ต้นแบบฮาร์ดแวร์ตรวจสอบลายนิ้วมือบนตัวประมวลผลสัญญาณดิจิทัล

Fingerprint Verification Hardware Prototype Using Digital Signal Processor

คำนำ

ในปัจจุบันนี้ความเจริญทางเทคโนโลยีสารสนเทศมีบทบาทมากในชีวิตประจำวัน ทำให้สังคมเรามีการติดต่อสื่อสารกันมากขึ้น ไม่ว่าจะเป็นการติดต่อระหว่างบุคคล การติดต่อระหว่างองค์กรหรือหน่วยงาน การติดต่อธุรกิจทางด้านธนาคาร หรือแม้แต่ทางด้านการทหาร ล้วนต้องการระบบรักษาความปลอดภัย เพื่อความเชื่อถือได้ของการติดต่อหรือทำธุรกิจ เทคโนโลยีการพิสูจน์บุคคลแบบอัตโนมัติแบบเดิมที่เราคุ้นเคยกันดี ไม่ว่าจะเป็นการใช้บัตรเครดิต บัตรประจำตัว หรือบัตรสมาร์ตการ์ด นั้นมีความยุ่งยากตรงที่ต้องพกพาบัตรเหล่านั้นไว้กับตัว อาจสูญหาย ถูกขโมยไปใช้ หรือนำไปปลอมแปลงเพื่อคัดลอกรหัสผ่านได้ เนื่องจากปัญหาเหล่านี้ทำให้แนวคิดในการใช้วิธีการตรวจสอบอัตลักษณ์หรือคุณลักษณะของส่วนต่าง ๆ ของร่างกายของแต่ละบุคคลเข้ามาทดแทน จุดเด่นและข้อได้เปรียบของการตรวจสอบอัตลักษณ์ คือ ปลอมแปลงได้ยาก ไม่โดนขโมย และไม่สูญหายเมื่อเปรียบเทียบกับบัตรเครดิตหรือสมาร์ตการ์ด อีกทั้งไม่ต้องจดจำรหัสผ่านหรือเลขรหัสต่าง ทำให้วิธีการตรวจสอบอัตลักษณ์เป็นที่สนใจในงานด้านรักษาความปลอดภัยเป็นอย่างมาก

การใช้ลายนิ้วมือเพื่อระบุหรือพิสูจน์บุคคล เป็นวิธีหนึ่งที่มีการนำมาใช้อย่างแพร่หลาย เนื่องจากลายนิ้วมือมีลักษณะเป็นรหัสประจำตัวที่เป็นเอกลักษณ์ของบุคคลหนึ่งบุคคลใดเท่านั้น ไม่เปลี่ยนแปลงไปตามเวลา ยากต่อการปลอมแปลงได้ และสะดวกในการใช้งาน ในปัจจุบันเทคโนโลยีมีความก้าวหน้ามากขึ้นโดยเฉพาะเทคโนโลยีคอมพิวเตอร์ ซึ่งถูกพัฒนาจนมีความรวดเร็ว และมีประสิทธิภาพสูงขึ้น จึงมีการพัฒนาระบบตรวจสอบลายนิ้วมืออัตโนมัติ (AFIS: Automated Fingerprint Identification System) โดยอาศัยการประมวลภาพดิจิทัล ให้เหมาะสมกับการใช้งานในงานต่างๆได้หลากหลาย เช่น ทางด้านรักษาความปลอดภัยในการเข้าถึงสถานที่หรือข้อมูล และ การบันทึกเวลาเข้าออกพนักงาน ซึ่งเป็นที่นิยมอย่างมากในปัจจุบัน

แม้ว่าผลิตภัณฑ์ทางด้านการตรวจสอบลายนิ้วมืออัตโนมัติได้มีการนำเข้ามาขายในเมืองไทยอย่างกว้างขวางในปัจจุบัน แต่ราคาขายโดยเฉพาะอุปกรณ์ตรวจสอบลายนิ้วมือที่ประมวลผลได้ด้วยตัวเองยังมีราคาสูงอยู่มาก การพัฒนาฮาร์ดแวร์อุปกรณ์ตรวจสอบลายนิ้วมือในประเทศไทยยังอยู่ในช่วง

เริ่มต้นและจำกัด เนื่องจากขั้นตอนวิธีมีความซับซ้อนสูง ต้องใช้ฮาร์ดแวร์ที่มีประสิทธิภาพในการคำนวณสูงและต้องมีพื้นที่หน่วยความจำที่ใหญ่พอที่จะประมวลภาพและเก็บฐานข้อมูลลายนิ้วมือในรูปแบบของรหัส รวมทั้งปัญหาทางด้านพลังงานที่จำกัดและการติดต่อสื่อสารกับระบบอื่น ทำให้การพัฒนาฮาร์ดแวร์ตรวจสอบลายนิ้วมือเป็นปัญหาที่ท้าทายนักวิจัยและพัฒนาอยู่ในปัจจุบัน

ในงานวิจัยนี้ได้ทำการพัฒนาฮาร์ดแวร์สำหรับระบบตรวจสอบลายนิ้วมือ ซึ่งถูกออกแบบและพัฒนาขึ้นโดยคณะวิจัยการประมวลสัญญาณและภาพมหาวิทยาลัยเกษตรศาสตร์ (Kasetsart Signal and Image Processing Laboratory) หรือ KSIP Lab ร่วมกับศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (National Electronics and Computer Technology Center) หรือ NECTEC เป็นฮาร์ดแวร์ที่มีขนาดเล็กและใช้พลังงานต่ำ ใช้ตัวประมวลผลสัญญาณดิจิทัล (Digital Signal Processing) หรือ DSP ซึ่งเป็นอุปกรณ์ที่เหมาะสม เนื่องจากมีขนาดเล็กและสามารถออกแบบให้ติดต่อกับอุปกรณ์ภายนอกได้หลายรูปแบบ นอกจากนี้ยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น เครื่องมือ DSP/BIOS (Digital Signal Processing / Basic Input Output System) ทำให้ง่ายต่อการพัฒนาซอฟต์แวร์ระบบด้วย โดยมีจุดประสงค์เพื่อนำไปใช้สำหรับการพัฒนาขั้นตอนวิธีการตรวจสอบลายนิ้วมือแบบเวลาจริงบนฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ซึ่งทางคณะวิจัย KSIP Lab ได้คิดค้นขึ้น และเพื่อเป็นต้นแบบระบบตรวจสอบลายนิ้วมือในเชิงพาณิชย์ โดยมีจุดเด่นตรงที่เป็นนวัตกรรมที่สร้างขึ้นโดยฝีมือของคนไทยทั้งหมดทั้งซอฟต์แวร์และฮาร์ดแวร์ ซึ่งสามารถควบคุมต้นทุนการผลิตได้ ช่วยลดการนำเข้าสินค้าจากต่างชาติ จุดเด่นอีกข้อหนึ่งคือ ฮาร์ดแวร์ที่พัฒนาขึ้นถูกออกแบบให้สามารถนำไปปรับใช้กับงานต่างๆ ที่ต้องการประสิทธิภาพในการคำนวณสูงได้หลายรูปแบบ เพื่อรองรับงานวิจัยอื่นๆ ได้ในอนาคต

วัตถุประสงค์

วัตถุประสงค์ของการพัฒนาต้นแบบฮาร์ดแวร์ตรวจสอบลายนิ้วมือบนตัวประมวลผลสัญญาณดิจิทัลมีดังนี้

1. เพื่อพัฒนาฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลแบบเวลาจริง โดยใช้ตัวประมวลผล DSP ของบริษัท Texas Instruments (TI) ซึ่งถูกออกแบบให้นำไปใช้ในการพัฒนาระบบตรวจสอบลายนิ้วมือ
2. เพื่อสร้างต้นแบบผลิตภัณฑ์สำหรับรักษาความปลอดภัยในการเข้าถึงสถานที่หรือข้อมูลด้วยการตรวจพิสูจน์ลายนิ้วมือ ซึ่งเป็นนวัตกรรมของคนไทย และนำไปสู่การผลิตเชิงพาณิชย์ได้
3. ฮาร์ดแวร์ที่พัฒนาขึ้นถูกออกแบบให้สามารถนำไปปรับใช้กับงานต่างๆ ที่ต้องการประสิทธิภาพในการคำนวณสูงได้หลายรูปแบบ เพื่อรองรับงานวิจัยอื่นๆ ได้ในอนาคต

การตรวจเอกสาร

ระบบตรวจสอบลายนิ้วมืออัตโนมัติ

การตรวจสอบอัตลักษณ์ (Biometric) หมายถึง การพิสูจน์หรือระบุตัวบุคคล โดยเฉพาะอย่างยิ่งเกี่ยวกับลักษณะเด่นทางสรีรศาสตร์หรือลักษณะทางกายภาพของแต่ละบุคคลซึ่งเป็นสิ่งที่มีเอกลักษณ์เฉพาะตัว เทคโนโลยีทางการตรวจสอบอัตลักษณ์หลายชนิดถูกนำมาใช้กันอย่างกว้างขวางในเชิงพาณิชย์ เช่น ใบหน้า ลายนิ้วมือ ลักษณะรูปร่างของฝ่ามือ ลายม่านตา ถ่ายเส้นเลือด รหัสดีเอนเอ เสียงพูด และลายมือชื่อ ฯลฯ เพราะการพิสูจน์บุคคลด้วยระบบไบโอเมตริก มีโอกาสที่เกิดความผิดพลาดได้น้อย ไม่สูญหาย ไม่สามารถปลอมแปลงหรือปลอมแปลงได้ยาก และไม่เปลี่ยนแปลงเมื่อเวลาผ่านไป ซึ่งให้ความเชื่อถือได้มากกว่าระบบรักษาความปลอดภัยในแบบเก่า เช่น กุญแจ รหัสผ่าน หรือบัตรแถบแม่เหล็ก ระบบไบโอเมตริกนั้นมีคุณสมบัติที่ดีกว่าทั้งทางด้านประสิทธิภาพ และความสะดวกสบายในการใช้งาน ด้วยเหตุผลเหล่านี้ระบบไบโอเมตริกจึงถูกนำมาใช้กันอย่างแพร่หลาย ยกตัวอย่างเช่น ความปลอดภัยในการผ่านเข้าออกอาคาร การใช้งานคอมพิวเตอร์ การเข้าใช้เครื่องรับจ่ายเงินอัตโนมัติ บัตรประชาชน ใบขับขี่ ระบบเครือข่าย เป็นต้น สำหรับการพิสูจน์บุคคลด้วยลายนิ้วมือนั้นมีใช้มานานกว่า 100 ปีแล้วในทางด้านกฎหมาย แต่สำหรับระบบการตรวจสอบลายนิ้วมืออัตโนมัตินั้นเพิ่งจะเริ่มมีการพัฒนาครั้งแรกเมื่อ 50 ปีที่ผ่านมานี่เอง (Maltoni *et al*., 2003)

โครงสร้างของระบบตรวจสอบลายนิ้วมืออัตโนมัติ ประกอบด้วยส่วนหลัก ๆ 4 ส่วน คือ หน่วยรับภาพลายนิ้วมือ หน่วยประมวลผล หน่วยจัดเก็บฐานข้อมูล และหน่วยเชื่อมต่ออุปกรณ์ภายนอก ดังแสดงในภาพที่ 1

1. หน่วยรับภาพลายนิ้วมือ (Sensing Element) หน่วยรับภาพลายนิ้วมือจะเป็นฮาร์ดแวร์หลักที่มีหน้าที่ กราดตรวจลายนิ้วมือและดัดแปลงภาพที่ได้ไปอยู่ในรูปข้อมูลแบบดิจิทัลเสียก่อน จากนั้นจึงส่งข้อมูลภาพลายนิ้วมือไปยังหน่วยประมวล ฮาร์ดแวร์กราดตรวจลายนิ้วมือที่นิยมใช้กันจะเป็นลักษณะของสารกึ่งตัวนำ (Semiconductor) และ เชิงแสง (Optical)

2. หน่วยประมวลผล (Processing Element) โดยทั่วไปแล้วหน่วยประมวลผลมีหลายประเภท ซึ่งตัวประมวลผลที่นิยมใช้สำหรับงานตรวจสอบลายนิ้วมือ ได้แก่ โปรเซสเซอร์ ARM (Acorn RISC Machine), ตัวประมวลผลสัญญาณดิจิทัล และ คอมพิวเตอร์ส่วนบุคคล เป็นต้น ลักษณะของซอฟต์แวร์

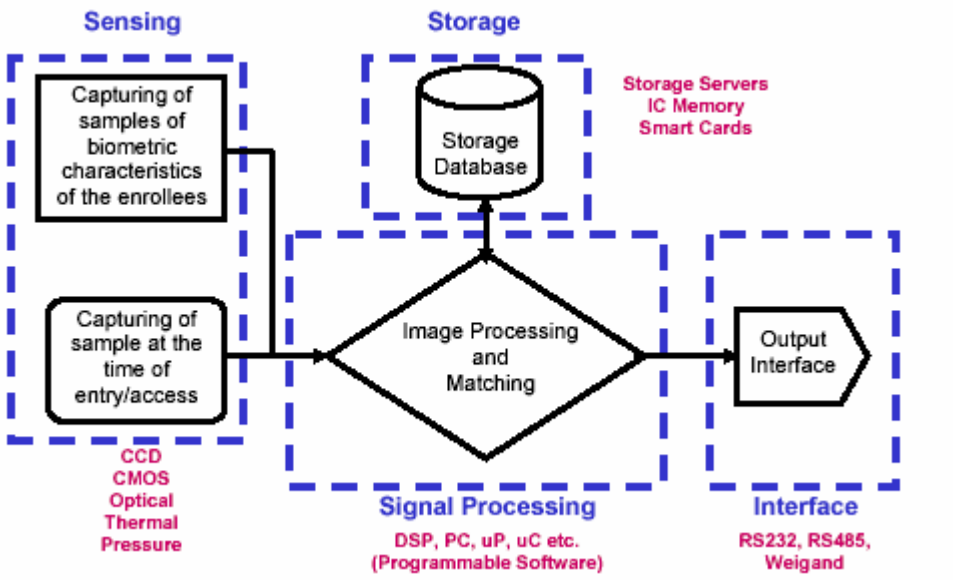
บนตัวประมวลสามารถแบ่งได้เป็น 2 ส่วน คือ ซอฟต์แวร์จัดการระบบ และซอฟต์แวร์ขั้นตอนวิธีหรืออัลกอริทึม

ซอฟต์แวร์จัดการระบบเป็นโปรแกรมเฉพาะบนตัวประมวลผล ซึ่งใช้สำหรับจัดการทรัพยากรทั้งหมดในระบบ โดยซอฟต์แวร์จัดการระบบจะทำหน้าที่จัดการทรัพยากรต่างๆ ในระบบ เช่น ควบคุมเวลาทำงานของระบบ จัดการการใช้งานหน่วยความจำของระบบ รวมทั้งควบคุมส่วนติดต่ออุปกรณ์ภายนอกระบบ ได้แก่ อุปกรณ์ติดต่อผู้ใช้งาน (User Interface Device) อุปกรณ์กราดตรวจลายนิ้วมือ (Fingerprint Sensor Device) เป็นต้น

ซอฟต์แวร์ขั้นตอนวิธีตรวจสอบลายนิ้วมือเป็นโปรแกรมที่ใช้ในการประมวลผลภาพลายนิ้วมือที่ได้รับมาจากหน่วยรับภาพลายนิ้วมือ โดยทั่วไปจะประกอบด้วยการวิเคราะห์คุณภาพลายนิ้วมือ การปรับปรุงคุณภาพลายนิ้วมือ การดึงลักษณะเด่นของลายนิ้วมือออกมา เช่น จุดมินูเทียร์ (Minutiae) และจุดโฟกัส (Focus) เพื่อนำไปเปรียบเทียบกับลายนิ้วมือที่เก็บไว้ในฐานข้อมูลต่อไป จากนั้นจึงนำผลลัพธ์ที่ได้จากการประมวลผลส่งกลับไปยังซอฟต์แวร์จัดการระบบต่อไป

3. หน่วยจัดเก็บฐานข้อมูล (Storage Element) หน้าที่หลักของหน่วยจัดเก็บฐานข้อมูลคือ เก็บแม่แบบ (Template) ของลายนิ้วมือที่มีการลงทะเบียนไว้ เพื่อนำไปใช้เปรียบเทียบกับลายนิ้วมือที่นำมาตรวจสอบ โดยทั่วไปแล้วหน่วยจัดเก็บฐานข้อมูลจะใช้หน่วยความจำประเภท RAM (Random-Access Memory) หรือ EPROM (Erasable Programmable Read-Only Memory) ในกรณีต่อกับเครื่องคอมพิวเตอร์อาจใช้ในลักษณะของเครือข่ายบริการข้อมูล (Data Server) หรือเพื่อสะดวกในการพกพาอาจเก็บข้อมูลลงในบัตรอัจฉริยะ (Smart Card) ได้เช่นกัน

4. หน่วยเชื่อมต่ออุปกรณ์ภายนอก (Interface Element) หน่วยเชื่อมต่ออุปกรณ์ภายนอกมีหน้าที่ติดต่อกับเครื่องมือหรืออุปกรณ์ภายนอกระบบ เช่น อุปกรณ์ติดต่อผู้ใช้งาน และอุปกรณ์ควบคุมการเปิดปิดประตูเข้าออก เป็นต้น ลักษณะการติดต่อสื่อสารอย่างง่ายโดยผ่านพอร์ตอนุกรมซึ่งใช้โปรโตคอล RS232 หรือโปรโตคอล USB (Universal Serial Bus) สำหรับอัตราการส่งข้อมูลที่สูงขึ้นสามารถใช้โปรโตคอล TCP/IP ผ่านการสื่อสารแบบใช้สาย เช่น 10/100 Ethernet หรือผ่านการสื่อสารแบบไร้สาย เช่น 802.11b Protocol, ISM RF Band, RFID, Buletooth, Cellular Protocol เป็นต้น



ภาพที่ 1 องค์ประกอบของระบบตรวจสอบลายนิ้วมืออัตโนมัติ

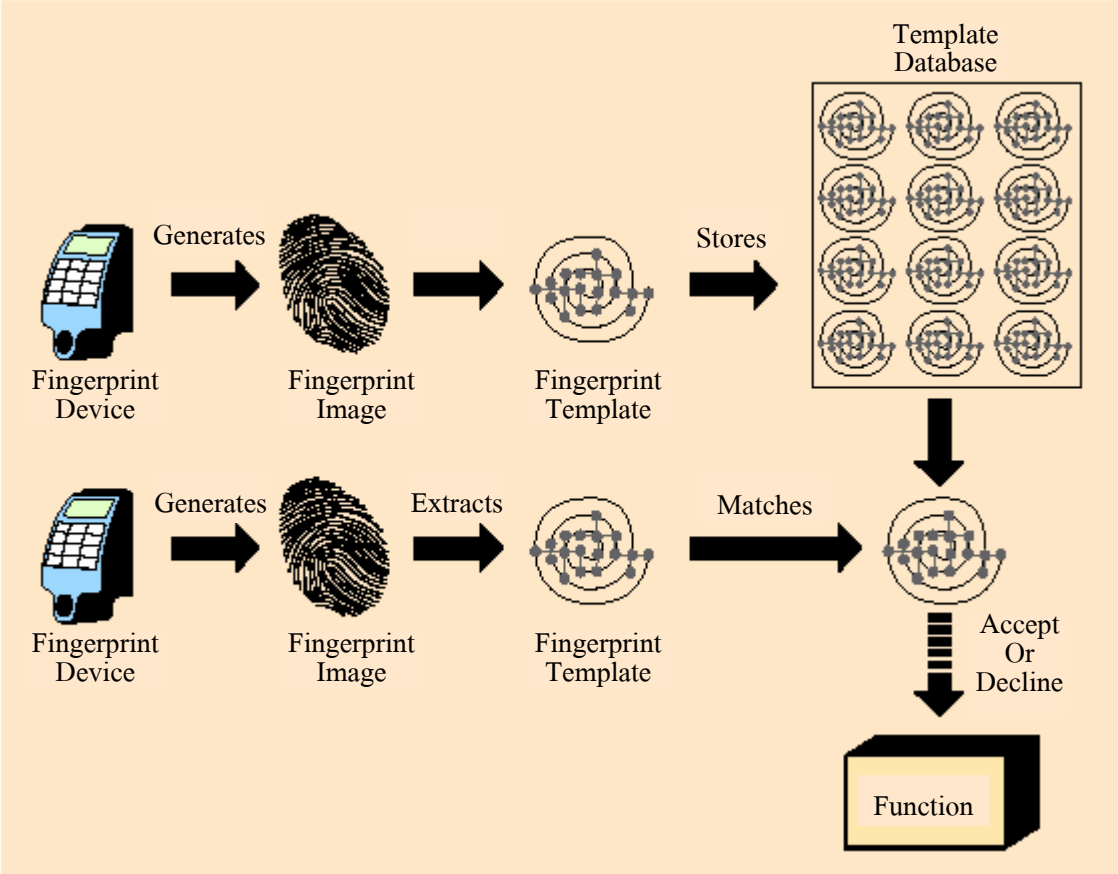
ที่มา : Texas Instruments. (2003)

