ลายมือชื่อนิสิต

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ลายมือชื่อ อ.ที่ปรึกษาร่วม

5870285121 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS: SIGN LANGUAGE / HDMI / ZYNQ / BICUBIC INTERPOLATION / DEAF PERSON / IMAGE EXPANDER

AUANGKUN RANGSIKUNPUM: A Design of a 2x2 Sign Language Image Enlarger for Full HD Television Using Bicubic Interpolation. ADVISOR: ASST. PROF. SUREE PUMRIN, Ph.D., CO-ADVISOR: ASSOC. PROF. EKACHAI LEELARASMEE, Ph.D., 58 pp.

This thesis presents a design of device for expanding sign language images inserted in some television programs in order to help deaf people approaching the information without disrupting normal viewers. It is used bicubic interpolation to enlarge sign language window by 2x2 times of its size. The 2x2 expansion simplifies the bicubic interpolation formula to just division by two operations or right shift operations in binary number which enables effectively implementing on hardware. The proposed architecture is implemented on Zedbaord with programmable Zynq-7000 SoC combining dual-core ARM Cortex-A9 processors with FPGA. The processor is only responsible to initialize the device while the programmed FPGA performs the real-time image expansion. Using its onboard push buttons, the position and size of the expanding area are adjustable for capturing the sign language window. In this prototype, the expanding area size can be adjusted form 115x115 pixels to 371x371 pixels. Experimental results show that the prototype can operate well with Full-HD or 1920x1080 pixels 60 Hz HDMI video source. This means the image expansion process can be executed within one frame or 16.7 ms with clock frequency of 148.5 MHz.

Department: Electrical Engineering

Field of Study: Electrical Engineering

Academic Year: 2016

Student's Signature

Advisor's Signature

Co-Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือเป็นอย่างดีของ ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และ รองศาสตราจารย์ ดร.เอกชัย ลีลารัมย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ซึ่งท่านทั้งสองได้ให้คำแนะนำ ข้อคิดเห็นและการสนับสนุนต่างๆ ที่เป็นประโยชน์อย่างยิ่งในการทำวิจัย

ขอขอบคุณการสนับสนุนอุปกรณ์การทำงานวิจัยจากห้องปฏิบัติการวิจัย ระบบสมองกลฝังตัวและการออกแบบวงจรรวม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และทุนอุดหนุนการศึกษาจากโครงการศิษย์ก้นกุฏิ ระดับปริญญาโท ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ประจำปีการศึกษา 2558

ขอขอบคุณ ท่านอาจารย์ เพื่อนๆ พี่ๆ และน้องๆ ในห้องปฏิบัติการวิจัยการออกแบบและประยุกต์วงจรรวมทุกคนสำหรับความช่วยเหลือ ช่วยรับฟังปัญหา ให้คำปรึกษาต่างๆ และแบ่งปันประสบการณ์ดีๆ ด้วยดีมาตลอด

ขอขอบคุณพีภัทธิรา อุทัยชนะ และพีๆ ในบริษัท Design Gateway ที่ให้คำปรึกษาในเชิงลึกของการออกแบบวงจรและการออกแบบ FPGA

สุดท้ายนี้ ข้าพเจ้าใคร่ขอกราบขอบพระคุณบิดา-มารดา อันเป็นที่เคารพรัก ที่คอยดูแลเอาใจใส่ สนับสนุนในด้านการเงิน และให้กำลังใจแก่ข้าพเจ้าด้วยดีเสมอมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญรูปภาพ.....	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของวิทยานิพนธ์	2
1.4 วิธีดำเนินงานวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 มาตรฐานและทฤษฎีที่เกี่ยวข้อง	4
2.1 โครงสร้างสัญญาณภาพเคลื่อนไหวแบบดิจิทัล.....	4
2.2 ระบบส่งสัญญาณมัลติมีเดียความละเอียดสูง (High-Definition Multimedia Interface : HDMI)	6
2.2.1 ช่องทางการสื่อสารโปรโทคอลของ HDMI.....	7
2.2.2 Extended Display Identification Data (EDID).....	8
2.3 มาตรฐาน ITU-R BT.656.....	8
2.4 วิธีการขยายภาพ.....	11
2.4.1 วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood).....	12
2.4.2 วิธีการประมาณค่าแบบไบลิเนียร์ (Bilinear interpolation method).....	13
2.4.3 วิธีการประมาณค่าแบบไบคิวบิก (Bicubic interpolation method).....	15

2.4.4	วิธีการขยายภาพแบบอื่นๆ.....	16
2.4.5	วิธีการขยายภาพ 2 เท่าด้วยวิธีการประมาณค่าไบคิวบิก.....	16
2.4.6	การวัดค่าคุณภาพของการขยายภาพด้วย PSNR.....	17
บทที่ 3	บอร์ดทดลอง	20
3.1	บอร์ดทดลอง Zedboard	20
3.1.1	การติดต่อสื่อสารแบบ AXI.....	22
3.1.2	ส่วนการแสดงผลภาพขาออก HDMI ของ Zedboard.....	24
3.2	บอร์ดเสริม FMC-HDMI.....	25
บทที่ 4	วงจรขยายภาพภาษามือ	27
4.1	การใช้งานวงจรขยายภาพเคลื่อนไหวล้ามภาษามือ.....	27
4.2	การออกแบบวงจรขยายภาพภาษามือ.....	28
4.2.1	ตัวรับสัญญาณ HDMI (HDMI receiver)	29
4.2.2	ตัวส่งสัญญาณ HDMI (HDMI transmitter)	31
4.2.3	วงจรตัวควบคุมการขยายภาพ (Expansion controller).....	32
บทที่ 5	ผลการทดลองและสรุปผล.....	40
5.1	การวิเคราะห์วงจรขยายภาพภาษามือ	40
5.2	การทดลองใช้งานเครื่องขยายภาพภาษามือ	44
5.3	สรุปผล.....	47
5.4	ข้อเสนอแนะ	48
รายการอ้างอิง	49
ภาคผนวก.....	52
ภาคผนวก ก	บทความที่ได้ตีพิมพ์ใน ECTI-CON 2017.....	53
ประวัติผู้เขียนวิทยานิพนธ์	58

สารบัญตาราง

ตารางที่ 2.1 เปรียบเทียบค่า PSNR กับรูปภาพล่ามภาษามือ 3 ภาพด้วยวิธีการประมาณค่า ต่างๆ.....	19
ตารางที่ 5.1 เส้นทางในวงจรที่มีการหน่วงมากที่สุด	41
ตารางที่ 5.2 การใช้ทรัพยากรภายใน FPGA.....	41
ตารางที่ 5.3 การเปรียบเทียบโครงสร้างวงจรขยายพด้วยวิธีการประมาณค่าในช่วงแบบไปคิวบิก ของวงจรขยายภาพล่ามภาษามือกับโครงสร้างอื่นๆ.....	43



สารบัญรูปภาพ

รูปที่ 1.1 การใช้งานเครื่องขยายภาพภาษามือ.....	2
รูปที่ 2.1 สัญญาณเสริมที่จำเป็นในการแสดงภาพในแต่ละเฟรม	4
รูปที่ 2.2 การแสดงเส้นภาพในแบบ interlaced	5
รูปที่ 2.3 การแสดงเส้นภาพในแบบ progressive	5
รูปที่ 2.4 ภาพตัดขวางของส่วนหัวของ HDMI	6
รูปที่ 2.5 โหมดการทำงานต่างๆของ TDMS ในหนึ่งเฟรมความละเอียด 720×480 จุดภาพ.....	9
รูปที่ 2.6 รูปแบบการส่งสัญญาณภาพตามมาตรฐาน ITU.R BT656	10
รูปที่ 2.7 ตัวอย่างการขยายภาพขนาด 2×2 เท่า	11
รูปที่ 2.8 การชักตัวอย่างซ้ำเพื่อขยายภาพในแนวนอนขนาด $k = 2$ เท่า	12
รูปที่ 2.9 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 4 จุดคือ I_{11}, I_{12}, I_{21} และ I_{22} เมื่อขยายภาพด้วยวิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด	12
รูปที่ 2.10 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 4 จุดคือ I_{11}, I_{12}, I_{21} และ I_{22} เมื่อขยายภาพด้วยวิธีประมาณค่าแบบไบลิเนียร์.....	13
รูปที่ 2.11 การประมาณค่าแบบไบลิเนียร์ 1 มิติ.....	14
รูปที่ 2.12 แสดงการประมาณค่าแบบไบลิเนียร์ 2 มิติด้วยการทำการประมาณค่าแบบไบลิเนียร์ 1 มิติ 2 แนวแกน	14
รูปที่ 2.13 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 16 จุดเมื่อขยายภาพด้วยวิธีประมาณค่าแบบไบควบิก	15
รูปที่ 2.14 การประมาณค่าแบบไบควบิก 1 มิติ.....	15
รูปที่ 2.15 วงจรการประมาณค่าด้วยไบควบิกหนึ่งมิติ (1DIU).....	17
รูปที่ 2.16 การทำการหาค่า PSNR สำหรับการขยายภาพ 2 เท่า	18
รูปที่ 2.17 ภาพล่ามภาษามือทั้ง 3 ภาพที่นำมาหาค่า PSNR	19
รูปที่ 3.1 บอร์ดทดลอง Zedboard.....	20

รูปที่ 3.2 โมเดลอย่างง่ายของ Zynq architecture	21
รูปที่ 3.3 โครงสร้างการช่องทางการเขียน	22
รูปที่ 3.4 โครงสร้างการช่องทางการอ่าน.....	23
รูปที่ 3.5 ตัวอย่าง handshake VALID ก่อน READY :ที่เวลา T1 ต้นทางจะให้ VALID เป็น ‘1’ จากนั้นที่เวลา T2 ปลายทางจะให้ค่า READY เป็น ‘1’ จากนั้นต้นทางจะคงค่า INFORMATION ให้เสถียรจนการส่งเกิดขึ้นที่เวลา T3.....	24
รูปที่ 3.6 รูปแบบของสัญญาณภาพ 16 bit YCbCr 4:2:2	25
รูปที่ 3.7 บอร์ดเสริม FMC-HDMI	25
รูปที่ 3.8 แผนภาพบล็อกของ FMC-HDMI	26
รูปที่ 4.1 แสดงการใช้งานวงจรถ่ายภาพภาษามือ.....	27
รูปที่ 4.2 แสดงการรับส่งสัญญาณภาพเคลื่อนไหวของวงจรถ่ายภาพเพื่อผู้พิการทางการได้ยิน	28
รูปที่ 4.3 โครงสร้างของวงจรถ่ายภาพภาษามือ.....	28
รูปที่ 4.4 การใช้งานตัวรับสัญญาณ HDMI.....	29
รูปที่ 4.5 แผนภาพบล็อกของวงจร ADV7611.....	30
รูปที่ 4.6 การใช้งานตัวส่งสัญญาณ HDMI	31
รูปที่ 4.7 แผนภาพบล็อกของวงจร ADV7511.....	32
รูปที่ 4.8 แผนภาพบล็อกของวงจรตัวควบคุมการขยายภาพ.....	33
รูปที่ 4.9 โครงสร้างของวงจรตัวต่อประสานปุ่มกด (button interface).....	34
รูปที่ 4.10 การใช้งานหน่วยความจำ DDR3 ขณะช่วงเวลาภาพเฟรมที่ n และภาพเฟรมที่ n+1	35
รูปที่ 4.11 วงจรนับ address ของ AXI Write และ AXI Read.....	36
รูปที่ 4.12 แผนภาพบล็อกของวงจรถ่ายภาพ	36
รูปที่ 4.13 การทำการประมาณค่าไบควิกด้วยการทำการประมาณค่าหนึ่งมิติสองครั้ง.....	37
รูปที่ 4.14 โครงสร้างของตัวขยายภาพในส่วนของการขยายภาพ.....	38
รูปที่ 4.15 โครงสร้างของวงจรควบคุมของวงจรถ่ายภาพ	39

รูปที่ 5.1 แผนภาพบล็อกวงจรตัวควบคุมการขยายภาพใน Vivado.....42

รูปที่ 5.2 การติดตั้งการทดลองเครื่องขยายล่ำมภาษามือ.....44

รูปที่ 5.3 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ำมภาษามือขนาด 187x243 จุดภาพ..45

รูปที่ 5.4 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ำมภาษามือขนาด 219x243 จุดภาพ..46

รูปที่ 5.5 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ำมภาษามือขนาด 235x251 จุดภาพ..46

รูปที่ 5.6 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ำมภาษามือขนาด 259x299 จุดภาพ..47



บทที่ 1

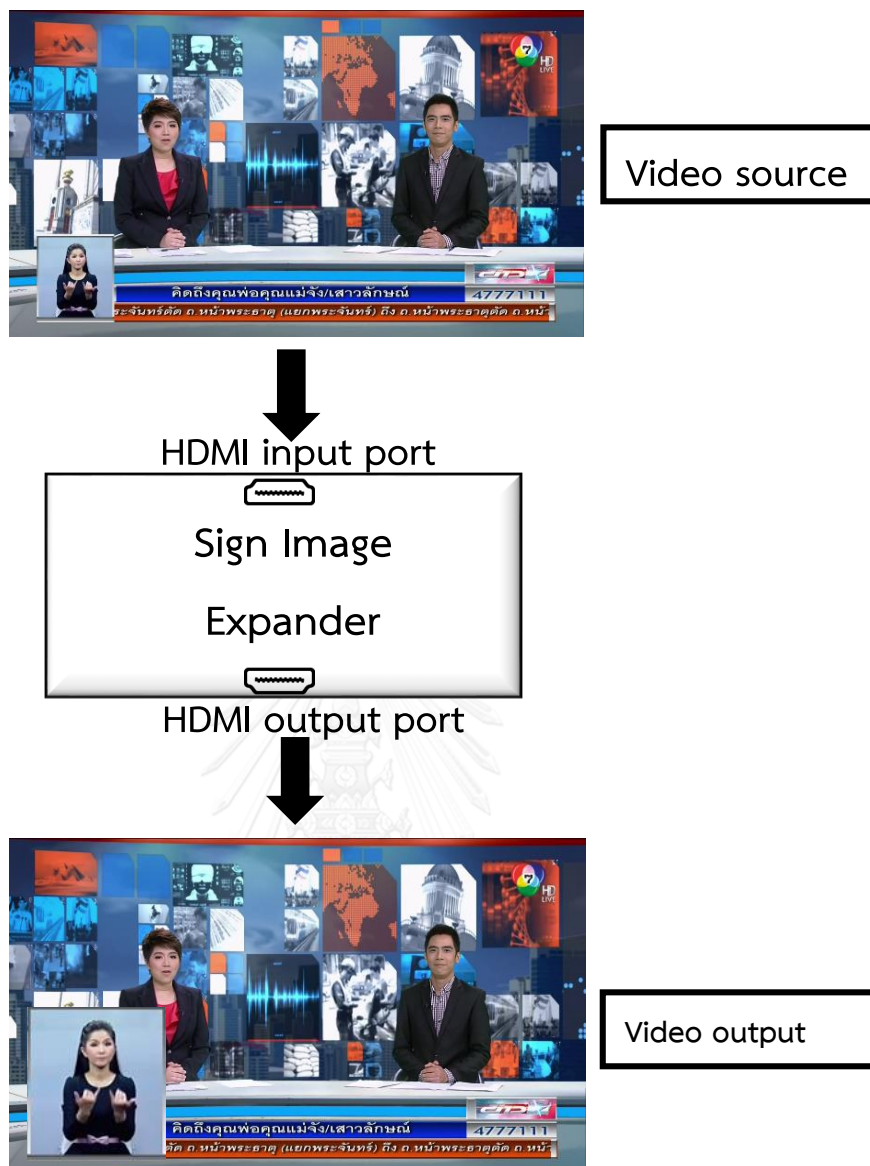
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีมีความสำคัญในการเข้ามาช่วยอำนวยความสะดวกในการทำกิจกรรมต่างๆ โดยเฉพาะอย่างยิ่งกับผู้ที่มีความบกพร่องทางร่างกายที่จำเป็นต้องพึ่งเทคโนโลยีเพื่อให้การดำเนินชีวิตในแต่ละวันเป็นไปได้อย่างปกติ รายการโทรทัศน์บางรายการเช่น รายการข่าว เป็นต้นได้มีการแทรกภาพภาษามือเพื่อให้ผู้พิการทางการได้ยินได้ทราบถึงเนื้อหาของรายการโทรทัศน์นั้นๆได้ แต่การแทรกภาพภาษามือนั้นมีขนาดที่ค่อนข้างเล็ก ทำให้ผู้พิการทางการได้ยินไม่สามารถอ่านปากและภาษามือได้อย่างชัดเจน การรับข้อมูลจากรายการโทรทัศน์นั้นๆจึงไม่เข้าใจได้ทั้งหมด ในการแก้ปัญหาภาพภาษามือที่เล็กเกินไปนั้นอาจแก้ไขได้โดยให้รายการโทรทัศน์นั้นๆเพิ่มขนาดภาพภาษามือให้ใหญ่กว่าเดิม แต่การกระทำดังกล่าวจะส่งผลให้ไปรบกวนภาพเนื้อหาหลักของรายการและทำให้เกิดความรำคาญแก่ผู้รับชมปกติได้

งานวิจัยจึงได้นำเทคโนโลยีมาช่วยแก้ปัญหาภาพภาษามือที่มีขนาดเล็กเกินไปสำหรับผู้พิการทางการได้ยิน โดยสร้างเครื่องขยายภาพภาษามือ ที่สามารถขยายภาพภาษามือได้ ซึ่งจะทำให้ผู้พิการสามารถมองเห็นภาพภาษามือได้อย่างชัดเจนมากขึ้นและไม่เป็นการไปรบกวนผู้รับชมปกติ

เครื่องขยายภาพภาษามือจะรับสัญญาณภาพจากผ่านทาง ระบบส่งสัญญาณมัลติมีเดียความละเอียดสูงขาเข้า (High-Definition Multimedia Interface input port : HDMI input port) และจะดึงส่วนของภาพภาษามือออกมา และทำการขยายภาพ จากนั้นก็นำภาพภาษามือที่ถูกขยายแล้วแทรกกลับเข้าไปในภาพเดิม แล้วนำสัญญาณภาพดังกล่าวนั้นส่งออกทาง ระบบส่งสัญญาณมัลติมีเดียความละเอียดสูงขาออก เข้า (High-Definition Multimedia Interface input port : HDMI output port) ดังรูปที่ 1.1



รูปที่ 1.1 การใช้งานเครื่องขยายภาพภาษามือ

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาเครื่องขยายภาพภาษามือ ที่ช่วยให้ผู้พิการทางการได้ยินได้รับชมภาษามือได้อย่างชัดเจน

1.3 ขอบเขตของวิทยานิพนธ์

1. ออกแบบวงจรต้นแบบของเครื่องขยายภาพภาษามือที่มีคุณสมบัติดังนี้
 - 1.1) ขยายภาพภาษามือได้ขนาด 2x2 เท่า
 - 1.2) ใช้วิธีการขยายภาพแบบไบคิวบิก
 - 1.3) สามารถปรับขนาดของภาพที่ต้องการขยายได้

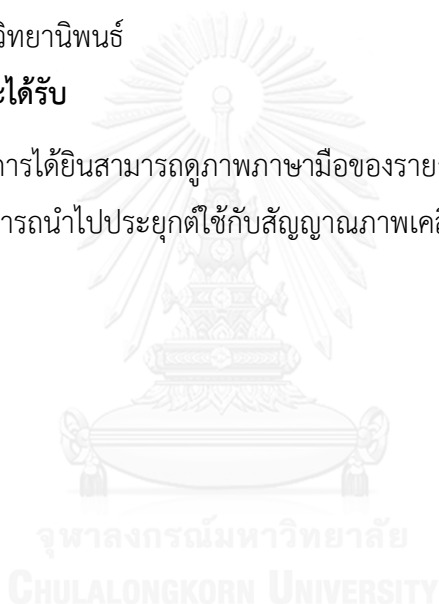
- 1.4) สามารถเลือกตำแหน่งของภาพที่ต้องการขยายได้
- 1.5) รองรับสัญญาณภาพความละเอียดสูงเต็มรูปแบบ (1080p: 1920×1080 pixels)
- 1.6) รองรับระบบสัญญาณภาพที่รับและส่งผ่านสาย HDMI

1.4 วิธีดำเนินงานวิจัย

1. ศึกษารูปแบบสัญญาณภาพเคลื่อนไหวแบบดิจิทัล
2. ศึกษาวิธีการขยายภาพในแบบต่างๆ
3. ออกแบบเครื่องขยายภาพให้ขยายภาพได้โดยกำหนดตำแหน่งของภาพที่พิกัดหนึ่งๆ
4. ออกแบบเครื่องขยายภาพให้ขยายภาพได้โดยสามารถปรับตำแหน่งและขนาดของภาพได้
5. ทดสอบและพัฒนาการทำงานของวงจร
6. สรุปผลและเขียนวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้ผู้พิการทางการได้ยินสามารถดูภาพภาษามือของรายการโทรทัศน์ได้อย่างชัดเจนขึ้น
2. ผู้ใช้งานทั่วไปสามารถนำไปประยุกต์ใช้กับสัญญาณภาพเคลื่อนไหวอื่นๆที่ต้องการขยายภาพบางส่วนได้



บทที่ 2

มาตรฐานและทฤษฎีที่เกี่ยวข้อง

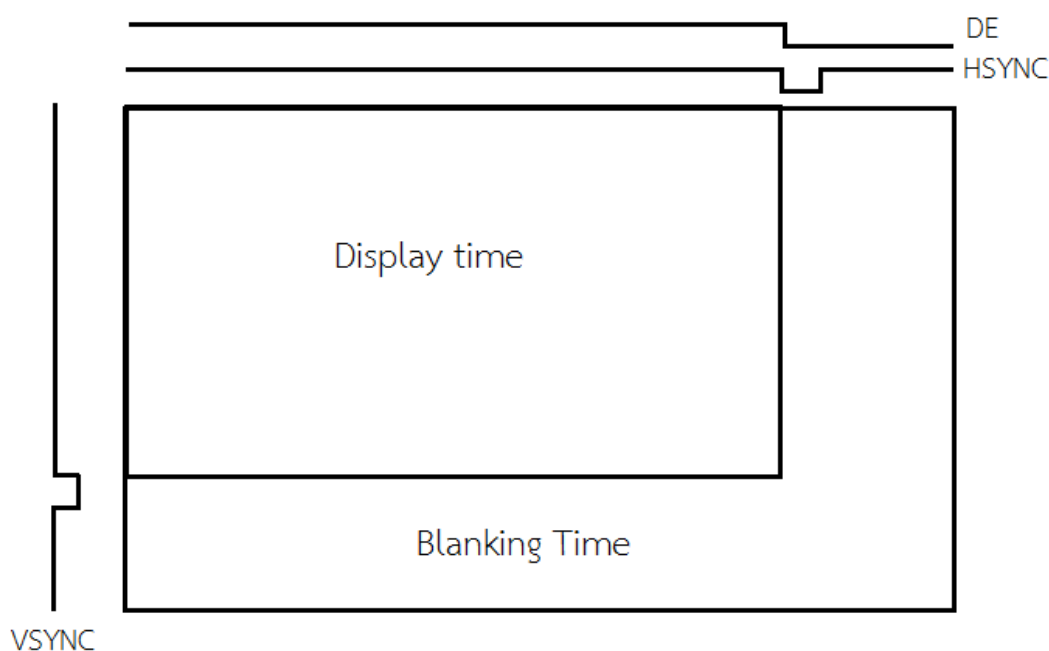
2.1 โครงสร้างสัญญาณภาพเคลื่อนไหวแบบดิจิทัล

ภาพเคลื่อนไหวสามารถมองเป็นภาพนิ่งหลายๆภาพมาต่อกันซึ่งแต่ละภาพนิ่งสามารถมองได้ว่าเป็นการเก็บจุดภาพในแบบ 2 มิติซึ่งจะแสดงภาพตามความถี่ค่าหนึ่ง โดยการส่งค่าจุดภาพแต่ละจุดจะถูกส่งมาจากมุมบนซ้ายของภาพไปทางขวามือที่เส้นจนถึงมุมล่างขวาของภาพ โดยการแสดงภาพในแต่ละภาพจำเป็นต้องมีสัญญาณเสริมที่คอยระบุถึงสถานะของข้อมูลที่กำลังส่งมาเช่น ข้อมูลที่กำลังส่งมาไม่มีข้อมูลภาพหรือไม่หรือควรจะเริ่มต้นการแสดงผลภาพในแถวถัดไปเป็นต้น [1, 2] ดังรูปที่ 2.1 โดย

DE (Data enable) เป็นสัญญาณที่บอกถึงว่าข้อมูลที่กำลังส่งมาเป็นข้อมูลจุดภาพเมื่อ DE มีค่าเป็น '1' ซึ่งเรียกว่า display time และช่วงที่ไม่มีข้อมูลจุดภาพ DE มีค่าเป็น '0' หรือเรียกว่า blanking time

