

การออกแบบชิปทีวีไมโครคอนโทรลเลอร์ที่สามารถแสดงข้อความภาษาไทย-อังกฤษ

A DESIGN OF A TV MICROCONTROLLER CHIP THAT CAN DISPLAY THAI-ENGLISH TEXTS

อัยยงค์ แทนสติย์ และ เอกชัย ลีลาธรรม์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย ถนนพญาไท กรุงเทพฯ 10330

โทร. 218-6488 โทรสาร 218-6488

บทคัดย่อ

บทความนี้อธิบายถึงสถาปัตยกรรมภายในของทีวีไมโครคอนโทรลเลอร์ที่ออกแบบด้วย VHDL และสังเคราะห์ลงบนชิป FPGA ชิปนี้จะประกอบไปด้วยหน่วยประมวลผลกลางขนาด 8 บิต, ตัวควบคุมการแสดงผลบนหน้าจอ, ตัวประมวลผลสัญญาณขึ้นต้นจากเครื่องควบคุมระยะไกล, ตัวมอดูเลตความกว้างพัลส์และทางเข้าออก (I/O พอร์ต) จำนวนหนึ่ง ทีวีไมโครคอนโทรลเลอร์ที่ออกแบบขึ้นมานี้จะใช้ชิปภายนอกอีกเพียง 3 ชิป คือ หน่วยความจำอ่านอย่างเดียว, หน่วยความจำเข้าถึงแบบสุ่ม และวงจรแปลงสัญญาณเชิงอุปมานเป็นเชิงเลข เพื่อให้สามารถจะควบคุมหน้าที่การทำงานต่างๆ ของเครื่องรับโทรทัศน์ได้เหมือนดังเช่นชิปทีวีไมโครคอนโทรลเลอร์ที่มีความสามารถสูงและผลิตในเชิงพาณิชย์ นอกจากนี้แล้ววงจรควบคุมการแสดงผลบนหน้าจอได้ถูกออกแบบขึ้นเป็นพิเศษเพื่อรองรับการแสดงข้อความภาษาไทยและภาษาอังกฤษ

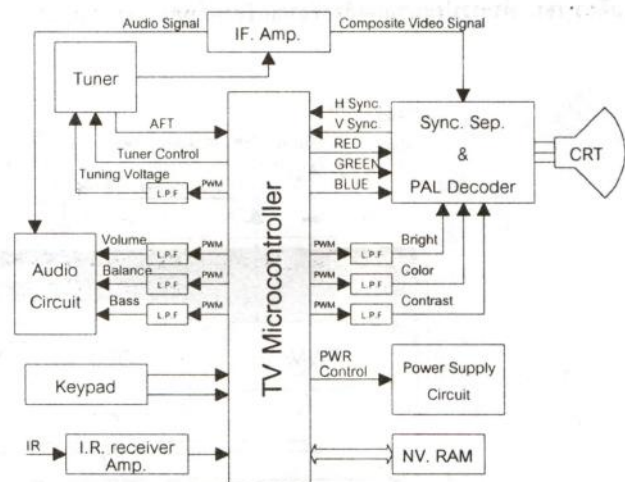
Abstract

This paper describes the internal architecture of a TV microcontroller designed by using VHDL and synthesized on an FPGA chip. The chip consists of an 8-bit CPU, an on screen display (OSD) controller, a remote control signal preprocessor, several pulse width modulators and I/O ports. Only 3 more external chips, i.e. ROM, RAM and A/D are required to make this microcontroller perform various TV control functions that are common in many advanced and commercialized TV microcontroller chips. Moreover, the OSD controller is specially designed to support the display of Thai and English text.

บทนำ

ทีวีไมโครคอนโทรลเลอร์เป็นวงจรรวม (IC) ชิปเดี่ยวที่ถูกรออกแบบให้ใช้สำหรับควบคุมการทำงานที่สำคัญๆ ของเครื่องรับโทรทัศน์ ได้แก่ การปรับช่องสถานี, การปรับภาพ และเสียง, การรับคำสั่งจากเครื่อง

ควบคุมระยะไกล (Remote Control) หรือปุ่มกดที่ตัวเครื่อง (Key Pad) และการสร้างเมนู (Menu) บนจอภาพ เป็นต้น ดังแสดงในรูปที่ 1



รูปที่ 1 โครงสร้างภายในเครื่องรับโทรทัศน์ที่ใช้ทีวีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานภายใน

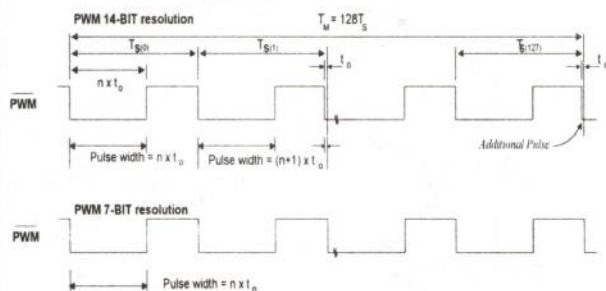
ข้อดีของการใช้ทีวีไมโครคอนโทรลเลอร์เป็นตัวควบคุมวงจรต่างๆ ภายในโทรทัศน์ คือ ทำให้แผงวงจรภายในเครื่องรับโทรทัศน์มีขนาดเล็ก และสามารถนำไปใช้กับเครื่องรับโทรทัศน์ที่มีประสิทธิภาพหรือมีวงจรย่อยต่างกันออกไปได้ โดยแก้ไขเพียงโปรแกรมควบคุมที่เก็บไว้ในตัวชิป ปัจจุบันจึงมีผู้ผลิตทีวีไมโครคอนโทรลเลอร์ออกมาหลายราย [1-4] แต่เป็นแบบที่ใช้หน่วยความจำชนิดโปรแกรมได้ครั้งเดียว (OTP) เพื่อเก็บโปรแกรมควบคุมและรูปแบบอักษร (Font) ไว้ภายในตัว จึงไม่เหมาะที่จะนำมาใช้ในงานวิจัยพัฒนาที่ต้องแก้ไขโปรแกรมควบคุมหลายครั้ง ยิ่งกว่านั้นทีวีไมโครคอนโทรลเลอร์เหล่านี้ไม่สามารถสนับสนุนการแสดงผลข้อความ (Text) ภาษาไทย แบบ 3 ระดับ คือ พยัญชนะ, สระบน และสระล่างได้