ขั้นตอนการทำงานของระบบตรวจสอบลายนิ้วมือ อาจแบ่งเป็นได้ 2 ขั้นตอนคือ

1. ขั้นตอนการลงทะเบียน (Enrollment Phase)
2. ขั้นตอนการเปรียบเทียบบุคคล (Verification Phase)

ขั้นตอนในการเริ่มต้นใช้งานระบบตรวจสอบลายนิ้วมือ คือ การลงทะเบียน (Enrollment) การลงทะเบียนที่ว่านี้ หมายถึงการที่ผู้ใช้แต่ละรายจะต้องถูกเก็บภาพตัวอย่างลายนิ้วมือผ่านตัวกราดตรวจลายนิ้วมือ จากนั้นทำการประมวลผลวิเคราะห์ และผ่านขั้นตอนการดึงลักษณะเด่น (Feature Extraction) คือขั้นตอนการหาข้อมูลที่เป็นลักษณะที่เป็นเอกลักษณ์ของภาพลายนิ้วมือในแต่ละบุคคล และแปลงเป็นโครงสร้างข้อมูลที่เหมาะสม จากนั้นจะเก็บข้อมูลที่ได้ลงในระบบฐานข้อมูลพร้อมรหัสชื่อของบุคคลที่ลงทะเบียนเพื่อเป็นแม่แบบ (Template) สำหรับนำไปเปรียบเทียบพิสูจน์บุคคลต่อไป

สำหรับขั้นตอนเปรียบเทียบบุคคลนั้น ในขั้นแรกผู้ใช้แต่ละรายจะต้องถูกเก็บภาพตัวอย่างลายนิ้วมือผ่านตัวกราดตรวจลายนิ้วมือในลักษณะเดียวกับการลงทะเบียน จากนั้นทำการประมวลผลวิเคราะห์ และแยกลักษณะเด่นออกมาแปลงเป็นโครงสร้างข้อมูลที่เหมาะสม แล้วทำการเปรียบเทียบคู่ลักษณะเด่น (Feature Matcher) ที่ได้กับข้อมูลที่มีอยู่ในฐานข้อมูล เพื่อสอบพิสูจน์ตัวบุคคลนั้นๆ ซึ่งข้อมูลทั้งคู่จะต้องตรงกันเท่านั้นผู้ใช้งานจึงจะผ่านเข้าไปยังระบบได้ หลักการทำงานของระบบตรวจสอบลายนิ้วมือแสดงไว้ดังภาพที่ 2



ภาพที่ 2 หลักการทำงานของระบบตรวจสอบลายนิ้วมืออัตโนมัติ
ที่มา : ศิริสาร (2546)

การใช้งานระบบตรวจสอบลายนิ้วมืออัตโนมัติ อาจแบ่งลักษณะการใช้งานได้ 2 รูปแบบคือ

1. การระบุตัวบุคคล (Identification) หรือการจับคู่เปรียบเทียบแบบหนึ่งต่อจำนวนหลายหน่วย (1:N) ระบบจะทำการจับคู่เปรียบเทียบข้อมูลที่ได้รับมากับข้อมูลทั้งหมดในฐานข้อมูล เพื่อระบุว่าข้อมูลที่ส่งมาเป็นของบุคคลใด การใช้งานในรูปแบบนี้เหมาะจะนำไปใช้กับระบบเกี่ยวกับการค้นหาบุคคล เช่น ในด่านตรวจคนเข้าเมือง การลงเวลาเข้าออกสำนักงาน งานสืบสวนทางอาชญากรรม เป็นต้น

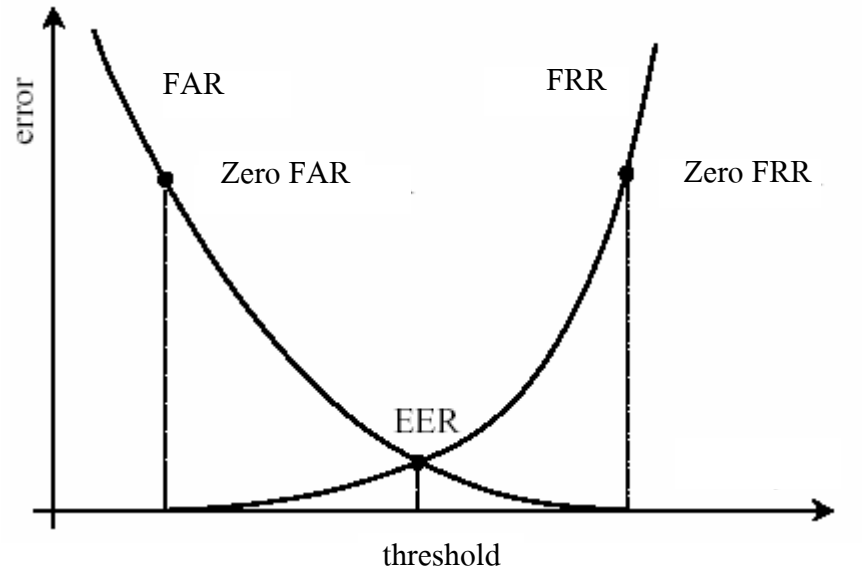
2. การตรวจพิสูจน์บุคคล (Verification) หรือการจับคู่เปรียบเทียบแบบหนึ่งต่อหนึ่ง (1:1) เพื่อเป็นการพิสูจน์ว่าเป็นบุคคลคนเดียวกันกับที่ได้รับอนุญาตจากระบบหรือไม่ ซึ่งการอ้างถึงข้อมูลเฉพาะที่เก็บไว้ในระบบฐานข้อมูลสามารถอ้างได้จากรหัสเฉพาะประจำตัว ได้แก่ ตัวเลขระบุตัวบุคคล (PIN: Personal Identification Number) หรือบัตรอัจฉริยะ (Smart Card) ระบบจะทำงานโดยเริ่มจาก

ผู้ใช้จะต้องป้อนรหัสประจำตัวที่ระบุถึงตัวผู้ใช้ก่อน แล้วจึงกราดตรวจลายนิ้วมือของตนเองให้กับระบบเพื่อเป็นการยืนยันตัวเองเพื่อบอกว่าใช่บุคคลที่อ้างถึงหรือไม่ ตัวอย่างการใช้งานระบบตรวจสอบลายนิ้วมือในรูปแบบนี้ เช่น การผ่านเข้าออกอาคารหรือสถานที่ เป็นต้น

การทำความเข้าใจถึงผลกระทบที่มีต่อความแม่นยำของระบบตรวจสอบลายนิ้วมือ สามารถเทียบวัดได้จากค่าอัตราการปฏิเสธผู้ใช้งานและอัตราการยอมรับผู้แปลกปลอม ที่แต่ละระบบจะแสดงค่าออกมา เพื่อบ่งบอกถึงคุณภาพในเรื่องความปลอดภัยหรือการตอบรับผู้ใช้งานที่มีการลงทะเบียนไว้

1. ค่าอัตราการปฏิเสธผู้ใช้งาน หรือ False Rejection Rate (FRR) คือ ค่าของอัตราการปฏิเสธการผ่านกับผู้ใช้งานที่อยู่ในระบบ ค่า FRR นี้จะมีอัตราส่วนแสดงอยู่ เช่น 1/100 หมายถึงการยอมรับใน 100 ครั้งการเข้าถึง จะไม่สามารถเข้าถึงได้ 1 ครั้ง

2. ค่าอัตราการยอมรับผู้แปลกปลอม หรือ False Acceptance Rate (FAR) คือค่าของอัตราการหลุดรอดของผู้แปลกปลอมจากการตรวจจับ ค่า FAR นี้ควรจะมีค่าน้อยมาก ๆ จะถือว่ามีคุณภาพดี เช่น 1/20,000 หมายถึง ใน 20,000 ครั้งที่ผู้แปลกปลอมพยายามจะเข้าระบบ จะยอมรับได้เพียง 1 ครั้งเท่านั้น



ภาพที่ 3 ความสัมพันธ์ระหว่างค่า FAR และ FRR ต่อค่าระดับขีดเริ่มเปลี่ยน

ที่มา : Maltoni *et al*. (2003)

โดยทั่วไประบบตรวจสอบลายนิ้วมือที่มีขายอยู่ในตลาดปัจจุบันจะมีค่า FRR อยู่ที่ประมาณ 0.1% ส่วนค่า FAR นั้น จะมีค่าอยู่ที่ประมาณ 0.001% ทั้งนี้ค่า FRR และ ค่า FAR เป็นค่าที่มีความสัมพันธ์แปรผกผันกัน คือ เมื่อค่า FAR มีค่าสูง ค่า FRR ก็จะมีค่าต่ำโดยอัตโนมัติ ณ ตำแหน่งที่ค่า FRR และ ค่า FAR มีค่าเท่ากัน เราจะเรียกค่านั้นว่า ค่า EER (Equal Error Rate) ซึ่งคือค่าความผิดพลาดที่เท่ากันของระบบ ซึ่งสามารถนำมาใช้เปรียบเทียบระบบได้ถ้าใช้ฐานข้อมูลที่เหมือนกัน ในระบบรักษาความปลอดภัยด้วยไบโอเมตริก ค่า FRR และค่า FAR จะเป็นค่าที่สามารถถูกปรับได้ตามความต้องการของผู้ติดตั้งระบบ ว่าต้องการให้ระบบมีความปลอดภัยมากน้อยเพียงใด โดยการเลือกค่าระดับขีดเริ่มเปลี่ยน (Threshold) หรือเลือกค่าเปอร์เซนต์ความเข้าคู่กันของลายนิ้วมือที่จะถือว่าเป็นลายนิ้วมือของคนคนเดียวกันให้เหมาะสมกับการใช้งาน ความสัมพันธ์ระหว่างค่า FAR FRR และ EER ต่อค่าระดับขีดเริ่มเปลี่ยนได้แสดงไว้ในภาพที่ 3

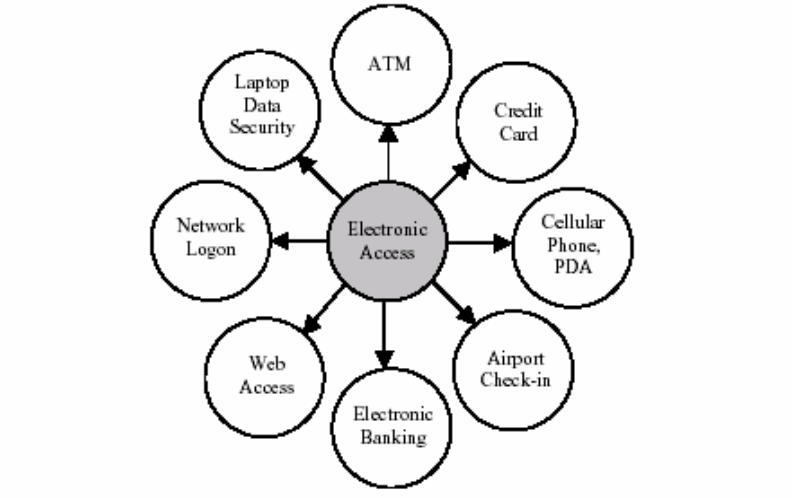
เทคโนโลยีการตรวจพิสูจน์บุคคลด้วยลายนิ้วมือ ได้ถูกพัฒนาและมีการนำมาใช้งานกันอย่างแพร่หลาย เนื่องจากเป็นระบบที่มีความเชื่อถือได้สูง รูปแบบการใช้งานระบบตรวจสอบลายนิ้วมือสามารถแบ่งออกได้เป็น 3 กลุ่มหลัก ๆ คือ (Jain *et al.*, 2004)

1. งานด้านนิติเวช (Forensic) เป็นระบบขนาดใหญ่กับฐานข้อมูลจำนวนมาก เช่น การพิสูจน์ ศพ, อาชญากรข้ามชาติ, การสืบสวนการกระทำผิดทางอาญา, การระบุตัวผู้ก่อการร้าย ระบบเหล่านี้ต้องการความเชื่อถือสูง ดังนั้นจึงจำเป็นต้องใช้ผู้เชี่ยวชาญช่วยในการตรวจสอบ และตัดสินใจโดยใช้ระบบเป็นเครื่องประมวลหาผู้ต้องสงสัยเบื้องต้น

2. งานด้านการปกครอง (Government) เป็นระบบขนาดใหญ่กับฐานข้อมูลจำนวนมาก เช่น การสำรวจสัมโนประชากร, ทำบัตรประชาชน, ใบขับขี่, หนังสือเดินทาง เป็นต้น การใช้งานในลักษณะนี้ลายนิ้วมือจะถือเป็นสิ่งแทนตัวของบุคคล เพื่อเป็นหลักฐานแสดงต่อองค์กรหรือหน่วยงานต่าง ๆ

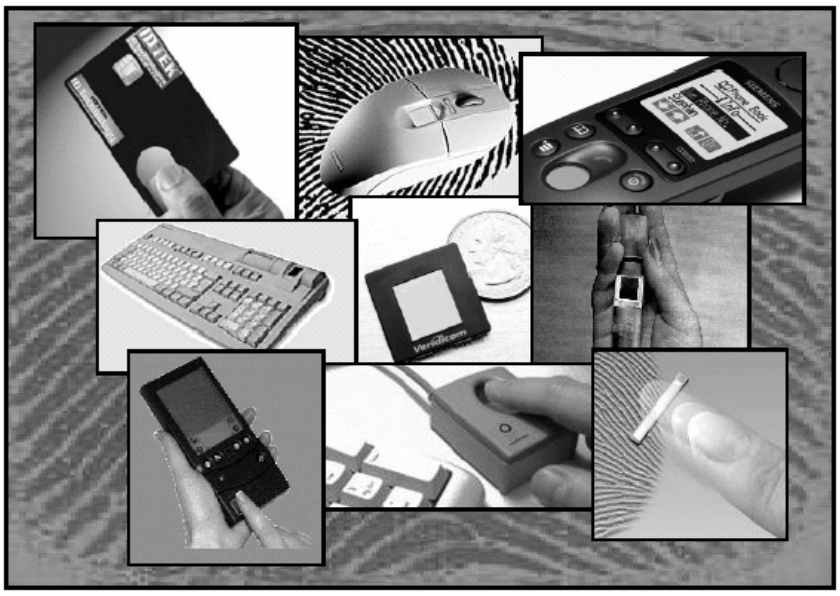
3. งานด้านธุรกิจ (Commercial) เป็นระบบขนาดกลางและเล็ก เช่น การใช้คอมพิวเตอร์เพื่อเข้าระบบเครือข่าย, ธุรกิจองค์กรอินเตอร์เน็ต, การแลกเปลี่ยนข้อมูลระหว่างองค์กร, การโอนเงินผ่านเครือข่าย, การธนาคาร, เครื่องรับจ่ายเงินอัตโนมัติ (ATM: Automated Teller Machine), ระบบล็อกประตูห้อง, การเรียนแบบออนไลน์ เป็นต้น การใช้งานในลักษณะนี้ ลายนิ้วมือจะเป็นสิ่งยืนยันสิทธิสำหรับกลุ่มผู้ใช้ในการเข้าใช้งานระบบ ซึ่งจะช่วยอำนวยความสะดวกสบายและทำให้ธุรกิจมีความคล่องตัวมากขึ้น

จะเห็นได้ว่าระบบตรวจสอบลายนิ้วมืออัตโนมัตินั้น สามารถประยุกต์ใช้กับงานด้านต่าง ๆ ได้อย่างมากมาย เนื่องจากเป็นระบบที่มีความน่าเชื่อถือสูง ใช้งานง่าย นอกจากนั้นตัวกราดตรวจลายนิ้วมือยังมีขนาดเล็กกะทัดรัด สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์ประเภทต่าง ๆ ได้อย่างหลากหลาย ดังภาพที่ 4 และภาพที่ 5



ภาพที่ 4 การประยุกต์ใช้งานของระบบตรวจสอบลายนิ้วมืออัตโนมัติ

ที่มา : Maltoni *et al.* (2003)



ภาพที่ 5 การใช้งานเซนเซอร์ลายนิ้วมือร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ

ที่มา : Maltoni *et al.* (2003)

ปัจจุบันผลิตภัณฑ์ตรวจพิสูจน์ลายนิ้วมืออัตโนมัติที่มีอยู่ในท้องตลาด ล้วนแต่เป็นผลิตภัณฑ์ของบริษัทต่างชาติทั้งสิ้น และสินค้าก็มีราคาแพงเนื่องจากเป็นเทคโนโลยีใหม่ รวมทั้งค่าลิขสิทธิ์ต่าง ๆ ตัวอย่างของผลิตภัณฑ์ที่มีจำหน่ายอยู่ในประเทศไทยได้แสดงไว้ในภาพที่ 6 ดังต่อไปนี้

K8	**AV200A**	**BF660C**	**BioEntry**
Bioclock III	**F4**	**NAC3000**	**Bioclock II**
NDL600	**SMK1**	**YM2000**	**F7**
T5	**AV300**	**Biolock**	**BKS4700**

ภาพที่ 6 ผลิตภัณฑ์ตรวจพิสูจน์ลายนิ้วมืออัตโนมัติที่มีจำหน่ายในประเทศไทย

ที่มา : Y. Matrix Co., Ltd. (2006)

ตารางที่ 1 เปรียบเทียบผลิตภัณฑ์ตรวจพิสูจน์ลายนิ้วมือแบบประมวลผลในตัวเอง

ผลิตภัณฑ์	รูปแบบการใช้งาน		จำนวนลายนิ้วมือที่เก็บได้	เวลาระบุลายนิ้วมือ (วินาที)	ประสิทธิภาพ		การเชื่อมต่อ			ราคาประมาณ (บาท)
	บันทึกเวลาทำงาน	ควบคุมการเปิดปิดประตู			FAR (%)	FRR (%)	Serial & Lan	USB	Wire less	
HIP Bio-clock I	●		1,000	2	0.0001	1	●			20,000
HIP Bio-clock II	●		1,000	2	0.0001	1	●			25,000
HIP Bio-clock III	●	●	1,500	2	0.0001	1	●			32,000
HIP A5	●		1,500	2	0.0001	1	●	●		26,000
HIP F7		●	500	2	0.0001	1	●			30,000
HIP F4	●	●	1,000	2	0.0001	1	●			34,000
HIP Biolock		●	30	0.5	0.0003	0.02				22,000
AV100	●	●	1,000	1	0.00001	0.001	●		●	35,000
AV200	●	●	2,000	1	0.00001	0.001	●		●	40,000
AV300	●	●	6,000	1	0.00001	0.001	●		●	55,000
NAC3000	●		4,000	1	0.00001	0.001	●			30,000
NDL600		●	100	1	-	-			●	15,000