HSYNC (Horizontal synchronization) เป็นสัญญาณที่บอกถึงให้เริ่มแสดงจุดภาพในแถวถัดไปโดย HSYNC อาจจะเป็นได้ทั้ง active high และ active low โดยปกติเป็น active low

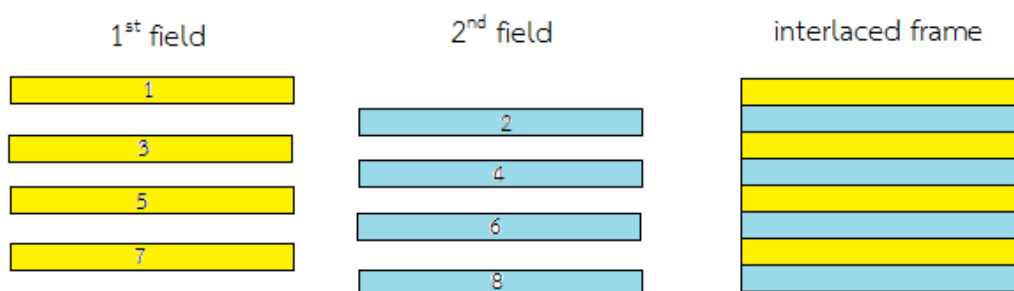
VSYNC (Vertical synchronization) เป็นสัญญาณที่บอกถึงการให้เริ่มแสดงจุดภาพในเฟรมถัดไปโดย VSYNC อาจจะเป็นได้ทั้ง active high และ active low โดยปกติเป็น active low



รูปที่ 2.1 สัญญาณเสริมที่จำเป็นในการแสดงผลภาพในแต่ละเฟรม

ในการบอกความละเอียดภาพของภาพเคลื่อนไหวจะบอกความละเอียดเป็นตัวเลขของความละเอียดในแนวตั้งแล้วตามด้วยอักษร p หรือ i ซึ่งจะเป็นการบอกถึงการแสดงเส้นภาพในแบบ progressive หรือ interlaced เช่น 720p หมายถึงขนาดภาพที่มีอัตราส่วน 16:9 แสดงภาพขนาดจริงมีขนาด 1280 x 720 จุดภาพและแสดงเส้นภาพแบบ progressive

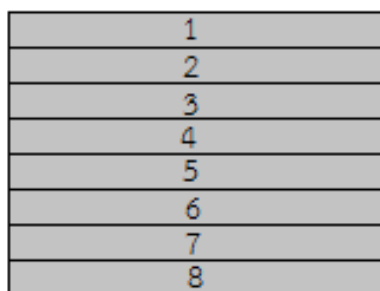
การแสดงเส้นภาพในแบบ Interlaced ถูกใช้ในมาตรฐานของสัญญาณภาพเคลื่อนไหวแอนะล็อก โดยจะส่งภาพที่ต้องการแสดงทีละครึ่งภาพที่เรียกว่า field โดยเรียก field ที่ส่งเส้นภาพเลขคี่ว่า 1st field และเส้นภาพเลขคู่ว่า 2nd field ดังรูปที่ 2.2



รูปที่ 2.2 การแสดงเส้นภาพในแบบ interlaced

การแสดงเส้นภาพในแบบ progressive ถูกใช้ในมาตรฐานของสัญญาณภาพเคลื่อนไหวดิจิทัลส่วนใหญ่โดยจะส่งมาทีละทั้งภาพดังรูปที่ 2.3 ซึ่งทำให้ได้ภาพที่ดูเรียบเนียนกว่าแบบ interlaced

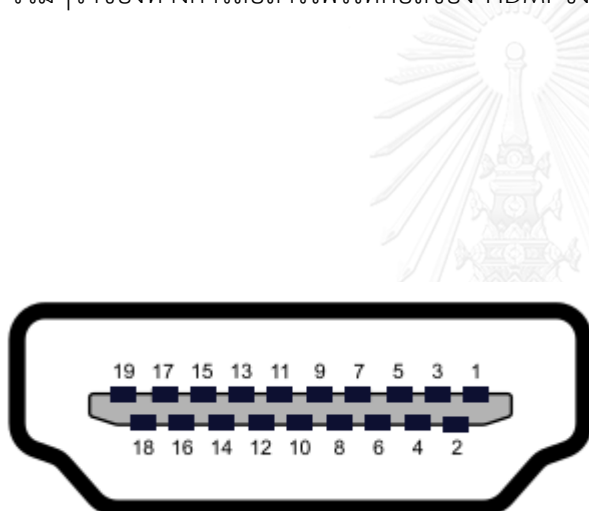
รูปที่ 2.3 การแสดงเส้นภาพในแบบ progressive



รูปที่ 2.3 การแสดงเส้นภาพในแบบ progressive

2.2 ระบบส่งสัญญาณมัลติมีเดียความละเอียดสูง (High-Definition Multimedia Interface : HDMI)

HDMI [3, 4] เป็นมาตรฐานในการส่งภาพและเสียงด้วยความเร็วสูงภายในสายเส้นเดียวในระบบดิจิทัล ซึ่งปัจจุบันได้ถูกนำมาใช้อย่างแพร่หลายในระบบภาพและเสียง ไม่ว่าจะเป็นโทรทัศน์ดิจิทัลหรือจอคอมพิวเตอร์ โดยมาตรฐาน HDMI ได้ถูกนำมาเริ่มใช้ในปี พ.ศ. 2545 เป็น HDMI Version 1.0 ที่รองรับความละเอียดภาพสูงสุดที่ 1080p หรือ Full High-Definition (FHD) ที่ความถี่การแสดงผลภาพ 60Hz หลังจากนั้น HDMI ก็ได้ถูกพัฒนาจนปัจจุบันคือ HDMI Version 2.0 ที่รองรับความละเอียดภาพสูงสุดที่ 4k (4096x2160 จุดภาพ) หรือ Ultra High-Definition (UHD) ภาพตัดขวางของหัว HDMI ถูกแสดงในรูปแบบที่ 2.4 หน้าทีในแต่ขาของ HDMI จะทำหน้าที่ในแต่ละแบบโดยเรียกรวมๆว่าช่องทางการสื่อสารโปรโทคอลของ HDMI ซึ่งจะอธิบายในหัวข้อถัดไป



S = Shield

C = Clock

ขาที่	การทำงาน	ขาที่	การทำงาน
1	TMDS 2+	11	TMDS C S
2	TMDS 2 S	12	TMDS C-
3	TMDS 2-	13	CEC
4	TMDS 1+	14	HEC & ARC
5	TMDS 1 S	15	SCL
6	TMDS 1-	16	SDA
7	TMDS 0+	17	Ground
8	TMDS 0 S	18	+5V
9	TMDS 0-	19	Hot plug Detect
10	TMDS C+		

รูปที่ 2.4 ภาพตัดขวางของส่วนหัวของ HDMI

2.2.1 ช่องทางการสื่อสารโปรโตคอลของ HDMI

เนื่องจาก HDMI มีช่องทางการสื่อสารโปรโตคอล (Communication channel protocol) หลายช่องทางจึงทำให้ออกจากการส่งภาพและเสียงแล้ว HDMI ยังมีคุณลักษณะเด่นอื่นที่ทำให้ผู้ใช้งานเกิดความสะดวกรสบายในการใช้งานเช่น ผู้ใช้งานสามารถต่อจอแสดงผลภาพเข้ากับแผ่นวงจรภาพ (Video card) ด้วยสาย HDMI แล้วสามารถใช้งานได้เลยโดยไม่ต้องปรับตั้งค่าใดๆ (plug and play) เป็นต้น

ใน HDMI Version 1.4 มีช่องทางการสื่อสารโปรโตคอล (Communication channel protocol) 5 ช่องทางที่ได้แก่

1.) DDC (Display Data Channel) ที่ทำให้จอแสดงผล (HDMI sink) และแผ่นวงจรภาพ (HDMI source) สามารถสื่อสารกันโดยจอแสดงผลจะอ่านค่าข้อมูล EDID (จะอธิบายในหัวข้อถัดไป) จากนั้นจะทำการปรับค่าความละเอียดได้อย่างอัตโนมัติเพื่อให้เหมาะสมกันระหว่างจอแสดงผลและแผ่นวงจรภาพ

2.) CEC (Consumer Electronics Control) เป็นช่องทางการสื่อสารที่มีผลทำให้ผู้ใช้งานสามารถควบคุมอุปกรณ์หลายๆอุปกรณ์ที่ต่อการใช้สาย HDMI โดยการใช้เพียงแคร์รีโมตคอนโทรลหนึ่งตัว เช่น สามารถควบคุมเครื่องเล่น DVD ได้ด้วยรีโมตคอนโทรลของโทรทัศน์ เป็นต้น

3.) ARC (Audio Return Channel) เป็นช่องทางการสื่อสารที่มีผลทำให้ผู้ใช้สามารถลดจำนวนสายต่อระหว่างอุปกรณ์ลงโดยใช้เพียงสาย HDMI เท่านั้นในการต่อระหว่างโทรทัศน์และชุดระบบเสียง กล่าวคือ ARC ทำให้โทรทัศน์รับสัญญาณภาพจากชุดระบบเสียงและขณะเดียวกันก็ส่งสัญญาณเสียงไปที่ชุดระบบเสียงโดยไม่ต้องใช้สาย S/PDIF เพิ่ม

4.) HEC (HDMI Ethernet Channel) เป็นช่องทางการสื่อสารที่ทำให้อุปกรณ์สามารถสื่อสารสองทางได้ด้วยความเร็วสูงสุด 100 Mb/sec ซึ่งทำให้สามารถนำไปใช้งานที่ใช้ไอพีเป็นหลัก (IP-based application) ได้

5.) TMDS (Transition-minimized differential signaling) เป็นเทคโนโลยีการส่งข้อมูลอนุกรมที่มีความเร็วสูง ที่ใช้ขั้นตอนวิธีเข้ารหัส (Coding algorithm) ที่ช่วยลดการเกิดสัญญาณรบกวน (Electromagnetic interference: EMI) ระหว่างสายได้ ซึ่งในช่อง TMDS จะมีทั้งข้อมูลภาพ ข้อมูลเสียง และข้อมูลเสริมอื่นๆ โดย TMDS ใน HDMI นั้นสามารถแบ่งได้เป็น 3 ช่องทางข้อมูล (TMDS data channels) คือ TMDS data 0, TMDS data 1 และ TMDS data 2 และอีก 1 สัญญาณนาฬิกา (TMDS clock channels) คือ TMDS clock

โหมดการทำงานของ TMDS data channels สามารถแบ่งได้เป็น 3 โหมดคือ Video Data period, Data Island period และ Control period ในช่วงโหมดการทำงาน Video Data period สัญญาณ

ภาพจะถูกส่ง ในช่วงโหมดการทำงาน Data Island period ข้อมูลเสียงและข้อมูลเสริมอื่นๆเช่น ข้อมูลเฟรม (InfoFrames), ข้อมูลที่อธิบายลักษณะของข้อมูลเสียง เป็นต้น จะถูกส่ง และในโหมดการทำงาน Control period จะทำงานก็ต่อเมื่อ ไม่ได้ส่งข้อมูลภาพ ข้อมูลเสียงและข้อมูลเสริมอื่นๆ โดยจำเป็นต้องให้ทำงานในโหมด Control period ขึ้นระหว่าง Data Island period และ Video Data period เสมอ ตัวอย่างการทำงานในโหมดของภาพหนึ่งเฟรมด้วยความละเอียด 720×480 จุดภาพได้ แสดงไว้ในรูปที่ 2.5

มาตรฐาน HDMI ไม่ได้ถูกออกแบบให้มีส่วนที่สามารถเก็บคำบรรยายภาพ (closed caption) ดังนั้นคำบรรยายภาพจะต้องถูกถอดรหัส (decode) ก่อน แล้วนำคำบรรยายภาพแทรกเข้าไปในภาพเดิมจากนั้นค่อยส่งผ่านสาย HDMI จึงจะสามารถดูคำบรรยายภาพในโทรทัศน์ดิจิทัลได้

2.2.2 Extended Display Identification Data (EDID)

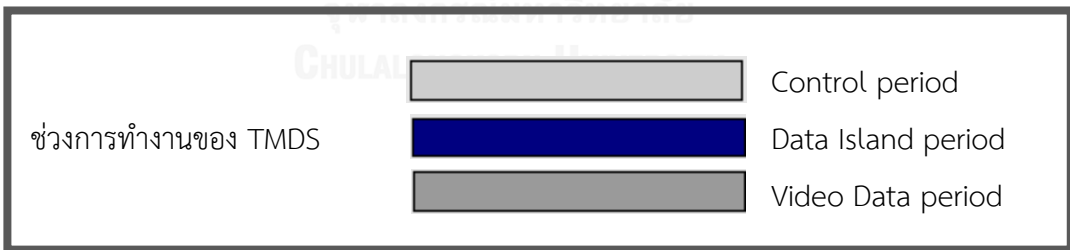
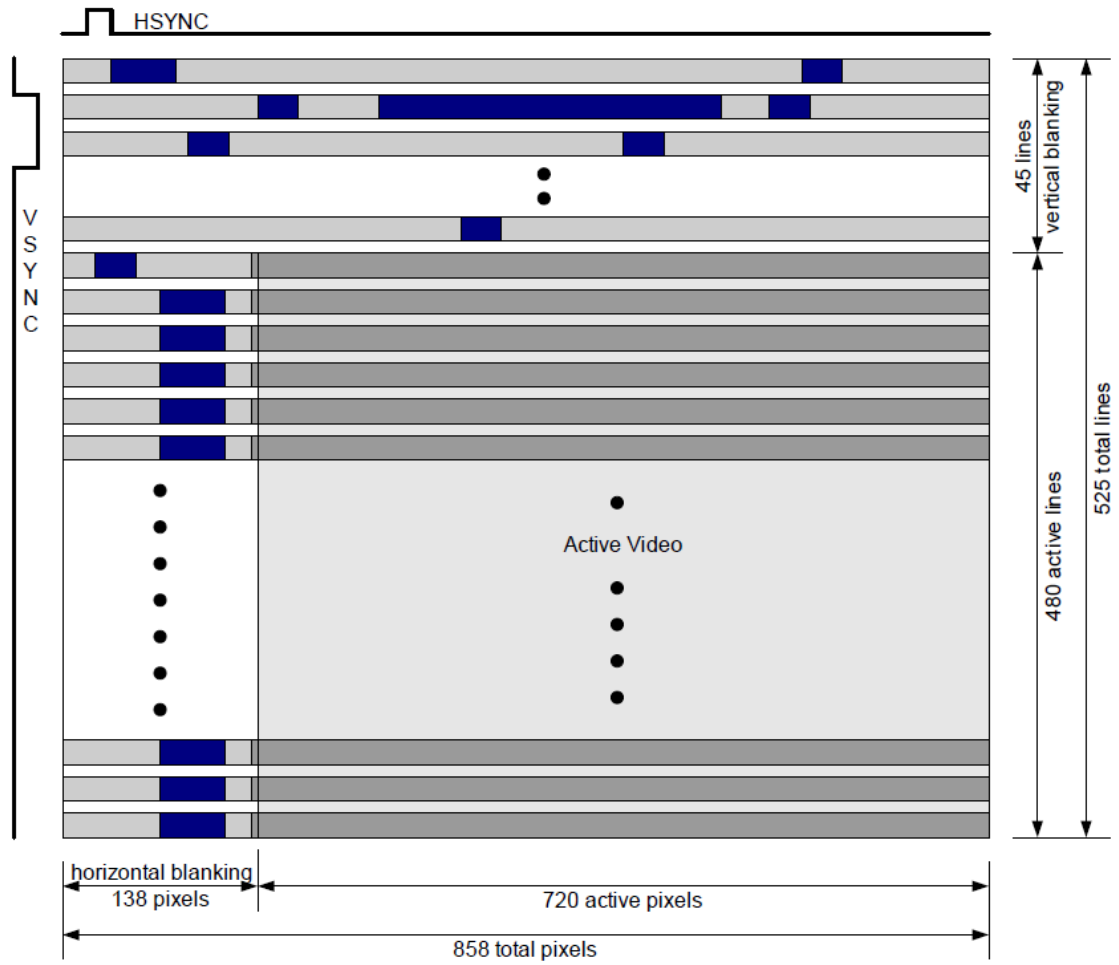
EDID เป็นโครงสร้างข้อมูลที่ถูกเก็บไว้โดยจอแสดงผลภาพ (HDMI sink) เพื่ออธิบายให้กับแหล่งกำเนิดสัญญาณภาพเคลื่อนไหว (HDMI source) เช่น การ์ดแสดงผล TV-set top box เป็นต้น ได้ทราบถึงความละเอียดภาพที่จอแสดงผลภาพนั้นๆสามารถรองรับได้รวมถึงยังมีข้อมูลอื่นๆที่ถูกเก็บไว้ใน EDID อีกด้วยเช่น ชื่อบริษัทและวันเดือนปีที่ผลิตของจอแสดงผลภาพ เป็นต้น

โดยปกติแล้วช่องทางในการส่งข้อมูล EDID จากจอแสดงผลภาพไปยังแหล่งกำเนิดสัญญาณภาพเคลื่อนไหวจะส่งผ่าน I2C บัส และการเก็บข้อมูล EDID จะถูกเก็บไว้ในหน่วยความจำที่ไม่สูญหายเมื่อไม่มีไฟฟ้า (non-volatile memory) เช่น EEPROM เป็นต้น

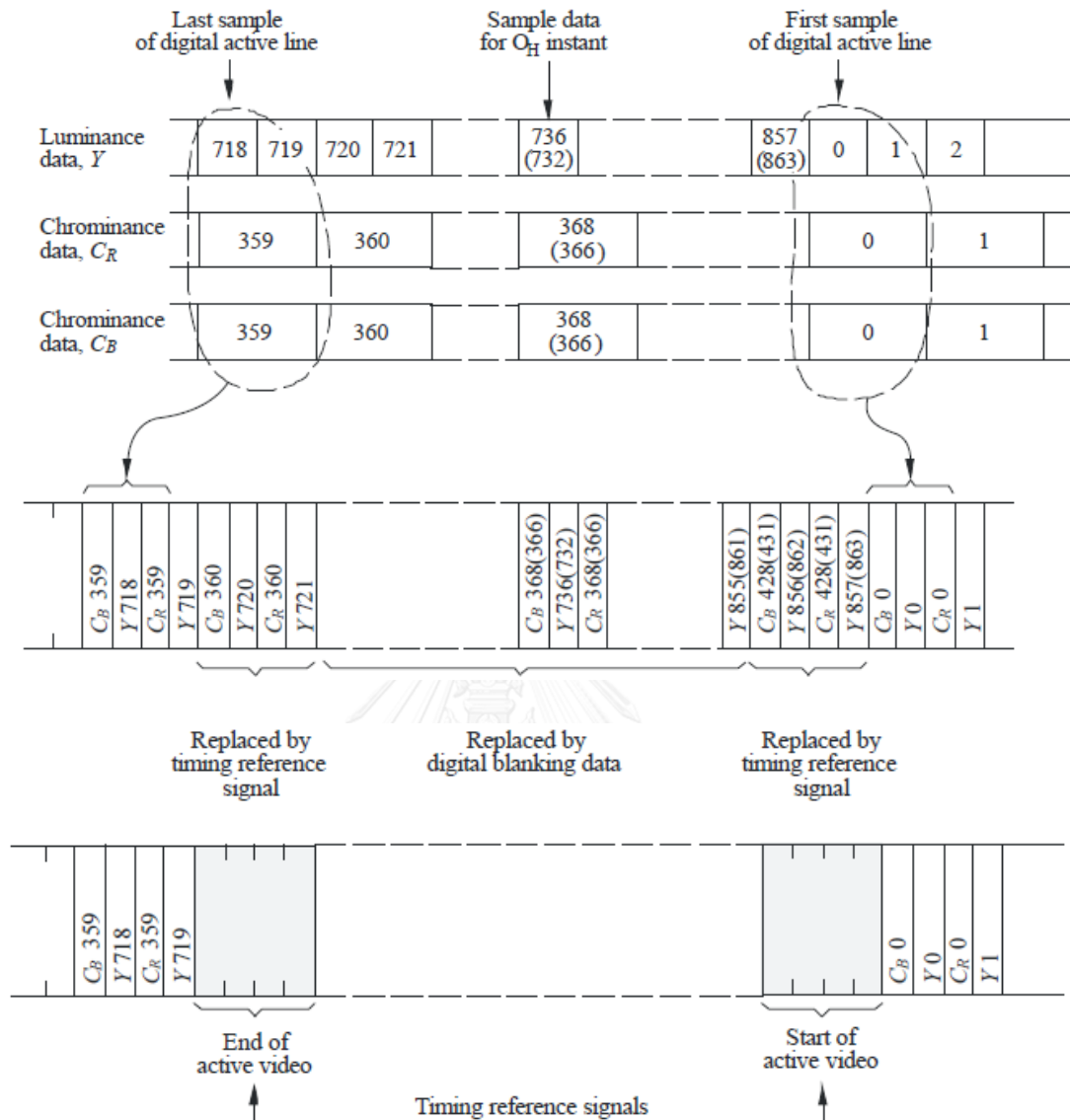
2.3 มาตรฐาน ITU-R BT.656

มาตรฐาน ITU-R BT.656 [5] เป็นมาตรฐานการแสดงผลสัญญาณภาพวีดิทัศน์ในระบบดิจิทัล ที่ได้รับความนิยม ซึ่งกล่าวถึงรูปแบบการส่งสัญญาณภาพให้ผู้รับและผู้ส่งสามารถเข้าใจได้ตรงกัน ส่วนใหญ่ในอดีตมาตรฐาน ITU-R BT.656 จะถูกใช้คู่กับ ITU-R BT.601 [6] ที่กล่าวถึงการแปลงสัญญาณวีดิทัศน์แบบแอนะล็อกมาเป็นแบบดิจิทัล แต่ในปัจจุบันนี้ที่โทรทัศน์ส่วนใหญ่เป็นแบบดิจิทัลมาตรฐาน ITU-R BT.656 นั้นก็ยังถูกนำมาใช้เป็นรูปแบบมาตรฐานในการแสดงผลสัญญาณภาพ ตัวอย่างเช่น ADV7611 ซึ่งเป็นชิปตัวรับสัญญาณ HDMI ต่อประสาน (HDMI interface receiver) สามารถตั้งค่าให้สัญญาณขาออกเป็นไปตามมาตรฐาน ITU-R BT.656 เป็นต้น

ข้อมูลจุดภาพในมาตรฐาน ITU-R BT.656 ประกอบไปด้วย ความเข้มของแสง (Luminance : Y) และความเข้มของสีอีก 2 องค์ประกอบ (Chrominance : C_B , C_R) โดยการแสดงผลข้อมูลจุดภาพ จะถูกแสดงที่ละองค์ประกอบสลับกันไปด้วยอัตราส่วน 4:2:2 ซึ่งเป็นอัตราส่วนของ Y: C_B : C_R โดยการเรียงข้อมูลภาพถูกแสดงในรูปที่ 2.6



รูปที่ 2.5 โหมดการทำงานต่างๆของ TDMS ในหนึ่งเฟรมความละเอียด 720x480 จุดภาพ



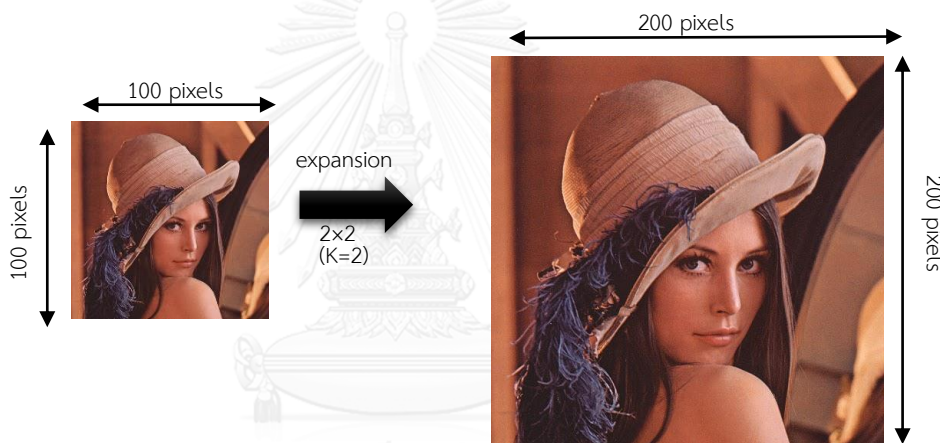
รูปที่ 2.6 รูปแบบการส่งสัญญาณภาพตามมาตรฐาน ITU.R BT656