ดังนั้นบทความนี้จึงนำเสนอการออกแบบทีวีไมโครคอนโทรลเลอร์ ที่สามารถทำงานได้เช่นเดียวกับที่มีการผลิตในเชิงพาณิชย์ แต่ใช้หน่วยความจำภายนอกสำหรับเก็บโปรแกรมควบคุม และ

ไมโครคอนโทรลเลอร์ต้องมีการประมวลผลสัญญาณเชิงอุปมานด้วย ดังนั้นจึงต้องมีวงจรนี้เป็นส่วนเพิ่มเติมขึ้นมา

ตัวกำเนิดสัญญาณมอดูเลตความกว้างพัลส์

ตัวกำเนิดสัญญาณมอดูเลตความกว้างพัลส์ มีหน้าที่สร้างสัญญาณมอดูเลตความกว้างพัลส์ ซึ่งสัญญาณมีลักษณะดังรูปที่ 3 โดยสัญญาณมอดูเลตความกว้างพัลส์ที่ใช้กับแต่ละวงจรจะต่างกันที่ความละเอียดในการแปลงสัญญาณ คือ การควบคุมภาพ และการควบคุมเสียง จะใช้ที่ความละเอียดค่าเพียง 7 บิตเท่านั้น แต่สำหรับการปรับช่องสถานี ซึ่งต้องการความละเอียดสูงจึงใช้ที่ 14 บิต โดยที่ความละเอียด 14 บิตนี้ ได้ใช้วิธี Special distributed pulses technique [5] ซึ่งเป็นวิธีกระจายพัลส์ขนาดกว้างให้เป็นพัลส์เล็กหลายลูก ทำให้มีองค์ประกอบความถี่สูงเพิ่มขึ้น ดังนั้นวงจรกรองผ่านต่ำที่ใช้สำหรับกรองความถี่สูงออก จึงไม่ต้องแม่นยำมากเท่าวิธีปกติ โดยสัญญาณที่ความละเอียด 14 บิต จะมีคาบนานเป็น 128 เท่าของคาบแบบ 7 บิต (T_M) ดังแสดงในรูปที่ 3 และจะถูกจัดแบ่งเป็น 7 บิตบนกับ 7 บิตล่าง โดย 7 บิตบนจะกำหนดความกว้างพัลส์ที่ความละเอียด 7 บิต ส่วน 7 บิตล่างจะเป็นตัวกำหนดตำแหน่งของพัลส์เพิ่มเติม (Additional Pulse) ที่คาบต่างๆ ใน 128 คาบย่อย (T_S) ซึ่งตำแหน่งที่จะเกิดขึ้นนั้น จะเป็นดังแสดงในตารางที่ 1

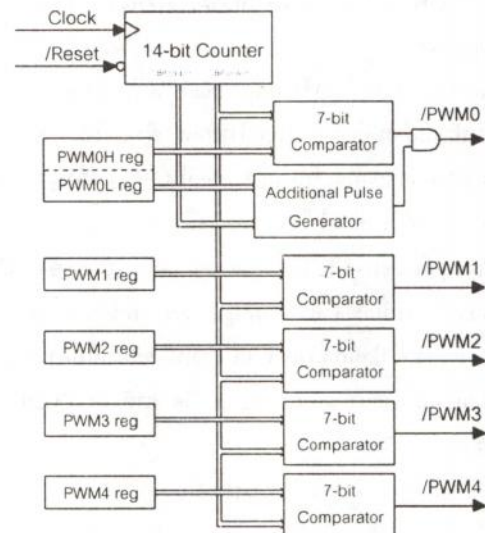


รูปที่ 3 สัญญาณมอดูเลตความกว้างพัลส์ (PWM)

ตำแหน่ง "1"	คาบย่อย (T_S) ที่จะมีพัลส์เพิ่มเติม
บิตที่ 0	64
บิตที่ 1	32, 96
บิตที่ 2	16, 48, 80, 112
บิตที่ 3	8, 24, 40, 56, 72, 88, 104, 120
บิตที่ 4	4, 12, 20, 28, ..., 100, 108, 116, 124
บิตที่ 5	2, 6, 10, 14, 18, 22, ..., 106, 110, 114, 118, 122, 126
บิตที่ 6	1, 3, 5, 7, 9, 11, 13, ..., 117, 119, 121, 123, 125, 127

ตารางที่ 1 ตำแหน่งของคาบย่อยที่ให้มีพัลส์เพิ่มเติม

โครงสร้างของตัวกำเนิดสัญญาณ PWM จะเป็นดังรูปที่ 4 ซึ่งประกอบไปด้วยวงจรนับขนาด 14 บิต, วงจรเปรียบเทียบขนาด 7 บิต จำนวน 5 วงจร เพื่อใช้เปรียบเทียบค่าที่ได้จาก PWM รีจิสเตอร์ต่างๆ ให้เป็นสัญญาณ PWM ที่มีความละเอียด 7 บิต จำนวน 4 ช่องสัญญาณ คือ PWM1-PWM4 และตัวสร้างพัลส์เพิ่มเติม (Additional Pulse Generator) ซึ่งจะเป็นส่วนที่ให้กำเนิดพัลส์เพิ่มเติมให้กับสัญญาณ PWM0 เพื่อให้เป็นสัญญาณ PWM ที่มีความละเอียด 14 บิต



รูปที่ 4 โครงสร้างภายในของตัวกำเนิดสัญญาณ PWM

ตัวประมวลผลขั้นต้นสัญญาณจากเครื่องควบคุมระยะไกล

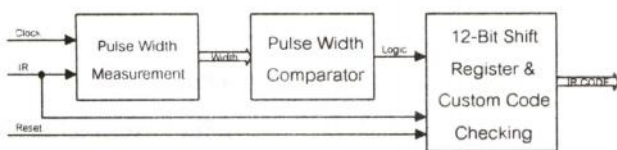
ตัวประมวลผลขั้นต้นสัญญาณจากเครื่องควบคุมระยะไกล มีหน้าที่ถอดรหัสแบบ Pulse Code จากตัวรับแสงอินฟราเรด ซึ่งรูปแบบของสัญญาณแบบ Pulse Code [6] ในรหัสที่ส่งมา 1 รหัสจะประกอบไปด้วยส่วนต่างๆ คือ