จากตารางที่ 1 จะเห็นว่าเครื่องตรวจพิสูจน์ลายนิ้วมือมีราคาประมาณเครื่องละ 2-3 หมื่นบาท ซึ่งไม่แตกต่างจากการใช้เครื่องตรวจบัตรแถบแม่เหล็กมากนัก แต่จะคุ้มค่ากว่าในระยะยาว เนื่องจากไม่ต้องเสียค่าใช้จ่ายเพิ่มเติมอีกในการทำบัตรประจำตัว ดังนั้นจึงเป็นผลิตภัณฑ์ที่กำลังเป็นที่นิยมอย่างมากในปัจจุบัน ผลิตภัณฑ์เหล่านี้ส่วนใหญ่ถูกนำเข้ามาจากประเทศ จีน เกาหลี และไต้หวัน เนื่องจากประเทศเหล่านี้สามารถผลิตได้ในราคาค่อนข้างต่ำ นอกจากเรื่องราคาแล้วยังต้องแข่งขันกันที่ความแม่นยำในการระบุตัวบุคคล รวมถึงจำนวนผู้ใช้สูงสุดที่สามารถบันทึกลงไปได้ ซึ่งหากงานวิจัยของโครงงานนี้สามารถพัฒนาให้ได้ทั้งคุณภาพและราคาที่ใกล้เคียงกับผลิตภัณฑ์เหล่านี้ได้ จะสามารถดึงลูกค้าให้หันกลับมาใช้สินค้าของไทยเพื่อช่วยลดการนำเข้าสินค้าจากต่างประเทศได้

การแข่งขันขั้นตอนวิธีการตรวจสอบลายนิ้วมือ

แม้ว่าจะมีการศึกษา ค้นคว้า และวิจัยทางด้านขั้นตอนการตรวจสอบลายนิ้วมืออัตโนมัติกันมาเป็นเวลาต่อเนื่องยาวนานกว่า 50 ปีแล้วก็ตาม ปัญหาของขั้นตอนวิธีการเปรียบเทียบลายนิ้วมือยังคงเป็นปัญหาอยู่ในปัจจุบัน ไม่ใช่ปัญหาที่ได้รับการแก้ไขและสำเร็จเรียบร้อยแล้วอย่างที่คนส่วนมากเข้าใจ ถึงแม้ว่าจะมีผลิตภัณฑ์การตรวจพิสูจน์ลายนิ้วมืออัตโนมัติได้ออกสู่ตลาดเป็นระยะเวลาหนึ่งแล้ว ปัญหาของขั้นตอนวิธี (Algorithm) ก็ยังคงมีอยู่จากปัญหาในการใช้งานจริง เนื่องจากลายนิ้วมือมีการบิดเบี้ยวตามลักษณะการกดหรือบิด การแตกแห้งหรือชื้นเกินไปตามลักษณะของผิวของแต่ละบุคคลรวมทั้งสภาพอากาศทำให้คุณภาพของภาพลายนิ้วมือเปลี่ยนแปลงการสัมผัสกับตัวกราดตรวจลายนิ้วมือที่เปื้อนน้ำมันจากนิ้วหรือฝุ่นละอองต่าง ๆ รวมทั้งคราบลายนิ้วมือเก่า รวมทั้งปัญหาการปลอมแปลงลายนิ้วมือ ทำให้งานวิจัยทางด้านนี้ยังคงต้องกระทำต่อเนื่องเพื่อแก้ปัญหาต่าง ๆ ที่เกิดขึ้นต่อไปอีกมาก ตัวอย่างลายนิ้วมือที่มีปัญหาดังแสดงในภาพที่ 7

(ก) รอยแตก (ข) แห้ง (ค) รอยเก่า (ง) ชื้น

ภาพที่ 7 ลายนิ้วมือที่มีปัญหากับเซนเซอร์ในปัจจุบัน (ก) รอยแตก (ข) แห้ง (ค) รอยเก่า (ง) ชื้น

งานวิจัยและพัฒนาทางด้านการตรวจสอบลายนิ้วมืออัตโนมัติยังคงเป็นงานวิจัยที่ได้รับความสำคัญมากในปัจจุบัน ซึ่งสามารถพิจารณาได้ขั้นตอนวิธีใหม่ ๆ ในการแก้ปัญหายังคงจะมีการนำเสนอทางการประชุมวิชาการและวารสารวิชาการนานาชาติอย่างต่อเนื่อง รวมทั้งมีการจัดการแข่งขันขั้นตอนวิธีการตรวจสอบลายนิ้วมือหรือ Fingerprint Verification Competition (FVC) เป็นการแข่งขันในระดับนานาชาติซึ่งร่วมกันจัดโดย University of Bologna (Italy), Michigan State University (USA), and San Jose State University (USA) ซึ่งเปิดโอกาสให้บริษัท สถานศึกษา หรือนักพัฒนาอิสระ ส่งโปรแกรมปฏิบัติการ (Execution Code) โดยไม่ต้องเปิดเผยโปรแกรมต้นฉบับ (Source Code) รวมทั้งสามารถส่งในกรณีที่ไม่เปิดเผยชื่อก็ได้ ทำให้มีผู้เข้าร่วมแข่งขันจำนวนมาก

ขึ้นเรื่อย ๆ ทุกครั้ง โดยมีการแข่งขันมาทั้งหมด 3 ครั้งแล้วคือ ปี ค.ศ. 2000, ค.ศ. 2002, ค.ศ. 2004 แต่ละครั้งในการแข่งขันจะมีฐานข้อมูลลายนิ้วมือให้ทดสอบทั้งหมด 4 ฐานข้อมูลลายนิ้วมือ โดยจะใช้ฐานข้อมูลที่เก็บใหม่จากเซนเซอร์ที่เลือกมาจากบริษัทต่าง ๆ แตกต่างกันรวมทั้งมีลายนิ้วมือแบบสังเคราะห์ขึ้นเองด้วย รายละเอียดสามารถดูได้จากตารางที่ 2 ซึ่งเป็นการเปรียบเทียบเซนเซอร์ฐานข้อมูลลายนิ้วมือแต่ละครั้งของการแข่งขัน จำนวนผู้เข้าแข่งขัน และรายละเอียดของเครื่องคอมพิวเตอร์ที่ใช้ในการประเมิน

ตารางที่ 2 เปรียบเทียบฐานข้อมูล ตัวเซนเซอร์ และฮาร์ดแวร์ที่ใช้เป็นมาตรฐานในการเปรียบเทียบการแข่งขัน FVC

	FVC2000	**FVC2002**	**FVC2004**
Number of Databases	**4**	**4**	**4**
Fingerprint Sensors Used (Approximately 500 dpi)	- ST (Capacitive 256×364) - Keytronic (Optical - 300×300) - Identicator (Optical 448×478) - SfinGe v2.0 (Synthetic Generator 240×320)	- Identix (Optical 388×374) - Biometrika (Optical 296×560) - Precise B. (Capacitive 300×300) - SfinGe v2.51 (Synthetic Generator 288×384)	- CrossMatch V300 (Optical 640×480) - Digital Persona (Optical 328×364) - Atmel Sweep. (Thermal 300×480) - SfinGe v3.0 (Synthetic Generator 288×384)
Algorithm Evaluated	11(+3)	31	67 (41 Open, 26 Light)
HW/SW used for running the evaluation	PentiumIII (450MHz) Windows NT	PentiumIII (933MHz) Windows 2000	AMD Athlon 1600+ (1.41GHz) Windows XP

ที่มา : FVC. (2000), FVC. (2002), FVC. (2004)

การแข่งขันครั้งล่าสุด FVC2004 ได้แบ่งแยกประเภทของขั้นตอนวิธีเป็นแบบ Open กับแบบ Light ซึ่งแบบ Open หมายถึงขั้นตอนวิธีจะมีพื้นที่หน่วยความจำในการประมวลผลไม่จำกัดและขนาดของแม่แบบ (Template) ไม่จำกัด ระยะเวลาในการลงทะเบียนไม่เกิน 10 วินาที (บนเครื่องตามตารางที่ 2) และระยะเวลาการเปรียบเทียบหนึ่งต่อหนึ่งไม่เกิน 5 วินาที แต่แบบ Light จะมีการจำกัดพื้นที่หน่วยความจำในการประมวลผลไม่เกิน 2M ไบต์ และขนาดของแม่แบบไม่เกิน 2K ไบต์ จำกัดระยะเวลาในการลงทะเบียนไม่เกิน 0.5 วินาที (บนเครื่องตามตารางที่ 2) และระยะเวลาการเปรียบเทียบหนึ่งต่อหนึ่งไม่เกิน 0.3 วินาที สังเกตุว่าแบบ Light จะเป็นซอฟต์แวร์ที่เหมาะจะพัฒนาให้ใช้ในฮาร์ดแวร์เฉพาะที่ทำงานได้แบบเอกเทศ (Standalone) มีทรัพยากรหน่วยความจำที่จำกัด ซึ่งเป็นที่น่าสนใจและเกี่ยวข้องกับโครงงานนี้โดยตรง สำหรับแบบ Open นั้นถูกออกแบบให้

ใช้กับคอมพิวเตอร์ส่วนบุคคล ทำงานแบบทรัพยากรไม่จำกัด ผลการแข่งขัน FVC2004 แบบ Open แสดงดังตารางที่ 3 และแบบ Light แสดงดังตารางที่ 4

ตารางที่ 3 ผลการแข่งขัน FVC2004 เรียงตามลำดับ EER (Open)

Rank	Organization	Type	Country	Average EER%
1	Bioscrypt Inc.	Industrial	Canada	2.07
2	Sonda Ltd.	Industrial	Russia	2.10
3	The Chinese Academy of Sciences	Academic	China	2.30
4	Gevarius	Industrial	Russia	2.45
5	Jan Lunter	Independent Developer	France	2.90
6	NITGEN Co.Ltd.	Industrial	Korea	3.13
7	Neurotechnologija Ltd.	Industrial	Lithuania	3.24
8	Suprema Inc.	Industrial	Korea	3.31
9	Anonymous (Expect Siemens)	Industrial	-	3.71
10	IDENCOM Germany GmbH	Industrial	Germany	4.03
11	Beijing HanWang Technology Co., Ltd.	Industrial	China	4.04
12	Deng Guoqiang	Developer	China	4.27
13	Testech Inc.	Industrial	Korea	4.33
14	Anonymous (SAGEM)	Industrial	-	4.41
15	Ariel Unanue	Developer	Argentina	4.49
16	Miaxis Biometrics Co., Ltd.	Industrial	China	4.89
17	Changsha XingTong Technology Development Co., Ltd.	Industrial	China	5.66
18	Integral Ltd.	Industrial	Ukraine	6.12
19	Anonymous	Developer	-	6.23
20	Zaklad Techniki Mikroprocesorowej	Industrial	Poland	6.42
21	Morphosoric	Industrial	Germany	6.56
22	Anonymous	Industrial	-	6.80
23	Anonymous	Industrial	-	7.17
24	Anonymous	Industrial	-	7.27
25	Anonymous	Academic	-	7.48
26	DATAMICRO Co., Ltd.	Industrial	Russia	7.64
27	Anonymous	Academic	-	8.24
28	Anonymous	Academic	-	8.29
29	Futronic Technology Co. Ltd.	Industrial	China	8.30
30	Li Lijuan	Developer	China	8.38
31	Anonymous	Industrial	-	9.57
32	Anonymous	Industrial	-	9.62
33	Nyoun	Industrial	Korea	10.03
34	Anonymous	Industrial	-	11.37
35	Anonymous	Industrial	-	12.29
36	Anonymous	Developer	-	12.68
37	Anonymous	Industrial	-	13.08
38	Anonymous	Developer	-	16.84
39	Anonymous	Academic	-	28.44
40	Anonymous	Industrial	-	36.77
41	Anonymous	Academic	-	37.83

ที่มา : FVC. (2004)

ตารางที่ 4 ผลการแข่งขัน FVC2004 เรียงตามลำดับ EER (Light)

Rank	Organization	Type	Country	Average EER%
1	Suprema Inc.	Industrial	Korea	3.51
2	Ji Hui	Independent Developer	China	3.69
3	Beijing HanWang Technology Co., Ltd.	Industrial	China	3.96
4	Bioscrypt Inc.	Industrial	Canada	4.29
5	Testech Inc.	Industrial	Korea	4.33
6	NITGEN Co.Ltd.	Industrial	Korea	4.86
7	The Chinese Academy of Sciences	Academic	China	4.91
8	Deng Guoqiang	Independent Developer	China	5.26
9	IDENCOM Germany GmbH	Industrial	Germany	5.29
10	Neurotechnologija Ltd.	Industrial	Lithuania	5.64
11	Miaxis Biometrics Co., Ltd.	Industrial	China	6.21
12	Activcard	Industrial	Canada	6.82
13	DATAMICRO Co., Ltd.	Industrial	Russia	8.04
14	Anonymous	Industrial	-	9.05
15	Changsha XingTong Technology Development Co., Ltd.	Industrial	China	9.79
16	Zaklad Techniki Mikroprocesorowej	Industrial	Poland	11.18
17	Anonymous	Independent Developer	-	12.68
18	Li Lijuan.	Independent Developer	China	15.05
19	Anonymous	Industrial	-	21.38
20	Anonymous	Academic	-	26.86
21	Anonymous	Industrial	-	27.96
22	Anonymous	Industrial	-	32.71
23	Integral Ltd.	Industrial	Ukraine	37.72
24	Anonymous	Independent Developer	-	50.00
25	Anonymous	Independent Developer	-	50.00
26	Futronic Technology Co. Ltd.	Industrial	China	50.00

ที่มา : FVC. (2004)

จากตารางจะเห็นได้ว่าค่าที่ต่ำที่สุดของค่าเฉลี่ย EER อยู่ที่ 2.07% ในตารางที่ 3 (Open) และ 3.51% ในตารางที่ 4 (Light) เมื่อนำค่ามาเปรียบเทียบกับการทดสอบเชิงทฤษฎีในการเปรียบเทียบลายนิ้วมือในบทความ (Pankanti, 2002) ซึ่งได้แสดงว่า ความน่าจะเป็นที่สองลายนิ้วมือที่มีจุดมินูเทียร์ หรือจุดแยกกับจุดหยุดในลายนิ้วมือทั้งหมด 36 จุด และมีเหมือนกันจำนวน 12 จุดนั้นเท่ากับ $6.1x10^{-8}$ หรือเทียบเป็นเปอร์เซนต์ความผิดพลาดอยู่ที่ 0.0000061% เห็นว่าช่องห่างระหว่างทฤษฎีกับปฏิบัติยังห่างกันมาก แสดงว่าขั้นตอนวิธียังสามารถพัฒนาได้อีก จึงสรุปได้ว่า งานวิจัยทางด้านขั้นตอนวิธีการตรวจสอบลายนิ้วมือยังคงเป็นปัญหาที่ท้าทายความสามารถของนักวิจัยทั่วโลกต่อไป

การพัฒนาฮาร์ดแวร์ตรวจสอบลายนิ้วมือ

การพัฒนาเทคโนโลยีทางด้านฮาร์ดแวร์การตรวจสอบลายนิ้วมือได้ก้าวหน้าไปมาก โดยสามารถแบ่งได้เป็นการพัฒนาเทคโนโลยีเซนเซอร์ และเทคโนโลยีฮาร์ดแวร์ประมวลผลภาพ ดังมีรายละเอียดดังต่อไปนี้

1. เทคโนโลยีเซนเซอร์ตรวจจับภาพลายนิ้วมือ (Fingerprint Sensor)

เทคโนโลยีเซนเซอร์สำหรับการเก็บภาพลายนิ้วมือยังคงได้รับการพัฒนาอย่างต่อเนื่อง ในปัจจุบันเทคโนโลยีการเก็บภาพลายนิ้วมือแยกเป็นประเภทต่าง ๆ ได้ดังต่อไปนี้

1.1 เทคโนโลยีเซนเซอร์แสง เป็นเทคโนโลยีที่เก่าแก่ที่และใช้หลักของการกดลายนิ้วมือลงบนผิวของปริซึมและใช้ CCD (Charge Coupled Device) หรือกล้องที่เป็นลักษณะของสารกึ่งตัวนำ ในการรับภาพลายนิ้วมือที่สะท้อนมาเรียกเทคนิคนี้ว่า FTIR (Frustrated Total Internal Reflection) จากเทคนิคการรับภาพแบบนี้จะทำให้ได้ภาพเป็นภาพต่อเนื่องหรือวีดิทัศน์ ปัจจุบันเทคโนโลยีแสงได้พัฒนาเป็นแบบบางโดยใช้เทคนิคเส้นใยแสง (Optical Fibers) และแบบที่ใช้กล้อง CCD รับภาพโดยตรง ข้อเสียของระบบนี้คือถ้าผิวของลายนิ้วมือนี้แห้งเกินไปหรือเปียกเกินไป รวมทั้งรอยน้ำมันและฝุ่น ก็จะทำให้ได้ภาพที่ไม่ชัดเจนมีปัญหากับขั้นตอนวิธีในการเปรียบเทียบได้

1.2 เทคโนโลยีเซนเซอร์สารกึ่งตัวนำ เป็นเทคโนโลยีที่สามารถสร้างเป็นวงจรรวมได้ภายในชิปเดียว โดยมีเทคโนโลยีแบบต่าง ๆ อาทิเช่น เซนเซอร์ตัวเก็บประจุ (Capacitive), เซนเซอร์ความร้อน (Thermal), เซนเซอร์สนามไฟฟ้า (Electric Field), และเซนเซอร์แรงกด (Piezoelectric) เป็นต้น เซนเซอร์ที่เป็นที่นิยมมีดังต่อไปนี้

- เซนเซอร์ตัวเก็บประจุ (Fujitsu, 2004) ที่สามารถวัดค่าตัวเก็บประจุที่แตกต่างกันระหว่างส่วนที่เป็นร่องและเป็นเส้นนูนในลายนิ้วมือได้ภายในชิปเดียว ปัญหาของเซนเซอร์ชนิดนี้คือ เสียหายง่ายเมื่อเจอไฟฟ้าสถิต และการขูดขีด สึกหรอ รวมทั้งมีปัญหากับความแห้งและชื้นของลายนิ้วมือ มีฟังก์ชันมาตรฐานในการติดต่อกับอุปกรณ์ทางฮาร์ดแวร์

- เซนเซอร์ความร้อน (Atmel, 2003) เป็นเทคโนโลยีที่อาศัยความแตกต่างของอุณหภูมิของนิ้วมือระหว่างร่องกับเส้นนูนของลายนิ้วมือ เนื่องจากเวลาใช้ถ้าวางนิ้วไว้นานจะทำให้อุณหภูมิปรับเท่ากันได้อย่างรวดเร็ว เซนเซอร์จึงต้องเป็นแบบรูดเพื่อตรวจจับอุณหภูมิที่แตกต่างอย่างรวดเร็วก่อนที่จะไม่เห็นความแตกต่างของอุณหภูมิ เซนเซอร์แบบนี้จะทนไฟฟ้าสถิตได้สูงขึ้น ปัญหาความแห้งและชื้นของลายนิ้วมือก็น้อยลงไป แต่ปัญหาของเซนเซอร์จะอยู่ที่ผู้ใช้ที่ต้องเรียนรู้เทคนิคการรูด และภาพที่ได้ต้องทำการสร้างกลับด้วย

- เทคโนโลยีเซนเซอร์อุลตราซาวน์ (Ultra-Scan, 2006) ใช้เทคนิคของคลื่นอุลตราโซนิคกราดตรวจทั่วผิวของลายนิ้วมือเพื่อวัดความลึกของร่องที่แตกต่างจากเส้นนูน และสร้างภาพลายนิ้วมือจากระยะที่วัดได้ ข้อดีคือไม่มีผลกับลายนิ้วมือที่แห้งหรือเปียก รวมทั้งทนต่อลายนิ้วมือที่เปื้อนน้ำมัน แต่ข้อเสียคือมีขนาดใหญ่และใช้เวลาในการสแกนนาน รวมทั้งมีราคาแพง