ในแต่ละเส้นสัญญาณภาพจะประกอบไปด้วยสัญญาณจริงและสัญญาณไร้ภาพ ในช่วงสัญญาณจริงนั้นข้อมูลจะถูกส่งออกมาทีละองค์ประกอบเป็นลำดับดังนี้ C_B, Y, C_R, Y, C_B, Y, C_R, Y ดังจะเห็นในรูปที่ 7 โดย C_B 359 และ C_R 359 จะเป็นองค์ประกอบความเข้มสีที่คู่กับ Y 718 ซึ่งเป็นองค์ประกอบความเข้มแสงนั่นเอง ส่วนในช่วงสัญญาณไร้ภาพจะมีช่วงที่เรียกว่าสัญญาณเวลา (Timing reference signal) อยู่ 2 ช่วงคือ ช่วงจบสัญญาณภาพ (End of active video: EAV) และช่วงเริ่มต้นสัญญาณภาพ (Start of active video: SAV) ซึ่งทั้งสองช่วงจะสามารถระบุได้ถึง เส้นภาพอยู่ในฟิลด์คู่หรือคี่ อยู่ในช่วงซิงค์แนวตั้งหรือไม่ และเป็นช่วงเริ่มต้นมีสัญญาณภาพหรือเป็นช่วงจบสัญญาณภาพหรือไม่

โดยปกติมาตรฐาน ITU.R BT656 จะถูกแบบให้รองรับกับสัญญาณภาพความละเอียด 480i และ 576i ซึ่งเป็นความละเอียดของสัญญาณแอนะล็อก แต่ใน ADV7611 ที่รองรับสัญญาณภาพความละเอียด 720p และ 1080p แต่ยังคงส่งรูปแบบสัญญาณภาพขาออกได้ตามมาตรฐาน ITU.R BT656 นั้นจะหมายถึงว่ามีการส่งสัญญาณ EAV และ SAV ตามมาตรฐาน ITU.R BT656 แต่อาจจะไม่ได้ส่งสัญญาณภาพที่ละองค์ประกอบเหมือนมาตรฐาน ITU.R BT656

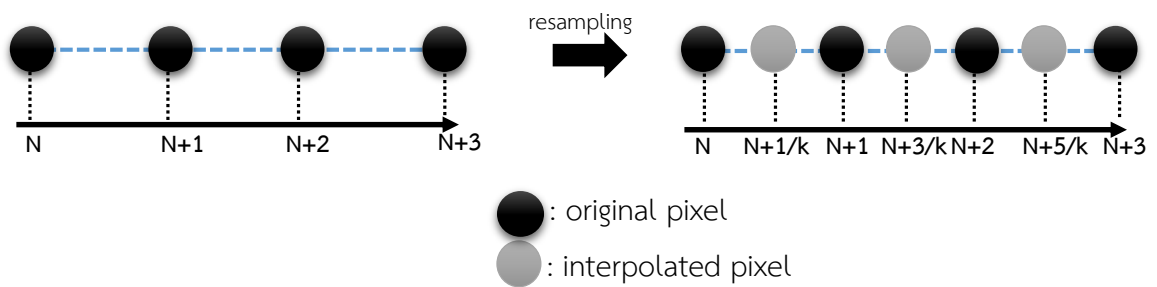
2.4 วิธีการขยายภาพ

ในการขยายภาพขนาด $k \times k$ เท่าจะหมายความว่าขยายขนาดภาพจากเดิม $M \times N$ จุดภาพเป็นขนาด $kM \times kN$ ซึ่งในเครื่องขยายวิดีโอที่ศัณษานาเหมือนนี้จะใช้ค่า $k = 2$ นั้นหมายความว่าเราจะได้ขนาดภาพใหม่เป็นค่า $2M \times 2N$ จุดภาพนั่นเองดังตัวอย่างรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการขยายภาพขนาด 2x2 เท่า

การขยายภาพก็คือการนำค่าแต่ละจุดภาพเดิมมาทำการซัดตัวอย่างซ้ำ (resampling) ซึ่งถ้าอัตราการซัดตัวอย่าง (sample rate) มากก็จะเป็นการขยายภาพแต่ถ้าน้อยก็จะเป็นการลดขนาดภาพ สำหรับการขยายภาพด้วย $k = 2$ นั้นจะแสดงในรูปที่ 2.8 โดยวงกลมสีดำจะแสดงถึงจุดภาพเดิมและวงกลมสีเทาคือจุดภาพที่ได้จากการซัดตัวอย่างซ้ำ หากการซัดตัวอย่างภาพตรงกับจุดภาพเดิมจุดภาพก็จะไม่เปลี่ยนแปลง แต่ถ้าการซัดตัวอย่างภาพไม่ตรงกับจุดภาพเดิมจะต้องมีการคำนวณในการหาค่าโดยใช้วิธีการประมาณค่าในช่วง (interpolation method)

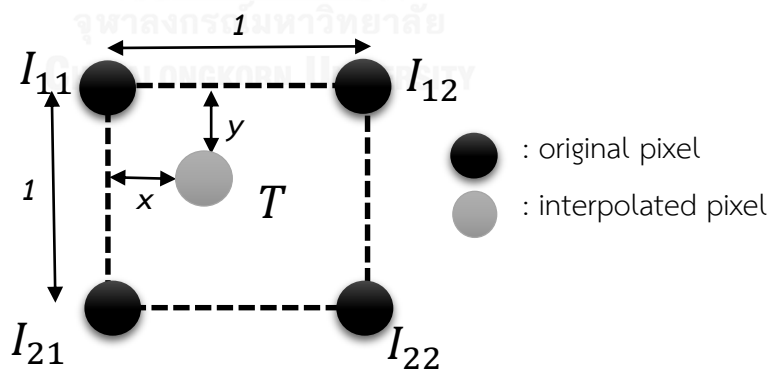


รูปที่ 2.8 การซ้ดตัวอย่างซ้ำเพื่อขยายภาพในแนวนอนขนาด $k = 2$ เท่า

วิธีการหาค่าโดยใช้วิธีการประมาณค่าในช่วง (interpolation method) มีอยู่หลายวิธีการ ซึ่งวิธีการขยายภาพที่น่าสนใจและได้รับความนิยมที่นำมาเสนอมี่ดังนี้คือ วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood) [7] วิธีประมาณค่าแบบไบลิเนียร์ (Bilinear interpolation) [8] วิธีประมาณค่าแบบไบคิวบิก (Bicubic interpolation) [9] และวิธีการขยายภาพแบบอื่นๆ ซึ่งรายละเอียดของวิธีนั้นมีดังต่อไปนี้

2.4.1 วิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด (Nearest neighborhood)

การขยายภาพแบบประมาณค่าจากค่าที่อยู่ใกล้ที่สุด จะสร้างจุดภาพใหม่โดยการเลือกใช้ค่าจุดภาพเดิมที่อยู่ใกล้ที่สุดจาก 4 จุดรอบๆที่จะประมาณค่า ดังแสดงในรูปที่ 2.9 โดยวงกลมสี่เหลี่ยมคือจุดภาพที่เกิดจากการประมาณและวงกลมสี่เหลี่ยมดำคือจุดภาพเดิม



รูปที่ 2.9 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 4 จุดคือ I_{11}, I_{12}, I_{21} และ I_{22} เมื่อขยายภาพด้วยวิธีประมาณค่าจากค่าที่อยู่ใกล้ที่สุด

โดย I_{11}, I_{12}, I_{21} และ I_{22} เป็นข้อมูลจุดเดิมที่ต้องการหาค่าจุดภาพใหม่ T ซึ่งห่างจากจุดภาพ I_{11} เป็นระยะ x ในแนวนอน และ y ในแนวตั้ง T จะสามารถคำนวณได้จากสมการต่อไปนี้

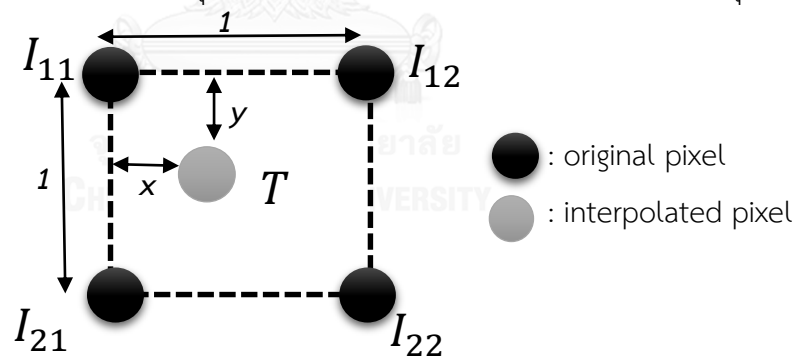
$$T = \begin{cases} I_{11}; & 0 \leq x \leq 0.5, 0 \leq y \leq 0.5 \\ I_{12}; & 0 \leq x \leq 0.5, 0 < y < 1 \\ I_{21}; & 0 < x < 1, 0 \leq y \leq 0.5 \\ I_{22}; & 0 < x < 1, 0 < y < 1 \end{cases}$$

สมการที่ (2.1)

จากสมการที่ 2.1 จะพบว่า จุดภาพที่ประมาณได้จะมีค่าเท่ากับจุดภาพเดิมที่อยู่ใกล้กับจุดภาพที่ต้องการประมาณมากที่สุด ดังนั้นการขยายภาพด้วยวิธีนี้จะใช้แค่ระยะทางเท่านั้นไม่ต้องใช้การคำนวณใดๆ ดังนั้นการขยายภาพจะสามารถทำได้อย่างรวดเร็วมาก แต่ผลเสียก็คือคุณภาพของภาพจะเกิดปรากฏการณ์แตกเป็นเหลี่ยมๆ เพราะที่เกิดจากการแสดงจุดภาพซ้ำเท่านั้นไม่มีการลดหรือเพิ่มขนาดภาพที่ประมาณแต่อย่างใด

2.4.2 วิธีการประมาณค่าแบบไบลิเนียร์ (Bilinear interpolation method)

การขยายภาพแบบประมาณแบบไบลิเนียร์สร้างจุดภาพใหม่โดยการเลือกใช้ค่าจุดภาพเดิมที่อยู่ใกล้ที่สุดจาก 4 จุดรอบๆที่จะประมาณค่าเหมือนกับวิธีการประมาณค่าจากค่าที่อยู่ใกล้ที่สุด ดังแสดงในรูปที่ 2.10 โดยวงกลมสี่เหลี่ยมคือจุดภาพที่เกิดจากการประมาณและวงกลมสี่เหลี่ยมคือจุดภาพเดิม



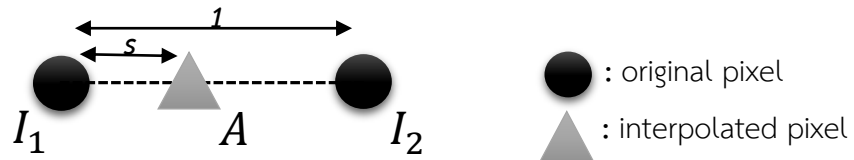
รูปที่ 2.10 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 4 จุดคือ I_{11}, I_{12}, I_{21} และ I_{22} เมื่อขยายภาพด้วยวิธีประมาณค่าแบบไบลิเนียร์

โดยวิธีการประมาณแบบไบลิเนียร์จะนำวิธีการประมาณแบบเชิงเส้นระหว่างจุดทั้ง 4 จุดคือ I_{11}, I_{12}, I_{21} และ I_{22} ทั้งนี้การประมาณค่าแบบเชิงเส้นระหว่างจุดทั้ง 4 จุดจะใช้ประมาณค่าในช่วงแบบ 2 มิติ (2-D interpolation) ซึ่งจะแปลงได้เป็นการประมาณค่าในช่วงแบบ 1 มิติ (1-D interpolation) 2 แนวแกนคือทำในแนวนอนและแนวตั้งซึ่งสมการของการประมาณค่าแบบไบลิเนียร์แบบ 1 มิติคือ

$$A = (1 - s) \times I_1 + s \times I_2$$

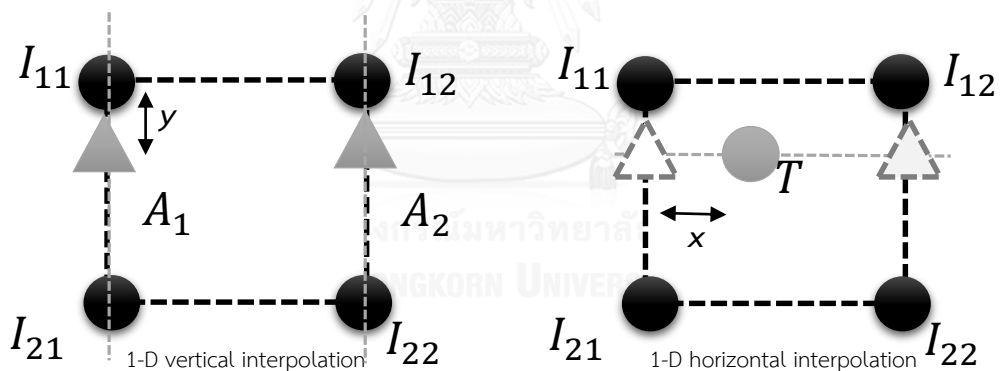
สมการที่ (2.2)

ซึ่ง A คือจุดภาพที่ต้องการประมาณค่า I_1 และ I_2 เป็นจุดภาพเดิมและ s คือระยะห่างระหว่าง A และ I_1 ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 การประมาณค่าแบบไบลิเนียร์ 1 มิติ

ในการหาจุดภาพ T ในรูปที่ 2.10 จะต้องใช้วิธีการประมาณค่าแบบไบลิเนียร์ 2 มิติซึ่งอาจทำได้โดยการทำการประมาณค่าแบบไบลิเนียร์แบบ 1 มิติในแนวตั้งก่อนซึ่งจะได้ค่า A_1 และ A_2 จากนั้นทำการประมาณค่าแบบไบลิเนียร์แบบ 1 มิติในแนวนอนกับค่าประมาณที่ได้ ดังรูปที่ 2.12

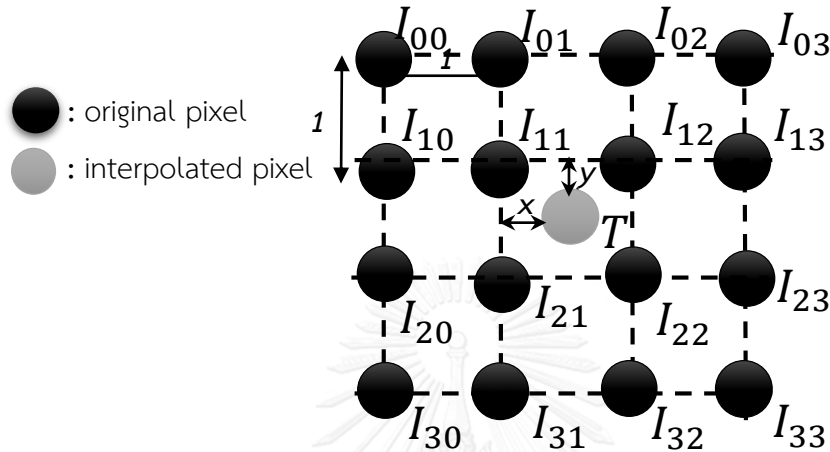


รูปที่ 2.12 แสดงการประมาณค่าแบบไบลิเนียร์ 2 มิติด้วยการทำการประมาณค่าแบบไบลิเนียร์ 1 มิติ 2 แนวแกน

จากสมการของการประมาณค่าแบบไบลิเนียร์จะเห็นว่าจะมีการนำค่า 4 จุดรอบๆของจุดที่ต้องการประมาณค่าแบบเชิงเส้น ซึ่งจะทำให้ให้ภาพที่ขยายเรียบขึ้นแต่ก็มีข้อเสียคือทำให้ภาพเบลอและขาดความคมชัด

2.4.3 วิธีการประมาณค่าแบบไบคิวบิก (Bicubic interpolation method)

การขยายภาพโดยการประมาณค่าแบบไบคิวบิกนี้จะคำนวณจุดภาพที่ต้องการประมาณจากจุดภาพรอบๆทั้งหมด 16 จุดภาพดังรูปที่ 2.13 ซึ่งจะแตกต่างจากวิธีการประมาณค่าทั้ง 2 วิธีก่อนหน้านี้ที่ใช้เพียง 4 จุดภาพ



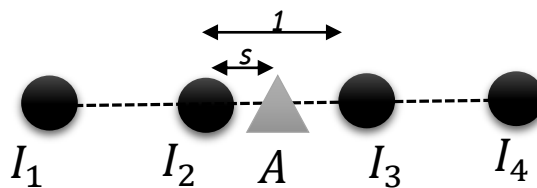
รูปที่ 2.13 การหาค่าจุดภาพที่ต้องการ T ซึ่งคำนวณจากจุดภาพรอบๆ 16 จุดเมื่อขยายภาพด้วยวิธีการประมาณค่าแบบไบคิวบิก

วิธีการประมาณค่าแบบไบคิวบิกสามารถแบ่งเป็นการประมาณค่าในช่วง 1 มิติ 2 แนวแกนได้ เหมือนการประมาณค่าไบลิเนียร์โดยสมการของการประมาณค่าแบบไบคิวบิกคือ

$$\begin{aligned}
 A &= c_0 \times I_0 + c_1 \times I_1 + c_2 \times I_2 + c_3 \times I_3 \\
 c_0 &= -[(1 - s) \times (1 - s) \times s] \\
 c_1 &= (1 - s) + [(1 - s) \times (1 - s) \times s] \\
 c_2 &= s + [(1 - s) \times s \times s] \\
 c_3 &= -[(1 - s) \times s \times s]
 \end{aligned}$$

สมการที่ (2.3)

โดย A คือจุดภาพที่ต้องการประมาณค่า I_1, I_2, I_3 และ I_4 เป็นจุดภาพเดิมและ s คือระยะห่างระหว่าง A และ I_1 ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 การประมาณค่าแบบไบคิวบิก 1 มิติ

สำหรับการประมาณค่าแบบไปควบิก 2 มิติอาจทำได้โดยการใช้การประมาณค่าแบบไปควบิก 1 มิติในแนวตั้งก่อน จากนั้นจึงทำการประมาณค่าแบบไปควบิก 1 มิติในแนวนอนเหมือนกับวิธีการประมาณค่าแบบโบลีเนียร์ที่แสดงในรูปที่ 9 ที่การประมาณค่าแบบโบลีเนียร์ 1 มิติแต่ละครั้งจะใช้จุดภาพเดิม 2 จุดภาพ แต่สำหรับการประมาณค่าแบบไปควบิกการทำประมาณค่าไปควบิก 1 มิติแต่ละครั้งจะใช้จุดภาพเดิม 4 จุดภาพ

ภาพที่ได้จากการขยายภาพแบบไปควบิกจะได้ภาพที่มีคุณภาพกว่าทั้ง 2 วิธีก่อนหน้านี้ เนื่องจากการใช้จำนวนข้อมูลภาพเดิมมากกว่าและใช้สมการที่มีความซับซ้อนมากกว่า โดยภาพที่ได้จะคมชัดกว่าโบลีเนียร์ และภาพไม่แตกเป็นสี่เหลี่ยมเหมือนกับวิธีการประมาณค่าที่ใกล้ที่สุด

2.4.4 วิธีการขยายภาพแบบอื่นๆ

เพื่อการขยายภาพได้แบบตามเวลาจริง (real time) และให้ผลลัพธ์เป็นภาพขยายที่มีคุณภาพที่อยู่ในระดับยอมรับได้ วิธีการขยายภาพอื่นๆได้ถูกคิดค้นในช่วงทศวรรษที่ผ่านมา โดยวิธีการขยายที่น่าสนใจก็ได้แก่ วิธีการปรับขนาดภาพแบบวินสเกล (Winscale) เป็นวิธีการปรับเปลี่ยนขนาดภาพโดยพิจารณาเป็นพื้นที่ๆไปแทนที่จะคิดเป็นจุดภาพ [10] วิธีการขยายภาพโดยวิธีการคอนโวลูชันเชิงเส้นที่ถูกขยาย (extended linear convolution) ที่เป็นวิธีการที่พยายามเลียนแบบวิธีการประมาณค่าแบบไปควบิกแต่พยายามทำให้สมการที่คำนวณจุดภาพใหม่อยู่ในรูปสมการกำลังหนึ่ง [11] วิธีการขยายภาพที่มีการนำการปรับปรุงขอบภาพมาใช้ [12, 13]

วิธีการขยายภาพแบบอื่นๆที่ได้ทำการทบทวนวรรณกรรมดังกล่าวได้มีข้อเสียแบบเดียวกันคือคุณภาพของภาพที่ขยายแล้วจะมีคุณภาพน้อยกว่าหรือเท่ากับคุณภาพของภาพที่ขยายแล้วของวิธีการขยายภาพแบบไปควบิก ซึ่งวิธีการขยายภาพแบบไปควบิกสามารถลดความซับซ้อนของการคำนวณได้ ซึ่งจะอธิบายในบทย่อถัดไป โดยสรุปได้ว่าวิธีการขยายภาพแบบไปควบิกที่ลดความซับซ้อนของการคำนวณเป็นที่วิธีการขยายภาพที่น่าใช้ที่สุด เพราะได้คุณภาพภาพที่ขยายแล้วที่สูงและมีการใช้การคำนวณที่ไม่ซับซ้อน

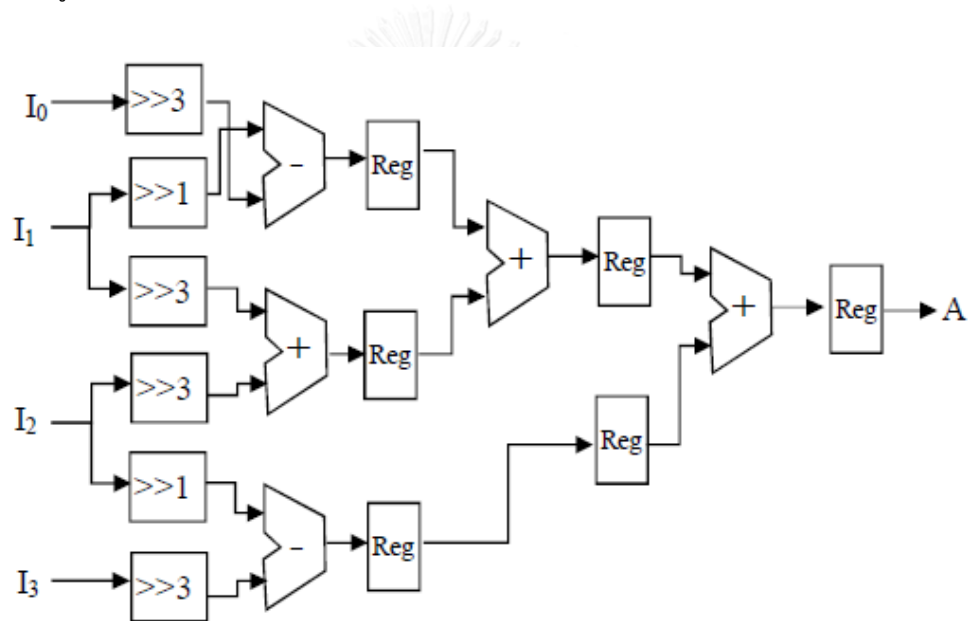
2.4.5 วิธีการขยายภาพ 2 เท่าด้วยวิธีการประมาณค่าไปควบิก

สำหรับการขยายภาพ 2 เท่าด้วยวิธีการประมาณค่าไปควบิกค่า S ในสมการการประมาณค่าไปควบิกแบบหนึ่งมิติหรือสมการที่ 2.3 จะมีค่าเท่ากับ 0.5 ซึ่งสามารถซึ่งจะสามารถทำให้สมการเหลือเพียงการคูณด้วย 0.5 หรือการหารด้วย 2 เท่านั้นดังนี้

$$\begin{aligned}
 A &= c_0 \times I_0 + c_1 \times I_1 + c_2 \times I_2 + c_3 \times I_3 \\
 c_0 &= -[0.5 \times 0.5 \times 0.5] \\
 c_1 &= 0.5 + [(0.5) \times (1 - s) \times s] \\
 c_2 &= 0.5 + [(1 - s) \times s \times s] \\
 c_3 &= -[0.5 \times 0.5 \times 0.5]
 \end{aligned}$$

สมการที่ (2.4)

ซึ่งในการหาคูณด้วย 0.5 หรือหารด้วย 2 ในระบบคือการเลื่อนบิตไปทางขวา (shift right arithmetic) ดังนั้นในการนำไปใช้กับฮาร์ดแวร์จะสามารถออกแบบให้เป็นระบบการทำงานแบบสายท่อสามชั้นตอน (three-state pipeline) ได้ดังรูปที่ 2.15 ซึ่งตั้งชื่อว่า 1-D Interpolation Unit หรือ 1DIU ซึ่งจะถูกนำไปใช้ในการออกแบบวงจรขยายภาพภาษาเมื่อในบทที่ 4



รูปที่ 2.15 วงจรการประมาณค่าด้วยไบควิกบิตหนึ่งมิติ (1DIU)