- Head pulse มีคาบนาน 6.4 มิลลิวินาที เป็นการบอกตำแหน่งเริ่มต้นของรหัส
 - Custom code จำนวน 6 บิต เริ่มจาก LSB ไป MSB ซึ่งเป็นรหัสเฉพาะของอุปกรณ์ตัวรับ เพื่อมิให้มีการรบกวนการทำงานกันของอุปกรณ์ควบคุมระยะไกลตัวอื่น
 - Data จำนวน 6 บิต เริ่มจาก LSB ไป MSB เป็นข้อมูลประจำไม่ถูกรบกวน
 - Reverse custom code และ Reverse data จำนวน 12 บิต เป็นรหัส custom code และ data ที่กลับข้อมูล ลอจิกจาก 0 เป็น 1 และจาก 1 เป็น 0 เพื่อให้มีระยะเวลาการส่งรหัส 1 รหัสใช้เวลาที่เท่ากัน
- ข้อมูลลอจิก 0 จะมีช่วงเวลาที่ เป็น 5 โวลต์ นาน 0.8 มิลลิวินาที และช่วงเวลาที่ เป็น 0 โวลต์ นาน 2.4 มิลลิวินาที ในขณะที่ข้อมูลลอจิก 1

จะมีช่วงเวลาที่ เป็น 5 โวลต์ นาน 0.8 มิลลิวินาที และช่วงเวลาที่ เป็น 0 โวลต์ นาน 0.8 มิลลิวินาที ดังนั้นการถอดรหัสข้อมูลแบบ Pulse Code นี้จึงประกอบไปด้วยส่วนต่างๆ ดังนี้ คือ

- ส่วนวัดความกว้างพัลส์ (Pulse Width Measurement) ซึ่งเป็นวงจรจับระยะเวลาคาบของรหัสแต่ละบิต โดยใช้สัญญาณนาฬิกาเป็นตัวนับ ส่วนนี้จะให้สัญญาณ Width ซึ่งเป็นข้อมูลระยะเวลาของคาบส่งต่อไปยังส่วนเปรียบเทียบความกว้างพัลส์ (Pulse Width Comparator)
- ส่วนเปรียบเทียบความกว้างพัลส์ (Pulse Width Comparator) เป็นวงจรเปรียบเทียบค่าของความกว้างพัลส์ ซึ่งถ้าระยะเวลา 1 คาบ นาน 3.2 มิลลิวินาทีจะให้ค่าที่ Logic เป็น '0' แต่ถ้าระยะเวลาคาบ นาน 1.6 มิลลิวินาทีจะให้ค่าที่ Logic เป็น '1'
- รีจิสเตอร์เลื่อนขนาด 12 บิต และตรวจ Custom Code ซึ่งจะเลื่อนค่าของ Logic ไปทีละ 1 ค่า ตามสัญญาณ IR ที่เข้ามา ซึ่งถ้า Custom Code ขนาด 6 บิตถูกต้องค่าที่ IR CODE จะให้เป็นค่าของรหัสที่ผู้ใช้กดมา แต่หากว่า Custom Code ผิด ค่าที่ IR CODE จะเป็น "000000"

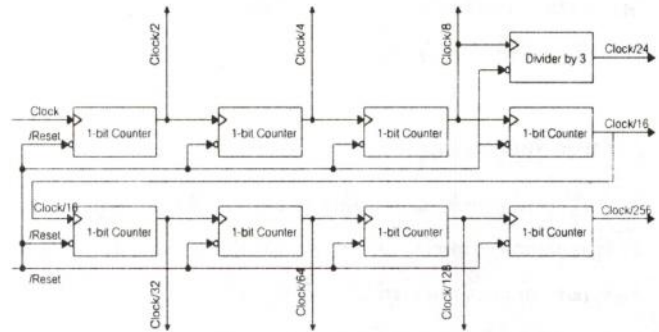
โดยมีโครงสร้างภายในดังรูปที่ 5



รูปที่ 5 โครงสร้างภายในของตัวประมวลผลขั้นต้นสัญญาณจากเครื่องควบคุมระยะไกล

ตัวกำเนิดและหารความถี่

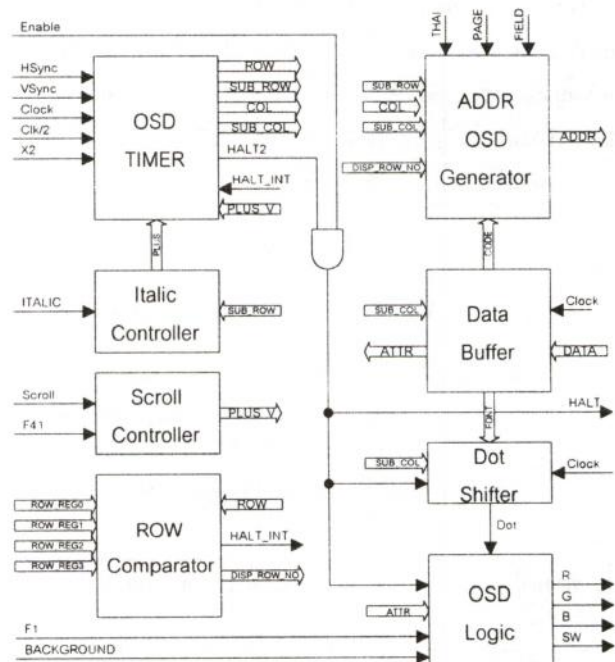
ตัวกำเนิดและหารความถี่ ตัวกำเนิดความถี่นั้นเป็นวงจรที่มีอยู่ใน Xilinx FPGA [7] อยู่แล้วเพียงแต่กำหนดค่าให้เลือกใช้งาน หรือว่าไม่ใช้งานเท่านั้น ซึ่งมีความถี่ให้เลือกใช้ คือ 8 MHz, 500 kHz, 16 kHz และ 490 Hz แต่จะเลือกใช้งานได้พร้อมกันเพียง 2 ความถี่เท่านั้น ส่วนตัวหารความถี่จะประกอบไปด้วยวงจรนับขนาด 1 บิต ต่อเรียงกันเพื่อหารความถี่ลงทีละครึ่งหนึ่ง โดยมีลักษณะโครงสร้างดังรูปที่ 6