เทคโนโลยีต่าง ๆ ที่กล่าวมาแล้วเป็นเทคโนโลยีที่ได้รับการยอมรับและมีผลิตภัณฑ์ออกจำหน่ายในท้องตลาดแล้ว ผลิตภัณฑ์ตรวจสอบลายนิ้วมืออัตโนมัติในปัจจุบัน นิยมใช้เทคโนโลยีเซนเซอร์แสงเนื่องจากมีความทนทานต่อการใช้งาน และเทคโนโลยีสารกึ่งตัวนำเนื่องจากมีราคาถูก ซึ่งมีให้เลือกมากมายในท้องตลาด

2. เทคโนโลยีฮาร์ดแวร์ประมวลผลภาพ (Image Processing Hardware)

ฮาร์ดแวร์ประมวลผลภาพเวลาจริงในปัจจุบัน สามารถใช้คอมพิวเตอร์ส่วนบุคคลในการประมวลผลได้ แต่เนื่องจากไม่ได้สร้างมาเพื่อการประมวลผลภาพโดยเฉพาะ ทำให้มีราคาแพงและกินกำลังไฟฟ้าสูง ไม่เหมาะสมกับงาน การสร้างฮาร์ดแวร์เฉพาะจึงเป็นอีกทางเลือกหนึ่ง เพราะจะทำให้ต้นทุนถูกกว่าและประหยัดไฟกว่า ตัวประมวลผลที่ถูกเลือกใช้ในงานประมวลผลภาพ ได้แก่ โปรเซสเซอร์ ARM (Acorn RISC Machine) และ DSP (Digital Signal Processing) ซึ่งความเร็ว และมีประสิทธิภาพสูงในการประมวลผลภาพ

โปรเซสเซอร์ ARM เป็นตัวประมวลผลชนิด RISC (Reduced Instruction Set Computer) มีลักษณะเด่นที่ สามารถประมวลผลด้วยความเร็วสูง แต่ใช้กำลังงานไฟฟ้าต่ำ สามารถเขียนโปรแกรมบนระบบปฏิบัติการ Linux ที่สามารถนำมาทำงานบนตัวประมวลผลได้ แต่มีโครงสร้างไม่เหมาะกับการประมวลผลที่ซับซ้อน และประมวลผลในแบบจุดตรึง (Fixed-Point)

เท่านั้น จึงเหมาะสำหรับนำไปใช้ในงานประเภทที่ต้องการประหยัดพลังงาน หรือใช้ในงานจำพวกโปรแกรมประยุกต์ ตัวอย่างเช่น Mobile Phone, PDA (Personal Digital Assistant), Networking เป็นต้น

โปรเซสเซอร์ DSP เป็นตัวประมวลที่มีทั้งแบบจุดลอยตัว (Floating-Point) และแบบจุดตรึง (Fixed-Point) มีลักษณะเด่นที่ มีความสามารถทางการคำนวณในเชิงคณิตศาสตร์ ประมวลผลแบบคู่ขนาน (Parallel Architecture) ทำให้มีประสิทธิภาพและความเร็วในการประมวลผลสูงกว่าโปรเซสเซอร์ ARM ในกรณีที่กำหนดความถี่สัญญาณเท่ากัน จึงเหมาะสำหรับนำไปใช้ในงานประเภทที่ต้องการความเร็วสูง และใช้การคำนวณในเชิงคณิตศาสตร์ที่ซับซ้อนสูง ตัวอย่างเช่น การส่งสัญญาณเสียงผ่านระบบเครือข่าย (Voice Over IP), การกำจัดเสียงสะท้อน (Echo Canceler), การประมวลผลภาพดิจิทัล (Digital Image Processing) เป็นต้น

ตัวประมวลผลสัญญาณดิจิทัล TMS320C6000

ตัวประมวลผลสัญญาณดิจิทัลของบริษัท Texas Instruments เริ่มต้นพัฒนาจากเบอร์ TMS32010 ในปี 1982 , TMS320C25 ในปี 1986 และ TMS320C50 ในปี 1991 ในแต่ละรุ่นของตัวประมวลผล (C1x, C2x, C5x) มีลักษณะแตกต่างกันที่ความเร็วในการประมวลผล แต่สิ่งที่เหมือนกันคือ เป็นตัวประมวลผล 16 บิต ประมวลผลแบบจุดตรึง (Fixed-Point) และใช้สถาปัตยกรรม Harvard ซึ่งผู้พัฒนาซอฟต์แวร์สามารถย้ายขั้นตอนวิธีไปยังอีกแพลตฟอร์ม (Platform) หนึ่งได้ ทำให้ไม่เสียเวลาในการพัฒนาให้เข้ากับฮาร์ดแวร์ใหม่ หลังจากนั้นในปี 1980 จึงได้มีการพัฒนาตระกูล C3x และ ตระกูล C4x ตามมา ซึ่งสามารถประมวลผลได้ในแบบจุดลอยตัว (Floating-Point) เหมาะสำหรับงานที่ต้องการความละเอียดสูงมากขึ้น และยังใช้สถาปัตยกรรม Harvard เช่นเดียวกัน

ตัวประมวลผล TMS320C6201 (ตระกูล C62x) ได้ถูกพัฒนาขึ้นในปี 1997 ซึ่งเป็นตัวแรกของตระกูล C6x ที่ประมวลผลแบบจุดตรึง มีความแตกต่างกับตัวประมวลผลตระกูล C1x, C2x และ C5x ตรงที่ ตระกูล C6x นั้นใช้สถาปัตยกรรม Very-Long-Instruction-Word (VLIW) ซึ่งตัวประมวลผลได้ถูกพัฒนาขึ้นให้มีขนาด 32 บิต ดังนั้นจึงไม่สามารถย้ายแพลตฟอร์มกับตัวประมวลผล C1x, C2x และ C5x ได้ หลังจากนั้นตัวประมวลผลตระกูล C67x ซึ่งเป็น

ตัวประมวลผลแบบจุดลอยตัว และตัวประมวลผลตระกูล C64x ซึ่งเป็นตัวประมวลผลแบบจุดยึด จึงได้ถูกพัฒนาขึ้นในเวลาต่อมา

การเลือกใช้ตัวประมวลผลสัญญาณดิจิทัลที่มีการประมวลผลแบบเวลาจริงนั้นขึ้นอยู่กับว่าจะนำไปใช้ในงานในลักษณะใด โดยทั่วไปแล้วอาศัยปัจจัยหลักๆ ในการพิจารณา คือ ราคา ประสิทธิภาพการทำงาน ความเร็วในการประมวลผล พลังงานที่ใช้ ความง่ายต่อการใช้งาน และระยะเวลาในการพัฒนา ตัวประมวลผลในตระกูล C6x ถูกพัฒนาขึ้นโดยเน้นเรื่องของความเร็วในการประมวลผล สามารถทำงานได้สูงถึงระดับ MIPS (Millon Instructions Per Second) เหมาะกับงานที่มีการคำนวณสูง โดยมีตัวประมวลผลตระกูล C62x และ C64x เป็นชนิดจุดตรึง เหมาะสำหรับงานที่กินพลังงานต่ำและใช้แบตเตอรี่ ส่วนตัวประมวลผลในตระกูล C67x เป็นชนิดจุดลอยตัว เหมาะสำหรับงานที่มีความซับซ้อนในการคำนวณสูงและใช้การคำนวณในเชิงคณิตศาสตร์ โดยในตารางที่ 5 และ ตารางที่ 6 ได้แสดงรายการของตัวประมวลผลในตระกูล C6x ทั้งหมดของบริษัท Texas Instruments ในปัจจุบัน เปรียบเทียบให้เห็นถึงความแตกต่างกัน เช่น ความเร็ว เวลาที่ใช้ต่อหนึ่งชุดคำสั่ง พลังงานที่ใช้ พื้นที่หน่วยความจำ รูปแบบการเชื่อมต่อกับภายนอก ลักษณะของตัวชิป (Packaging) และ ราคา ยกตัวอย่างเช่น รุ่น TMS320C6201 ทำงานที่ 200 MHz จะสามารถคำนวณได้ 1600 MIPS หรือ 400 MACs (Million Multiplies and Accumulates Per Second) และรุ่น TMS320C6701 ทำงานที่ 167 MHz จะสามารถคำนวณได้ 1000 MFLOPS (Million Floating-Point Operation Per Second) ในภาพที่ 8 และ ภาพที่ 9 ได้แสดงการเปรียบเทียบความเร็วกับตัวประมวลผลสัญญาณดิจิทัลในบริษัทอื่นๆ เช่น Intel, Analog Devices, Freescale เป็นต้น

สำหรับงานพัฒนาฮาร์ดแวร์อุปกรณ์ตรวจสอบลายนิ้วมืออัตโนมัติ ได้ตัดสินใจเลือกใช้ตัวประมวลผลสัญญาณดิจิทัล TMS320C6713 ซึ่งเป็น DSP ในตระกูล C67x ทำงานที่ความเร็วสัญญาณนาฬิกา 225 MHz จะสามารถคำนวณได้สูงสุด 1.35 GFLOPS หรือ 1800 MIPS สามารถย้ายซอฟต์แวร์ไปพัฒนาบนแพลตฟอร์ม TMS320C6701 TMS320C6711 และ TMS320C6712 ได้ ซึ่งใช้สถาปัตยกรรม VLIW และ TI's VelociTI™ เช่นเดียวกับตัวประมวลผลในตระกูล C6x

ตารางที่ 5 คุณลักษณะของตัวประมวลผลสัญญาณแบบจุดตรึง (TMS320C64x™ DSP Generation)

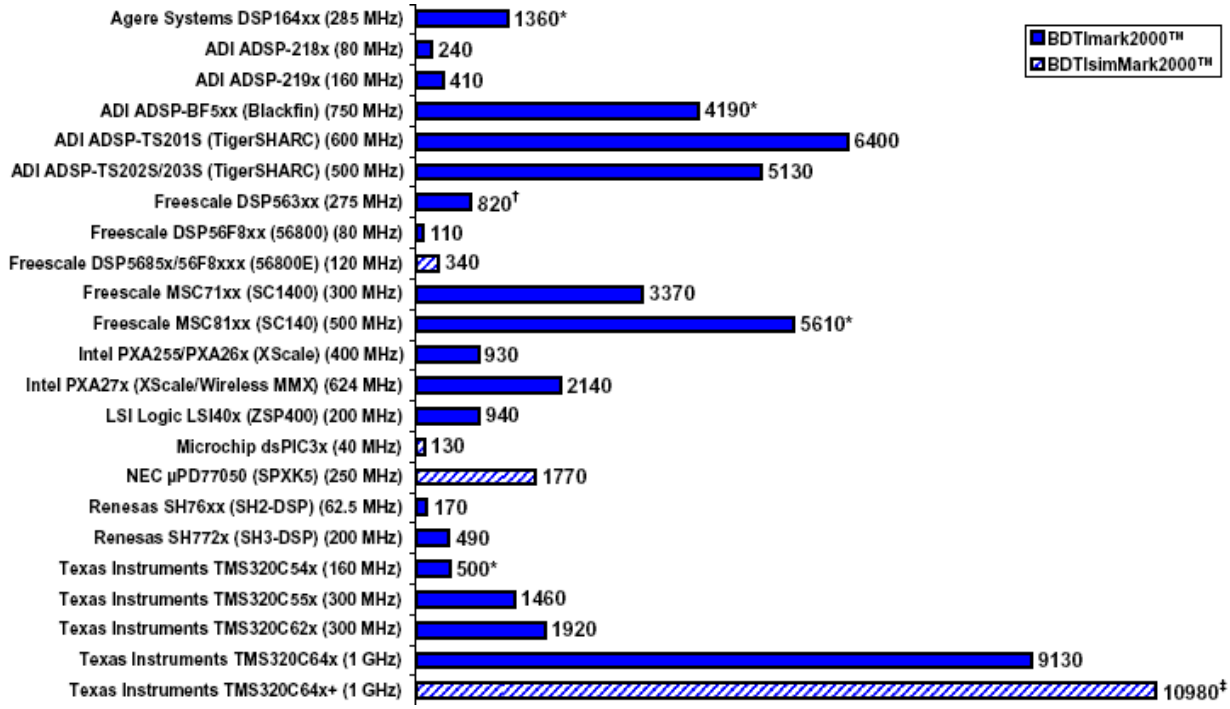
Part Number	Internal RAM (Bytes) L1 Program Cache/ L1 Data Cache/ L2 Unified RAM/Cache	McBSP	Enhanced DMA (Channel)	COM	Timers	MHz	MIPS	Power (W)		Voltage (V)		Packaging	1 KU ($U.S.)
								CPU and L1	Total	Core	I/O		
TMS320C6410GTS400	16K/16K/128K	2	64	HPI 32/16	3	400	3200	0.58	1.0	1.2	3.3	288 BGA	16.89
TMS320C6410ZTSA400	16K/16K/128K	2	64	HPI 32/16	3	400	3200	0.58	1.0	1.2	3.3	288 BGA	16.89
TMS320C6413GTS500	16K/16K/256K	2	64	HPI 32/16	3	500	4000	0.58	1.1	1.2	3.3	288 BGA	27.06
TMS320C6412AGDK5	16K/16K/256K	2	64	PCI/HPI/EMAC	3	500	4000	0.66	1.3	1.2	3.3	548 BGA	39.88
TMS320C6412AGNZ5	16K/16K/256K	2	64	PCI/HPI/EMAC	3	500	4000	0.66	1.3	1.2	3.3	548 BGA	39.88
TMS320C6412AGDK6	16K/16K/256K	2	64	PCI/HPI/EMAC	3	600	4800	0.93	1.9	1.4	3.3	548 BGA	43.91
TMS320C6412AGNZ6	16K/16K/256K	2	64	PCI/HPI/EMAC	3	600	4800	0.93	1.9	1.4	3.3	548 BGA	43.91
TMS320C6412AGDK7	16K/16K/256K	2	64	PCI/HPI/EMAC	3	720	5760	0.93	2.15	1.4	3.3	548 BGA	68.69
TMS320C6412AGNZ7	16K/16K/256K	2	64	PCI/HPI/EMAC	3	720	5760	0.93	2.15	1.4	3.3	548 BGA	68.69
TMS320C6418GTS600	16K/16K/512K	2	64	HPI 32/16	3	600	4800	0.82	1.7	1.4	3.3	288 BGA	50.34
TMS320C6418ZTSA500	16K/16K/512K	2	64	HPI 32/16	3	500	4000	0.58	1.1	1.4	3.3	288 BGA	50.34

ที่มา : Texas Instruments. (2006)

ตารางที่ 6 คุณลักษณะของตัวประมวลผลแบบ Floating-Point DSPs (TMS320C67x™ DSP Generation)

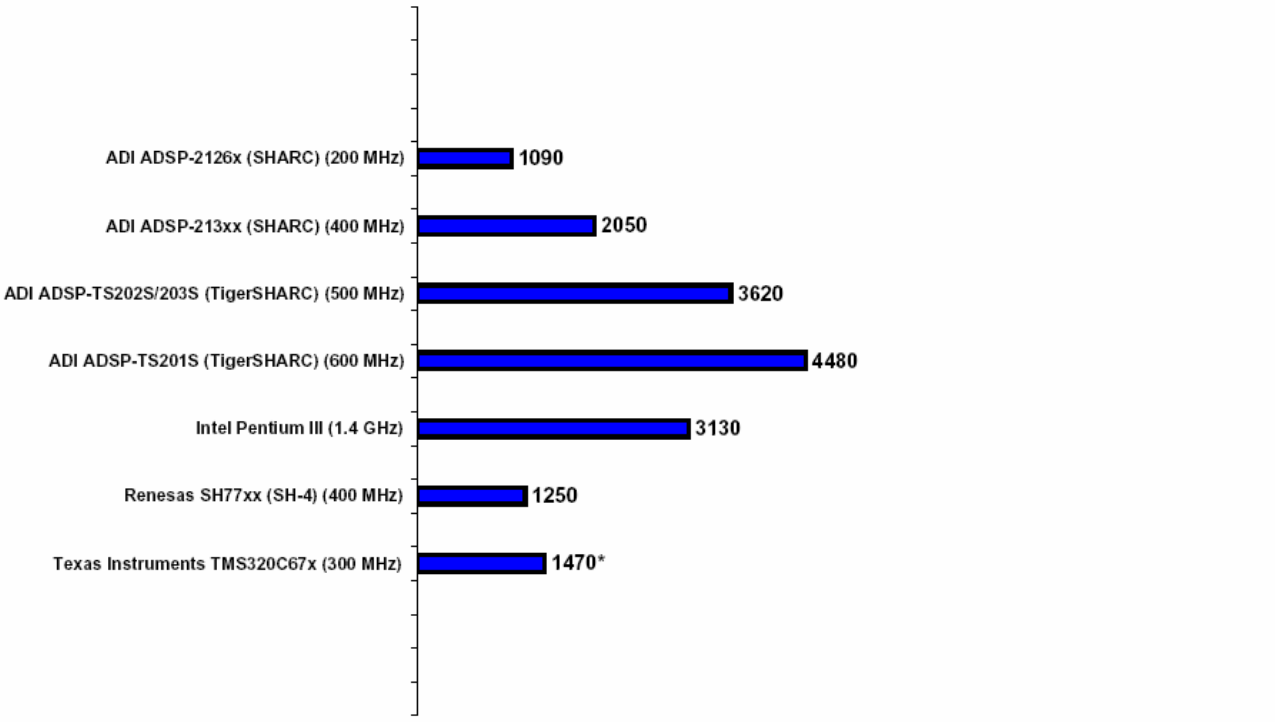
Part Number	RAM (Bytes) Data/Prog	McBSP	McASP	DMA	COM	SPI/ I^2C	MHz	Cycle (ns)	MFLOPS	Typical Activity Total Internal Power (W)	Voltage (V)		Packaging	1 KU ($U.S.)
											Core	I/O		
TMS320C6712DGDP150	4K/4K/64K	2	-	16	-	-	150	6.7	900	0.7	1.2	3.3	272 BGA	13.77
TMS320C6722RFP200	32K/128/284K	-	2	MAX	-	2/2	200	5	1200	0.8	1.2	3.3	144 PQFP	11.24
TMS320C6722RFPA225	32K/128/284K	-	2	MAX	-	2/2	225	4.4	1350	0.8	1.2	3.3	144 PQFP	13.05
TMS320C6722RFP250	32K/128/284K	-	2	MAX	-	2/2	250	4	1500	0.9	1.2	3.3	144 PQFP	13.05
TMS320C6711DGDP200	4K/4K/64K	2	-	16	HPI	-	200	5	1200	0.9	1.2	3.3	272 BGA	17.12
TMS320C6726RFPA225	32K/256/384K	-	3	MAX	-	2/2	225	4.4	1350	0.8	1.2	3.3	144 PQFP	15.93
TMS320C6726RFP250	32K/256/384K	-	3	MAX	HPI	2/2	250	4	1500	0.9	1.2	3.3	144 PQFP	15.93
TMS320C6713BPYP200	4K/4K/64K	2	2	16	UHPI	-	200	5	1200	1.0	1.2	3.3	208 PQFP	20.02
TMS320C6727GDHA250	32K/256/384K	-	3	MAX	UHPI	2/2	300	3.3	1800	1.1	1.2	3.3	256 BGA	22.54
TMS320C6727ZDH250	32K/256/384K	-	3	MAX	UHPI	2/2	250	4	1500	1.0	1.2	3.3	256 BGA	19.94
TMS320C6727ZDH300	32K/256/384K	-	3	MAX	HPI	2/2	300	3.3	1800	1.1	1.2	3.3	256 BGA	22.54
TMS320C6713BPYP225	4K/4K/256K	2	2	16	HPI	-	225	4.4	1350	1.0	1.2	3.3	208 PQFP	23.09
TMS320C6713BGDP225	4K/4K/256K	2	2	16	HPI	-	225	4.4	1350	1.1	1.4	3.3	272 BGA	26.30
TMS320C6713BGDP300	4K/4K/256K	2	2	16	HPI	-	300	3.3	1800	1.6	1.8	3.3	272 BGA	34.97
TMS320C6701GLC150	64K/64K	2	-	4	HPI	-	150	6.7	900	1.3	1.9	3.3	352 BGA	82.24