2.4.6 การวัดค่าคุณภาพของการขยายภาพด้วย PSNR

สำหรับการวิเคราะห์คุณภาพของการขยายนั้นจะนำค่าของ PSNR (peak signal-to-noise ratio) [12] มาเปรียบเทียบกับในแต่ละวิธีการขยายภาพ โดยค่า PSNR ที่มีค่าน้อยจะบอกถึงปริมาณสัญญาณรบกวนของภาพที่ถูกขยายแล้วมีค่ามาก PSNR มีหน่วย dB และสามารถคำนวณได้จาก

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

สมการที่ (2.5)

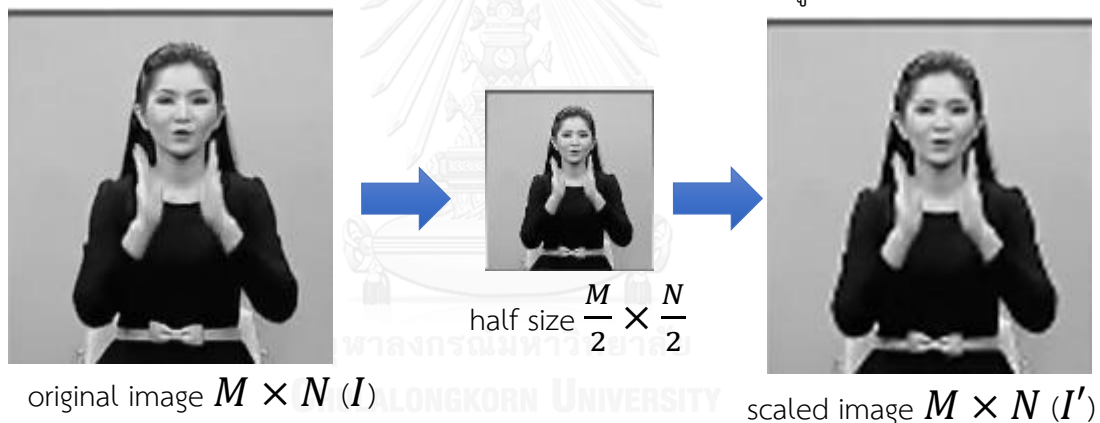
โดย MAX จะหมายถึงค่ามากสุดในแต่ละจุดภาพ สำหรับจุดภาพ 8 บิต MAX จะมีค่าเท่ากับ 255 ส่วน MSE จะหมายถึงค่าเฉลี่ยกำลังสองความคลาดเคลื่อน (mean square error) ซึ่งคำนวณได้จาก

$$MSE = \frac{\sum_i \sum_j (I(i,j) - I'(i,j))^2}{M \times N}$$

สมการที่ (2.6)

เมื่อ $I(i,j)$ คือจุดภาพต้นฉบับ $I'(i,j)$ คือจุดภาพที่ถูกขยายแล้ว ที่ตำแหน่ง (i,j) M และ N คือความกว้างและความสูงของภาพต้นฉบับ

ในการทำการหาค่า PSNR ของการขยายภาพ 2 เท่า นั้น จะทำการลดขนาดภาพลงเป็นขนาด 0.5 เท่า หลังจากนั้น จะทำการขยายภาพ 2 เท่า กลับมาที่ขนาดภาพเดิม ดังรูปที่ 2.16



รูปที่ 2.16 การทำการหาค่า PSNR สำหรับการขยายภาพ 2 เท่า

เพื่อยกตัวอย่างการประเมินหาคุณภาพของการขยายภาพ 2 เท่า โดยจะทดสอบกับวิธีการประมาณค่า 3 วิธีคือ การประมาณค่าจากค่าที่อยู่ใกล้ที่สุด การประมาณค่าแบบไบลิเนียร์ และการประมาณค่าแบบไบควบิก ภาพล่ามภาษามือ 3 ภาพ (แสดงในรูปที่ 2.17) ได้ถูกนำมาเพื่อหาค่า PSNR โดย 3 วิธีดังกล่าว โดยผลการหาค่า PSNR ได้ถูกแสดงในตารางที่ 2.1



ภาพล่ามภาษามือที่ 1

162 x 178 pixels



ภาพล่ามภาษามือที่ 2

146 x 164 pixels



ภาพล่ามภาษามือที่ 3

158 x 165 pixels

รูปที่ 2.17 ภาพล่ามภาษามือทั้ง 3 ภาพที่นำมาหาค่า PSNR

ตารางที่ 2.1 เปรียบเทียบค่า PSNR กับรูปภาพล่ามภาษามือ 3 ภาพด้วยวิธีการประมาณค่าต่างๆ

	Nearest neighbor	Bilinear	Bicubic
ภาพล่ามภาษามือ 1	22.9869	28.6168	29.0772
ภาพล่ามภาษามือ 2	28.4742	34.8368	35.9638
ภาพล่ามภาษามือ 3	28.4608	32.2183	32.4962

จากตารางที่ 2.1 แสดงให้เห็นว่าวิธีการประมาณค่าด้วยไบคิวบิกสามารถให้ค่า PSNR ที่มากที่สุดซึ่งหมายความว่าภาพที่ถูกขยายด้วยวิธีการประมาณค่าด้วยไบคิวบิกมีค่าสัญญาณรบกวนน้อยซึ่งหมายถึงภาพที่ถูกขยายมีความใกล้เคียงกับภาพดั้งเดิมที่สุดนั่นเอง

บทที่ 3

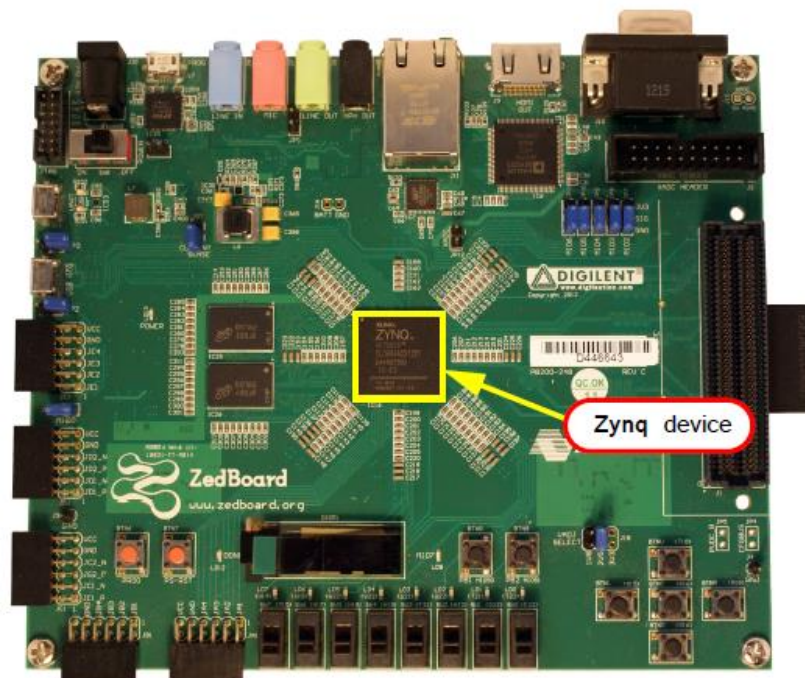
บอร์ดทดลอง

ในการออกแบบและทดสอบการทำงานของวงจรรขยายภาพภาษาเมื่อจำเป็นต้องมีบอร์ดทดลองที่มี หน่วยความจำที่เหมาะสม มีส่วนของช่องเชื่อมต่อ HDMI ขาเข้าและขาออก และที่สำคัญคือมี FPGA ที่ใช้เพื่อทำการออกแบบวงจรรขยายภาพได้แบบ real-time

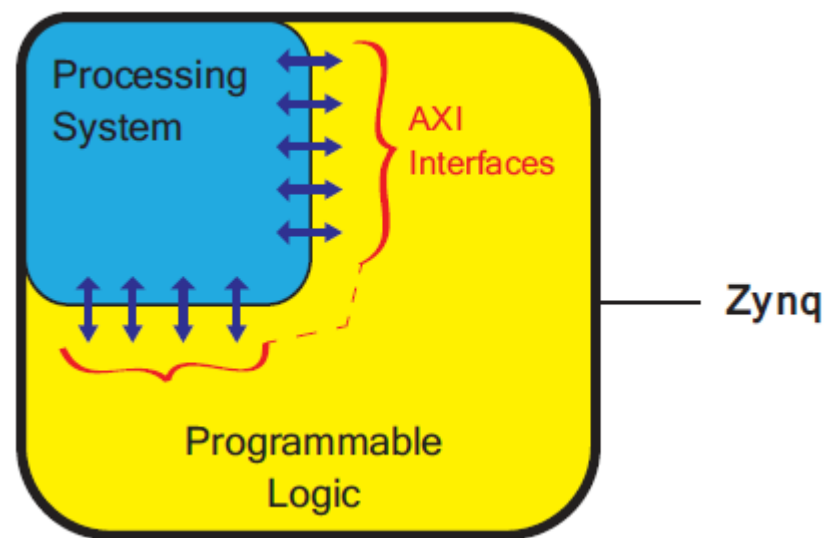
3.1 บอร์ดทดลอง Zedboard

Zedboard [14] เป็นบอร์ดทดลองที่มีชิพ XC7Z020 Zynq-7000 device แสดงในรูปที่ 3.1 ซึ่งตัวชิพนี้จะประกอบไปด้วย 2 ส่วนหลักคือ Processing System (PS) เป็นหน่วยประมวลผล dual-core ARM Cortex-A9 และ Programmable Logic (PL) ซึ่งเทียบเท่ากับ FPGA นั้นเอง Zedboard ยังเป็นบอร์ดที่ราคาไม่สูงมากและมีการใช้อย่างแพร่หลายสามารถสืบค้นตัวอย่างได้ง่าย

PS จะเป็นส่วนที่ทำงานพวกทำงานเป็นปกติซ้ๆ หรือ/และระบบปฏิบัติการ ส่วน PL จะใช้ทำการคำนวณตรรกะที่ใช้ความเร็วสูงหรือการทำงานเป็นแบบขนาน การสื่อสารกันระหว่าง PL และ PS จะใช้ผ่านทางกรณีวิธี AXI (Advanced eXtensible Interface) ดังรูปที่ 3.2



รูปที่ 3.1 บอร์ดทดลอง Zedborad



รูปที่ 3.2 โมเดลอย่างง่ายของ Zynq architecture

มีส่วนขาเข้าขาออก (I/O) ที่สามารถนำไปใช้ได้มากมายคือ:

- GPIO: 9 x LEDs, 8 x switches, 7 x push buttons
- Audio codec
- Video (HDMI out)
- Video (VGA)
- OLED display
- Pmod interfaces x 5
- Ethernet
- USB-OTG (peripherals)
- USB-JTAG (programming)
- USB-UART (communication)
- SD card slot
- FMC interface
- XADC header
- Xilinx JTAG header
- 256 Mbit flash memory
- 512 MB DDR3 memory

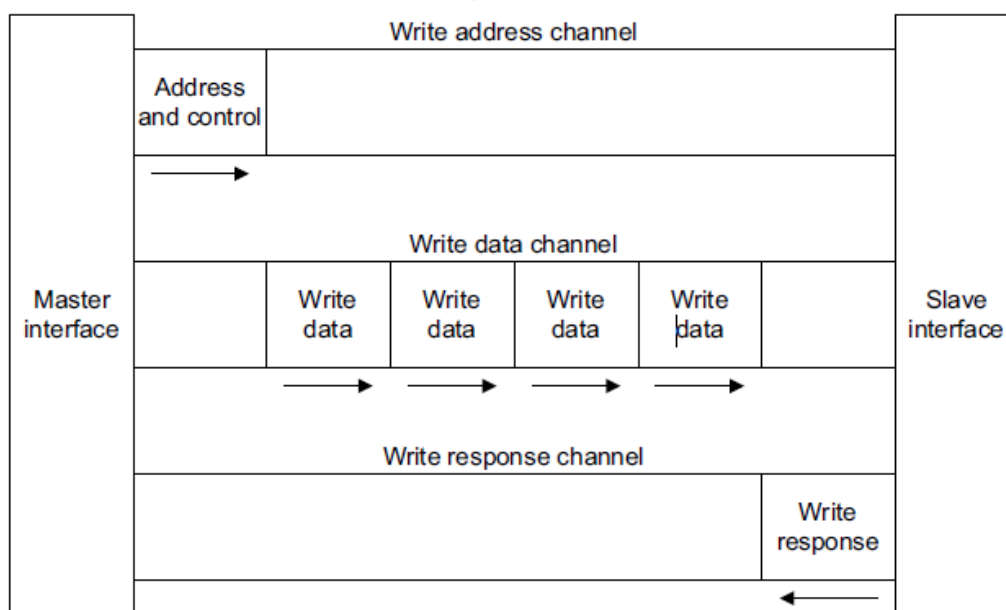
ชิพ ZC7Z020 Zynq ใน Zedboard ในส่วนของ FPGA นั้นจะมีทรัพยากรที่สามารถนำไปใช้ได้คือ 13,300 logic slices, 220 DSP48E1s, และ 140 BlockRAMs ซึ่งเพียงพอต่อการออกแบบวงจรขยายภาพ อีกทั้ง Zedboard ยังมี 512 MB DDR3 ที่น่าจะถูกนำมาใช้เป็นตัวเก็บข้อมูลภาพชั่วคราวได้ จะเห็นว่า Zedboard เหมาะที่จะนำมาทดสอบวงจรขยายภาพเพื่อผู้พิการทางการได้ยิน ทั้งนี้ในส่วนของการนำสัญญาณภาพเข้าทาง HDMI input นั้นจำเป็นต้องมีบอร์ดเสริมเนื่องจาก Zedboard มีเพียง HDMI output เท่านั้น ซึ่งจะกล่าวถึงบอร์ดเสริมในภายหลัง

3.1.1 การติดต่อสื่อสารแบบ AXI

เนื่องจาก DDR3 ของ Zedboard อยู่ในส่วนของ PS แต่ส่วนที่นำสัญญาณภาพมาทำการคำนวณจะอยู่ในส่วนของ PL ทำให้การใช้งาน DDR3 จะต้องมีการสื่อสารข้ามกันระหว่าง PS และ PL ซึ่งการสื่อสารดังกล่าวจะต้องติดต่อผ่านทาง AXI [15]

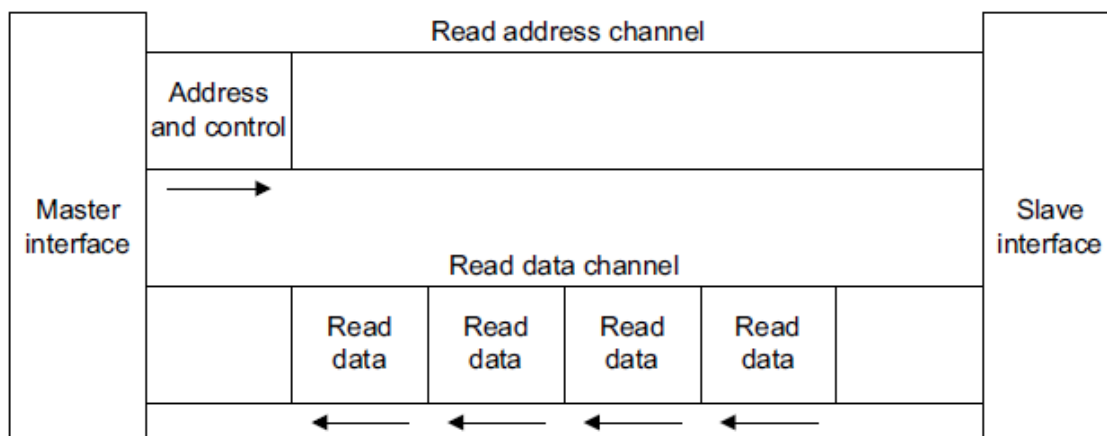
การรับส่งข้อมูลผ่าน AXI interface จะเรียกตัวที่ควบคุมว่า Master และตัวที่ถูกควบคุมว่า Slave ทั้งนี้ การสื่อสารของ AXI จะแบ่งส่วนกันระหว่างส่วน address กับส่วนของข้อมูลอย่างชัดเจน และรองรับการส่งข้อมูลที่ละเอียดละเอียดย (burst-based transactions) โดยโครงสร้างของ AXI สามารถแบ่งเป็น 2 ส่วนคือ ส่วนของการเขียน (write transaction) และ ส่วนของการอ่าน (read transaction)

ส่วนโครงสร้างของการเขียน จะประกอบด้วย 3 ช่องทางการสื่อสาร คือ Write address channel, Write data channel และ Write response channel ดังรูปที่ 3.3



รูปที่ 3.3 โครงสร้างการช่องทางการเขียน

ส่วนโครงสร้างของการอ่าน จะประกอบไปด้วย 2 ช่องทางการสื่อสาร คือ Read address channel และ Read data channel ดังรูปที่ 3.4



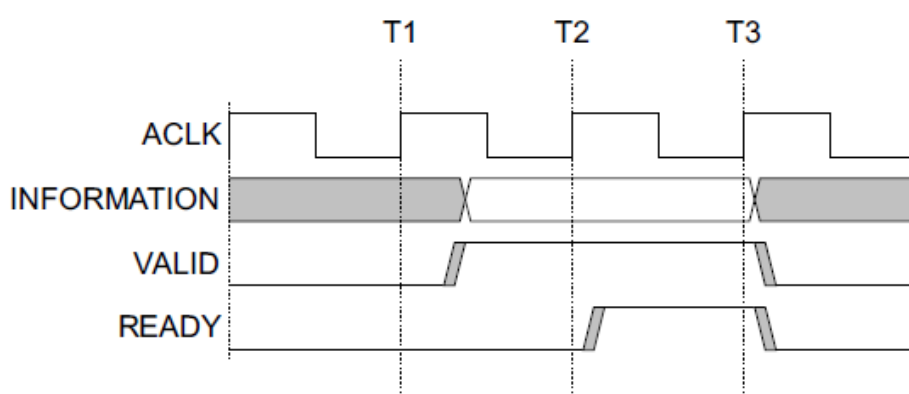
รูปที่ 3.4 โครงสร้างการช่องทางการอ่าน

ช่องทางการสื่อสารของ AXI

ทั้ง 5 ช่องทางการสื่อสาร (3 ช่องสำหรับการเขียน และ 2 ช่องสำหรับการอ่าน) จะเป็นอิสระต่อกันซึ่งจะมีสัญญาณข้อมูลที่ชื่อ VALID และ READY ที่ทำให้เกิดการสื่อสารแบบ two-way handshake ได้ ที่ต้นทางจะใช้ สัญญาณ VALID เพื่อแสดงว่า address, data หรือ control information สามารถนำไปใช้ได้ จากนั้นปลายทางจะใช้สัญญาณ READY เพื่อแสดงว่าสามารถที่จะรับข้อมูลนั้นแล้วดังตัวอย่างรูปที่ 3.5 และทั้ง read data channel และ write data channel จะมีสัญญาณ LAST เพื่อบอกว่าเป็นข้อมูลตัวสุดท้าย

- 1.) **Read and write address channels** ทั้งการเขียนและการอ่านจะมี address channel เป็นของตัวเอง address channel จะมีทั้ง address และสัญญาณควบคุมที่จะทำการส่ง
- 2.) **Read data channel** จะมีทั้งข้อมูลอ่านกับการตอบสนองการอ่าน (the read data and the read response information) จาก slave ไปยัง master โดยบัสข้อมูลจะมีความกว้างข้อมูลได้เป็น 8, 16, 32, 64, 128, 256, 512 หรือ 1024 bits ส่วนสัญญาณการตอบสนองการอ่านจะบอกถึงสถานะการอ่านข้อมูลว่าเสร็จแล้ว

- 3.) **Write data channel** จะมีข้อมูลเขียนจาก master ไปที่ slave โดยบัสข้อมูลจะมีความกว้างข้อมูลได้เป็น 8, 16, 32, 64, 128, 256, 512 หรือ 1024 bits และมีสัญญาณ byte lane strobe สำหรับทุกๆ 8 bits ข้อมูลเพื่อบอกว่า bytes ไหนของบัสข้อมูลใช้ได้
- 4.) **Write response channel** slave จะใช้ write response channel เพื่อตอบสนองการเขียนสำเร็จในแต่ละครั้ง

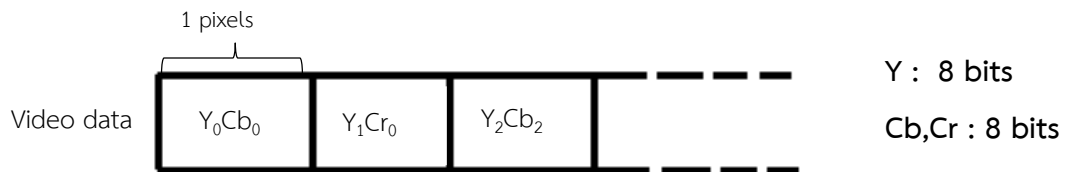


รูปที่ 3.5 ตัวอย่าง handshake VALID ก่อน READY :ที่เวลา T1 ต้นทางจะให้ VALID เป็น '1' จากนั้นที่เวลา T2 ปลายทางจะให้ค่า READY เป็น '1' จากนั้นต้นทางจะคงค่า INFORMATION ให้เสถียรจนการส่งเกิดขึ้นที่เวลา T3

3.1.2 ส่วนการแสดงผลภาพขาออก HDMI ของ Zedboard

HDMI output ของ Zedboard มีชิพ ADV7511 [16] ของบริษัท Analog device เป็นตัวส่งสัญญาณภาพ HDMI ที่รองรับความละเอียดสัญญาณภาพสูงสุดที่ 1080p ที่ 60 Hz โดยจริงๆ แล้ว ADV7511 สามารถแปลงสัญญาณขาเข้าได้หลายรูปแบบเช่น 24 bit RGB หรือ 24 bit YCbCr เป็นต้น แต่เนื่องจาก Zedboard มีการเดินสายจาก ADV7511 เพียง 16 สายจึงทำให้การตั้งสัญญาณขาเข้านั้นสามารถรับที่ความละเอียดสูงสุดเพียง 16 bits

สัญญาณขาเข้าของ ADV7511 สามารถปรับเปลี่ยนรูปแบบได้จากการผ่านทาง I2C เป็นตัวตั้งค่าทั้งนี้สำหรับวงจรรขยายภาพภาษามือนี้จะใช้รูปแบบขาเข้า ADV7511 เป็น 16 bit YCbCr 4:2:2 sperated syncs โดยสัญญาณภาพจะมีการส่งข้อมูลภาพเป็นดังรูปที่ 3.6



รูปที่ 3.6 รูปแบบของสัญญาณภาพ 16 bit YCbCr 4:2:2

ADV7511 จะต้องถูกตั้งค่าผ่าน I2C ให้ทำงานอย่างถูกต้องตามต้องการ โดยสามารถปรับโหมดการทำงานของ ADV7511 ได้อย่างหลากหลายเช่นให้ส่งสัญญาณภาพ HDMI ที่ความละเอียด 1080p ที่ 60 Hz, รับสัญญาณเข้าเป็น 16 bits YCbCr 4:2:2 separated syncs, ตั้งให้โหมดหลับเมื่อไม่ใช้งาน เป็นต้น

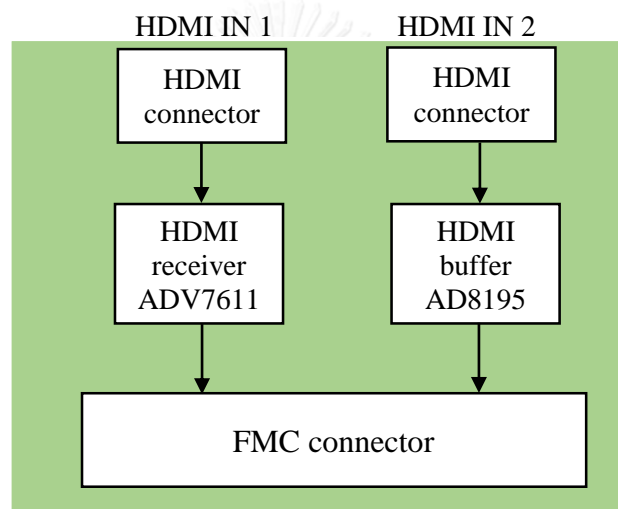
3.2 บอร์ดเสริม FMC-HDMI



รูปที่ 3.7 บอร์ดเสริม FMC-HDMI

เนื่องจากที่ได้กล่าวไปว่าจะใช้ Zedboard เป็นบอร์ดทดลองวงจรหลัก จำเป็นจะต้องหาบอร์ดเสริมที่ทำหน้าที่เป็นตัวรับสัญญาณเข้า HDMI (HDMI input) โดยเลือก FMC-HDMI [17] รูปที่ 3.7 เพราะสามารถทำหน้าที่นี้ได้ โดยจะนำ FMC-HDMI ไปต่อกับส่วนของ FMC interface ของบอร์ด Zedboard ก็สามารถใช้งานได้ทันที