รูปที่ 6 โครงสร้างภายในของตัวหารความถี่

ตัวแสดงผลบนหน้าจอ

เป็นส่วนที่ทำหน้าที่สร้างสัญลักษณ์ หรือตัวอักษรให้ปรากฏบนหน้าจอ โดยการส่งสัญญาณแม่สีไปยังวงจร PAL Decoder ซึ่งตัวแสดงผลบนหน้าจอมีโครงสร้างดังรูปที่ 7



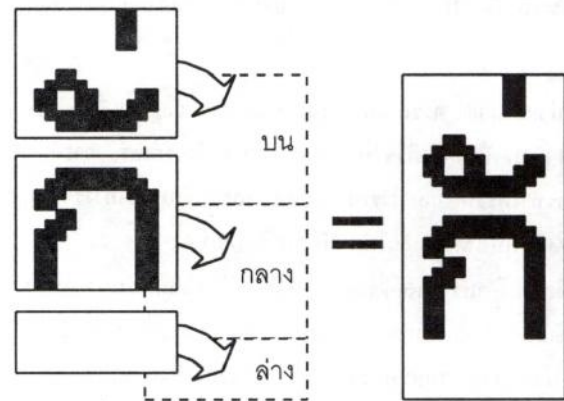
รูปที่ 7 โครงสร้างภายในของตัวแสดงผลบนหน้าจอ ซึ่งแต่ละส่วนภายในมีหน้าที่ และการทำงานดังนี้

- ตัวจับเวลาการแสดงผลบนหน้าจอ (OSD Timer) เป็นส่วนที่ทำหน้าที่บอกถึงตำแหน่งในการแสดงบนจอภาพให้กับส่วนต่างๆ ของตัวแสดงผลบนหน้าจอ
- ตัวควบคุมอักษรเอียง (Italic Controller) เป็นตัวกำหนดค่าให้การนับในแต่ละสครัมภ์ย่อยของการแสดง 1 ตัวอักษรให้เร็วหรือช้าต่างกัน ทำให้ตำแหน่งที่แสดงเลื่อนไปทางซ้าย และขวา เกิดเป็นตัวอักษรเอียงขึ้นมา

- ตัวควบคุมการเลื่อน (Scroll Controller) ใช้สำหรับควบคุมการเลื่อนขึ้นของบรรทัด ซึ่งการทำงานของตัวควบคุมการเลื่อนนี้คล้ายกับตัวควบคุมอักษรเอียง คือ ทำการกำหนดค่าในการนับตำแหน่งของบรรทัดย่อยให้เร็วขึ้นกว่าปกติ ซึ่งจะทำให้ตัวอักษรที่แสดงอยู่ปรากฏที่ตำแหน่งสูงขึ้นจากเดิมตามค่าที่กำหนดให้
- ตัวเปรียบเทียบตำแหน่งบรรทัด (Row Comparator) เป็นส่วนคอยเปรียบเทียบค่าจากรีจิสเตอร์ ROW0 ถึง ROW3 ซึ่งเป็นรีจิสเตอร์บอกตำแหน่งของการแสดงบรรทัดบนจอภาพ ซึ่งหากค่าบรรทัดจากตัวจับเวลาบนหน้าจอตรงกันกับรีจิสเตอร์ ROW ใดๆ เพียง 1 รีจิสเตอร์เท่านั้น จะทำให้ตัวแสดงผลบนหน้าจอทำงาน
- ตัวกำเนิดตำแหน่งที่อยู่ (Address OSD Generator) เป็นส่วนให้ตำแหน่งที่อยู่ในหน่วยความจำของข้อมูลการแสดงผลบนจอ ซึ่งจะให้ค่าตำแหน่งขนาด 16 บิต เพื่อใช้อ่านข้อมูลรหัสการแสดงผล (Display Code), รูปแบบอักษรแสดงผล (Display Font)
- ที่พักข้อมูล (Data Buffer) เป็นที่เก็บข้อมูลการแสดงผลชั่วคราวเพื่อให้ส่วนอื่นนำไปใช้งานต่อ ซึ่งประกอบไปด้วยข้อมูลของลักษณะการแสดงผล (Attribute), รหัสการแสดงผล (Display Code) และรูปแบบอักษรแสดงผล (Display Font)
- ตัวเลื่อนจุด (Dot Shifter) เป็นส่วนที่รับรูปแบบอักษร (Font) เข้ามา แล้วเลื่อนข้อมูลรูปแบบอักษรนี้ออกไป ทีละจุดตามจังหวะของสัญญาณนาฬิกาความถี่ 12 MHz ซึ่งจุดเหล่านี้จะให้กำเนิดเป็นตัวอักษรขึ้นบนจอ
- ตัวแสดงผลตรรกบนหน้าจอ (OSD Logic) เป็นตัวรับจุดที่ได้จากตัวเลื่อนจุดนำมา ตรวจสอบกับลักษณะการแสดงผล (Attribute) แล้วให้กำเนิดเป็นสัญญาณแม่สีแดง, เขียว และน้ำเงิน คือ R,G และ B ตามลำดับ โดยมีสัญญาณ SW เป็นตัวกำหนดจังหวะการแทรกสัญญาณแม่สี