ที่มา : Texas Instruments. (2006)



ภาพที่ 8 การเปรียบเทียบความเร็วตัวประมวลผล Fixed-Point DSP ของ BDTImark2000

ที่มา : BDTImark2000. (2006)

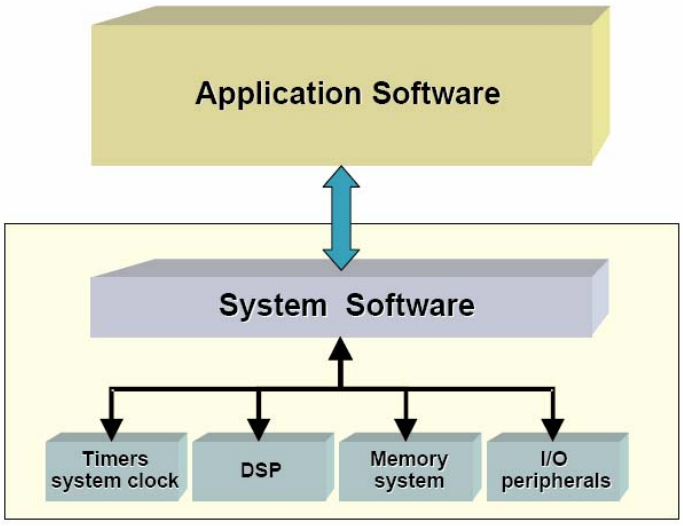


ภาพที่ 9 การเปรียบเทียบความเร็วตัวประมวลผล Floating-Point DSP ของ BDTImark2000

ที่มา : BDTImark2000. (2006)

การเขียนโปรแกรมประยุกต์ในเวลาจริง (Real-time Software Applications) บนตัวประมวลผลสัญญาณดิจิทัลของบริษัท Taxes Instruments ได้เตรียมเคอร์เนล (Kernel) ไบออสบนตัวประมวลผลสัญญาณดิจิทัล (DSP/BIOS) ซึ่งเป็นเครื่องมือที่ผู้พัฒนาโปรแกรมบนตัวประมวลผลสัญญาณดิจิทัลใช้จัดการโครงสร้างการทำงานภายในโปรแกรมบนตัวประมวลผลสัญญาณดิจิทัล และจัดการทรัพยากรต่างๆภายในระบบ นอกจากนั้นยังสามารถทำการวิเคราะห์ประสิทธิภาพของระบบได้อีกด้วย

ซอฟต์แวร์บนตัวประมวลผลสัญญาณดิจิทัลนั้นจะแบ่งตามหน้าที่การทำงานออกได้เป็นสองส่วน ดังภาพที่ 10 ซึ่งได้แก่ ซอฟต์แวร์ระบบ (System Software) และซอฟต์แวร์งานประยุกต์ (Application Software) ซึ่งส่วนของซอฟต์แวร์ระบบจะทำหน้าที่จัดการทรัพยากรต่างของระบบรวมทั้งอุปกรณ์รอบข้าง (Peripherals) ในระบบด้วย ส่วนซอฟต์แวร์งานประยุกต์นั้นเป็นส่วนที่ทำงานทางด้านประมวลผลต่างๆ



ภาพที่ 10 ส่วนประกอบของซอฟต์แวร์ระบบฝังตัว
ที่มา : Texas Instruments. (2001)

งานต่าง ๆ บนตัวประมวลผลสัญญาณดิจิทัลนั้นต้องการซอฟต์แวร์ระบบเพื่อช่วยในการจัดการกับทรัพยากรต่าง ๆ ในระบบ รวมทั้งทำหน้าที่กำหนดค่าเริ่มต้น (Initialization) และควบคุมส่วนของฮาร์ดแวร์ต่าง ๆ ในระบบด้วย ซึ่งอาจกล่าวได้ว่าซอฟต์แวร์ระบบเป็นตัวกลางในการเข้าถึง

ระหว่างซอฟต์แวร์งานประยุกต์ และทรัพยากรระบบ ดังนั้นเมื่อมีการนำซอฟต์แวร์งานประยุกต์ไปใช้กับฮาร์ดแวร์แพลตฟอร์ม (Platform) อื่น ๆ จะต้องคำนึงถึงคุณลักษณะของระบบนั้น ๆ ด้วย

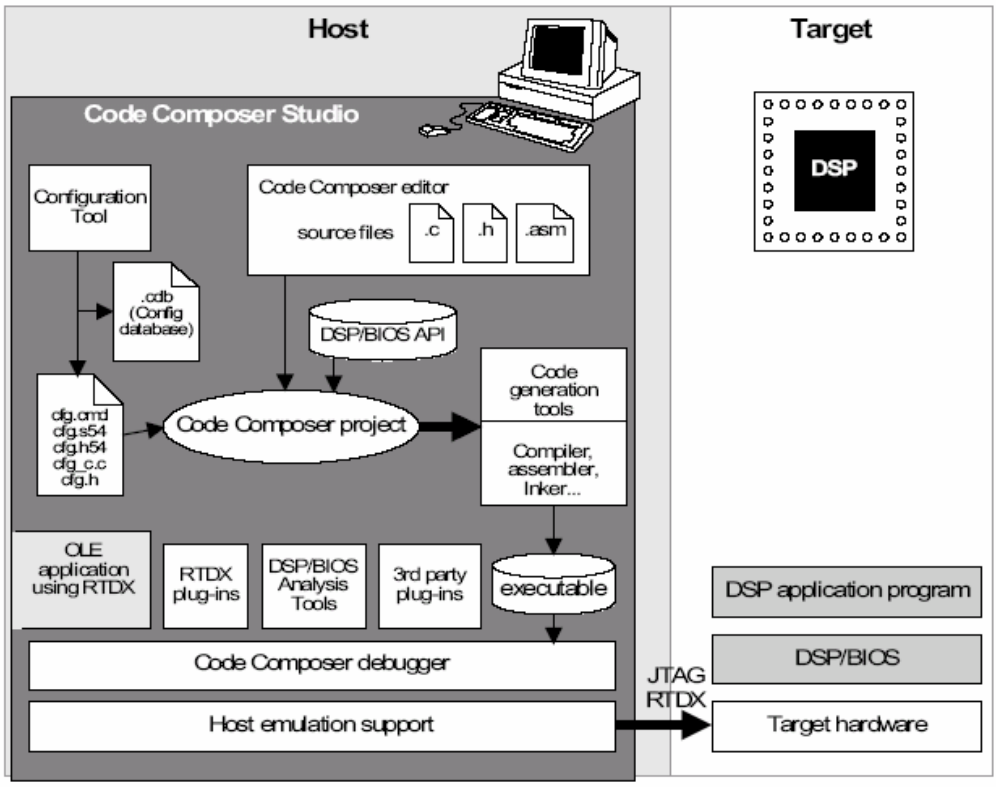
ในระบบทั่วไปซอฟต์แวร์ระบบนั้นประกอบด้วยฟังก์ชันการกำหนดค่าเริ่มต้นฮาร์ดแวร์ การใช้งานอุปกรณ์รอบข้าง และงานบริการการขัดจังหวะของฮาร์ดแวร์ (Hardware Interrupt Service) นอกจากนี้ยังทำการจัดลำดับ (Scheduling) การทำงานของงานย่อย (Thread) ต่างๆแบบเวลาจริง (Real-time) และควบคุมการใช้ทรัพยากรให้สามารถใช้งานได้ในลักษณะพร้อมกัน (Concurrent) การจัดการและควบคุมระบบของซอฟต์แวร์ระบบนั้น สามารถเขียนโปรแกรมโดยใช้ไบออสบนตัวประมวลผลสัญญาณดิจิทัล ซึ่งจะเตรียมเครื่องมือในการจัดการส่วนต่างๆในระบบ การใช้ไบออสบนตัวประมวลผลสัญญาณดิจิทัลมาเขียนโปรแกรมควบคุมระบบบนตัวประมวลผลสัญญาณดิจิทัลจะมีข้อดีต่างๆดังนี้

- สามารถช่วยจัดการการประมวลผลให้เป็นแบบหลายงานย่อย (Multi-thread) ได้ง่าย
- มีฟังก์ชันมาตรฐานในการติดต่อกับอุปกรณ์ทางฮาร์ดแวร์
- สามารถกำหนดการใช้งานทรัพยากรระบบได้
- สามารถทำงาน และวิเคราะห์การทำงานในเวลาจริงได้
- ง่ายต่อการย้ายโปรแกรมไปทำงานที่ตัวประมวลผลดิจิทัลแพลตฟอร์มอื่น เนื่องจากมีการใช้ไบออสบนตัวประมวลผลสัญญาณดิจิทัลที่คล้ายกัน

ไบออสบนตัวประมวลผลสัญญาณดิจิทัลเป็นเครื่องมือที่มาพร้อมกับชุดเครื่องมือรวมสำหรับการเขียนโปรแกรม (Code Composer Studio) ของบริษัท TI โดยที่ชุดเครื่องมือรวมสำหรับการเขียนโปรแกรมนั้นเป็นโปรแกรมที่ใช้ในการเขียน และสร้างซอฟต์แวร์สำหรับทำงานบนตัวประมวลผลสัญญาณดิจิทัล และแก้จุดบกพร่อง (Debug) ต่าง ๆ ของซอฟต์แวร์ ซึ่งส่วนประกอบของไบออสบนตัวประมวลผลสัญญาณดิจิทัลสามารถแสดงได้ดังภาพที่ 11

ระบบบนตัวประมวลผลสัญญาณดิจิทัลจะแบ่งเป็น 2 ส่วน คือ เครื่องแม่ข่าย (Host) และอุปกรณ์เป้าหมาย (Target) โดยที่แม่ข่ายคือชุดเครื่องมือรวมสำหรับการเขียนโปรแกรมบนเครื่องคอมพิวเตอร์ ซึ่งผู้พัฒนาสามารถเขียนโปรแกรมด้วยภาษาซี (C) หรือซีพลัสพลัส (C++) หรือแอสแซมบลี (Assembly) ได้ และในชุดเครื่องมือรวมสำหรับการเขียนโปรแกรมจะ

ประกอบด้วยส่วนต่อประสานโปรแกรมประยุกต์สำหรับไบออสบนตัวประมวลผลสัญญาณดิจิทัล (DSP/BIOS API) ซึ่งเป็นฟังก์ชันในการติดต่อหรือควบคุมไบออสบนตัวประมวลผลสัญญาณดิจิทัล



ภาพที่ 11 ส่วนประกอบต่างๆของไบออสบนตัวประมวลผลสัญญาณดิจิทัล
ที่มา : Texas Instruments. (2002)

เครื่องมือโครงแบบ (Configuration tool) เป็นเครื่องมือที่ใช้ในการกำหนดคุณลักษณะของอ็อบเจกต์ (Objects) ต่างๆในโปรแกรม ตัวแปลโปรแกรม (Compiler) ตัวแปลโปรแกรมภาษาแอสแซมบลี (Assembler) และโปรแกรมเชื่อมโยง (Linker) เครื่องมือต่าง ๆ เหล่านี้ใช้ในการแปลโปรแกรมภาษาซี หรือแอสแซมบลี และเชื่อมโยงไฟล์ต่าง ๆ ตามลำดับ เมื่อได้ไฟล์ที่สามารถทำงานบนตัวประมวลผลได้แล้วไฟล์นั้นจะถูกส่งไปยังอุปกรณ์เป้าหมายหรือตัวประมวลผลสัญญาณดิจิทัลผ่านทางเจแทค (JTAG) หรืออาร์ทีดีเอ็กซ์ (RTDX) และขณะที่ตัวประมวลผลกำลังทำงานอยู่ ชุดเครื่องมือรวมสำหรับการเขียนโปรแกรมได้เตรียมเครื่องมือวิเคราะห์ไบออสบนตัวประมวลผลสัญญาณดิจิทัล (DSP/BIOS Analysis Tool) ในการวิเคราะห์ระบบขณะทำงานในเวลาจริงได้

อุปกรณ์และวิธีการ

อุปกรณ์

1. เครื่องคอมพิวเตอร์ส่วนบุคคล
2. ชุดเครื่องมือรวมสำหรับการเขียนโปรแกรม (Code Composer Studio) รุ่นที่ 2.2
3. ชุดเครื่องมือรวมสำหรับการเขียนโปรแกรม (Xilinx ISE) รุ่น 6.3 i
4. โปรแกรมออกแบบลายวงจร PADS2004
5. สายดาวน์โหลดข้อมูล JTAG Emulator ของบริษัท Spectrum Digital
6. เครื่องบัดกรีชิปชนิด BGA/CSP Array Package ซึ่งเป็นของศูนย์วิจัย NECTEC
7. ส่วนประกอบฮาร์ดแวร์สำหรับพัฒนาฮาร์ดแวร์ตรวจสอบลายนิ้วมือ เช่น วงจรรวมตัวประมวลผลสัญญาณดิจิทัล วงจรรวมหน่วยความจำ เป็นต้น
8. ชุดเครื่องมือลงอุปกรณ์ฮาร์ดแวร์ เช่น เครื่องบัดกรีแบบปรับอุณหภูมิ กล้องจุลทรรศน์
9. เครื่องมือวัดต่าง ๆ เช่น ออสซิลโลสโคป และดิจิทัลมัลติมิเตอร์

วิธีการ

แผนการทำวิจัย

ในการออกแบบและพัฒนาระบบฮาร์ดแวร์ตรวจสอบลายนิ้วมือ สามารถแบ่งการพัฒนาได้ 3 ส่วนใหญ่ คือ การสร้างฮาร์ดแวร์ต้นแบบ การเขียนโปรแกรมควบคุมฮาร์ดแวร์ และการทดสอบ ดังแสดงในภาพที่ 12 โดยมีรายละเอียดของแต่ละส่วนดังต่อไปนี้

1. การสร้างฮาร์ดแวร์ต้นแบบ (Hardware Prototype Development)

การออกแบบฮาร์ดแวร์ต้นแบบเป็นส่วนที่จะต้องลงทุนสูงและมีความเสี่ยงสูงที่จะเกิดปัญหาล่าช้า เนื่องจากถ้าเกิดความผิดพลาดต้องย้อนกระบวนการทั้งหมดใหม่อีกครั้ง ดังนั้นถ้าต้องการลดความเสี่ยงจากความผิดพลาดในการออกแบบ จะต้องมีการวางแผนเตรียมการเบื้องต้นในการออกแบบอย่างดีและมีจุดตรวจสอบทุกขั้นตอน โดยมีรายละเอียดขั้นตอนการทำงานดังต่อไปนี้

1.1 การวางแผนเบื้องต้น (Preliminary Hardware Design Planning)

เป็นการวางแผนการพัฒนาฮาร์ดแวร์ โดยเริ่มจากการมองระบบออกเป็นส่วนย่อย ออกแบบการเชื่อมต่อของแต่ละส่วนย่อย และระบุรายละเอียดของแต่ละส่วนย่อย โดยเขียนเป็นบล็อกผังงาน (Block Diagram) ของทั้งระบบ สรุปงานแต่ละส่วนว่ามีอินพุตและเอาต์พุตเป็นอย่างไร

1.2 การเลือกอุปกรณ์และทำการเปรียบเทียบ (Component Search, Comparison and Selection)

เป็นการหาอุปกรณ์ในท้องตลาดในปัจจุบันและในอนาคตอันใกล้ เพื่อทำการเปรียบเทียบอุปกรณ์ต่าง ๆ ทั้งทางด้านความเหมะสมกับระบบที่วางแผนไว้ ความง่ายต่อการใช้งาน ง่ายต่อการหาซื้อ ความทนทานต่อการใช้งาน ราคา และการสนับสนุนต่าง ๆ จากบริษัท รวมทั้งความมั่นคงของบริษัท

1.3 การศึกษาอุปกรณ์ และวงจรสำเร็จรูป (Chips and Component Study)

เป็นการศึกษาตัววงจรรวมและอุปกรณ์ต่าง ๆ เพื่อการออกแบบการเชื่อมต่อฮาร์ดแวร์ ประกอบด้วยการเขียน การเชื่อมต่อของอุปกรณ์ต่าง ๆ และการตรวจสอบด้วยแผนภาพสัญญาณเชิงเวลา (Timing Diagram) ของแต่ละอุปกรณ์

1.4 การออกแบบวงจร และแผนภาพสัญญาณเชิงเวลา (Hardware Design Schematic Diagram and Timing Diagram Design)

เป็นการเขียนแผนภาพวงจร (Schematic Diagram) ลงในซอฟต์แวร์เพื่อเตรียมการออกแบบแผงวงจรลายทองแดง และการตรวจสอบด้วยแผนภาพสัญญาณเชิงเวลาระหว่างอุปกรณ์ที่นำมาเชื่อมต่อกัน

1.5 การวางอุปกรณ์บนแผงวงจรลายทองแดง (Chips and Components Layout)

เป็นการวางตัวอุปกรณ์ต่าง ๆ ลงบนแผงวงจรพิมพ์ (PCB; Printed Circuit Board) การพิจารณาข้อกำหนดและการรบกวนทางความถี่ในการวางอุปกรณ์ รวมทั้งการกำหนดขนาดของ PCB ให้เหมาะสม

1.6 การออกแบบวงจรลายทองแดง (PCB Design and Routing)

เป็นการออกแบบลายวงจรบนแผงวงจรพิมพ์ โดยมีหลักการตามหนังสืออ้างอิง (Archambeault, 2002) และ (Montrose, 1996)

1.7 การทำวงจรลายทองแดง (PCB Development)

เป็นการสร้างลายวงจรบนแผงวงจรพิมพ์ ตามที่ได้ออกแบบไว้ สำหรับงานวิจัยนี้จะจัดจ้างบริษัทที่ทำงานด้านนี้โดยเฉพาะให้ทำแผ่นวงจรลายทองแดงให้

1.8 การประกอบฮาร์ดแวร์ และการทดสอบ (Hardware Composition and Testing)

เป็นการประกอบวงจรและทดสอบจุดเชื่อมต่อต่าง ๆ และการทดสอบเบื้องต้นก่อนการทดสอบด้วยโปรแกรมทดสอบ

2. การเขียนโปรแกรมควบคุมฮาร์ดแวร์ (Software Control Library)

โปรแกรมควบคุมฮาร์ดแวร์ต้องอยู่ในความรับผิดชอบของผู้พัฒนาฮาร์ดแวร์ เนื่องจากต้องออกแบบเพื่อใช้ในการทดสอบว่าฮาร์ดแวร์ทำงานตามที่ได้ออกแบบไว้หรือไม่ โดยทั่วไปแล้วจุดนี้จะเป็นรอยต่อระหว่างผู้พัฒนาฮาร์ดแวร์ กับผู้พัฒนาซอฟต์แวร์ จะต้องกำหนดให้ชัดเจนและไม่คลุมเครือ และต้องกำหนดมาตรฐานขึ้น เป็นอันดับต้น ๆ ในการทำงานพัฒนา เพื่อให้การพัฒนาทั้งสองด้านไม่สะดุด และทำงานร่วมกันได้อย่างเป็นระบบ โดยมีขั้นตอนในรายละเอียดดังต่อไปนี้

2.1 การออกแบบโปรแกรมทดสอบฮาร์ดแวร์ (Hardware Test Program Design)