แผนภาพบล็อกของ FMC-HDMI แสดงอยู่ในรูปที่ 3.8 โดยภายใน FMC-HDMI จะมีชิพ ADV7611 [18] ที่ช่องขาเข้า HDMI IN 1 และ AD8195 ที่ช่องขาเข้า HDMI IN 2 ทั้งสองชิพเป็นชิพของบริษัท Analog device ซึ่ง AD8195 จะทำหน้าที่เป็น buffer ของสัญญาณ HDMI ขาเข้าเท่านั้น ไม่ได้ทำการแปลงค่าสัญญาณแต่อย่างใด กล่าวคือรับมาเป็นสัญญาณ TMDS ขาออกของ AD8195 ก็จะเป็นสัญญาณ TMDS นั้นๆ ส่วนตัว ADV7611 เป็นตัวรับสัญญาณภาพ HDMI ได้สูงสุดที่ความละเอียดภาพ 1080p ที่ 60 Hz โดยจะสามารถปรับเปลี่ยนสัญญาณภาพขาออกจากได้หลายรูปแบบ เช่นเดียวกับ ADV7511 ทั้งนี้เพื่อความสอดคล้องกับ HDMI ขาเข้า ADV7611 จะต้องตั้งค่าผ่าน I2C เพื่อให้สัญญาณภาพขาออกเป็นรูปแบบเดียวกันคือ 16 bits YCbCr 4:2:2 seperated syncs ดังนั้นการใช้งานบอร์ดเสริม FMC HDMI นั้นจะใช้งานในส่วนของ HDMI IN 1 ซึ่งก็คือชิพ ADV7611 นั่นเอง



รูปที่ 3.8 แผนภาพบล็อกของ FMC-HDMI

นอกจากการตั้งค่า ADV7611 ในเรื่องของรูปแบบสัญญาณภาพขาออกแล้วกับความละเอียดภาพแล้ว ADV7611 ยังสามารถปรับตั้งค่าโหมดการทำงานเสริมอื่นๆได้อีกตามความต้องการ ตัวอย่างเช่น สามารถตรวจสอบได้ว่าการต่อสาย HDMI เข้ามาแล้ว หรือสามารถรู้ข้อมูล HDMI ที่เข้ามามีความละเอียดภาพเท่าใด เป็นต้น

บทที่ 4

วงจรขยายภาพภาษามือ

4.1 การใช้งานวงจรขยายภาพเคลื่อนไหวล่ามภาษามือ

วงจรขยายภาพภาษามือจะต้องติดตั้งในระบบภาพโทรทัศน์ที่เป็นสัญญาณภาพ HDMI ที่ความละเอียด 1920x1080 จุดภาพ โดยเมื่อเริ่มต้นการใช้งานผู้ใช้จะสามารถปรับขนาดของกรอบการขยายได้และสามารถเลือกตำแหน่งการขยายภาพได้ โดยผ่านทางปุ่มกดบน Zedboard โดยกรอบของการขยายจะมีรูปร่างเป็นกรอบสี่เหลี่ยม เมื่อปรับตำแหน่งกรอบที่ต้องการขยายเสร็จแล้วให้เลื่อน dip switch ขึ้นเพื่อเริ่มการขยายภาพในกรอบสี่เหลี่ยมดังกล่าวขึ้น 2 เท่า

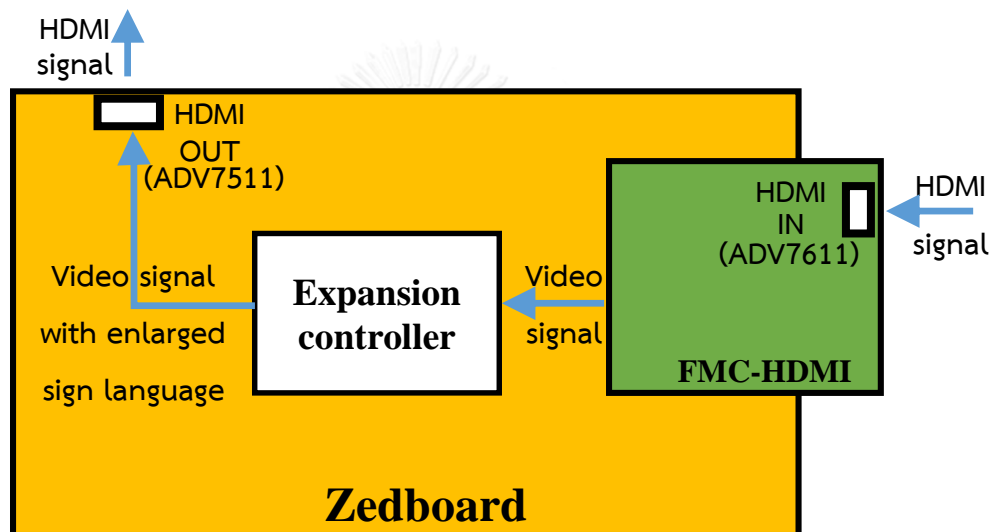


รูปที่ 4.1 แสดงการใช้งานวงจรขยายภาพภาษามือ

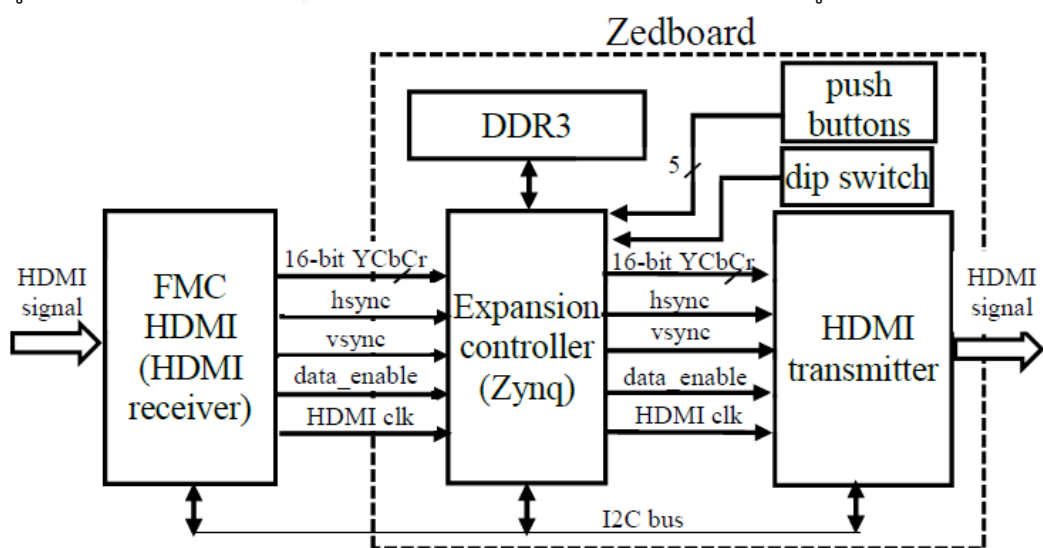
จากรูปที่ 4.1 ในโหมดปรับขนาดกรอบการขยายการกดปุ่ม P3 และ P4 จะทำการปรับขนาดของกรอบการขยายเพิ่มไปทางขวาและลงล่างตามลำดับ และการกดปุ่ม P1 และ P2 เพื่อทำการปรับขนาดของกรอบการขยายลดลงไปทางด้านบนและด้านซ้ายตามลำดับ (ลูกศรสีแดง) จากนั้นกดปุ่ม P0 เพื่อทำการปรับตำแหน่งของกรอบการขยายโดยกด P1, P2, P3 และ P4 เพื่อเลื่อนไปทางบน ซ้าย ขวา ล่าง ตามลำดับ (ลูกศรสีเขียว) ผู้ใช้งานสามารถกดปุ่ม P0 เพื่อเปลี่ยนโหมดเป็นการปรับตำแหน่งหรือปรับกรอบการขยายได้ทุกเมื่อ และดัน dip switch ขึ้นเมื่อต้องการเริ่มการขยายการขยาย 2 เท่า

4.2 การออกแบบวงจรขยายภาพภาษามือ

ในการออกแบบทั้งหมดจะทำในบอร์ดทดลอง Zedboard และบอร์ดเสริม FMC-HDMI โดยสัญญาณภาพที่รับเข้ามาจะเข้ามาทางช่องทาง HDMI input ของ FMC-HDMI จากนั้นจะทำการขยายภาพในส่วนของภาพภาษามือแล้วทำการแทรกภาพภาษามือเข้าไปในสัญญาณภาพเดิมจากนั้นส่งสัญญาณภาพดังกล่าวออกทางช่องทาง HDMI output ของ Zedboard ดังรูปที่ 4.2 โดยจะสามารถแบ่งส่วนประกอบของวงจรขยายภาพเคลื่อนไหวล่ามภาษามือได้เป็น 3 ส่วนหลักๆคือ ตัวรับสัญญาณ HDMI (HDMI receiver) วงจรควบคุมการขยายภาพ (Expansion controller) และตัวส่งสัญญาณ HDMI (HDMI transmitter) ดังรูปที่ 4.3 โดยสามารถอธิบายการทำงานในแต่ละส่วนได้ดังนี้

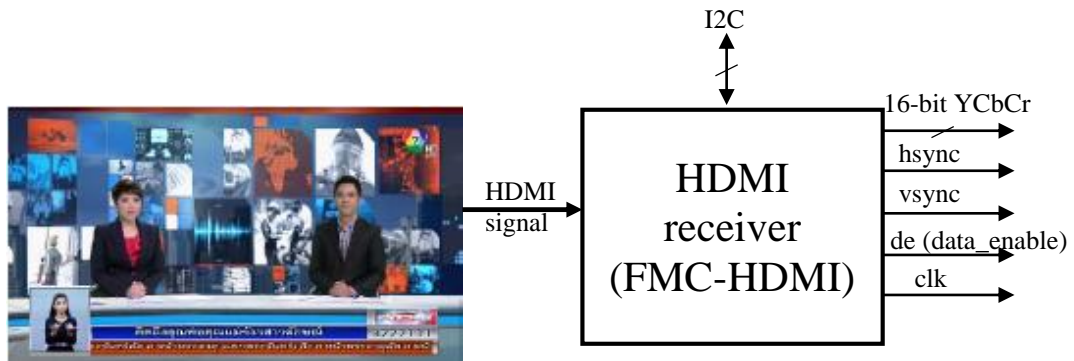


รูปที่ 4.2 แสดงการรับส่งสัญญาณภาพเคลื่อนไหวของวงจรขยายภาพเพื่อผู้พิการทางการได้ยิน



รูปที่ 4.3 โครงสร้างของวงจรขยายภาพภาษามือ

4.2.1 ตัวรับสัญญาณ HDMI (HDMI receiver)

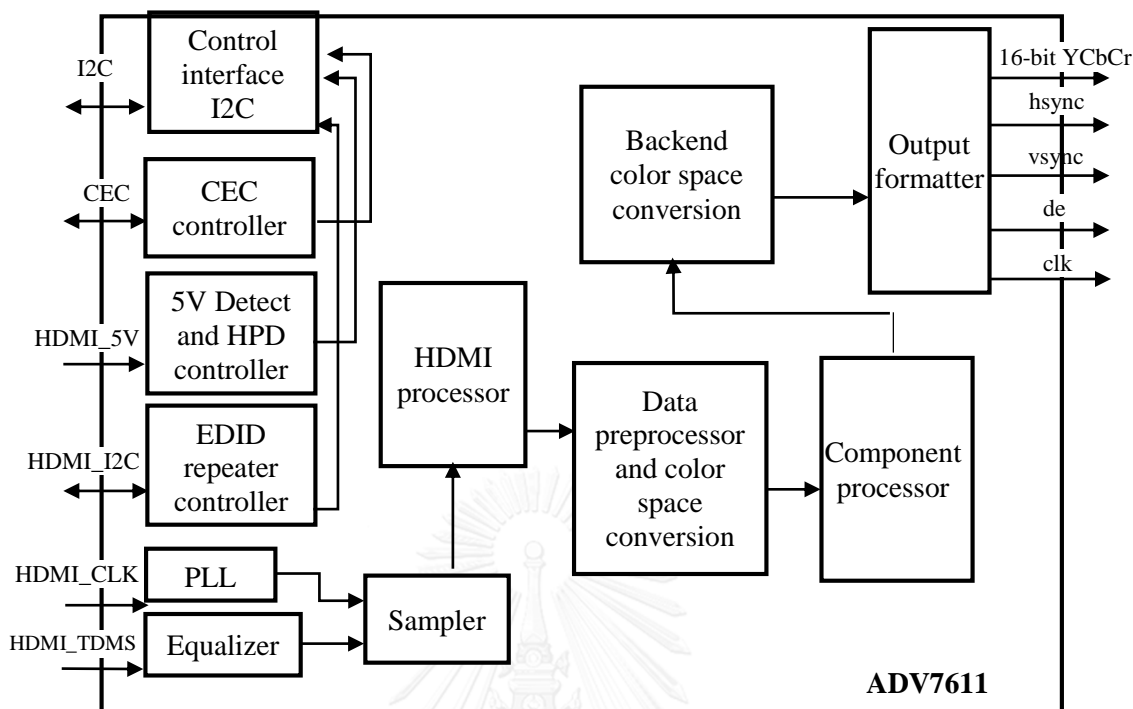


รูปที่ 4.4 การใช้งานตัวรับสัญญาณ HDMI

ตัวรับสัญญาณ HDMI ซึ่งคือบอร์ด FMC-HDMI ซึ่งจะใช้ชิพ ADV7611 โดยจะทำการแปลงสัญญาณ HDMI เป็นสัญญาณภาพที่มีองค์ประกอบของ 3 องค์ประกอบคือ สัญญาณความสว่าง (Y) 1 สัญญาณ และสัญญาณสีอีก 2 สัญญาณ (Cb,Cr) ซึ่งจะองค์ประกอบสีจะผลัดกันออกมาในรูปแบบ 4:2:2 ดังแสดงในรูปบทก่อนหน้าในที่รูปที่ 23 ซึ่งจะมีการเปลี่ยนข้อมูลภาพด้วยความถี่ 148.5 MHz ซึ่งสัญญาณภาพจะมีความละเอียดในระดับความละเอียดสูงเต็มรูปแบบ (Full-HD) ซึ่งสัญญาณภาพที่ได้จากตัวรับสัญญาณ HDMI จะมาคู่กับสัญญาณเวลาซึ่งได้แก่ สัญญาณ hsync ที่จะบอกจังหวะการขึ้นบรรทัดใหม่ของสัญญาณภาพ สัญญาณ vsync บอกจังหวะการขึ้นเฟรมใหม่ของสัญญาณภาพ และสัญญาณ data_enable ที่บอกช่วงเวลาว่าสัญญาณในช่วงไหนมีข้อมูลภาพอยู่โดยการทำงานของตัวรับสัญญาณ HDMI จะแสดงอยู่ในรูปที่ 4.4

ในตัวรับสัญญาณ HDMI ยังมี EDID แรม (EDID RAM) อยู่ภายในด้วยเพื่อทำการเก็บข้อมูลค่า EDID ทั้งหมดไว้ ซึ่งหน้าที่ของ EDID ก็ได้กล่าวไปแล้วในบทที่ 2 ซึ่งจะสามารถทำการตั้งค่า EDID RAM ด้วย I2C บัส นอกจากนี้ I2C บัสยังถูกนำมาใช้เพื่อตั้งค่าการทำงานทั้งหมดของ ADV7611 อีกด้วย ดังตัวอย่างดังนี้

- การตั้งค่าข้อมูลภาพข้อมูลขาออกเป็นสัญญาณภาพ 16-bit YCbCr 4:2:2 ที่มีสัญญาณเวลา (hsync, vsync และ data_enable) แยกมาด้วย
- ตั้งค่าไม่ใช้งานในส่วนของ HDCP (High-bandwidth Digital Content Protection)

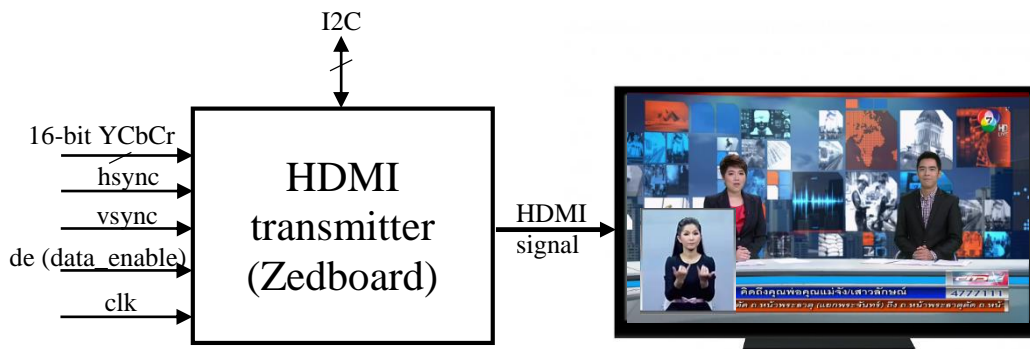


รูปที่ 4.5 แผนภาพบล็อกของวงจร ADV7611

แผนภาพบล็อกของวงจร ADV7611 สามารถแสดงได้รูปที่ 4.5 โดยส่วนประกอบภายในจะสามารถรองรับการทำงานพิเศษของ HDMI คือตัวควบคุม CEC (CEC controller) จะทำให้ HDMI สามารถใช้งานระบบ CEC ได้ ตัวตรวจหาไฟ 5 โวลต์และตัวควบคุม Hot-plug detector (5V Detect and HPD controller) จะสามารถทำให้วงจรสามารถตรวจสอบว่ามีการใช้งาน HDMI อยู่หรือไม่และสามารถให้ทำงานได้ปกติเมื่อถอดและเสียบสาย HDMI ใหม่ ตัวควบคุม EDID repeater (EDID repeater controller) จะเป็นตัวที่สามารถทำให้ EDID ภายในทำงานได้ปกติและสามารถเข้ามาอ่านข้อมูลได้จากช่องทาง HDMI_I2C โดยการตั้งค่าการทำงานนั้นๆจะใช้วงจรควบคุมต่อประสาน I2C (Control interface I2C)

ในส่วนของสัญญาณที่เป็นข้อมูลภาพนั้นจะเข้ามาทางช่องทาง HDMI_TDMS ที่จะผ่านตัว Equalizer ซึ่งจะทำหน้าที่ชดเชยการสูญเสียของข้อมูลที่เกิดจากการส่งสัญญาณความถี่สูงโดยเฉพาะอย่างยิ่งกับการใช้สาย HDMI ที่ยาว จากนั้นจะนำมาทำการแปลงสัญญาณ TMDS มาให้อยู่ในรูปข้อมูลของสีภาพซึ่งจะสามารถตั้งค่าได้ทั้ง RGB YUV และ YCbCr และสามารถปรับ color sampling ได้อีกด้วยซึ่งในที่นี้ก็จะตั้งค่าให้มีการทำงานมีข้อมูลสีภาพเป็น 16-bit YCbCr ที่ color sampling 4:2:2 อีกทั้งทำการสร้างสัญญาณเวลา (Hsync, Vsync และ Data_enable) ที่สามารถให้จังหวะการแสดงผลภาพได้ตรงกับสัญญาณข้อมูลสีภาพ การแปลงข้อมูลทั้งหมดจะเกิดในขั้นตอนต่อจากตัวบล็อก Sampler

4.2.2 ตัวส่งสัญญาณ HDMI (HDMI transmitter)



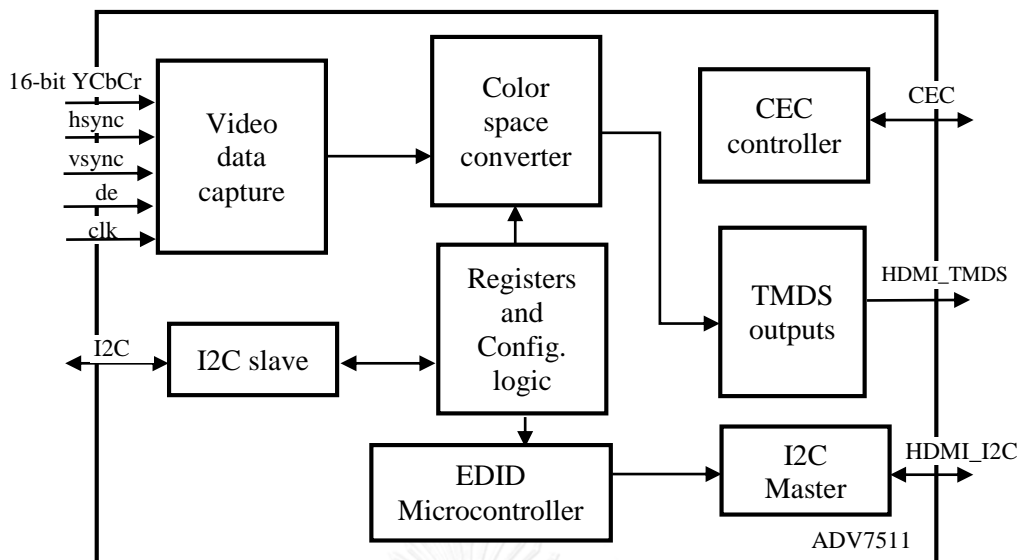
รูปที่ 4.6 การใช้งานตัวส่งสัญญาณ HDMI

ตัวส่งสัญญาณ HDMI จะซึ่งจะใช้ชิพ ADV7511 ที่อยู่บน Zedboard โดยจะรับสัญญาณภาพที่รับมาจากวงจรควบคุมการขยายภาพ เพื่อแปลงจาก YCbCr เป็นสัญญาณ HDMI เพื่อสามารถนำสัญญาณ HDMI ดังกล่าวไปใช้ร่วมกับโทรทัศน์ความละเอียดสูง ซึ่งรูปภาพของล่ามภาษามือจะมีขนาดใหญ่ขึ้น 2 เท่าดังแสดงในรูปที่ 4.6

การตั้งค่าการใช้งานตัวส่งสัญญาณ HDMI จะสามารถทำได้โดยการโปรแกรมผ่านบัส I2C เหมือนกับการตั้งค่าการใช้งานตัวส่งสัญญาณ HDMI เช่น

- ตั้งค่าการรับข้อมูลภาพให้อยู่ในรูปแบบ 16-bit YCbCr เพื่อให้มีรูปแบบของตัวส่งสัญญาณตรงกับตัวรับสัญญาณ HDMI
- ตั้งค่าไม่ใช้งานในส่วนของ HDCP (High-bandwidth Digital Content Protection)
- ตั้งค่าให้สามารถรับสัญญาณเวลาให้ตรงตามลักษณะของสัญญาณภาพความละเอียดสูงเต็มรูปแบบ (Full-HD)

แผนภาพบล็อกของวงจร ADV7511 สามารถแสดงได้ดังรูปที่ 4.7 ในส่วนของสัญญาณภาพที่รับมาจะเข้า ตัวจับสัญญาณภาพเคลื่อนไหว (Video data capture) ซึ่งจะสามารถรองรับกับข้อมูลภาพได้หลายรูปแบบทั้ง RGB และ YCbCr ทั้ง 4:2:2 และ 4:4:4 ทั้งนี้ในส่วนนี้จะถูกตั้งให้รับสัญญาณภาพรูปแบบ 16-bit YCbCr 4:2:2 เพื่อให้สอดคล้องกับตัวรับสัญญาณ HDMI หลังจากนั้นสัญญาณภาพ จะทำการเปลี่ยนรูปแบบสัญญาณภาพก่อนจะถูกแปลงเป็นสัญญาณภาพ TMDS ซึ่งเป็นส่วนของสัญญาณภาพของ HDMI นั้นเอง