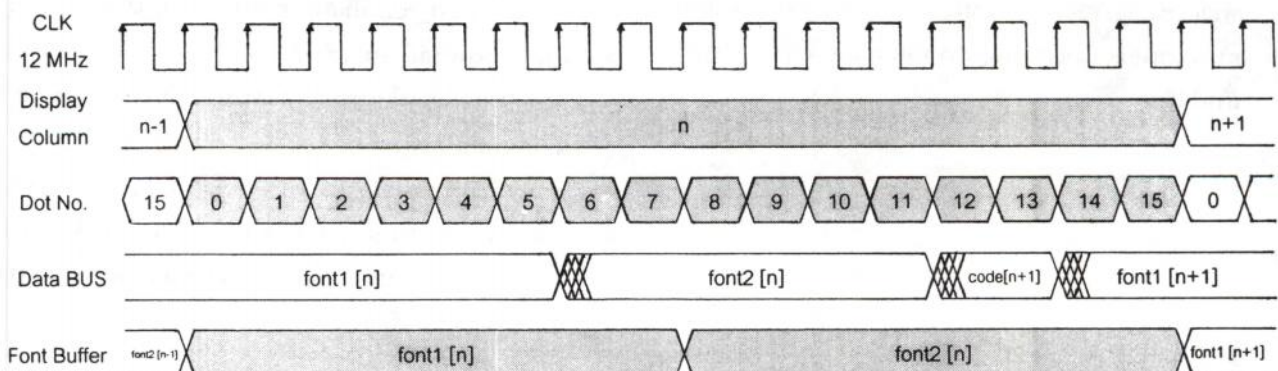
การแสดงผลบนหน้าจอจะแบ่งระดับในการแสดงเป็น 3 ระดับ คือ ระดับบน, ระดับกลาง และระดับล่าง ทั้งนี้เพื่อให้สะดวกต่อ

การใช้งานกับภาษาไทย ซึ่งการแสดงในแต่ละระดับจะอ่านรูปแบบอักษรที่มีในระดับนั้นๆ มาแสดง ดังในรูปที่ 8 โดยรูปแบบอักษรที่สามารถแสดงใน 1 สดมภ์นั้นจะมีความละเอียดกว้าง 16 จุด สูง 32 จุด โดยเก็บรูปแบบอักษรนี้ไว้ที่หน่วยความจำอ่านอย่างเดียว



รูปที่ 8 การแสดงผลบนหน้าจอโดยแบ่งเป็น 3 ระดับ

ตัวแสดงผลบนหน้าจอจะมีจังหวะในการแสดงในแต่ละสดมภ์ของบรรทัดย่อย เป็นดังรูปที่ 9 ซึ่งเมื่อรีจิสเตอร์บรรทัด กับบรรทัดที่โทรทัศน์กำลังแสดงตรงกัน โดยการตรวจสอบของตัวเปรียบเทียบตำแหน่งบรรทัดแล้ว ในสดมภ์ย่อยที่ 12 และ 13 ตัวแสดงผลบนหน้าจอจะอ่านข้อมูลการแสดงผลของสดมภ์ถัดไปมา โดยจะอ่านรูปแบบอักษรไบต์แรกมาเก็บไว้ชั่วคราวภายในตั้งแต่สดมภ์ย่อยที่ 0 ถึง 7 ก่อนที่จะเลื่อนออกไปทีละจุด และที่สดมภ์ย่อยที่ 6 ก็จะทำให้ตำแหน่งของรูปแบบอักษรไบต์ที่สองไปยังหน่วยความจำ แล้วเมื่อถึงสดมภ์ย่อยที่ 8 จะอ่านเข้ามาเก็บไว้ชั่วคราวแทนที่ข้อมูลไบต์แรก จากนั้นค่อยเลื่อนข้อมูลจุดนั้นออกไปแสดงบนหน้าจอ ทำเช่นนี้ไปในทุกๆ การแสดงผล 1 สดมภ์

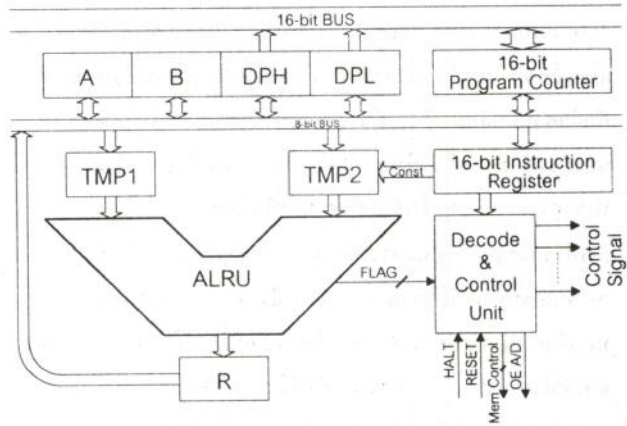


รูปที่ 9 จังหวะของการแสดงผลใน 1 สดมภ์

หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางเป็นตัวที่คอยควบคุมการทำงานของส่วนต่างๆ ภายในทีวีไมโครคอนโทรลเลอร์ โดยจะทำงานตามคำสั่งซึ่งเก็บไว้ในหน่วยความจำอย่างเดียวนั้น โครงสร้างภายในหน่วยประมวลผลกลางจะเป็นดังรูปที่ 10 ซึ่งจะประกอบไปด้วยส่วนต่าง ๆ ที่มีหน้าที่ดังนี้

- หน่วยคำนวณ, ตรรก และหมุน (Arithmetic Logic Rotate Unit : ALRU) เป็นส่วนที่ใช้ในการคำนวณทางคณิตศาสตร์, การดำเนินการทางตรรก และการหมุนข้อมูล โดยรับข้อมูลมาทางรีจิสเตอร์ TMP1 กับ TMP2 และให้ผลลัพธ์ที่รีจิสเตอร์ R
- รีจิสเตอร์คำสั่ง (Instruction Register) เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บคำสั่งที่อ่านมาจากบัสข้อมูลขนาด 8 บิต 2 ครั้ง เพื่อส่งไปยังหน่วยถอดรหัสและควบคุมต่อไป โดยถ้าเป็นคำสั่งที่ดำเนินการระหว่างรีจิสเตอร์กับค่าคงที่ที่จะส่งค่าคงที่จากคำสั่งนั้นไปยังหน่วยคำนวณ, ตรรก และหมุน เพื่อประมวลผล
- หน่วยถอดรหัสและควบคุม (Decode and Control Unit) เป็นส่วนที่ถอดรหัสจากรีจิสเตอร์คำสั่ง และควบคุมการทำงานในส่วนต่างๆ คือ ควบคุมการอ่าน, การเขียนข้อมูล ทั้งภายใน และภายนอกหน่วยประมวลผลกลาง โดยจะสั่งให้หยุดการทำงานของหน่วยประมวลผลกลางชั่วคราวหากมีสัญญาณ Halt จากภายนอกเข้ามา
- รีจิสเตอร์ (Register) ประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต ที่สามารถใช้งานได้จำนวน 4 รีจิสเตอร์ คือ รีจิสเตอร์ A, B, DPL และ DPH ซึ่ง 2 รีจิสเตอร์คือ DPL (Data Pointer Low byte) และ DPH (Data Pointer High byte) จะสามารถนำมาเรียงต่อกันเป็น รีจิสเตอร์ขนาด 16 บิต เพื่อใช้ในการอ้างอิงตำแหน่งของหน่วยความจำสำหรับอ่านข้อมูล, เขียนข้อมูล หรือกระโดดได้
- โปรแกรมเคาน์เตอร์ (Program Counter : PC) เป็นรีจิสเตอร์ขนาด 16 บิต เพื่อใช้อ้างอิงตำแหน่งของโปรแกรมที่เก็บไว้ในหน่วยความจำผ่านทางบัสตำแหน่งขนาด 16 บิต โดยค่าของโปรแกรมเคาน์เตอร์จะถูกกำหนดจากรีจิสเตอร์ DPH และ DPL หรือได้รับการคำนวณจากหน่วยคำนวณ, ตรรก และหมุน ซึ่งขึ้นอยู่กับชุดคำสั่งที่ได้รับ