เป็นการวางแผนการทดสอบฮาร์ดแวร์ด้วยการเขียนโปรแกรมควบคุม จะประกอบด้วยผังงาน (Flowchart) โปรแกรมการทดสอบและรายละเอียดการทดสอบ และข้อสรุปโปรแกรมแต่ละส่วนจะสามารถตรวจสอบอะไรได้บ้างในระบบฮาร์ดแวร์

2.2 การทดสอบฮาร์ดแวร์โดยโปรแกรม (Hardware Testing Program)

เป็นการเขียนโปรแกรมทดสอบฮาร์ดแวร์ตามที่ได้วางแผนไว้ และทดสอบกับฮาร์ดแวร์โดยตรง จะประกอบด้วยการทำงานของฮาร์ดแวร์ที่ทำงานตามฟังก์ชันทดสอบ (Test Function) ต่าง ๆ ที่ได้วางแผนไว้ใน 2.1

2.3 การออกแบบจำลองการทำงานของวงจรรวม (CPLD Design and Simulation)

เป็นการออกแบบวงจรรวม CPLD (Complex Programmable Logic Device) ในการทำงานตามฟังก์ชั่นต่าง ๆ ที่ต้องการในฮาร์ดแวร์ จะประกอบด้วย บล็อกผังงาน แผนภาพสัญญาณเชิงเวลา และทำการจำลองการทำงาน (Simulation)

2.4 การทดสอบวงจรรวมแบบโปรแกรมได้ (CPLD Implementation and Testing)

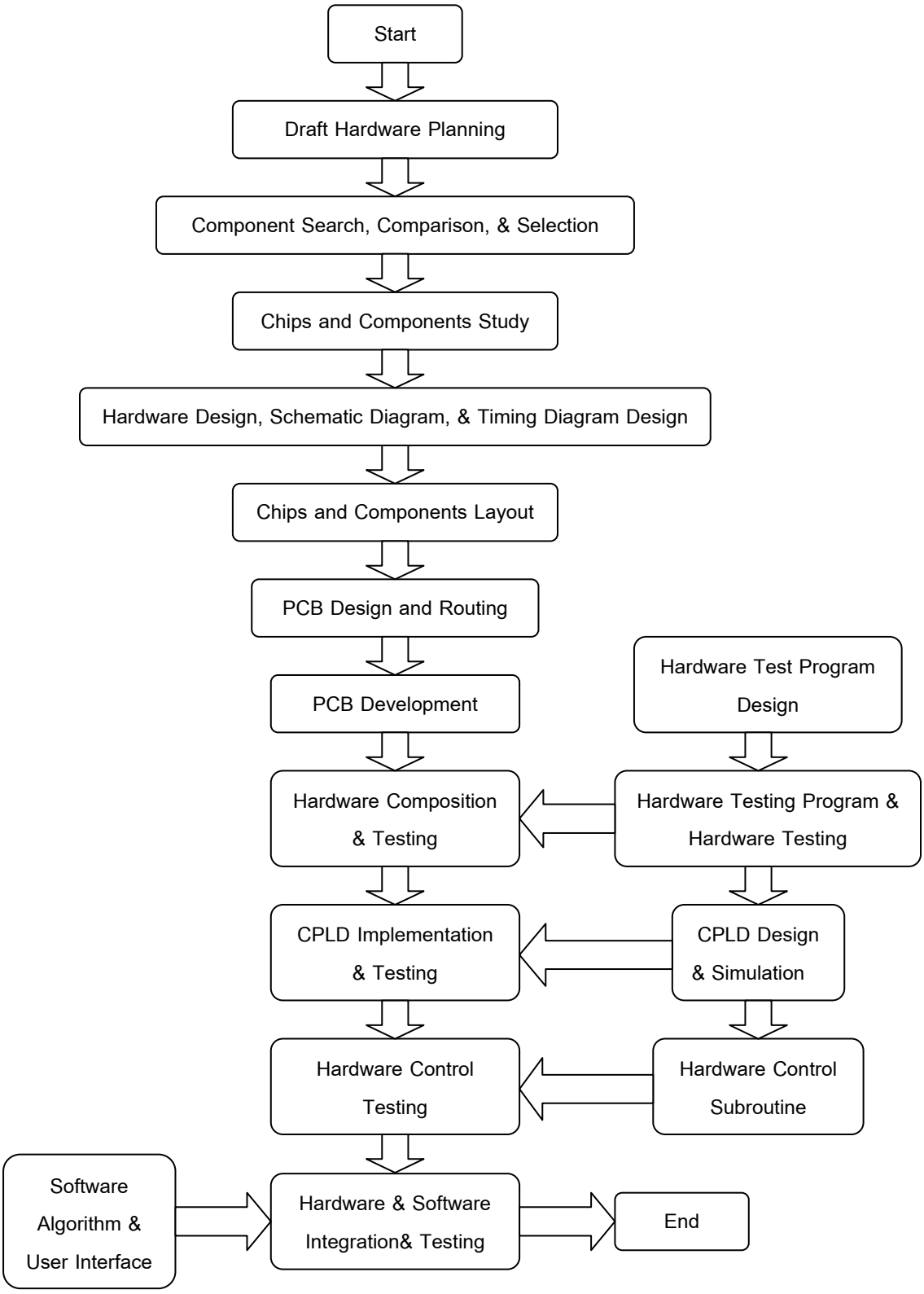
เป็นการทดสอบวงจรรวม CPLD กับระบบฮาร์ดแวร์จริง ให้ทำงานได้ตามหน้าที่ที่ได้ออกแบบไว้

2.5 โปรแกรมย่อยการควบคุมฮาร์ดแวร์เฉพาะ (Hardware Control Subroutine)

เป็นการเขียนโปรแกรมย่อย (Subroutine) ควบคุมในระดับฮาร์ดแวร์ เพื่อสร้างเป็นคลังโปรแกรม (Library) จะประกอบด้วยผังงานโปรแกรมการควบคุม และผลการทดสอบโปรแกรมการควบคุม

2.6 การทดสอบการรวมระบบ (Hardware and Software Integration and Testing)

เป็นขั้นตอนสุดท้ายในการทดสอบการทำงานทั้งระบบ รวมทั้งการทดสอบขั้นตอนวิธี และโปรแกรม ที่ทางด้านวิศวกรรมซอฟต์แวร์เขียนอยู่ในระดับบนให้ทำงานร่วมกับระดับล่างได้ตามที่ได้ออกแบบไว้



ภาพที่ 12 ขั้นตอนการออกแบบและพัฒนาฮาร์ดแวร์ต้นแบบ

การออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

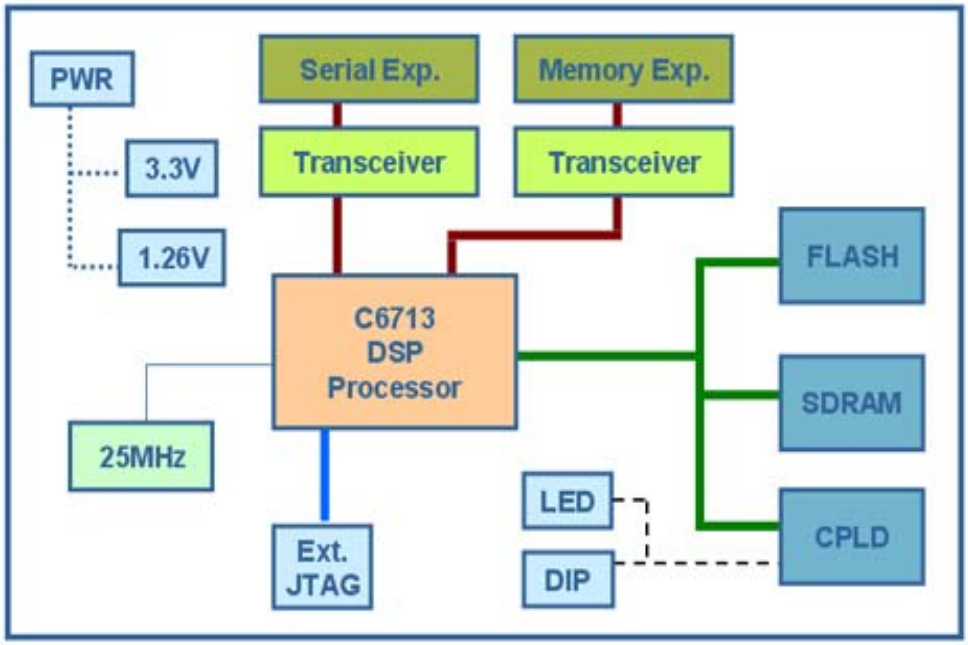
การประยุกต์ใช้งานทางด้านประมวลภาพดิจิทัลในระบบตรวจสอบลายนิ้วมือนั้น ผู้พัฒนาจะต้องออกแบบระบบฮาร์ดแวร์ที่เหมาะสมกับการใช้งานในที่ต่างๆ เช่น การตรวจสอบลายนิ้วมือสำหรับการเข้าถึงสถานที่หรือข้อมูล ผู้ออกแบบควรเลือกอุปกรณ์ฮาร์ดแวร์ที่มีขนาดเล็ก และใช้พลังงานต่ำ เนื่องจากอุปกรณ์ประมวลผลมีหลายชนิด ตัวประมวลผลสัญญาณดิจิทัล (Digital Signal Processor, DSP) เป็นอุปกรณ์ที่เหมาะสม เพราะมีขนาดเล็ก และสามารถออกแบบให้ติดต่อกับอุปกรณ์ภายนอกได้หลายรูปแบบ ซึ่งการนำตัวประมวลผลสัญญาณดิจิทัลมาประยุกต์ใช้งานทางด้านไบโอเมตริก จะช่วยลดขนาดของอุปกรณ์ทางด้านฮาร์ดแวร์ และทำให้สะดวกในการติดตั้งระบบในสถานที่ต่างๆ

การออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลนี้ ต้องออกแบบให้เหมาะสมกับประเภทของงานซึ่งลักษณะของงานที่ต้องทำ ได้แก่ การอ่านภาพลายนิ้วมือ การประมวลผลภาพลายนิ้วมือ การจัดการฐานข้อมูลขนาดใหญ่ การติดต่อผู้ใช้งาน การควบคุมการเปิดประตู และ การจัดการการสื่อสารผ่านพอร์ตอนุกรม RS232 เป็นต้น ซึ่งการทำงานค่อนข้างเป็นอิสระต่อกันและต้องใช้ความเร็วในการประมวลผล ทำให้ต้องเลือกใช้อุปกรณ์ที่สามารถทำงานทั้งหมดนี้ได้อย่างเหมาะสมที่สุด การออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลได้ถูกอธิบายไว้ในหัวข้อต่าง ๆ ดังต่อไปนี้

1. โครงสร้างของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

สำหรับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลต้นแบบ ได้ถูกออกแบบให้ใช้ตัวประมวลสัญญาณดิจิทัล (DSP) เบอร์ TMS320C6713 ของบริษัท Texas Instrument ทำงานที่ความเร็วสัญญาณนาฬิกา 225 MHz ประกอบไปด้วยหน่วยความจำแฟลช (Flash ROM) ขนาด 2M ไบต์ หน่วยความจำ SDRAM (Synchronized Dynamic Random Access Memory) ขนาด 16M ไบต์ วงจรรวม CPLD (Complex Programmable Logic Device) ขนาด 3,200 Gate รวมทั้งวงจรรวมป้องกันการค้าง (Watchdog) และวงจรรวมฐานเวลาจริง (Real-Time-Clock) มีชื่อเรียกว่าฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล (Digital Signal Processor Board) ถูกสร้างขึ้นเพื่อทดสอบประสิทธิภาพและใช้งานร่วมกับซอฟต์แวร์ประมวลผลลายนิ้วมือในระบบตรวจสอบลายนิ้วมือ เครื่องตรวจสามารถแยกกันทำงานเป็นเอกเทศ (Stand Alone) โดยอัตโนมัติ หรือทำงานเชื่อมโยงข้อมูลเป็นเครือข่าย (Network) ก็ได้ โดยมีคอมพิวเตอร์เป็นแม่ข่าย 1 เครื่องที่สามารถพ่วงกับเครื่องอ่านได้

255 ตัว ผ่านมาตรฐานการสื่อสารผ่านพอร์ตอนุกรม RS232 หรือ เครือข่ายอีเธอร์เน็ต (Ethernet) โดยต่อผ่านฮาร์ดแวร์เชื่อมต่อเครือข่าย (LAN Board) สามารถใช้งานร่วมกับฮาร์ดแวร์อื่นๆ อาทิเช่น ฮาร์ดแวร์กราดตรวจลายนิ้วมือ (Fingerprint Capture Board) ฮาร์ดแวร์แป้นพิมพ์และแสดงผล (I/O Processor Board) ฮาร์ดแวร์ควบคุมการเปิดปิดประตู (Door-lock Control Board) เพื่อติดต่อกับผู้ใช้และสั่งเปิดหรือปิดประตูได้ โครงสร้างของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลแสดงดังภาพที่ 13

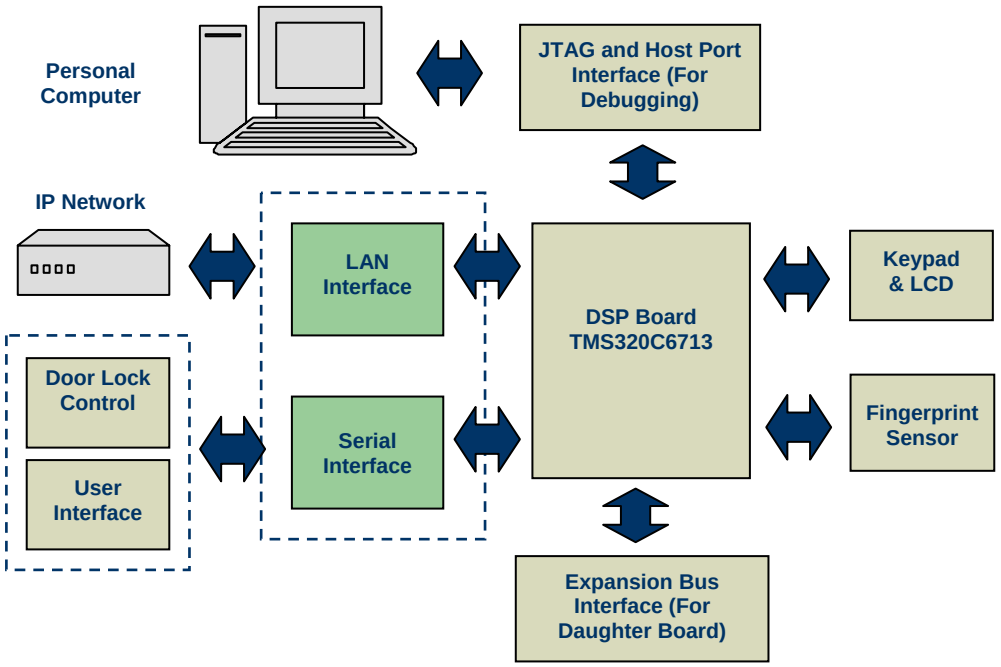


ภาพที่ 13 ส่วนประกอบต่างๆของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

สำหรับวงจรรวมที่ใช้ในแผงวงจร มีดังนี้

1) TMS320C6713BGDP-225 DSP Processor
2) MBM29LV160BE-90TN Flash 16×1Mbits
3) MT48LC4M32B2TG-75 SDRAM 32×4Mbits
4) XC95144XL-TQ100 CPLD 3,200 Gate
5) SN74LVTH162245DGGR Bi-direction Driver 16 Bits
6) SN74LVTH2245PWR Bi-direction Driver 8 Bits
7) SN74LVTH125PWR Bi-direction Driver 4 Bits
8) TPS70302PWP Dual Output Voltage Regulator 3.3 Volt & 1.26 Volt
9) DS13338C-33 Real Time Clock 2-wire Serial
10) MAX3232CSE RS-232 Transceivers

2. <u>บัสเชื่อมต่อระหว่างอุปกรณ์ภายนอก</u>



<u>ภาพที่ 14</u> การเชื่อมต่อระหว่างอุปกรณ์ภายนอกกับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

2.1 บัสเชื่อมต่อกับฮาร์ดแวร์กราดตรวจลายนิ้วมือ

1) สัญญาณ D0 – D7 ของ FP_Sensor Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ED0 – ED7 จาก DSP

2) สัญญาณ A0 ของ FP_Sensor Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EA2 จาก DSP

3) สัญญาณ /XCE0 ของ FP_Sensor Bus จะถูกถอดรหัสจากสัญญาณ /CE2, EA19, EA20, และ EA21 จาก DSP

4) สัญญาณ /RD ของ FP_Sensor Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /ARE จาก DSP

5) สัญญาณ /WR ของ FP_Sensor Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /AWE จาก DSP

6) สัญญาณ /INT4 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EXT_INT4 จาก DSP

2.2 บัสเชื่อมต่อกับฮาร์ดแวร์เชื่อมต่อเครือข่าย

1) สัญญาณ LAN_CLKX ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ CLKX0 จาก DSP

2) สัญญาณ LAN_FSX ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ FSXX0 จาก DSP

3) สัญญาณ LAN_DX ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DX0 จาก DSP

4) สัญญาณ LAN_FSR ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ FSR0 จาก DSP

5) สัญญาณ LAN_DR ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DR0 จาก DSP

6) สัญญาณ /INT5 ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EXT_INT5 จาก DSP

2.3 บัสเชื่อมต่อการสื่อสารแบบอนุกรม

1) สัญญาณ TX0 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DX0 จาก DSP
2) สัญญาณ RX0 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DR0 จาก DSP
3) สัญญาณ TX1 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DX1 จาก DSP
4) สัญญาณ RX1 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ DR1 จาก DSP
5) สัญญาณ IN0 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ TINP0 จาก DSP
6) สัญญาณ OUT0 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ TOUT0 จาก DSP
7) สัญญาณ IN1 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ TINP1 จาก DSP
8) สัญญาณ OUT1 ของ Serial Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ TOUT1 จาก DSP
9) สัญญาณ /INT7 ของ LAN Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EXT_INT7 จาก DSP

2.4 บัสเชื่อมต่อกับแอลซีดี

1) สัญญาณ D0 – D7 ของ LCD Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ED0 – ED7 จาก DSP

2) สัญญาณ RS ของ LCD Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ GP1 จาก DSP

3) สัญญาณ EN ของ LCD Bus จะถูกถอดรหัสจากสัญญาณ /CE2, EA19, EA20, และ EA21 จาก DSP

4) สัญญาณ BL ของ LCD Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ GP3 จาก DSP

2.5 บัสเชื่อมต่อกับแผงแป้นพิมพ์

1) สัญญาณ COL0 – COL3 ของ Keypad Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ COL0 – COL3 จาก CPLD

2) สัญญาณ ROW0 – ROW3 ของ Keypad Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ROW0 – ROW3 จาก CPLD และจะต่อกับแหล่งจ่ายไฟ (Pull-up) ด้วยตัวต้านทาน 10K Ohm สำหรับทุกขา

2.6 บัสเชื่อมต่ออุปกรณ์ภายนอก

1) สัญญาณ D0 – D31 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ED0 – ED31 จาก DSP

2) สัญญาณ A0 – A19 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EA2 – EA21 จาก DSP

3) สัญญาณ /XCE2 ของ Expansion Bus จะถูกถอดรหัสจากสัญญาณ /CE2, EA19, EA20, และ EA21 จาก DSP

4) สัญญาณ /XCE3 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /CE3 จาก DSP

5) สัญญาณ /BE0 - /BE3 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /BE0 - /BE3 จาก DSP

6) สัญญาณ /RD ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /ARE จาก DSP

7) สัญญาณ /WR ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /AWE จาก DSP

8) สัญญาณ /OE ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /AOE จาก DSP

9) สัญญาณ /RESET ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ /RESET จาก DSP

10) สัญญาณ CLK ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ CLKOUT2 จาก DSP

11) สัญญาณ ECLK ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ECLKOUT จาก DSP

12) สัญญาณ /WAIT ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ ARDY จาก DSP