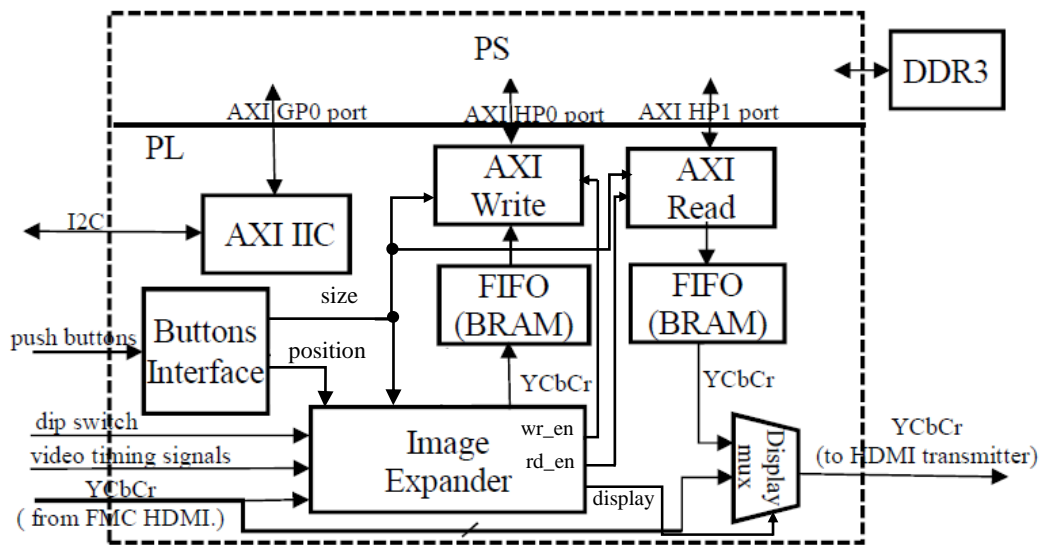


รูปที่ 4.7 แผนภาพบล็อกของวงจร ADV7511

ทั้งนี้ ADV7511 ยังมีตัวควบคุม CEC ที่ทำให้สามารถใช้งาน CEC ได้เช่นกัน และเนื่องจากการจะโปรแกรม ADV7511 จะต้องใช้ I2C จึงมีส่วนของ I2C slave ที่คอยรับการโปรแกรมตามเกณฑ์วิธีของ I2C อีกทั้ง มี EDID ไมโครคอนโทรลเลอร์ ที่คอยทำการอ่านค่า EDID จากตัวรับสัญญาณ HDMI ผ่าน I2C บัสของ HDMI (HDMI_I2C)

4.2.3 วงจรตัวควบคุมการขยายภาพ (Expansion controller)

เนื่องจาก Zedboard จะมีชิพ ZYNQ ที่สามารถทำการโปรแกรมให้ทำตามคำสั่งที่ต้องการได้ ในการออกแบบวงจรควบคุมการขยายภาพนั้นจึงจะออกแบบบน ZYNQ โดยภายใน ZYNQ จะประกอบไปด้วย 2 ส่วน คือ Processing System (PS) และ Programmable Logic (PL) โดย PS จะทำหน้าที่รับผิดชอบในการโปรแกรมเพื่อตั้งค่าวงจรตัวรับสัญญาณ HDMI และวงจรตัวส่งสัญญาณ HDMI ส่วนของ PL จะทำหน้าที่ขยายภาพและทำการส่งหรือรับสัญญาณภาพเคลื่อนไหวไปหรือจาก DDR3 โดยแผนภาพบล็อกของวงควบคุมการขยายภาพจะแสดงอยู่ในรูปที่ 4.8 โดยสัญญาณนาฬิกา (clock) ที่ใช้ทั้งหมดจะใช้ที่ความถี่ 148.5 MHz ที่เป็นความถี่ตรงกับความถี่ข้อมูลสัญญาณจุดภาพ



รูปที่ 4.8 แผนภาพบล็อกของวงจรตัวควบคุมการขยายภาพ

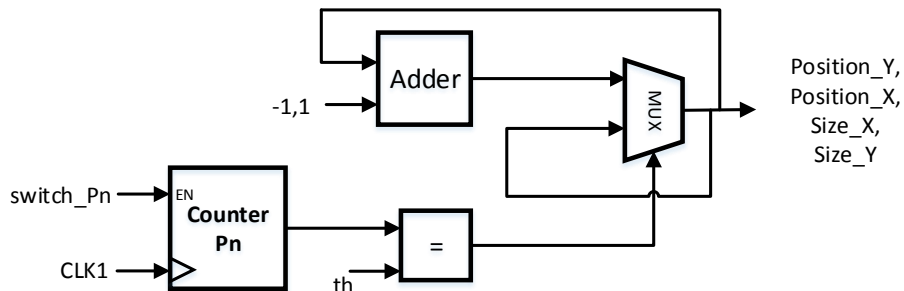
PS นั้นมีหน้าที่เพียงอย่างเดียวคือการโปรแกรมตัวรับและตัวส่งสัญญาณ HDMI ทำได้โดยการใช้ AXI IIC interface IP [19] หรือ AXI IIC ดังในรูปที่ 33 ซึ่งเป็นบล็อก IP ของ Xilinx โดย IP นี้จะทำให้สามารถทำการเขียนหรืออ่านค่าของอุปกรณ์ภายนอกได้ผ่านทาง IIC บัสด้วยการเขียนโปรแกรมภาษา C กล่าวคือเขียนภาษา C เพื่อใช้งานหน่วยประมวลผล ARM ในการส่งการสื่อสารแบบ IIC นั้นเอง ซึ่งทำให้ง่ายต่อการตั้งค่าชิพ ADV7511 และ ADV7611 ที่ต้องมีการตั้งค่าผ่าน I2C บัสไปที่ address ต่างๆเป็นจำนวนมาก

ภายใน PL หรือ FPGA จะประกอบไปด้วยวงจรต่างๆคือ วงจรประสานปุ่มกด (button interface) วงจรตัวขยายภาพ (Image Expander) วงจรตัวเขียน AXI (AXI Write) วงจรตัวอ่าน AXI (AXI Read) และตัว multiplexer แสดงผล โดยจะขออธิบายแต่ละส่วนดังนี้

1.) วงจรประสานปุ่มกด (button interface)

วงจรประสานปุ่มกดนั้นจะทำการแปลงข้อมูลการใช้งานปุ่มกด (push buttons) ของผู้ใช้งานจะถูกแปลงเป็นข้อมูลตำแหน่งและขนาดของภาพล่ามภาษามือโดยวงจรที่ชื่อว่า วงจรประสานปุ่มกด (button interface) ที่มีโครงสร้างดังรูปที่ 4.9 โดยปุ่มกดจะไปทำการสั่งให้ตัวนับทำงานและทำการปรับเปลี่ยนตำแหน่งหรือขนาดของล่ามภาษามือเมื่อกดปุ่มนานไปถึงระยะหนึ่งกล่าวคือเมื่อตัวนับมีค่าเท่ากับค่าคงที่ค่าหนึ่ง (threshold : th) วงจรจะทำการเพิ่มตำแหน่งหรือลดตำแหน่งของภาพล่ามภาษามือหรืออาจจะเป็นการปรับเปลี่ยนขนาดของภาพของล่ามภาษามือขึ้นกับโหมดในขณะนั้นเช่นจากรูป 26 ที่ถ้าเป็นโหมดการปรับตำแหน่งการกดปุ่ม P2 จะเป็นการ -1 ในตำแหน่ง X (position_X)

และกดปุ่ม P4 จะเป็นการ +1 ในตำแหน่ง Y (position_Y) หรือถ้าตอนนี้เป็นโหมดในการปรับเปลี่ยนขนาดของภาพภาษาเมื่อการกดปุ่ม P2 จะเป็นการลดกรอบการขยายภาพ และ P4 จะเป็นการเพิ่มกรอบการขยายภาพ เป็นต้น



รูปที่ 4.9 โครงสร้างของวงจรตัวต่อประสานปุ่มกด (button interface)

2.) วงจรตัวเขียน AXI (AXI Write) ตัวอ่าน AXI (AXI Read) และ multiplexer

แสดงผล

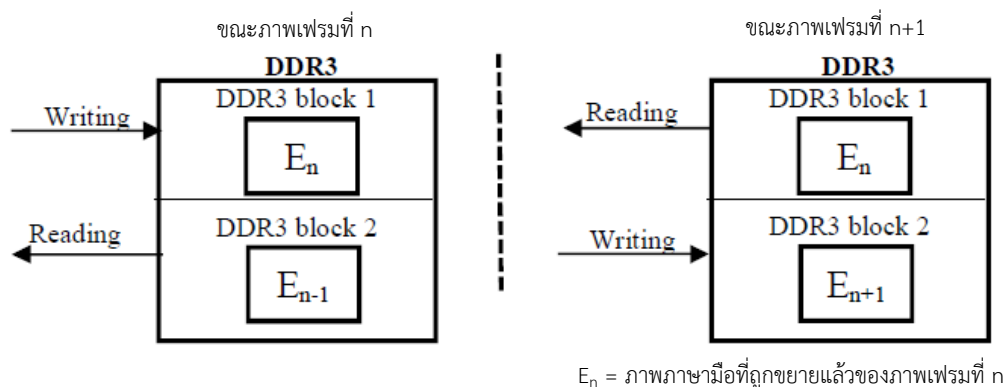
รูปภาพแต่ละเฟรมของล่ามภาษามือที่ถูกส่งเข้ามาจะถูกนำมาขยาย 2 เท่าก่อนด้วยตัวขยายภาพ (Image Expander) จากนั้นจะถูกนำไปเก็บไว้ในหน่วยความจำเพื่อนำไปแสดงในเฟรมถัดไป

ในพิจารณาหน่วยความจำที่ต้องใช้ในการเก็บภาพชั่วคราวของวงจรขยายภาพภาษาเมื่อนั้นจะพิจารณาจากภาพภาษามือที่ขยายแล้วที่มีความละเอียด 742×742 จุดภาพ ซึ่งเป็นขนาดภาพที่ใหญ่สุดที่วงจรขยายภาพภาษามือสามารถทำการขยายได้ โดยแต่ละจุดภาพจะมีข้อมูลอยู่ 16 บิตจะเป็นว่าต้องใช้หน่วยความจำทั้งหมด $742 \times 742 \times 16 = 8,809,024$ บิต โดยถ้าจะใช้ block rams ทั้งหมดภายใน FPGA ของ zedboard ซึ่งมีขนาด 36 Kb มีอยู่ทั้งหมด 140 หน่วย รวมเป็นหน่วยความจำที่รองรับได้ทั้งหมด $36 \times 1000 \times 140 = 5,040,000$ บิต ซึ่งถึงแม้การพิจารณานี้ยังไม่รวมถึงการใช้ block rams ที่จะนำมาใช้ในการทำเป็นตัวหน่วยเก็บความจำชั่วคราวในการทำการคำนวณการขยายภาพ ก็ จะเห็นว่าการใช้ block rams เพียงอย่างเดียวไม่เพียงพอ ดังนั้นจึงจำเป็นต้องมีการใช้หน่วยความจำ DDR3 ร่วมด้วย

ตัวเขียน AXI (AXI write) จะทำหน้าที่เขียนข้อมูลที่ได้รับมาไปที่หน่วยความจำ DDR3 ด้วยเกณฑ์วิธี AXI4 ซึ่งจะเป็นวิธีที่ต้องมีการรอสัญญาณจับมือ (handshaking signal) ดังนั้นตัวเก็บข้อมูลชั่วคราวแบบไฟโฟ (FIFO buffer) จึงจำเป็นที่ต้องแทรกไว้เพื่อทำการเก็บข้อมูลภาพไว้ชั่วคราวก่อนการทำจับมือกันของสัญญาณเสร็จสิ้น และภาพที่ขยายนั้นก็จะถูกอ่านออกมาจากหน่วยความจำ DDR3 ด้วยตัวอ่าน AXI (AXI read) ที่มีตัวเก็บข้อมูลชั่วคราวแบบไฟโฟเช่นกันด้วยเหตุผลเดียวกับตัวเขียน AXI จากนั้น Display multiplexer จะทำการเลือกสัญญาณภาพระหว่างสัญญาณภาพต้นฉบับกับสัญญาณภาพที่ถูกขยายแล้ว

การพิจารณาการจอง address ของหน่วยความจำ DDR3 จะสามารถคำนวณได้ดังนี้ ในการเลื่อนแต่ละ 1 บิตของ address ของหน่วยความจำ DDR3 จะเก็บข้อมูลได้ 8 บิต เช่น เก็บข้อมูลที่ละ 16 บิต จะต้องเก็บที่ address x“00000000” และ x“00000001” และจะเก็บข้อมูล 16 บิตถัดไปได้ที่ address x“00000002” ดังนั้นถ้าพิจารณาการเก็บภาพที่ขยายแล้วที่ใหญ่ที่สุดที่มีข้อมูล 8,809,024 บิต จะต้องมีการเลื่อน address เพื่อจะใช้เก็บข้อมูลถัดไปที่ address $8,809,024/8 = x“0010CD48”$ แต่การใช้หน่วยความจำ DDR3 เพื่อเก็บข้อมูลได้นั้นจะเริ่มที่ address x“00100000” ซึ่งแสดงว่าการเริ่มเก็บข้อมูลภาพถัดไปได้จะเริ่มที่ $x“00100000” + x“0010CD48” = x“0020CD48”$ เพื่อให้ไม่เกิดการล้นของ address อย่างแน่นอนจึงจะขอเริ่มเก็บข้อมูลถัดไปที่ x“00300000”

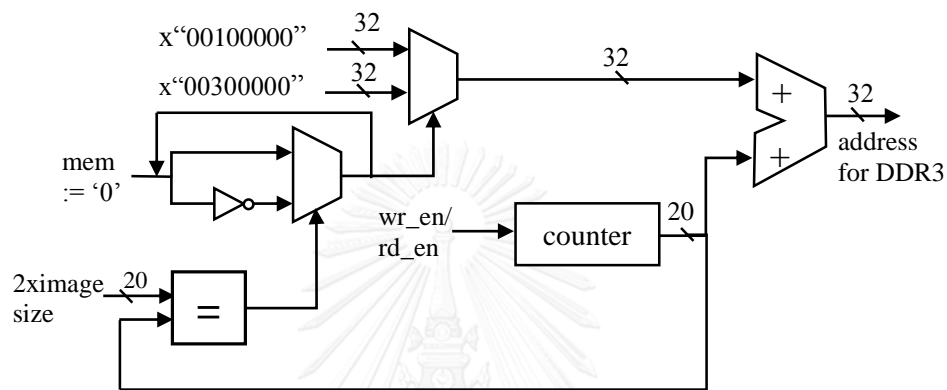
ดังนั้นหน่วยความจำ DDR3 จึงจะถูกแบ่งเป็น 2 บล็อก โดยบล็อกแรก (DDR3 block 1) เริ่มจาก address ที่ x“00100000” และบล็อกที่สอง (DDR3 block 2) เริ่มที่ address ที่ x“00300000” ซึ่งเป็นค่า 32 บิต การแบ่งหน่วยความจำ DDR3 เป็น 2 บล็อกนี้เพื่อที่จะสามารถทำการอ่านหน่วยความจำ DDR3 ที่บล็อกหนึ่งๆได้พร้อมกับการทำการเขียนหน่วยความจำ DDR3 อีกบล็อกได้ โดยการเขียนและการอ่านหน่วยความจำ DDR3 แสดงไว้ในรูปที่ 4.10 กล่าวคือในขณะที่ภาพภาษามือที่ขยายแล้วของภาพเฟรมที่ n (E_n) กำลังถูกเขียนลงบล็อกที่หนึ่ง ภาพภาษามือที่ขยายแล้วของภาพเฟรมที่ n-1 (E_{n-1}) จะถูกอ่านจากบล็อกที่สองเพื่อนำไปแสดงภาพ หลังจากนั้นเมื่อภาพภาษาที่ขยายแล้วของภาพเฟรมที่ n+1 (E_{n+1}) กำลังเข้ามาจะถูกนำไปเขียนที่บล็อกที่สองและขณะเดียวกันภาพภาษามือที่ขยายแล้วเฟรมที่ n (E_n) ก็จะถูกอ่านออกมากจากบล็อกที่หนึ่ง บล็อกที่หนึ่งและบล็อกที่สองจะสลับกันทำหน้าที่ถูกอ่านและถูกเขียนไปเรื่อยๆและจะเห็นว่าภาพที่ถูกนำมาแสดงนั้นจะมีการหมุนเวลาไปหนึ่งเฟรมนั่นเอง



รูปที่ 4.10 การใช้งานหน่วยความจำ DDR3 ขณะช่วงเวลาภาพเฟรมที่ n และภาพเฟรมที่ n+1

โครงสร้างของวงจรนับ address ของวงจรตัวเขียน AXI และตัวอ่าน AXI แสดงอยู่ในรูปที่ 4.11 โดยค่า mem จะแสดงถึง DDR3 บล็อกซึ่งค่า mem = '0' จะหมายถึงบล็อก 1 และ mem =

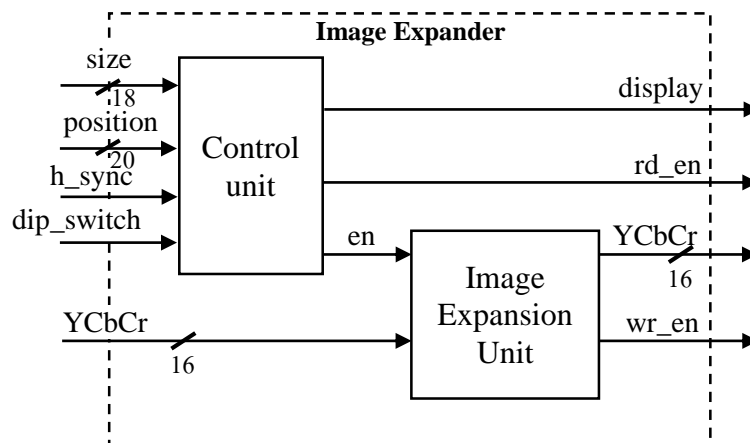
‘1’ จะแสดงถึงบล็อกที่ 2 โดยตัวนับจะนับค่า wr_en และ rd_en สำหรับตัวเขียน AXI และตัวอ่าน AXI ตามลำดับ เมื่อตัวนับนับค่าถึงค่าที่เป็นขนาดของภาพสองเท่าแล้วจะทำการสลับการใช้บล็อกของ DDR3 ซึ่งจะเห็นได้ว่าเมื่อวงจรถับนับถึงเท่ากับขนาดของภาพสองเท่าแล้วค่า mem จะกลับค่าด้วย not gate ซึ่งจะทำให้การสลับค่าเริ่มต้นของ address นั้นเอง โดยค่าภาพที่ใหญ่ที่สุดที่สามารถรับได้คือ 371x371 จุดภาพซึ่งหมายถึงภาพที่ขยายแล้ว 742x742 จุดภาพหรือ 550,564 จุดภาพ ซึ่งก็มีค่าอยู่ในช่วง 20 บิตดังนั้นค่าขนาดภาพสองเท่าจึงต้องใช้ 20 บิต



รูปที่ 4.11 วงจรนับ address ของ AXI Write และ AXI Read

3.) วงจรขยายภาพ (Image Expander)

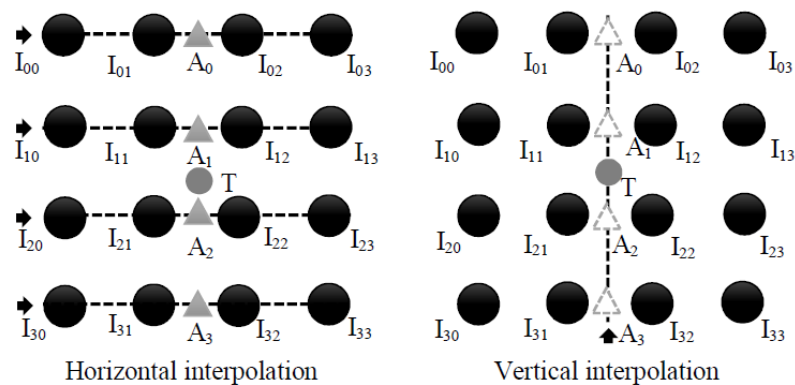
วงจรขยายภาพ (Image Expander) จะเป็นวงจรหลักในวงจรขยายภาพล่ามภาษามือที่มีหน้าที่คือการขยายภาพเป็นขนาดสองเท่าด้วยวิธีการประมาณค่าแบบไบคิวบิก โดยแผนภาพบล็อกของวงจรขยายภาพถูกแสดงในรูปที่ 4.12 สามารถแบ่งได้เป็น 2 ส่วนคือส่วนการขยายภาพและส่วนควบคุม โดยจะสามารถอธิบายได้ดังนี้



รูปที่ 4.12 แผนภาพบล็อกของวงจรขยายภาพ

3.1) ส่วนของการขยายภาพ

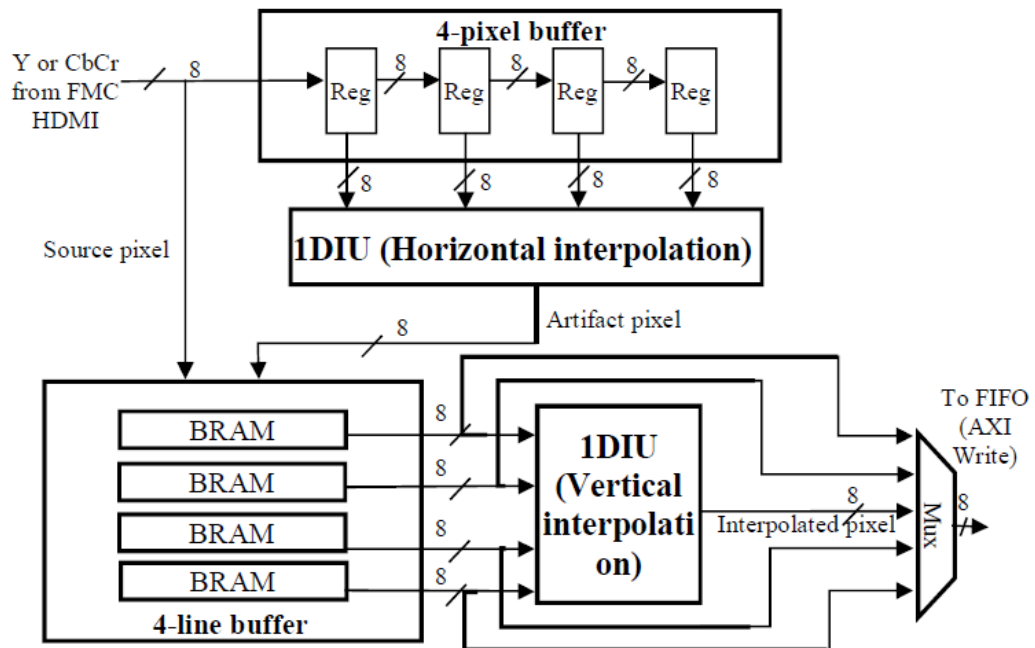
จากที่ได้กล่าวในบทที่ 2 ว่าการประมาณค่าด้วยวิธีไบคิวบิกสามารถแบ่งได้เป็น 2 ขั้นตอน การประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติตามแนวนอนจากนั้นจึงทำการประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติตามแนวตั้งดังรูปที่ 4.13 ซึ่งขอเรียกจุดภาพที่ได้จากการทำการประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติตามแนวนอนว่า A (artifact pixel) และจุดภาพที่ได้จากการทำการประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติสองครั้งแล้วว่า T (interpolated pixel)



รูปที่ 4.13 การทำการประมาณค่าไบคิวบิกด้วยการทำการประมาณค่าหนึ่งมิติสองครั้ง

โครงสร้างของตัวขยายภาพในส่วนของการขยายภาพแสดงไว้ในรูปที่ 4.14 เนื่องจากตัวขยายภาพรับข้อมูลจุดภาพทีละเส้นภาพแบบต่อเนื่องในการทำการประมาณค่าด้วยวิธีไบคิวบิกจึงจำเป็นต้องมีตัวเก็บข้อมูลภาพไว้ชั่วคราวโดยการเก็บข้อมูลภาพแบบต่อเนื่อง 4 จุดภาพเรียกว่า 4-pixel buffer จากนั้นจึงจะสามารถทำการประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติในแนวนอนด้วยกล่อง 1DIU (1-D Interpolation Unit) ที่อธิบายไว้ในบทที่ 2 เมื่อผ่านส่วนนี้จะได้ค่าจุดภาพที่คำนวณแล้ว A (Artifact pixel) หลังจากนั้นจุดภาพถูกไปเก็บเป็นแถวๆ 4 แถวโดยเรียกว่า 4-line buffer เมื่อเก็บครบ 4 แถวแล้วทยอยส่งเข้า 1DIU เพื่อทำการประมาณค่าด้วยวิธีไบคิวบิกหนึ่งมิติในแนวตั้งจากนั้นจะได้ค่า T (Interpolated pixel) หลังจากนั้นตัว 5:1 multiplexer จะทำการเลือกจุดภาพระหว่าง Artifact pixel และ Interpolated pixel ที่เหมาะสมเพื่อสร้างภาพขยายและก็นำไปเก็บไว้ที่หน่วยความจำ DDR3 โดยการขยายภาพนี้จะทำการขยายแยกกันทีละองค์ประกอบสีทั้ง Y และ CbCr

ทั้งนี้วงจรขยายภาพในส่วนของการขยายภาพนี้จะทำการสร้างสัญญาณ wr_en ออกมาด้วย ซึ่งไม่ได้เขียนแสดงไว้ในรูป โดยการสร้าง wr_en = '1' นั้นจะถูกสร้างมาพร้อมกับสัญญาณภาพที่สร้างมาได้ในส่วนสุดท้ายของวงจรเพื่อส่งต่อไปให้กับ AXI Write เพื่อทำการเขียนต่อไป