รูปที่ 10 โครงสร้างภายในของหน่วยประมวลผลกลาง

ชุดคำสั่งของหน่วยประมวลผลกลาง

ชุดคำสั่งของหน่วยประมวลผลกลางแบ่งเป็น 5 ประเภทดังตารางที่ 2 คือ

Transfer	Arithmetic	Logic	Rotate	Jump
MOV	ADD	AND	ROL	JMP
LOD	SUB	OR	ROR	JNC
STO	ADC	XOR	RLC	JC
LD_ADC	SUC		RRC	JNZ
LD_PCH				JZ
LD_PCL				JR

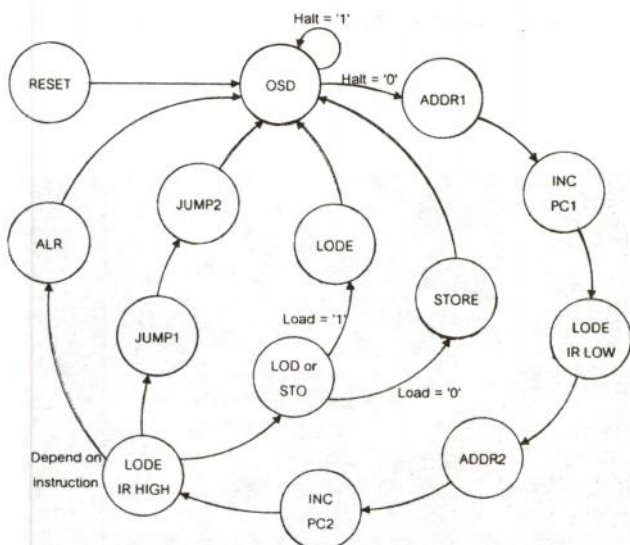
ตารางที่ 2 ชุดคำสั่งของหน่วยประมวลผลกลาง

- การโอนย้ายข้อมูล ประกอบด้วยคำสั่ง MOV เป็นการย้ายข้อมูล, LOD เป็นการอ่านข้อมูล, STO เป็นการเขียน ข้อมูล, LD_ADC เป็นการอ่านค่าที่ได้จากวงจรแปลงสัญญาณเชิงอนุพันธ์เป็นสัญญาณเชิงเลข, LD_PCH เป็นการอ่านค่าโปรแกรมเคาน์เตอร์ไบต์สูง และ LD_PCL เป็นการอ่านค่าโปรแกรมเคาน์เตอร์ไบต์ต่ำ
- การคำนวณทางคณิตศาสตร์ ประกอบด้วยคำสั่ง ADD เป็นการบวก, SUB เป็นการลบ, ADC เป็นการบวกพร้อมตัวทด และ SUC เป็นการลบพร้อมตัวทด
- การดำเนินการทางตรรก ประกอบด้วยคำสั่ง AND, OR และ XOR
- การหมุนข้อมูล ประกอบด้วยคำสั่ง ROR เป็นการหมุนขวา, ROL เป็นการหมุนซ้าย, RRC เป็นการหมุนขวาพร้อมตัวทด และ RLC เป็นการหมุนซ้ายพร้อมตัวทด

- การกระโดด ประกอบด้วยคำสั่ง JMP เป็นการกระโดดไปยังตำแหน่งที่รีจิสเตอร์ DPH กับ DPL อ้างอิงถึง, JNC เป็นการกระโดดสัมพันธ์เมื่อไม่มีแฟล็กตัวทด, JC เป็นการกระโดดสัมพันธ์เมื่อมีแฟล็กตัวทด, JNZ เป็นการกระโดดสัมพันธ์เมื่อไม่มีแฟล็กศูนย์, JZ เป็นการกระโดดสัมพันธ์เมื่อมีแฟล็กศูนย์ และ JR เป็นการกระโดดสัมพันธ์แบบไม่มีเงื่อนไข

จังหวะการทำงานของหน่วยประมวลผลกลาง

การทำงานของหน่วยประมวลผลกลางจะทำงานที่ความถี่ครั้งหนึ่งของสัญญาณนาฬิกาที่ตัวแสดงผลบนหน้าจอใช้งาน คือที่ 6 MHz โดยแต่ละขั้นตอนการทำงานจะใช้ระยะเวลา 1 คาบสัญญาณนาฬิกา ซึ่งสามารถอธิบายขั้นตอนการทำงานแต่ละขั้นได้ด้วยแผนภูมิการเปลี่ยนสแตต ดังรูปที่ 11 ซึ่งแต่ละสแตตมีการทำงานดังต่อไปนี้



รูปที่ 11 แผนภูมิสแตตการทำงานของหน่วยประมวลผลกลาง

1. Reset เป็นสแตตเริ่มต้นการทำงานของหน่วยประมวลผลกลาง ในสแตตนี้จะเป็นการตั้งค่าให้กับรีจิสเตอร์ทุกตัวภายในให้มีค่าเป็นศูนย์ สแตตนี้จะเกิดขึ้นหลังการจ่ายไฟให้กับชิป หรือมีการกดปุ่ม Reset