13) สัญญาณ /INT4– INT7 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ EXT_INT4 – EXT_INT7 จาก DSP

14) สัญญาณ SCL และ SDA ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ SCL0 และ SDA0 ของ DSP

15) สัญญาณ GP0 – GP3 ของ Expansion Bus จะถูกจับคู่เชื่อมต่อกับสัญญาณ GP10, GP11, GP13, และ GP15 จาก DSP

3. โครงสร้างของหน่วยความจำ

หน่วยความจำในตัวประมวลผลสัญญาณดิจิทัลตระกูล TMS320C6x ใช้สำหรับเก็บโปรแกรมและข้อมูลในการทำงานของระบบ รวมทั้งใช้ควบคุมส่วนต่างๆของระบบโดยกำหนดจากรีจิสเตอร์ควบคุม (Control Register) ซึ่งตัวประมวลผลสัญญาณดิจิทัลได้จัดแบ่งเป็นส่วนต่าง ๆ ดังตารางที่ 7

ตารางที่ 7 การแบ่งพื้นที่หน่วยความจำบนตัวประมวลผลสัญญาณดิจิทัล

MEMORY BLOCK DESCRIPTION	BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
Internal RAM (L2)	192K	0000 0000 – 0002 FFFF
Internal RAM/Cache	64K	0003 0000 – 0003 FFFF
External Memory Interface (EMIF) Registers	256K	0180 0000 – 0183 FFFF
L2 Registers	128K	0184 0000 – 0185 FFFF
HPI Registers	256K	0188 0000 – 018B FFFF
McBSP 0 Registers	256K	018C 0000 – 018F FFFF
McBSP 1 Registers	256K	0190 0000 – 0193 FFFF
Timer 0 Registers	256K	0194 0000 – 0197 FFFF
Timer 1 Registers	256K	0198 0000 – 019B FFFF
Interrupt Selector Registers	512	019C 0000 – 019C 01FF
Device Configuration Registers	4	019C 0200 – 019C 0203
EDMA RAM and EDMA Registers	256K	01A0 0000 – 01A3 FFFF
GPIO Registers	16K	01B0 0000 – 01B0 3FFF
I2C0 Registers	16K	01B4 0000 – 01B4 3FFF
I2C1 Registers	16K	01B4 4000 – 01B4 7FFF
McBSP0 Registers	16K	01B4 C000 – 01B4 FFFF
McBSP1 Registers	16K	01B5 0000 – 01B5 3FFF
PLL Registers	8K	01B7 C000 – 01B7 FFFF
Emulation Registers	256K	01BC 0000 – 01BF FFFF
McBSP0 Data Port	64M	3000 0000 – 33FF FFFF
McBSP1 Data Port	64M	3400 0000 – 37FF FFFF
McASP0 Data Port	1M	3C00 0000 – 3C0F FFFF
McASP1 Data Port	1M	3C10 0000 – 3C1F FFFF
EMIF CE0	256M	8000 0000 – 8FFF FFFF
EMIF CE1	256M	9000 0000 – 9FFF FFFF
EMIF CE2	256M	A000 0000 – AFFF FFFF
EMIF CE3	256M	B000 0000 – BFFF FFFF

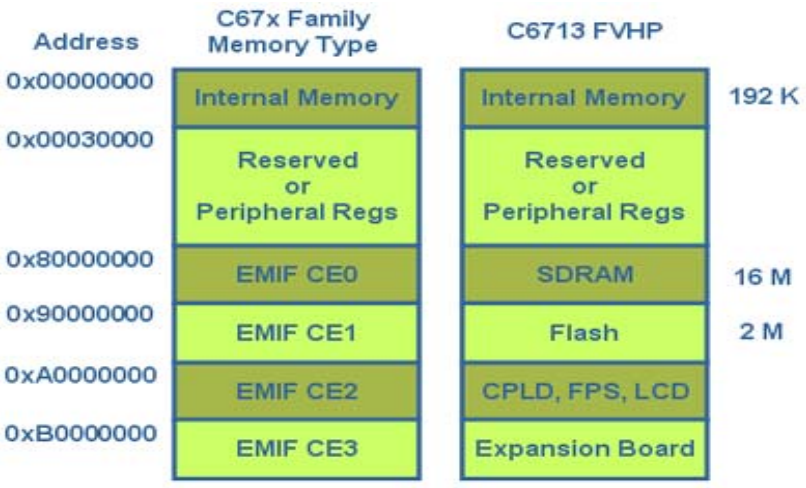
ที่มา : Texas Instruments. (2005)

จากตารางที่ 7 ค่าตำแหน่งของหน่วยความจำแสดงด้วยตัวเลขฐาน 16 โดยเริ่มจากตำแหน่ง 0 ซึ่งเป็นพื้นที่ของหน่วยความจำภายในขนาด 192K ไบต์ สามารถใช้เป็นหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลได้หรือใช้เป็นแคชระดับสอง (Cache L2) เพื่อเพิ่มความเร็วในการเข้าถึงข้อมูล ในส่วนต่อมาตั้งแต่ตำแหน่ง 0x01800000 ถึง 0x3C1FFFFF เป็นตำแหน่งที่ใช้สำหรับเข้าถึงรีจิสเตอร์ควบคุมแบบต่าง ๆ และสุดท้ายตั้งแต่ตำแหน่ง 0x80000000 ขึ้นไปจะถูกจัดให้เป็นพื้นที่ของหน่วยความจำภายนอก ซึ่งตระกูล TMS320C6x นั้น สามารถทำได้โดยใช้มอดูลจัดการหน่วยความจำ EMIF ซึ่งลักษณะวิธีการสำหรับการติดต่อกับอุปกรณ์ต่าง ๆ นั้นมีอยู่หลายรูปแบบ เช่น การติดต่อแบบ ASRAM (Asynchronous Static RAM), SBSRAM (Synchronous Burst Static RAM), SDRAM (Synchronous Dynamic RAM), FLASH ROM

การใช้งานหน่วยความจำภายนอกของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล แบ่งเป็นพื้นที่ต่างๆ ดังนี้

1) SDRAM 32 × 4 MBit ถูกเลือกด้วย / CE 0
2) FLASH 16 × 1 MBit ถูกเลือกด้วย / CE 1
3) FP Sensor, LCD, CPLD, Expansion ถูกเลือกด้วย / CE 2
4) Memory Expansion Interface ถูกเลือกด้วย / CE 3

ภาพที่ 15 แสดงการจับคู่เชื่อมต่อหน่วยความจำ ทางซ้ายมือแสดงพื้นที่เลขที่อยู่ทั้งหมดของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ทางด้านขวาแสดงรายละเอียดการนำแต่ละพื้นที่ไปใช้งาน



ภาพที่ 15 แผนผังหน่วยความจำของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

4. <u>รายละเอียดของวงจรรวมที่ใช้ในแผงวงจร</u>

4.1 TMS320C6713 Digital Signal Processor

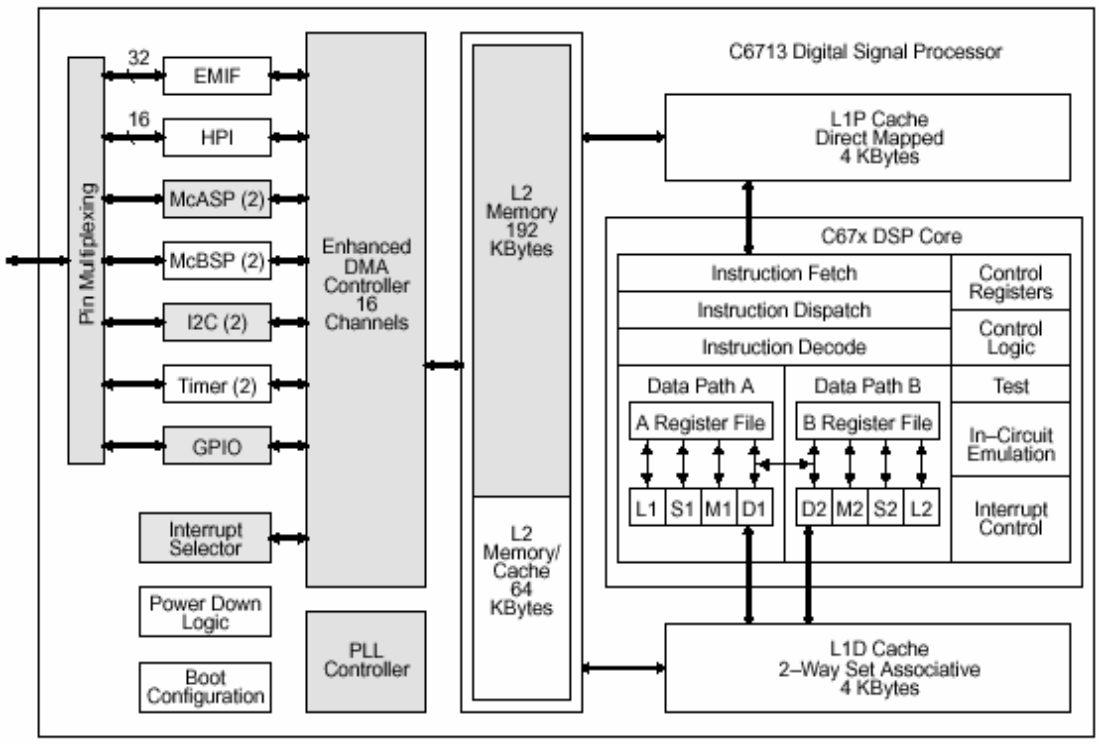
ตัวประมวลผลสัญญาณดิจิทัลเบอร์ TMS320C6713 เป็นตัวประมวลผลที่มีประสิทธิภาพสูงของตระกูล TMS320C6x DSP ที่เป็นชนิดจุดลอยตัว (Floating-Point) ของบริษัท Texas Instrument (TI) ซึ่งมีโครงสร้างภายในที่ใช้สถาปัตยกรรมของ Advanced Very-Long-Instruction-Word (VLIW) ที่ถูกพัฒนาขึ้นโดย TI จึงทำให้ TMS320C6713 DSP เป็นตัวเลือกที่ดีเยี่ยมสำหรับการทำงานที่ซับซ้อน (Multichannel and Multifunction Applications) และมีขั้นตอนวิธีคำนวณ (Algorithm) เกี่ยวกับเลขคณิตที่ซับซ้อน และต้องการความรวดเร็วในการทำงาน เนื่องจากโครงสร้างการทำงานภายใน TMS320C6713 DSP ที่ประกอบด้วยหน่วยประมวลผล 8 หน่วย ประกอบด้วย 2 หน่วยการคูณ และอีก 6 หน่วยการคำนวณทางเลขคณิต โดยใน 6 หน่วย (L1, L2, S1, S2, M1, M2) สามารถดำเนินการในระดับจุดลอยตัวทำงานขนานไปพร้อมกันได้ในทุก ๆ หนึ่งรอบสัญญาณนาฬิกา (1 Cycle) ยกตัวอย่างการทำงานที่ความถี่ 225 MHz TMS320C6713 DSP สามารถอ่านคำสั่งขนาด 32 บิต ได้ถึง 8 คำสั่งพร้อมกันภายในเวลา 1 / 225 MHz หรือ 4.45 นาโนวินาที (ns) นั่นคือ 1800 Million Instructions Per Second (MIPS) หรือได้ถึง 1350 Million Floating-Point Operations Per Second (MFLOP)

หน่วยความจำภายในของ TMS320C6713 DSP เป็นหน่วยความจำที่มีขนาดใหญ่และมีโครงสร้างหน่วยความจำแคชแบบสองระดับ (Two-Level Cache Architecture) ที่มีประสิทธิภาพสูงของ TMS320C6713 DSP ซึ่งประกอบด้วยหน่วยความจำแคช L1 ขนาด 4K ไบต์ สำหรับหน่วยความจำโปรแกรม (L1P) หน่วยความจำแคช L1 ขนาด 4K ไบต์ สำหรับหน่วยความจำข้อมูล (L1D) และหน่วยความจำแคช L2 ขนาด 64 K ไบต์ สำหรับเพิ่มประสิทธิภาพในการติดต่อกับหน่วยความจำภายนอก และยังมีหน่วยความจำอีก 192 K ไบต์ ที่แบ่งกันใช้ระหว่างหน่วยความจำภายในและหน่วยความจำแคช L2 รวมทั้งมี EDMA (Enhanced Direct Memory Access) เพื่อเพิ่มประสิทธิภาพในการจัดการข้อมูลกับหน่วยความจำภายนอกอย่างอิสระได้ถึง 16 งานพร้อมๆกัน และมีโครงสร้างสำหรับเชื่อมต่ออุปกรณ์ภายนอกได้อย่างหลากหลาย ดังนี้

- 32-bit External Memory Interface (EMIF)
- 16-bit Host Port Interface (HPI)
- 16 pins General-purpose Input/Output (GPIO)

- Two Multichannel Buffer Serial Ports (McBSPs)
- Two Multichannel Audio Serial Ports (McASPs)
- Two Inter-integrated Control Circuit (I2C)
- Two 32-bit Timers
- Boot Configuration
- Power-down Logic
- Full IEEE Standard 1149.1 Boundary-Scan-Compatible (JTAG)

มอดูลเชื่อมต่อหน่วยความจำภายนอก (EMIF) ขนาด 32 บิต ของ TMS320C6713 DSP นั้นสามารถเชื่อมต่อกับหน่วยความจำภายนอกมาตรฐานได้ทั้งแบบ ASRAM (Asynchronous Static RAM), SBSRAM (Synchronous Burst Static RAM) และแบบ SDRAM (Synchronous Dynamic RAM) โดยสามารถอ้างพื้นที่ได้ถึง 1G ไบต์ ประกอบด้วยพื้นที่ในส่วนของ CE0, CE1, CE2 และ CE3 อย่างละ 256M ไบต์ แผนภาพโครงสร้างภายในของ TMS320C6713 DSP แสดงดังภาพที่ 16



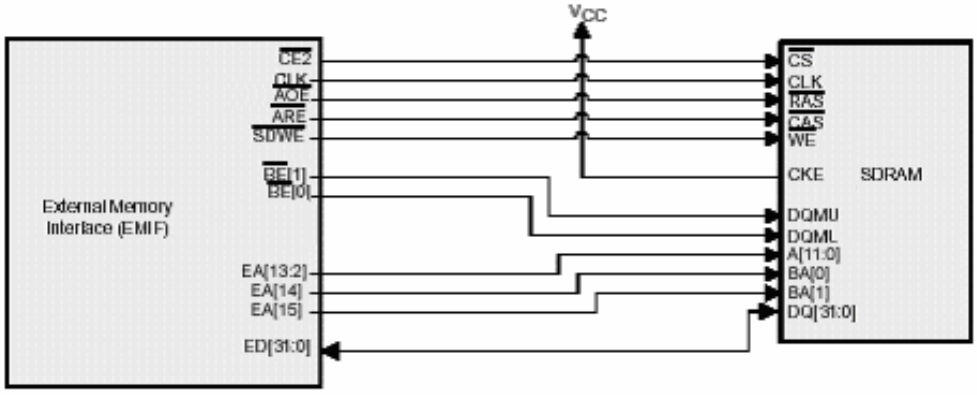
ภาพที่ 16 สถาปัตยกรรมภายในของตัวประมวลผล TMS320C6713 DSP

ที่มา : Texas Instruments. (2003)

4.2 MT48LC4M32B2TG

เป็นหน่วยความจำชนิด SDRAM ขนาด 1Meg×32×4 Bit 86-Pin TSOP (Thin Small Outline Package) ของบริษัท Micron Technology (Micron Technology, 2001) ต้องการไฟเลี้ยง 2.7-3.6 โวลต์ เชื่อมต่อกับ TMS320C6713 DSP ผ่านมอดูล EMIF ใช้การเชื่อมต่อแบบ 32 บิต SDRAM โดยเลือกลักษณะหน่วยความจำ (Memory Type) ของพื้นที่ของ CE0 เป็นชนิด 32 บิต SDRAM และกำหนดค่าเวลา (Timing) ในการติดต่อสื่อสารได้โดยเขียนค่าลงในรีจิสเตอร์ CECTL0 ซึ่งจะมีผลกับหน่วยความจำในช่วงเลขที่อยู่ 80000000h-8FFFFFFFh การเชื่อมต่อ SDRAM กับมอดูล EMIF ของตระกูล C671x ทำได้โดยง่าย ด้วยการกำหนดค่าที่ถูกต้องให้กับรีจิสเตอร์ SDRAM ตามรูปแบบการเชื่อมต่อทางกายภาพและขนาดของหน่วยความจำที่ใช้

ในตารางที่ 8 แสดงถึงข้อมูลการเชื่อมต่อกับ SDRAM ขนาดต่าง ๆ ทั้งหมดที่ DSP รองรับได้ และยังอธิบายถึงการเชื่อมต่อทางกายภาพ ระหว่างขาของ EMIF (DSP) และขาของ SDRAM ไว้อย่างละเอียด ซึ่งที่เลขที่อยู่ A [x : 0] ของ SDRAM จะต้องเชื่อมต่อกับเลขที่อยู่ EA [x+2 : 2] ของ EMIF (DSP) เนื่องจากตระกูล C671x EMIF จะใช้ขา EA [1 : 0] เพื่อไปต่อกับขา BE คุณสมบัติสำคัญของมอดูล EMIF ของตระกูล C671x คือที่ขา A10 ของ SDRAM จะถูกใช้ทำหน้าที่เติมประจุ (Precharge) ให้กับ SDRAM ซึ่งเชื่อมต่อได้โดยตรงกับ EA12 ของมอดูล EMIF (DSP) ไม่จำเป็นต้องใช้ขา SDA10 เช่นเดียวกับตระกูล C670x ซึ่งทำให้สะดวกในการใช้งานยิ่งขึ้น นอกจากการเชื่อมต่อแบบ 32 บิต แล้วยังสามารถเลือกขนาดจำนวนสายของการเชื่อมต่อหน่วยความจำ SDRAM ได้เป็น 8 บิต หรือ 16 บิต ได้ รูปด้านล่างแสดงถึงการเชื่อมต่อแบบ 32 บิต ซึ่งใช้ขา CE2, SDRAMCLK, AOE, ARE, AWE, BE(1:0), EA(15:2) และ ED(31:0) ในการติดต่อกับ SDRAM



ภาพที่ 17 การเชื่อมต่อ SDRAM ทางฮาร์ดแวร์ของ DSP รุ่น C6x ผ่านทาง EMIF

ที่มา : Texas Instruments. (2004)

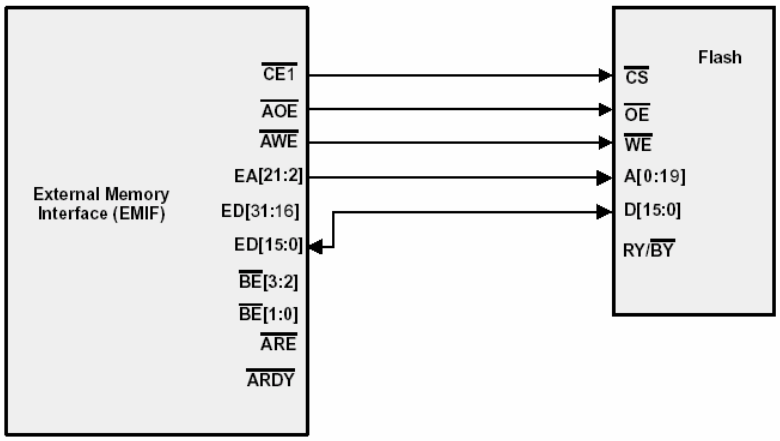
ตารางที่ 8 การเชื่อมต่อกับหน่วยความจำ SDRAM เข้ากับตัวประมวลผล C671x