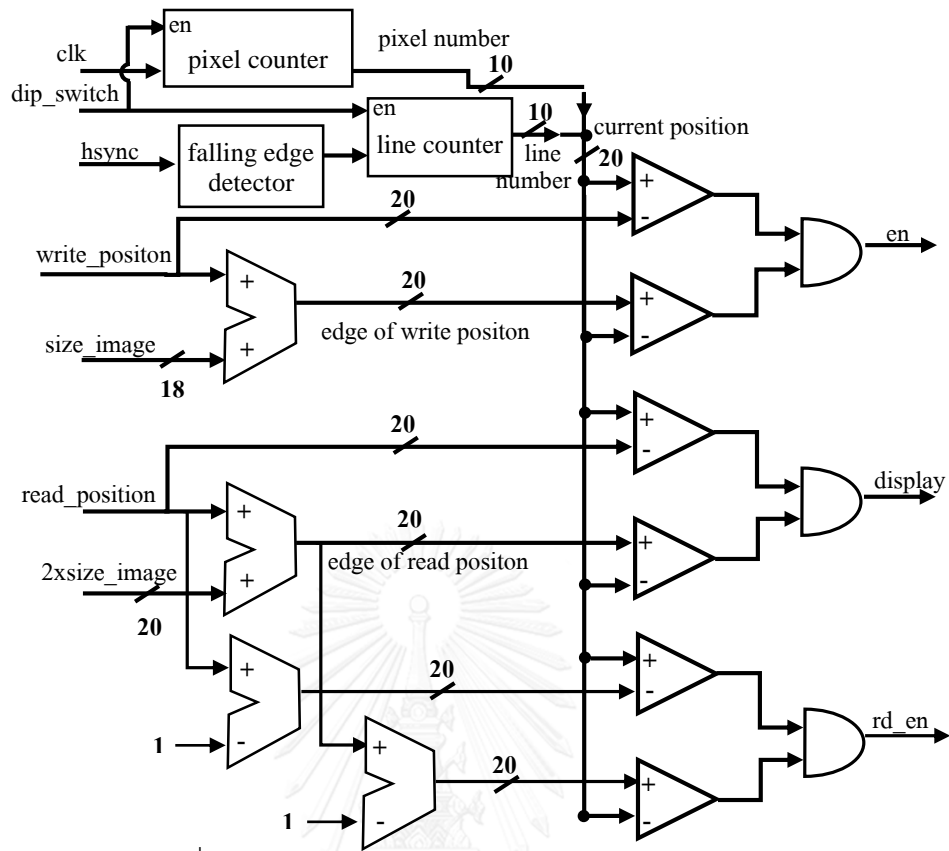


รูปที่ 4.14 โครงสร้างของตัวขยายภาพในส่วนของการขยายภาพ

3.2) ส่วนของการของการควบคุม

ในส่วนของการควบคุมภายในวงจรขยายภาพ (Image Expander) จะทำหน้าที่สร้างสัญญาณเพื่อควบคุมการทำงานทั้งหมดของวงจรขยายภาพได้แก่สร้างสัญญาณ en เพื่อส่งส่วนของการขยายภาพของวงจรขยายภาพทำงาน สัญญาณ rd_en เพื่อสั่งให้ AXI Read ทำงาน และสร้างสัญญาณ $display$ เพื่อสั่งให้ display multiplexer ทำการเลือกส่งสัญญาณภาพที่ทำการขยายแล้ว

โครงสร้างภายในของวงจรสร้างสัญญาณควบคุมภายในวงจรขยายภาพนั้นแสดงอยู่ในรูปที่ 4.15 ซึ่งประกอบไปด้วยวงจรถับ (counter) 2 ตัวคือ ตัวนับจุดภาพ (pixel counter) ซึ่งจะทำหน้าที่ในการนับตำแหน่งล่าสุดของจุดภาพในปัจจุบัน และตัวนับเส้นภาพ (line counter) ที่จะทำหน้าที่ในการนับตำแหน่งเส้นภาพของปัจจุบัน อีกทั้งยังมี วงจรตรวจการเปลี่ยนจาก '1' เป็น '0' (falling edge detector) เพื่อให้ตัวนับเส้นภาพได้สามารถใช้กับ $hsync$ ได้กล่าวคือเมื่อ $hsync$ มีการเปลี่ยนค่าจาก '1' เป็น '0' แล้วจะถือว่ามีการเปลี่ยนเส้นภาพไปหนึ่งเส้นภาพนั่นเอง โดยเมื่อเรารู้ทั้งตำแหน่งของจุดภาพและตำแหน่งของแถวภาพแล้วเราก็สามารถรู้ตำแหน่งของภาพได้อย่างแม่นยำนั่นเอง



รูปที่ 4.15 โครงสร้างของวงจรควบคุมของวงจรขยายภาพ

วงจรเปรียบเทียบจะทำการเปรียบเทียบเมื่อค่าที่ขาบวกมีค่ามากกว่าขาลบจะให้ผลลัพธ์เป็น '1' ซึ่งใช้ร่วมกับวงจรและ (and gate) จะทำให้สัญญาณ en มีค่าเป็น '1' ในช่วงของตำแหน่งที่อยู่ในระหว่างการแสดงภาพล่ามภาษามือ สัญญาณ display มีค่าเป็น '1' ในช่วงของตำแหน่งที่อยู่ในระหว่างเริ่มตำแหน่งล่ามภาษามือที่มีขนาดใหญ่ขึ้นสองเท่า และส่วนสัญญาณ rd_en จะมีค่าเป็นหนึ่งในช่วงก่อนสัญญาณภาพล่ามภาษามือ 1 แถวภาพเนื่องจากในการดึงข้อมูลภาพจาก DDR3 จะต้องมี การรอสัญญาณการจับมือก่อนจึงไม่สามารถดึงสัญญาณภาพมาได้ทันที ซึ่งการให้ค่า rd_en มีค่าเป็น '1' ก่อน 1 แถวและเก็บไว้ก่อนในหน่วยเก็บข้อมูลภาพชั่วคราว (buffer) ก่อนจะสามารถทำให้แสดงภาพได้ตรงตามตำแหน่งที่ต้องการได้เพราะว่าการเรียกข้อมูลภาพจากที่เก็บข้อมูลภาพชั่วคราวจะสามารถนำออกมาได้ทันที

บทที่ 5

ผลการทดลองและสรุปผล

5.1 การวิเคราะห์วงจรขยายภาพภาษามือ

ในการออกแบบวงจรภายใน Zedboard ที่ใช้เทคโนโลยี ZYNQ นั้นได้ใช้ซอฟต์แวร์ Vivado 2016.2 โดยสามารถออกแบบวงจรตัวควบคุมการขยายภาพ (Expansion controller) เป็นแผนภาพบล็อกใน Vivado ได้ดังรูปที่ 5.1 วงจรที่ถูกออกแบบเองทั้งหมดนั้นจะอยู่ในบล็อกที่ชื่อว่า sign_image_expander_0 ส่วนบล็อกที่เหลือจะเป็น IP ของ Xilinx ทั้งหมด

โดยการใช้สัญญาณนาฬิกามีทั้งหมด 3 สัญญาณคือ 1.สัญญาณนาฬิกาของ AXI IIC (digilent_iic และ zedboard_iic) ใช้งานของสัญญาณนาฬิกา clk_fpga_0 ที่ 100 MHz 2.สัญญาณนาฬิกาของส่วนที่มีการเชื่อมต่อระหว่างหน่วยความจำ DDR3 (AXI Write และ AXI Read) ที่ใช้สัญญาณนาฬิกา clk_fpga_1 ที่ 150 MHz และ 3.สัญญาณนาฬิกาของส่วนการขยายภาพ ใช้สัญญาณนาฬิกา hdmi_clk ที่ 148.5 MHz

ในส่วนการพิจารณาส่วนของเส้นทางที่มีการหน่วง (delay path) มากสุด แบ่งเป็น 3 ส่วนตามสัญญาณนาฬิกาคือ clk_fpga_1 ในส่วนของ AXI IIC clk_fpga_1 ในส่วนที่มีการเชื่อมต่อระหว่างหน่วยความจำ DDR3 และ hdmi_clk ในส่วนของการขยายภาพ ซึ่งแสดงในตารางที่ 5.2

ความถี่สูงสุดของแต่ละส่วนวงจรสามารถทำงานได้โดยสามารถหาค่าจากส่วนกลับของเวลาหน่วงของเส้นทางที่มากที่สุดของวงจรมันดังสมการ

$$f_{max} = \frac{1}{t_{d,max}} \quad \text{สมการที่ (5.1)}$$

เมื่อ f_{max} คือความถี่สูงสุดที่วงจรนั้นๆสามารถทำงานได้และ $t_{d,max}$ คือเวลาหน่วงของเส้นทางที่มีค่ามากที่สุด

จากสมการ 5.1 สามารถคิดความถี่มากที่สุดของแต่ละส่วนวงจรสามารถทำงานได้ดังนี้ ในส่วนที่ 1 ส่วนของสัญญาณนาฬิกา clk_fpga_1 สามารถทำงานได้ที่ความถี่สูงสุดที่ $1/7.07 = 141.44$ MHz ซึ่งในส่วนนี้ไม่จำเป็นต้องใช้นาฬิกาที่มีความถี่สูงมากก็ได้เนื่องจากเป็นส่วนของการตั้งค่าเริ่มต้นเท่านั้น ส่วนที่ 2 ส่วนของสัญญาณนาฬิกา clk_fpga_1 สามารถทำงานได้ที่ความถี่สูงสุด $1/5.71 = 175.13$ MHz ซึ่งหมายความว่าถึงแม้หน่วยความจำ DDR3 ใน zedboard สามารถทำงานได้สูงถึง 533 MHz [20] แต่ในการส่งสัญญาณข้อมูลรับจาก/ไปยังหน่วยความจำ DDR3 มีคอขวดอยู่ที่ axi_mem_intercon ซึ่งเป็น IP ที่จำเป็นต้องใช้ในการใช้ protocol AXI และส่วนสุดท้ายส่วนที่ 3 สัญญาณนาฬิกา hdmi_clk ทำงานได้ที่ความถี่สูงสุด $1/6.03 = 165.83$ MHz ในส่วนนี้ถึงส่วนการ

ขยายภาพทำงานได้ที่ความถี่สูงมากเท่าใด ก็จำเป็นต้องให้ทำงานที่ความถี่ของการรับสัญญาณภาพที่ได้รับมาตาม (pixel clock) อยู่ดี

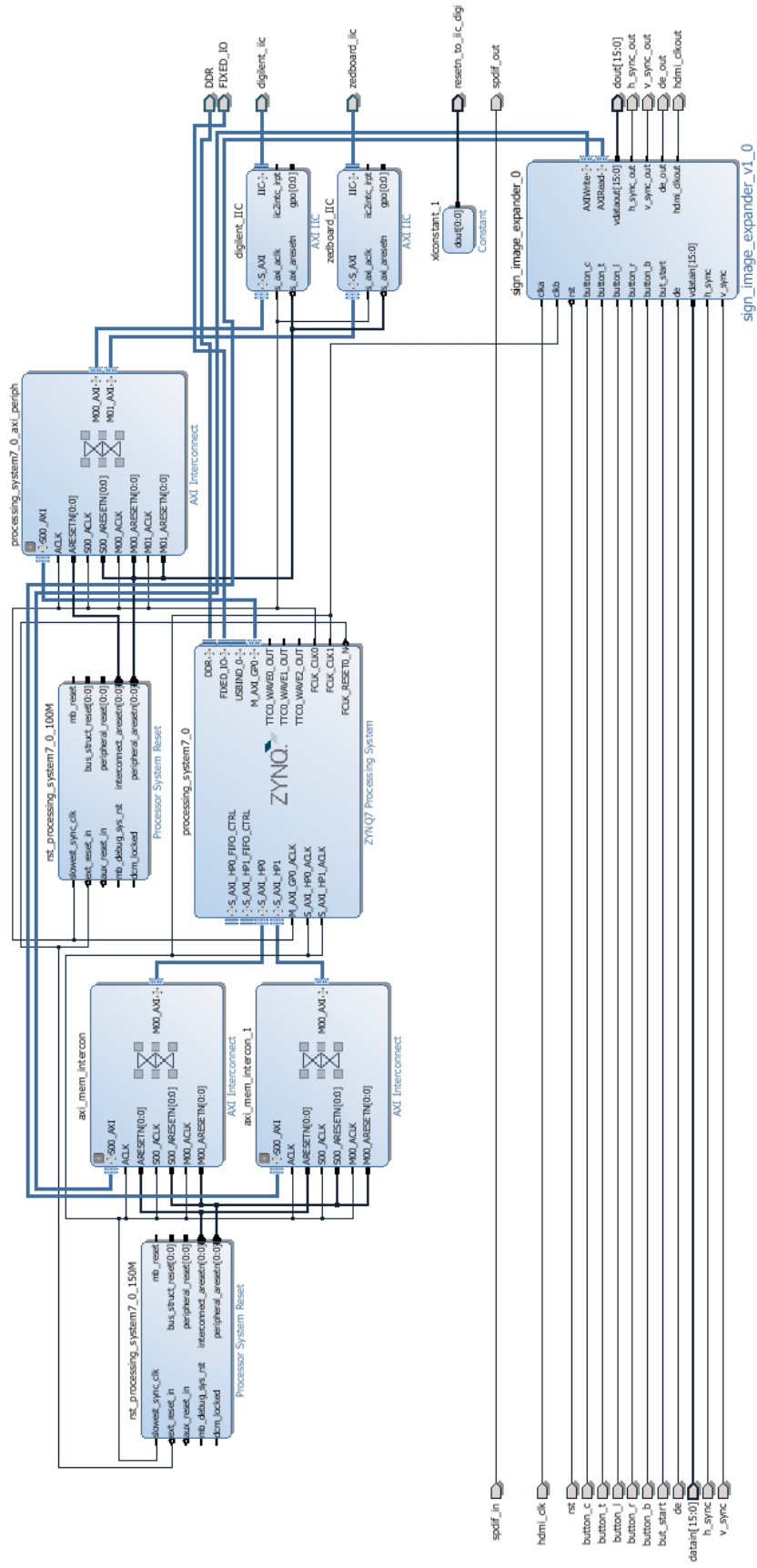
ตารางที่ 5.1 เส้นทางในวงจรที่มีการหน่วงมากที่สุด

สัญญาณนาฬิกา	จาก	ไปยัง	Delay (ns)
clk_fpga_0	processing_system7_0	digilent_IIC	7.07
clk_fpga_1	axi_mem_intercon	axi_mem_intercon	5.71
hdmi_clk	inst_blk_mem_gen	YCbCrr_interpol_unit	6.03

การใช้ทรัพยากรของ FPGA หรือ PL ของวงจรขยายภาพภาษามือถูกแสดงอยู่ในตารางที่ 5.2 จะเห็นได้ว่าเนื่องจากการลดความซับซ้อนของการขยาย 2 เท่าของวิธีการประมาณค่าแบบไบคิวบิกทำให้ไม่จำเป็นต้องใช้ตัวคูณหรือนั่นก็คือไม่จำเป็นต้องใช้ DSP48E ซึ่งเป็นวงจรสำหรับการคำนวณทางคณิตศาสตร์ในวงจร FPGA

ตารางที่ 5.2 การใช้ทรัพยากรภายใน FPGA

ชื่อวงจร	Registers	LUTs	Block RAMs (36 Kb)
AXI IIC	736	768	0
Button interface	217	276	0
Image Expander	167	281	4
AXI Write	390	248	2
AXI Read	225	158	1
วงจรอื่นๆ	1464	1047	0
รวม	3199(3%)	2778(5%)	7(5%)



รูปที่ 5.1 แผนภาพบล็อกวงจรตัวควบคุมการขยายภาพใน Vivado

โครงสร้างในส่วนของวิธีการขยายภาพของวงจรขยายภาพภาษาเมื่อได้ถูกนำมาเปรียบเทียบกับโครงสร้างการขยายภาพที่ใช้วิธีการประมาณค่าในช่วงแบบไปคิวบิกที่สามารถทำงานได้ตามเวลาจริง (real time) เช่นเดียวกันแต่มีการออกแบบในลักษณะโครงสร้างอื่นๆได้แก่ของ Nuno-Maganda [21], Mahale [22] และ Lin [9] โดยแสดงอยู่ในตารางที่ 5.3 ซึ่งข้อดีของวงจรขยายภาพที่นำเสนอก็คือมีการใช้ Block RAMs ที่น้อยกว่าและไม่มีการใช้งานตัววงจรที่ใช้ในการคำนวณทางคณิตศาสตร์ใดๆ ส่วนในเรื่องของความเร็วในการคำนวณก็สามารถทำได้แบบตามเวลาจริง (real-time) ของสัญญาณภาพความละเอียดภาพสูงเต็มรูปแบบ ที่ต้องการความถี่การทำงาน 148.5 MHz และแสดงภาพด้วยความถี่เฟรม 60 Hz (ให้เวลาคำนวณต่อเฟรม 16.7 ms)

ตารางที่ 5.3 การเปรียบเทียบโครงสร้างวงจรขยายภาพด้วยวิธีการประมาณค่าในช่วงแบบไปคิวบิกของวงจรขยายภาพล่ามภาษาเมื่อเทียบกับโครงสร้างอื่นๆ

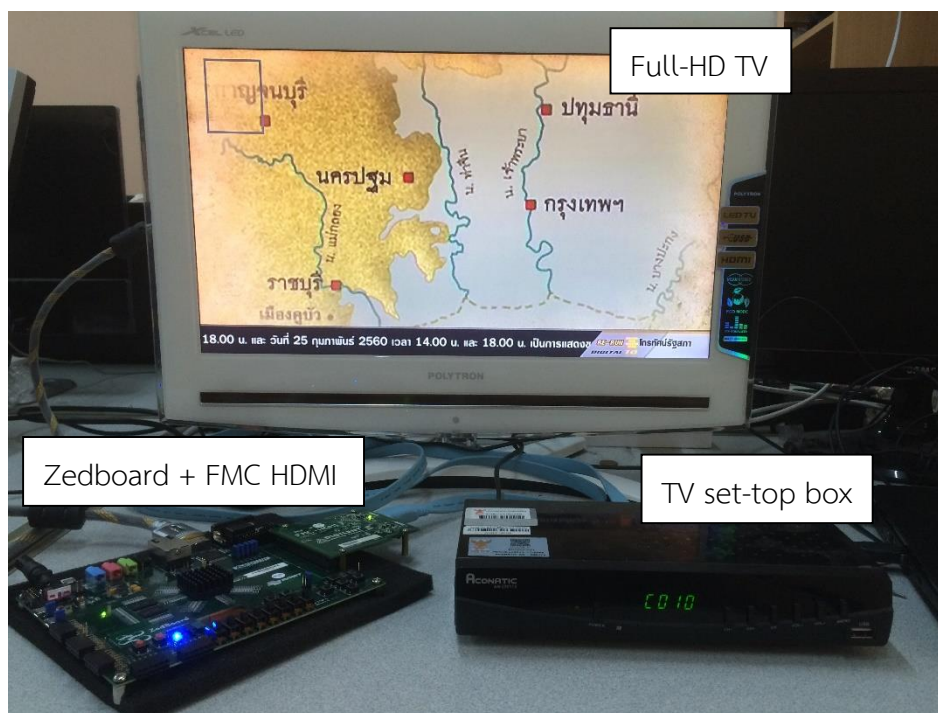
โครงสร้าง	เทคโนโลยี	ความละเอียดภาพของภาพที่ขยายแล้ว	ความถี่จุดภาพ (MHz)	เวลาในการประมวลผล	Scale factor	Block RAMs	DSP48Es
งานวิจัยนี้	FPGA Zynq-7000	742x742	148.5	16.7 ms	2 (คงที่)	7	0
Nuno-Maganda [21]	FPGA Virtex-II PRo	640x480	100	3.5 ms	สามารถปรับค่าได้	28	0
Mahale [22]	FPGA Virtex-6	2560x1920	75	17 ms	สามารถปรับค่าได้	78	64
Lin [9]	TSMC 0.13- μ m	N/A	279	N/A	สามารถปรับค่าได้	N/A	N/A

ในการโปรแกรมการทำงานของ processing system (PS) หรือ dual-core ARM cortex A9 นั้น ได้ใช้ซอฟต์แวร์ Xilinx Software Development Kit (XSDK) ซึ่งเป็นตัวโปรแกรมมีพื้นฐาน

มาจากโปรแกรม Eclipse และ CDT ใช้ภาษา C ในการเขียนโปรแกรม และสามารถใช้งานร่วมกับ Vivado ได้ ทำให้การออกแบบร่วมกันระหว่าง PS และ FPGA ทำได้ง่ายขึ้น

โดยโปรแกรมที่เขียนให้กับ PS จะทำหน้าที่ตั้งค่าเริ่มต้นให้กับ ตัวรับสัญญาณ HDMI และ ตัวส่งสัญญาณ HDMI ผ่าน IIC บัส

5.2 การทดลองใช้งานเครื่องขยายภาพภาษามือ



รูปที่ 5.2 การติดตั้งการทดลองเครื่องขยายล่ามภาษามือ

การติดตั้งการทดลองเครื่องขยายล่ามภาษามือแสดงอยู่ในรูปที่ 5.2 ซึ่งประกอบไปด้วยโทรทัศน์ที่รองรับภาพความละเอียดสูงเต็มรูปแบบ (Full-HD) ที่ทำหน้าที่เป็นจอแสดงผล set-top box ของโทรทัศน์ทำหน้าที่เป็นแหล่งกำเนิดภาพเคลื่อนไหวที่ให้ความละเอียดภาพที่ความละเอียดสูงเต็มรูปแบบ และบอร์ดทดลอง zedboard ที่ติดตั้งร่วมกับ FMC HDMI

สำหรับวงจรขยายภาพล่ามภาษามือรุ่นทดลองนี้ ถูกออกแบบให้รองรับการขยายภาพล่ามภาษามือปรับได้จาก 115x115 จุดภาพ ถึง 371x371 จุดภาพโดยสามารถปรับขนาดให้เพิ่มหรือลดได้ทั้งแนวนอนและแนวตั้งได้ทีละ 16 จุดภาพ และผู้ใช้งานสามารถขยับกรอบภาพการขยายเคลื่อนไปทั่วทั้งจอภาพโทรทัศน์

วงจรขยายภาพล่ามภาษามือได้ถูกทดลองกับรายการโทรทัศน์ที่มีภาพล่ามภาษามือต่างกัน ในรายการโทรทัศน์ที่ต่างกัน โดยภาพล่ามภาษามือมีขนาดดังนี้ 187x243 จุดภาพ 219x243 จุดภาพ

235x251 จุดภาพ และ 259x299 จุดภาพ สำหรับสัญญาณภาพระดับความละเอียดสูงเต็มรูปแบบ (Full-HD: 1080p) ที่ความถี่เฟรม 60 Hz โดยผลที่ได้ถูกแสดงอยู่ในรูปที่ 5.3 ถึงรูปที่ 5.6 จะเห็นว่า ภาพภาษามือที่ถูกขยาย 2 เท่ามีคุณภาพค่อนข้างดีและสามารถรับชมได้ง่ายขึ้น และการทำงานของ วงจรทำไปได้ราบรื่นไม่มีการสะดุด



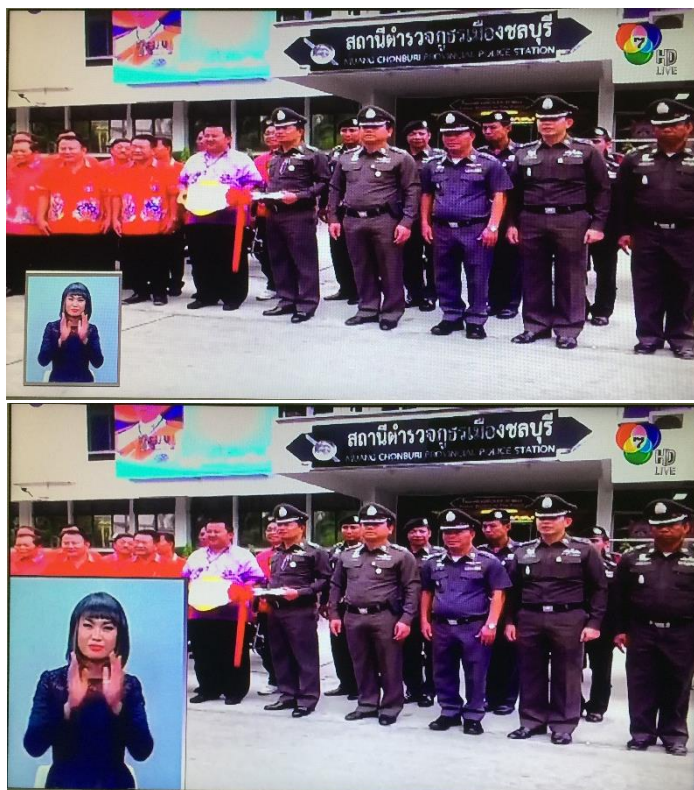
รูปที่ 5.3 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ามภาษามือขนาด 187x243 จุดภาพ