2. OSD เป็นสแตตที่หน่วยประมวลผลกลางจะหยุดทำงานชั่วคราว และยอมให้วงจรควบคุมการแสดงผลบนหน้าจอใช้งานทั้งบัสข้อมูล และบัสตำแหน่ง โดยจะคงอยู่ที่สแตตนี้หากสัญญาณ Halt จากวงจรควบคุมการแสดงผลบนหน้าจอมีค่าเป็น '1' และจะเปลี่ยนเป็นสแตตถัดไป หากสัญญาณ Halt มีค่าเป็น '0'

3. ADDR1 เป็นสแตตที่หน่วยประมวลผลกลางส่งค่าตำแหน่งของชุดคำสั่งไบต์ต่ำจากโปรแกรมเคาน์เตอร์ ไปยังบัสตำแหน่งเพื่อเตรียมการอ่านค่าชุดคำสั่งไบต์ต่ำมา และใน สแตตนี้จะเพิ่มค่าของโปรแกรมเคาน์เตอร์ที่ไบต์ต่ำโดยใช้หน่วยคำนวณ, ตรรก และหมุน

4. INC PC1 เป็นสแตตที่จะเก็บค่าของโปรแกรมเคาน์เตอร์ไบต์ต่ำ และเพิ่มค่าไบต์สูงของโปรแกรมเคาน์เตอร์หากมีการทดค่าจากไบต์ต่ำ

5. LOAD IR LOW เป็นสแตตที่หน่วยประมวลผล กลางจะอ่านค่าไบต์ต่ำของชุดคำสั่งเข้ามายัง รีจิสเตอร์คำสั่ง และเก็บค่าของโปรแกรมเคาน์เตอร์ไบต์สูงหากมีการทดค่าจากไบต์ต่ำ

6. ADDR2 เป็นสแตตที่หน่วยประมวลผลกลางส่งค่าตำแหน่งของชุดคำสั่งไบต์สูงจากโปรแกรมเคาน์เตอร์ ไปยังบัสตำแหน่งเพื่อเตรียมการอ่านค่าชุดคำสั่งไบต์สูงมา และใน สแตตนี้จะเพิ่มค่าของโปรแกรมเคาน์เตอร์ที่ไบต์ต่ำโดยใช้หน่วยคำนวณ, ตรรก และหมุน

7. INC PC2 เป็นสแตตที่จะเก็บค่าของโปรแกรมเคาน์เตอร์ไบต์ต่ำ และเพิ่มค่าไบต์สูงของโปรแกรมเคาน์เตอร์หากมีการทดค่าจากไบต์ต่ำ

8. LOAD IR HIGH เป็นสแตตที่หน่วยประมวลผล กลางจะอ่านค่าไบต์สูงของชุดคำสั่งเข้ามายังรีจิสเตอร์คำสั่ง และในสแตตนี้เป็นสแตตที่จะทำงานตามประเภทของชุดคำสั่ง ที่ได้รับ คือ

- ชุดคำสั่งประเภทคำนวณ, ตรรก และหมุน ส่วนลอจิกฮิสและควบคุมภายในหน่วยประมวลผลกลางจะกำหนดค่าให้กับรีจิสเตอร์ TMP1 กับ TMP2 และกำหนดการทำงานของหน่วยคำนวณ, ตรรก และเลื่อนตามชุดคำสั่งที่ได้รับมา และเก็บค่าโปรแกรมเคาน์เตอร์ไบต์สูงที่ได้รับการเพิ่มค่ามา ก่อนที่จะไปยังสแตต ALR ต่อไป
- ชุดคำสั่ง Load หรือ Store จะทำการเก็บค่าโปรแกรมเคาน์เตอร์ไบต์สูงที่ได้รับการเพิ่มค่ามา แล้วไปยังสแตต Load or Store ต่อไป
- ชุดคำสั่งประเภทกระโดด หากเป็นชุดคำสั่งที่เป็นการกระโดดแบบสัมพันธ์ ส่วนลอจิกฮิสและควบคุมภายในหน่วยประมวลผลกลางจะกำหนดค่าให้กับรีจิสเตอร์ TMP1 กับ TMP2 เพื่อทำการเพิ่มหรือลดค่าไบต์ต่ำของโปรแกรมเคาน์เตอร์ตามแต่การกระโดดสัมพันธ์นั้นๆ และไปยังสแตตต่อไป คือ สแตต Jump1 แต่ถ้าหากว่าเป็นชุดคำสั่ง JMP จะเป็นการกำหนดค่าให้กับโปรแกรมเคาน์เตอร์ด้วยรีจิสเตอร์ DPH กับ DPL และไปยังสแตต OSD เป็นสแตตต่อไป

9. ALR เป็นสแตตที่ส่วนลอจิกฮิส และควบคุมภายในหน่วยประมวลผลกลางจะนำผลลัพธ์ที่ได้จากการดำเนินการที่หน่วยคำนวณ, ตรรก และหมุน ในรีจิสเตอร์ R ไปเก็บยังรีจิสเตอร์ปลายทางตามคำสั่งที่ได้รับมา

10. LOAD or STORE เป็นสเตทที่เตรียมพร้อมในการอ่านหรือเขียนข้อมูล โดยจะส่งสัญญาณที่ใช้ในการควบคุมการอ่าน หรือเขียนออกไป ก่อนที่จะไปยังสเตท LOAD หรือ STORE ต่อไป ตามคำสั่งที่ได้รับ

11. LOAD เป็นสเตทที่อ่านข้อมูลจากหน่วยความจำ หรือ Map I/O เข้ามายังรีจิสเตอร์ จากนั้นจะไปยังสเตท OSD ต่อไป

12. STORE เป็นสเตทที่จะเขียนข้อมูลไปยังหน่วยความจำ หรือ Map I/O จากรีจิสเตอร์ จากนั้นจะไปยังสเตท OSD ต่อไป

13. JUMP1 เป็นสเตทที่จะเก็บค่าโปรแกรมเคาน์เตอร์ไบต์ต่ำที่ได้จากการคำนวณค่า จากนั้นไปยังสเตท JUMP2

14. JUMP2 เป็นสเตทที่จะเก็บค่าโปรแกรมเคาน์เตอร์ไบต์สูงที่ได้จากการคำนวณค่า จากนั้นไปยังสเตท OSD

เมื่อทำการออกแบบส่วนต่างๆ ทั้งหมดของทีวีไมโครคอนโทรลเลอร์แล้ว ได้ทำการสังเคราะห์วงจร และโปรแกรมการทำงานลงบน Xilinx FPGA เบอร์ 4010E-4 ซึ่งมีรายงานการใช้ทรัพยากรภายในชิปดังแสดงในตารางที่ 3