รูปที่ 5.4 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ามภาษามือขนาด 219x243 จุดภาพ



รูปที่ 5.5 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ามภาษามือขนาด 235x251 จุดภาพ



รูปที่ 5.6 ภาพรายการก่อน-หลังการขยายภาพที่มีภาพล่ามภาษามือขนาด 259x299 จุดภาพ

5.3 สรุปผล

งานวิจัยนี้นำเสนอการออกแบบวงจรที่สามารถขยายภาพภาษามือเป็นขนาด 2 เท่าได้โดยสามารถเลือกตำแหน่งและปรับขนาดของกรอบการขยายภาพที่ต้องการขยายได้ จากขนาดกรอบ 115x115 จุดภาพ ถึง 371x371 จุดภาพ โดยปรับเพิ่มหรือลดได้ที่ละแนวตั้งแนวนอนได้ที่ละ 16 จุดภาพ โดยใช้วิธีการประมาณค่าในช่วงแบบไบคิวบิกในการขยายภาพที่ให้ภาพที่คุณภาพดี และได้ประโยชน์จากการลดความซับซ้อนในการคำนวณของการขยายภาพ 2 เท่าที่ทำให้การออกแบบฮาร์ดแวร์มีความซับซ้อนน้อยและไม่เปลืองทรัพยากร โดยวงจรขยายภาพล่ามภาษามือรุ่นทดลองได้ทำการออกแบบบนบอร์ดทดลอง Zedboard โดยวงจรดังกล่าวสามารถทำงานได้ดีแบบตามเวลาจริง (real-time) กับสัญญาณภาพความละเอียดสูงเต็มรูปแบบ (1920x1080 จุดภาพ) ที่ความถี่เฟรม 60 Hz ซึ่งหมายความว่าสามารถคำนวณได้เสร็จภายในเวลา 16.7 ms และสามารถใช้งานได้กับโทรทัศน์ที่มีช่อง HDMI เท่านั้น

วงจรขยายภาพล่ามภาษามือนี้นี้จะช่วยให้ผู้พิการทางการได้ยินสามารถเข้าใจและมีความสุขกับเนื้อหาของรายการโทรทัศน์มากขึ้นโดยไม่ไปรบกวนผู้ใช้งานโทรทัศน์ปกติ

5.4 ข้อเสนอแนะ

ผู้จัดทำได้นำวงจรรขยายภาพล่ามภาษามือรุ่นทดลองไปให้ผู้พิการทางการได้ยินที่สมาคมคนหูหนวกแห่งกรุงเทพมหานครได้ทดลองใช้ โดยได้ให้ข้อเสนอแนะมาดังนี้

- ภาพภาษามือมีขนาดที่อยู่ในระดับที่พอใจแล้ว
- ราคาของวงจรรขยายภาพภาษามือยังมีราคาที่สูงเกินไป

ทั้งนี้สำหรับการลดราคาต้นทุนของภาพล่ามภาษามืออาจทำได้โดยการใช้บอร์ดทดลองที่มีส่วนขาเข้าขาออกที่ต้องการใช้เท่านั้นเช่นบอร์ดทดลองที่มีเพียงช่องขาเข้าขาออก HDMI เป็นต้นหรืออาจสามารถลดต้นทุนได้โดยการใช้บอร์ดที่ออกแบบมาเฉพาะที่จะใช้เพียงองค์ประกอบที่ต้องการใช้จริงๆก็จะสามารถช่วยราคาต้นทุนได้มาก



รายการอ้างอิง

1. Maxfield, C., *FPGAs: World Class Designs*. 2009: Elsevier Science.
2. Instruments, N. *Anatomy of a Video Signal*. [cited 2016 Dec]; Available from: <http://www.ni.com/white-paper/3020/en/>
3. *High-Definition Multimedia Interface Specification 1.3a*. 2006 [cited 2016 October]; Available from: <http://www.hdmi.org/manufacturers/specification.aspx>.
4. *HDMI :: Manufacturer :: HDMI 1.4*. . [cited 2016 October]; Available from: http://www.hdmi.org/manufacturer/hdmi_1_4/.
5. *ITU-R BT.656-4 standard* [cited 2016 Dec]; Available from: https://www.itu.int/dms_pubrec/itu-r/rec/bt/.
6. *ITU-R BT.601-5 standard* [cited 2016 Dec]; Available from: https://www.itu.int/dms_pubrec/itu-r/rec/bt/.
7. Jain, A.K., *Fundamentals of digital image processing*. Prentice-Hall information and system sciences series; Prentice-Hall information and system sciences series. 1989, Englewood Cliffs, NJ :: Prentice Hall.
8. Peters, R.A. *EECE 4353 Image Processing Lecture Notes: Resizing Images*. [cited 2016 Dec]; Available from: https://archive.org/details/Lectures_on_Image_Processing.
9. Chung-chi, L., et al. *The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing*. in *2008 IEEE International Symposium on Circuits and Systems*. 2008.
10. Chun-Ho, K., et al., *Winscale: an image-scaling algorithm using an area pixel model*. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003. **13**(6): p. 549-553.
11. Chung-chi, L., et al. *Real-time FPGA architecture of extended linear convolution for digital image scaling*. in *2008 International Conference on Field-Programmable Technology*. 2008.

12. Chen, S.L., *VLSI Implementation of an Adaptive Edge-Enhanced Image Scalar for Real-Time Multimedia Applications*. IEEE Transactions on Circuits and Systems for Video Technology, 2013. **23**(9): p. 1510-1522.
13. Chen, P.Y., C.Y. Lien, and C.P. Lu, *VLSI Implementation of an Edge-Oriented Image Scaling Processor*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009. **17**(9): p. 1275-1284.
14. Crockett, L.H., et al., *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. 2014: Strathclyde Academic Media. 484.
15. ARM. *AMBA AXI and ACE Protocol Specification*. AXI3, AXI4, and AXI4-Lite ACE and ACE-Lite [cited 2016 Dec]; Available from: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ih0022d/index.html>.
16. Devices, A. *ADV7511: 225 MHz, High Performance HDMI Transmitter with ARC Data Sheet*. [cited 2016 Dec]; Available from: <http://www.analog.com/media/en/technical-documentation/data-sheets/>
17. N.p., D. *HDMI: Dual HDMI Input Expansion Card*. [cited 2016 Dec]; Available from: <http://store.digilentinc.com/fmc-hdmi-dual-hdmi-input-expansion-card/>
18. Devices, A. *ADV7611: Low Power 165 MHz HDMI Receiver Datasheet* [cited 2016 Dec]; Available from: <http://www.analog.com/media/en/technical-documentation/data-sheets/>
19. Incorporation, X. *AXI IIC Bus Interface v2.0 LogiCORE IP Product Guide* [cited 2016 Dec]; Available from: https://www.xilinx.com/products/intellectual-property/axi_iic.html.
20. Ltd., A.T.S. *ZedBoard Hardware User's Guide*. 2014 [cited 2016 Dec]; Available from: <http://zedboard.org/>.
21. Nuno-Maganda, M.A. and M.O. Arias-Estrada. *Real-time FPGA-based architecture for bicubic interpolation: an application for digital image scaling*. in *2005 International Conference on Reconfigurable Computing and FPGAs (ReConFig'05)*. 2005.

22. Mahale, G., et al. *Hardware architecture of bi-cubic convolution interpolation for real-time image scaling*. in *2014 International Conference on Field-Programmable Technology (FPT)*. 2014.





ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาคผนวก ก
บทความที่ได้ตีพิมพ์ใน ECTI-CON 2017



A Design of Sign Video Image Expander for HDMI Source using Bicubic Interpolation

Auangkun Rangsikunpum, Ekachai Leclaramsee, Suree Pumrin
Embedded System and IC Design (ESID) Research Laboratory,
Department of Electrical Engineering, Chulalongkorn University,
Bangkok, Thailand
auangkun.r@gmail.com, ekachai.l@chula.ac.th, suree.p@chula.ac.th

Abstract— This paper describes a device for real-time expanding sign language images inserted in any TV programs by 2×2 scale factor. Bicubic interpolation is used as an image expansion method. The 2×2 scale factor simplifies the bicubic interpolation formula to just division by two operations which enables an efficient hardware implementation. As a prototype version, the proposed architecture is implemented on a Xilinx Zynq board with a Zynq-7020 SoC. The location and size of the expanding area can be adjusted to capture the sign language images using its onboard push buttons. Experimental results show that the proposed implementation can be compatible with 1080p (1920 \times 1080) 60 Hz HDMI video source. Target users are the deaf people who cannot clearly see the existing sign language images because of its small size.

Keywords—sign language, HDMI, Zynq, bicubic interpolation, deaf person, image expander

I. INTRODUCTION

Over 360 million people in the world [1] suffer from hearing loss. 280,000 of them are in Thailand [2]. These people use sign language for their daily communications. Thailand's regulation states that some television programs, such as news, must provide a sign language window to help their understanding. This sign language is usually inserted in a small square on one corner of the screen in order to avoid disrupting with normal viewers. Unfortunately, for many deaf people, this is still a problem in approaching the information as the sign language is too small to view.

The first attempt to alleviate this deaf problem was proposed [3] with a development of a TV set top box called Sign Image Expander that can enlarge the sign language image. It provides 9 selectable image scaling factors ranging from 1×1 to 2×2 using a combination of bilinear interpolation and nearest neighborhood techniques. However, it is only compatible with analog video signal which normally supports only low resolution video source.

Several image scaling methods can be found or proposed in the literature. For hardware-based real-time application, a trade-off between computation complexity and the quality of interpolation must be considered. The simplest method which results in an acceptable scaled image should be chosen. Although nearest neighbor [4] and bilinear interpolation [5] used in [3] are simple methods, they return resized image with

full of blocking and blurring effects. Bicubic interpolation [6] gives high quality images but takes lots of computation. To meet real-time requirements, some interpolation methods [7]-[9] have been proposed for VLSI implementations. However, they obtain worse image quality than bicubic method.

Therefore, this paper presents an attempt to improve the Sign Image Expander to get better results by using bicubic interpolation for 2×2 image expansion and be suitable with HDMI TV system. In this paper, the Sign Image Expander is an auxiliary device that enlarges only fixed 2×2 times of the original images and can be attached to any HDMI TV system using HDMI cable as shown in Fig. 1. The position and size of enlarging image window can be adjusted to support the variety of the sign language window on each television program. A prototype design is compatible with Full-HD (1080p) streaming source at 60 Hz frame rate.

The remainder of this paper is structured as follows: Section II describes the bicubic interpolation method and its implementation. Section III presents the design of Sign Image Expander which are application design and hardware implementation. Section IV and V shows the experimental results and concludes the work, respectively.

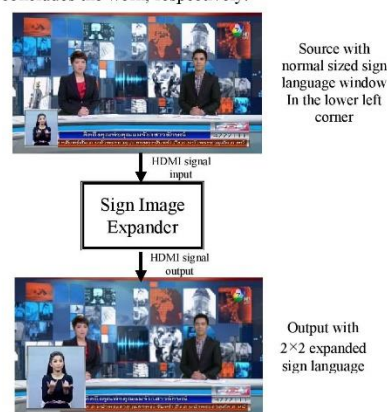


Figure 1. Conceptual design of the sign image expander.

II. BICUBIC INTERPOLATION METHOD

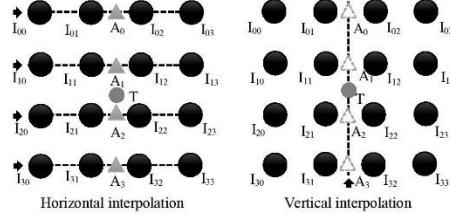


Figure 2. Bicubic interpolation at a pixel T can be carried out by 4 horizontal interpolations followed by one vertical interpolation

A method of reducing computation of bicubic interpolation [10] is selected as a 2×2 image expansion because it needs low computation cost and also produce high quality resized images. Bicubic interpolation uses 16 adjacent source pixels to generate an interpolated pixel. Each interpolated pixel requires five 1-dimensional bicubic interpolations consisting of four horizontal interpolations followed by one vertical interpolation. The interpolated pixel T is calculated using its 4×4 neighbor source pixels as shown in Fig. 2. Four horizontal interpolations, given in (1), are executed first to produce four artifact pixels A_0 , A_1 , A_2 , and A_3 according to the following formula

$$A_i = \sum_{j=0}^3 I_{ij} \times ch_j \quad i = 0, 1, 2, 3 \quad (1)$$

where ch_j are a horizontal coefficient of I_{ij} . These artifact pixels are used to calculate T using vertical interpolation given as

$$T = \sum_{i=0}^3 A_i \times cv_i \quad (2)$$

where cv_i are a vertical coefficient of A_i .

In 1-dimension interpolation, the calculating method of horizontal coefficients and vertical coefficients are identical. According to Fig. 3, 1-dimension bicubic interpolation equation can be written as

$$A = c_0 \times I_0 + c_1 \times I_1 + c_2 \times I_2 + c_3 \times I_3 \quad (3)$$

where c_0 , c_1 , c_2 , and c_3 , which are weighting coefficients of the source pixels I_0 , I_1 , I_2 , and I_3 , are defined in (4).

$$\begin{aligned} c_0 &= -[(1-s) \times (1-s) \times s] \\ c_1 &= (1-s) + [(1-s) \times (1-s) \times s] \\ c_2 &= s + [(1-s) \times s \times s] \\ c_3 &= -[(1-s) \times s \times s] \end{aligned} \quad (4)$$

where s is the distance between the source pixel I_1 and the interpolated pixel A .

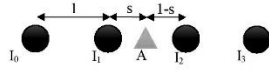


Figure 3. One-dimension bicubic interpolation

For a 2×2 enlarge factor, s and $1-s$ in (4) equal to 0.5 and the formulae in (3) reduces to (5) indicating that only division by 2 or multiplication by 0.5 are involved.

$$\begin{aligned} A &= -(0.5 \times 0.5 \times 0.5)I_0 + (0.5 + (0.5 \times 0.5 \times 0.5))I_1 \\ &\quad + (0.5 + (0.5 \times 0.5 \times 0.5))I_2 - (0.5 \times 0.5 \times 0.5)I_3 \end{aligned} \quad (5)$$

In binary number, multiplying by 0.5 is equivalent to right shift operation. Therefore, its hardware implementation, called the 1-D Interpolation Unit (1DIU), can be simply designed as three-state pipelined architecture without any multipliers as shown in Fig. 4.

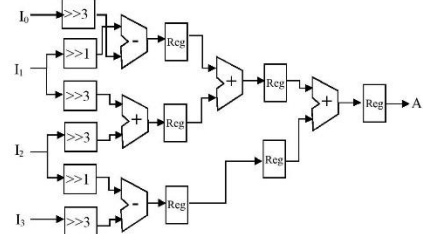


Figure 4. The 1-D Interpolation Unit (1DIU)

III. THE SIGN IMAGE EXPANDER PROTOTYPE

The design of the prototype aims to be compatible with HDMI interface Full-HD video sources. Moreover, the designed prototype can support the sign language originally displayed in various sizes and locations but the scale factor always remain the same at 2×2 .

A. Application Design

After the Sign Image Expander is installed in television system, a grey square window will be displayed on the top-left of the screen in order to show the boundary of the captured area for expansion as shown in Fig. 5. To adjust the size of the window, users can use the push button P0. Moreover, pushing push buttons P1, P2, P3 and P4 are to move the window up, left, right and down, respectively. Then, users can turn on dip switch to start enlarging the sign language images.

B. Hardware Implement

Zedboard is chosen for a main board of the Sign Image Expander prototype because its onboard Zynq-7020 SoC device which consists of a Dual ARM Cortex-A9 CPU based processor system (PS) and Artix-7 programmable logic (PL) or FPGA can enable real-time image enlargement. The PS which also includes memory controller for 512 MB DDR3 memory access can communicate with the PL through AXI bus interface. A FMC HDMI module is used for streaming the video data from the HDMI input port to the Zedboard.

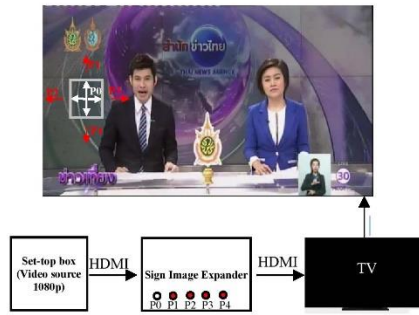


Figure 5. Application of the Sign Image Expander using push buttons

As shown in Fig. 6, there are mainly two parts in the Sign Image Expander. The first part is FMC HDMI module used as a HDMI receiver. It decodes HDMI signal into 16-bit YCbCr with video timing signals hsync, vsync and data enable signals. The second part is the Zedboard which is programmed to function as Expansion controller, HDMI transmitter, DDR3 memory interface and I/O interfaces for push buttons and dip switches. The functions of the Expansion controller are enlarging the sign language image using DDR3 memory as an expanded image buffer and programming the HDMI receiver and HDMI transmitter through IIC bus. The HDMI transmitter converts the video signal (16-bit YCbCr and video timing signals) back into HDMI signal.

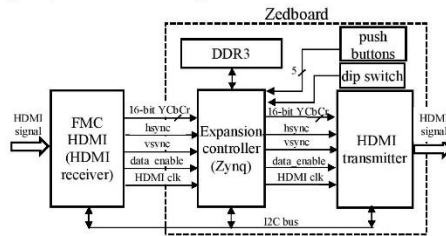


Figure 6. Block diagram of the Sign Image Expander

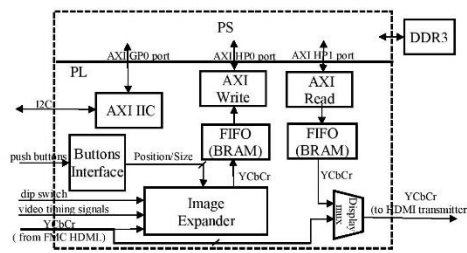


Figure 7. The architecture of the Expansion controller

The architecture of Expansion controller on Zynq is shown in Fig. 7. The PS is only responsible to initialize the FMC HDMI (HDMI receiver) and HDMI transmitter. Besides, the PL or FPGA performs the image expansion and feeds/receives the enlarged images to/from DDR3 memory.

Inside the PL, the images are first expanded by Image Expander and stored in DDR3 memory. The DDR3 memory is allocated into two separate blocks and the data flow is shown in Fig. 8. As the expanded image of the n^{th} video frame (E_n) is being written into one DDR3 block, the previous expanded image (E_{n-1}), which has already been stored in the another DDR3 block, is read and inserted in the current video frame. Therefore, the expanded image is displayed one frame delay compared to original video frame. The image data transmission to/from DDR3 is performed by AXI Write and AXI Read which act as an AXI master for AXI bus protocol. Since the AXI bus protocol uses handshaking signal in each transaction, FIFO data buffers (BRAM) are added to temporarily store incoming pixel data while handshaking process occurs. A display mux, which is a 2:1 data multiplexer, selects when the expanded pixel data or the incoming video data should be fed to the HDMI transmitter.

The hardware architecture of the image expansion of the Image Expander is shown in Fig. 9. It implements the algorithm described in Fig. 2. Inside the Image Expander, the source pixel data stream from FMC HDMI module is received line by line and stored in four registers serially in a 4-pixel buffer block. These 4 pixel data are fed simultaneously into 1DIU to perform horizontal interpolation according to Fig. 4, the result of which is called artifact pixel and is stored in the 4-line buffer block (BRAMs). To perform vertical interpolation according to (2), the artifact pixel data are read into another 1DIU. The interpolated output as well as the 4 artifact data go into a 5:1 multiplexer to select appropriate data associated with the 2x2 enlarged image and store in the DDR3 memory. This expansion mechanism is executed separately with each color component (8-bit Y and 8-bit CbCr).

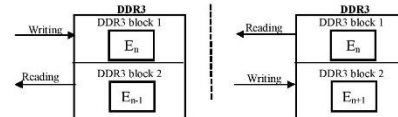


Figure 8. Data flow diagram for DDR3 usage during the n^{th} and $(n+1)^{th}$ video frame

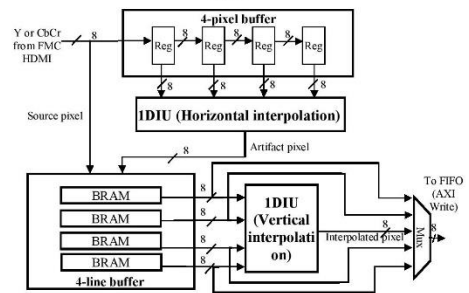


Figure 9. The architecture of the image expansion of the Image Expander

IV. EXPERIMENTAL RESULTS

The experimental setup, shown in Fig. 10, consists of Zedboard, FMC HDMI module, a set-top box and HDMI interface TV. The prototype Sign Image Expander is tested with four different TV channels where its enlarging sign area are of different sizes, i.e. 187×243 , 219×243 , 235×251 and 259×299 , and locate at different position on the screen. Here the push buttons are used to correctly identify the sign window to the expander and the screen outputs are shown in Fig. 11. It can operate well with 1080p 60 Hz video source. Obviously, the expanded sign language windows are much easier to see.

Table I highlights resource utilization of each unit inside the Sign Image Expander.

TABLE I. RESOURCE UTILIZATION

Unit name	Register	LUTs	BRAMs
AXI IIC	736	768	-
Buttons Interface	133	248	-
Image Expander	285	401	4
AXI Write	372	247	2
AXI Read	207	157	1
Others	1464	1042	-

V. CONCLUSION

In this paper, the design and hardware implementation of the Sign Image Expander that enlarges the sign language window in any TV programs is proposed. Because a scale factor of 2×2 is used with bicubic interpolation, the hardware implementation of an image expansion method can be significantly simplified. The prototype design is implemented on Xilinx Zynq board. It can operate in real-time with Full-HD at 60 frame per second video source. Users can locate and resize the expanding area as their needs. This device can benefit TV deaf viewer without interfacing with normal people.

ACKNOWLEDGMENT

This research has been supported by Embedded System and IC Design (ESID) Research Laboratory and Electrical Engineering Chulalongkorn University, Master Honours Program Scholarship.

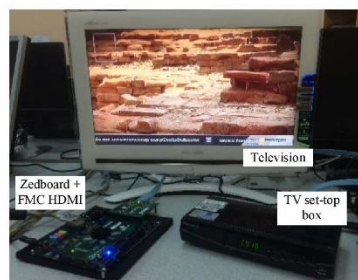


Figure 10. The whole experimental setup



Figure 11. The output images with 2×2 sign language image expansion in four news programs having different sign language image sizes and displayed at different locations.

REFERENCES

- [1] World Health Organization (WHO) <http://www.who.int/mediacentre/factsheets/fs300/en/>
- [2] Official Statistics Thailand. <http://service.nso.go.th/nso/web/survey/surpop2-5-4.html>
- [3] E. Leelaramee, "A TV sign image expander with built-in closed caption decoder," in *IEEE Transactions on Consumer Electronics*, vol. 51, no. 2, pp. 682-687, May 2005.
- [4] T.M. Lehmann, C. Gonner, K. Spitzer, "Survey: interpolation methods in medical image processing," *IEEE Trans. Medical Imaging*, vol. 18, no. 11, pp. 1049-1075, Nov. 1999.
- [5] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolation methods for image resampling," *IEEE Trans. Med. Imag.*, vol. MI-2, no. 3, pp. 31-39, Sep. 1983.
- [6] C. C. Lin, M. H. Sheu, H. K. Chiang, W. K. Tsai, and Z. C. Wu, "Real-time FPGA architecture of extended linear convolution for digital image scaling," in *Proc. IEEE Int. Conf. Field-Program. Technol.*, 2008, pp. 381-384.
- [7] C. H. Kim, S. M. Seong, J. A. Lee, and L. S. Kim, "Winscale: An imagescaling algorithm using an area pixel model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 549-553, Jun. 2003.
- [8] P. Y. Chen, C. Y. Lien, and C. P. Lu, "VLSI implementation of an edge-oriented image scaling processor," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 9, pp. 1275-1284, Sep. 2009.
- [9] C. C. Lin, M. H. Sheu, H. K. Chiang, W. K. Tsai, and Z. C. Wu, "Real-time FPGA architecture of extended linear convolution for digital image scaling," in *Proc. IEEE Int. Conf. Field-Program. Technol.*, 2008, pp. 381-384.
- [10] C. C. Lin, M. H. Sheu, H. K. Chiang, C. L. Liaw, and Z. C. Wu, "The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing," *2008 IEEE International Symposium on Circuits and Systems*, Seattle, WA, 2008, pp. 480-483.

ประวัติผู้เขียนวิทยานิพนธ์

นายเอื้ออังกูร รังสิกรรพุม เกิดวันที่ 13 ธันวาคม พ.ศ. 2535 ที่จังหวัดนครสวรรค์ สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2557 และต่อมาได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า แขนงวิชาการออกแบบและประยุกต์ วงจรรวม ที่คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2558