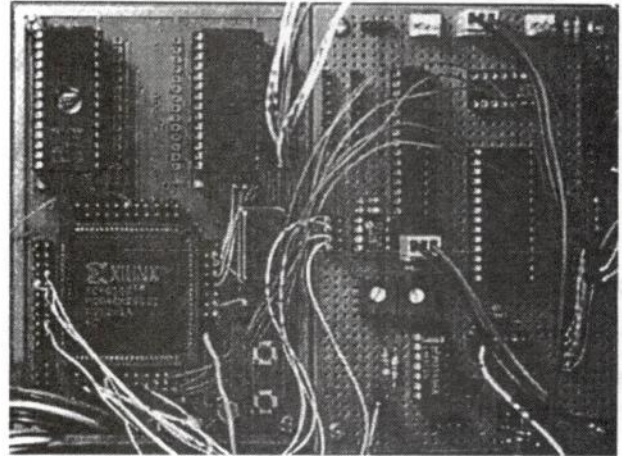
ทรัพยากรภายในชิป	จำนวนที่ใช้	จำนวนสูงสุด	% การใช้
External IOBs	56	61	91 %
Global Buffer IOBs	3	8	37 %
CLBs	344	400	86 %
Total CLB Flops	311	800	38 %
4 input LUTs	502	800	62 %
3 input LUTs	196	400	49 %
Oscillators	1	1	100 %
Pri-CLKs	2	4	50 %
Sec-CLKs	4	4	100 %

ตารางที่ 3 รายงานการใช้ทรัพยากรภายในชิป Xilinx 4010E

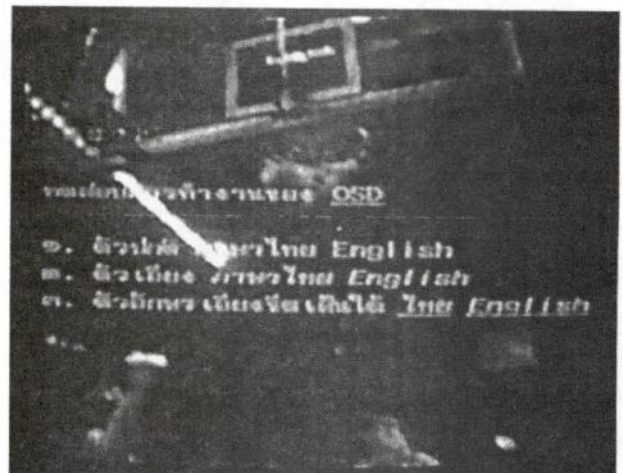
การทดสอบการทำงาน

เมื่อได้ทำการออกแบบ พร้อมทั้งจำลองการทำงานแล้ว ก็ได้สังเคราะห์วงจรทั้งหมด และทำการโปรแกรมข้อมูลของ Xilinx FPGA, โปรแกรมการทำงานของทีวีไมโครคอนโทรลเลอร์ และรูปแบบอักษรลงในหน่วยความจำอ่านอย่างเดียว ซึ่งบอร์ดต้นแบบของทีวีไมโครคอนโทรลเลอร์มีลักษณะดังรูปที่ 12

ทีวีไมโครคอนโทรลเลอร์สามารถควบคุมการทำงานของโทรทัศน์สี Distar ขนาด 14 นิ้ว รุ่น DT-1416AV II ได้ ซึ่งภาพทดสอบการแสดงผลบนหน้าจอเป็นดังรูปที่ 13



รูปที่ 12 บอร์ดต้นแบบของทีวีไมโครคอนโทรลเลอร์



รูปที่ 13 การทดสอบผลการแสดงผลบนหน้าจอ

สรุป

ทีวีไมโครคอนโทรลเลอร์ที่พัฒนาขึ้นมาใช้ Xilinx FPGA เบอร์ 4010E กับวงจรรวมภายนอกเพียง 3 วงจร คือหน่วยความจำอ่านอย่างเดียว, หน่วยความจำเข้าถึงแบบสุ่ม และวงจรแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข ซึ่งจากการทดสอบการทำงาน พบว่าทีวีไมโครคอนโทรลเลอร์นี้สามารถควบคุมเครื่องรับโทรทัศน์ได้จริง และเป็นต้นแบบที่จะใช้สำหรับการพัฒนาทีวีไมโครคอนโทรลเลอร์ซึ่งสามารถถอดรหัสคำบรรยายภาพแบบซ่อนได้ภาษาไทย-อังกฤษได้ในตัวต่อไปในอนาคต

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนจากสำนักงานคณะกรรมการ
สภาวิจัยแห่งชาติ, มูลนิธิโทรเพื่อการส่งเสริมวิทยาศาสตร์ประเทศไทย
และสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

เอกสารอ้างอิง

- [1] Hiroyasu Shindo, Tsutomu Nakazawa, Masaya Ohta, Kasaya Ohta, Kazuyuki Mitsuya and Masaru Tonoizuka, *Microcontrollers for closed caption system*, IEEE Transactions on Consumer Electronics, Vol. 38, August 1992, pp. 268-273.
- [2] Hing-Yip Tong, *A Single Chip Micro-Computer For A/V Monitor And TV Receiver*, IEEE Transactions on Consumer Electronics, November 1990, Vol. 36, pp. 825-831.
- [3] Zilog, *Digital Television Controllers*, Product Specifications Databook.
- [4] Toshiba Corporation, *Toshiba Original CMOS 8-bit Microcontroller TLCS-870 SERIES Databook*.
- [5] Tom S. Wei, Andre B. Walker, Gary K. Pacey, *A universal high performance digital performance television controller*, IEEE Transactions on Consumer Electronics, Vol. 36, August 1990, pp. 678-683.
- [6] Matsushita Denshi (S) PTE LTD., *MN6014, MN6014S Remote-Control Transmitter CMOS LSI*, Device Specification.
- [7] Xilinx, *SRAM-Based FPGA Products*, The Programmable Logic Data Book, August 15, 1996.