SDRAM Size	Banks	Width	Depth	Max Devices/ CE	Addressable Space (M Bytes)		Column Address	Row Address	Bank Select	Pre-charge
16M bits	2	x4	2M	8	16M	SDRAM	A9–A0	A10–A0	A11	A10
						EMIF	EA11–EA2	EA12–EA2	EA13	EA12
	2	x8	1M	4	8M	SDRAM	A8–A0	A10–A0	A11	A10
						EMIF	EA10–EA2	EA12–EA2	EA13	EA12
	2	x16	512K	2	4M	SDRAM	A7–A0	A10–A0	A11	A10
						EMIF	EA9–EA2	EA12–EA2	EA13	EA12
64M bits	4	x4	4M	8	64M	SDRAM	A9–A0	A11–A0	A13–A12	A10
						EMIF	EA11–EA2	EA13–EA2	EA15–EA14	EA12
	4	x8	2M	4	32M	SDRAM	A8–A0	A11–A0	A13–A12	A10
						EMIF	EA10–EA2	EA13–EA2	EA15–EA14	EA12
	4	x16	1M	2	16M	SDRAM	A7–A0	A11–A0	A13–A12	A10
						EMIF	EA9–EA2	EA13–EA2	EA15–EA14	EA12
	4	x32†	512K	1	8M	SDRAM	A7–A0	A10–A0	A12–A11	A10
						EMIF	EA9–EA2	EA12–EA2	EA14–EA13	EA12
128M bits	4	X8	4M	4	64M	SDRAM	A9–A0	A11–A0	A13–A12	A10
						EMIF	EA11–EA2	EA13–EA2	EA15–EA14	EA12
	4	X16	2M	2	32M	SDRAM	A8–A0	A11–A0	A13–A12	A10
						EMIF	EA10–EA2	EA13–EA2	EA15–EA14	EA12
	4	X32	1M	1	16M	SDRAM	A7–A0	A11–A0	A13–A12	A10
						EMIF	EA9–EA2	EA13–EA2	EA15–EA14	EA12
256M bits	4	x8	8M	4	128M	SDRAM	A9–A0	A12–A0	A14–A13	A10
						EMIF	EA11–EA2	EA14–EA2	EA16–EA15	EA12
	4	x16	4M	2	64M	SDRAM	A8–A0	A12–A0	A14–A13	A10
						EMIF	EA10–EA2	EA14–EA2	EA16–EA15	EA12
512M bits	4	X16	8M	2	128M	SDRAM	A9–A0	A12–A0	A14–A13	A10
						EMIF	EA11–EA2	EA14–EA2	EA16–EA15	EA12

ที่มา : Texas Instruments. (2004)

4.3 MBM29LV160BE

เป็นหน่วยความจำชนิดแฟลช (Flash ROM) ขนาด 16×1Mbits 48-Pin TSOP ของบริษัท Spansion (Spansion, 2003) อุปกรณ์ทำงานและทำการโปรแกรมโดยต้องการไฟเลี้ยง 3.3 โวลต์ สามารถลบและโปรแกรมซ้ำได้โดยใช้มาตรฐานของ EPROM (Erasable Programmable Read Only Memory) เชื่อมต่อกับ TMS320C6713 DSP ผ่านมอดูล EMIF ใช้การเชื่อมต่อ 16 บิต แบบไม่ประสานเวลา (Asynchronous Interface) โดยเลือกลักษณะหน่วยความจำของพื้นที่ของ CE1 เป็นชนิด 16-bit-asynchronous และกำหนดค่าช่วงเวลาในการติดต่อสื่อสาร โดยเขียนค่าลงในรีจิสเตอร์ CECTL1 ซึ่งจะมีผลกับหน่วยความจำในช่วงเลขที่อยู่ 90000000h-9FFFFFFFh การเชื่อมต่อทางกายภาพระหว่างมอดูล EMIF กับ Flash ROM เป็นการเชื่อมต่อแบบ 16 บิต ซึ่งใช้ขา CE1, AOE, AWE, EA[21:2] และ ED[15:0] ดังภาพที่ 18 รายละเอียดดูเพิ่มเติมได้ใน (Texas Instruments, 2002)



ภาพที่ 18 การเชื่อมต่อหน่วยความจำแฟลชทางฮาร์ดแวร์ของ DSP รุ่น C6x ผ่านทาง EMIF
ที่มา : Texas Instruments. (2003)

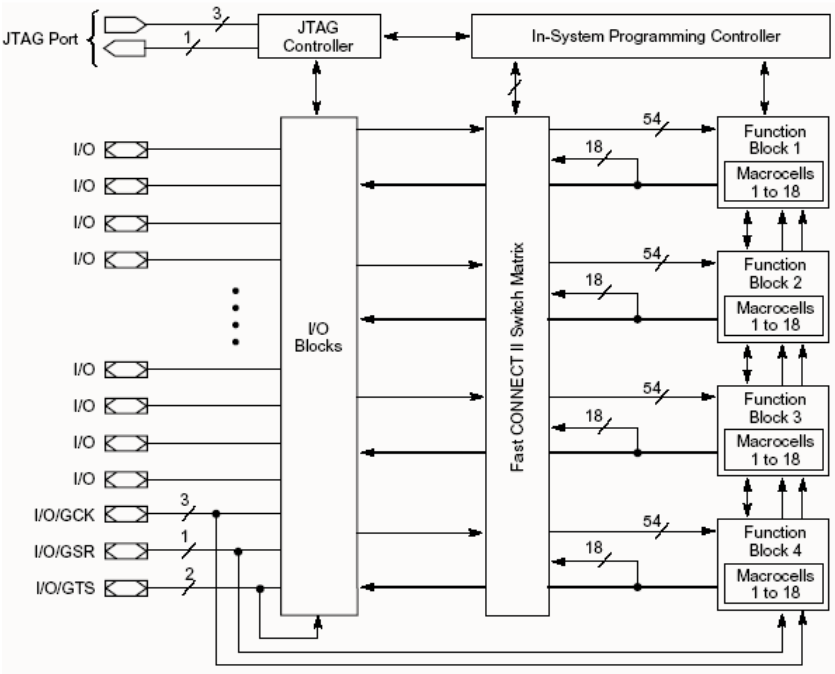
4.4 XC95144XL-TQ100

วงจรรวม CPLD (Complex Programmable Logic Device) เป็นอุปกรณ์ที่ได้มีการออกแบบวงจรโครงสร้างภายในเป็นวงจรลอจิกพื้นฐานต่าง ๆ แถวลำดับ (Array) AND, แถวลำดับ OR และ มาโครเซล (Macrocells) ต่อกันอยู่เป็นกลุ่ม มีทั้งวงจรเชิงผสม (Combination Logic) และวงจรเชิงลำดับ (Sequential Logic) อีกทั้งยังมีความยืดหยุ่นในการออกแบบวงจรสูง คือสามารถกำหนดโครงสร้างการทำงานภายในวงจรรวม CPLD ได้อย่างอิสระ โดยวงจรรวม CPLD นี้จะเป็นประเภทเดียวกันกับวงจรรวม FPGA (Field Programmable Gate Array) แต่จะต่างกันตรงที่

วงจรรวม CPLD เมื่อทำการโปรแกรมแล้วข้อมูลจะไม่ศูนย์หายแม้จะไม่มีไฟเลี้ยงจ่ายให้ก็ตาม ส่วนวงจรรวม FPGA ข้อมูลที่โปรแกรมจะเกิดการสูญหายเมื่อหยุดจ่ายไฟเลี้ยงให้กับตัววงจรรวม แต่วงจรรวมประเภทนี้จะมีโครงสร้างวงจรและฟังก์ชันการทำงานที่ซับซ้อนกว่าวงจรรวม CPLD ซึ่งโดยทั่วไปวงจรรวม FPGA จะถูกใช้เป็นต้นแบบในการสร้างหรือทดลองออกแบบวงจรรวมต่างๆ ซึ่งผู้ออกแบบสามารถทำได้เองโดยไม่ต้องไปพึ่งโรงงาน อีกทั้งการตรวจสอบ หรือการจำลองการทำงานยังทำได้ง่าย และยังมีเครื่องมือช่วยในการพัฒนางานทางด้านนี้มากพอสมควร

เนื่องจากวงจรรวม CPLD สามารถเก็บข้อมูลไว้ได้ตลอดแม้ไม่มีไฟเลี้ยงจ่ายให้ อีกทั้งราคายังถูกกว่าวงจรรวม FPGA ดังนั้นจึงค่อนข้างเหมาะสมสำหรับการนำมาใช้กับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล แม้ว่าวงจรรวม CPLD จะมีข้อจำกัดในเรื่องจำนวนเกต (Gate) และโครงสร้างภายใน โดยจะมีจำนวนเกตและความซับซ้อนของโครงสร้างน้อยกว่าวงจรรวม FPGA ทำให้การออกแบบวงจรรวมที่มีความซับซ้อนมากๆ ซึ่งต้องใช้เกตภายในจำนวนมากนั้น วงจรรวม CPLD ก็อาจจะไม่สามารถทำได้ อย่างไรก็ตามวงจรรวม CPLD ที่ต้องการใช้สำหรับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลนั้น ยังไม่มีความซับซ้อนมากนักวงจรรวม CLPD เบอร์ XC95144XL ถือว่าเป็นวงจรรวมที่มีจำนวนเกตภายในพอสมควร สามารถนำไปออกแบบวงจรรวมที่ซับซ้อนได้ในระดับหนึ่ง โดยจะมีคุณสมบัติต่างๆดังนี้

- 5 ns Pin-to-pin Logic Delays on all Pins
- System Frequency up to 178 MHz
- 144 Macrocells with 3,200 Usable Gates
- 100-pin TQFP (81 user I/O pins)
- Optimized for High-performance 3.3V Systems
- 5V Tolerant I/O Pins Accept 5V, 3.3V, and 2.5V Signals
- Endurance Exceeding 10,000 Program/erase Cycles
- Full IEEE Standard 1149.1 Boundary-scan (JTAG)



ภาพที่ 19 สถาปัตยกรรมภายในของวงจรรวม XC95144XL CPLD

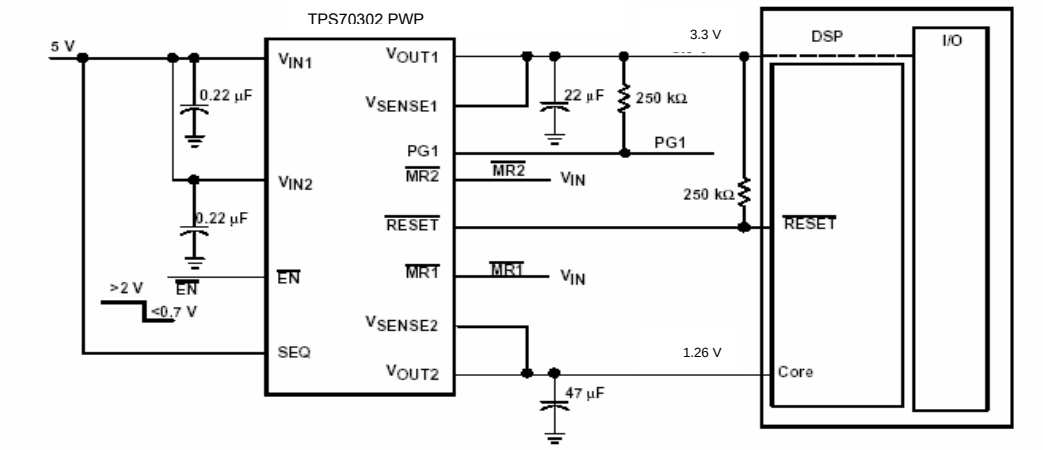
ที่มา : Xilinx. (2004)

4.5 TPS70302PWP

วงจรรวมคุมค่าแรงดันสองค่า (Dual-output Low-dropout Voltage Regulators) เป็นอุปกรณ์ที่ได้มีการออกแบบให้สามารถใช้งานได้กับตัวประมวลผล DSP, FPGA, ASIC (Application Specific Integrated Circuit) และตัวประมวลผลดิจิทัลอื่น ๆ ที่ต้องการค่าแรงดัน 2 ค่า คือ เมื่อจ่ายแรงดันอินพุต 5 โวลต์ ให้กับวงจรรวมแล้วจะได้แรงดันเอาต์พุต 3.3 โวลต์ และ 1.26 โวลต์ และยังสามารถจัดลำดับก่อนหลังของการป้อนแรงดันเอาต์พุตให้กับวงจรรวมได้ ซึ่งจัดลำดับของการป้อนแรงดันนี้เป็นมาตรฐานที่จำเป็นต้องใช้ในตัวประมวลผลสัญญาณดิจิทัล (DSP) ทุก ๆ เบอร์ของบริษัท Texus Instrument โดยมีคุณสมบัติต่าง ๆ ดังนี้

- Dual Output Voltages for Split-Supply Application
- Selectable Power Up Sequencing for DSP Application
- Output Current Range of 1 A on Regulator 1 and 2 A on Regulator 2
- Open Drain Power-On Reset with 120-ms Delay
- Two Manual Reset Inputs
- Undervoltage Lockout (UVLO) Feature

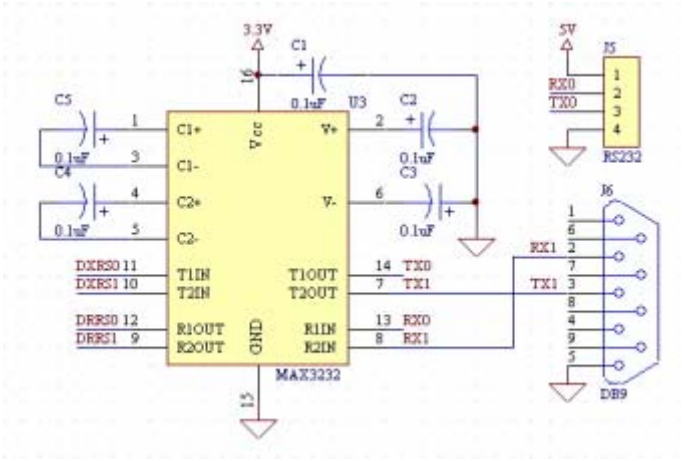
- 24-Pin PowerPAD TSSOP Package
- Thermal Shutdown Protection



ภาพที่ 20 การเชื่อมต่อทางฮาร์ดแวร์ของวงจรรวมคุมแรงดันสองค่า TPS70302
ที่มา : Texas Instruments. (2002)

4.6 MAX3232CSE

สำหรับวงจรสื่อสารสัญญาณผ่านมาตรฐานพอร์ตอนุกรมแบบ RS232 นั้นใช้วงจรรวมเบอร์ MAX3232 (RS232 Transceiver) ของบริษัท Maxim (Maxim Inc, 2003) ซึ่งทำหน้าที่แปลงระดับแรงดันจาก 3.3 โวลต์ ให้ได้แรงดันตามมาตรฐาน RS232 โครงสร้างภายในวงจรมี ตัวรับ (Receiver) และตัวขับ (Driver) อย่างละ 2 ตัว โดยคู่แรกเป็นตัวรับส่งเพื่อเชื่อมต่อระหว่างพอร์ต McBSP0 ของ DSP โดยผ่านทางตัวต่อขยายอนุกรมพอร์ต 0 (Serial 0 Expansion Connector) กับเครื่องคอมพิวเตอร์ส่วนบุคคล โดยผ่านทางตัวต่อแบบ DB9 และอีกคู่เป็นตัวรับส่งเพื่อเชื่อมต่อระหว่างพอร์ต McBSP1 ของ DSP โดยผ่านทางตัวต่อขยายอนุกรมพอร์ต 1 (Serial 1 Expansion Connector) กับฮาร์ดแวร์แป้นพิมพ์และแสดงผล (I/O Processor Board) ลักษณะการต่อวงจรได้แสดงไว้ในภาพที่ 21 ดังนี้

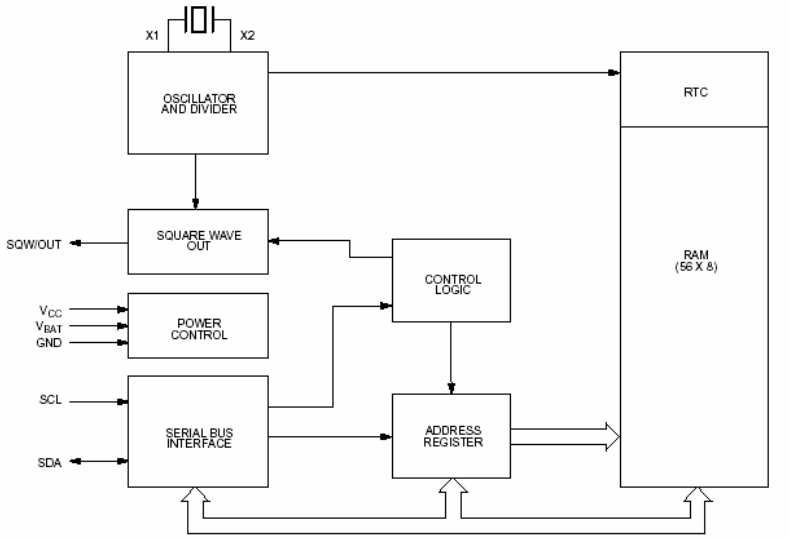


ภาพที่ 21 การเชื่อมต่อทางฮาร์ดแวร์ของตัวรับส่ง RS232 โดยใช้วงจรรวม MAX3232

ที่มา : Maxim. (2003)

4.7 DS1338C

เป็นวงจรรวมที่ทำหน้าที่สร้างฐานเวลาจริง (Real Time Clock) ซึ่งสามารถรับส่งข้อมูลให้กับไมโครคอนโทรลเลอร์ได้ โดยใช้บัสแบบ I2C (Inter Integrated Circuit) สามารถให้ข้อมูลเกี่ยวกับเวลา เช่น วินาที นาที ชั่วโมง (ทั้งแบบ 24 ชั่วโมงและแบบ 12 ชั่วโมงพร้อมระบุค่า (AM/PM) และบอก วัน เดือน ปี ได้

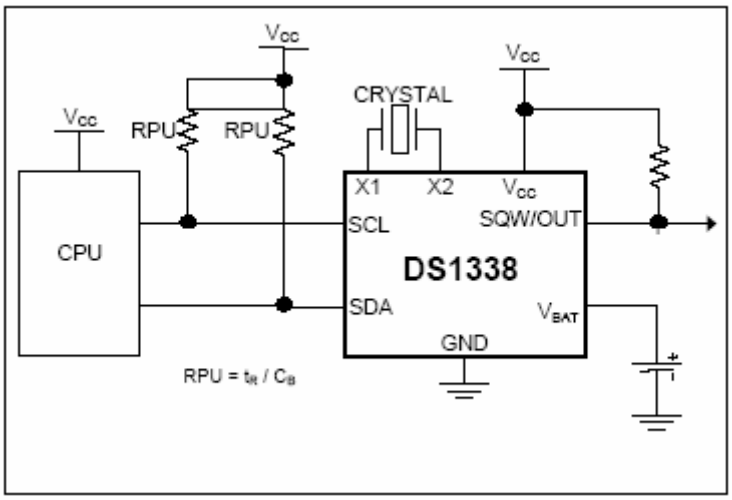


ภาพที่ 22 สถาปัตยกรรมภายในของวงจรรวมฐานเวลาจริง DS1338C

ที่มา : Maxim. (2004)

จากโครงสร้างภายในดังภาพที่ 22 จะพบว่าวงจรออสซิลเลเตอร์จะทำให้เวลาในวงจรรวมฐานเวลาจริงเดินไปอย่างต่อเนื่อง ถ้าหากต้องการอ่านค่าวันเวลาสามารถทำได้โดยอ่านข้อมูลจากหน่วยความจำ (RAM) ในตำแหน่งเวลาที่ต้องการ ถ้าหากต้องการตั้งค่าเวลาใหม่ก็ให้เขียนข้อมูลลงไปในหน่วยความจำตำแหน่งที่สอดคล้องกัน ภายในวงจรรวมจะมีวงจรควบคุมการจ่ายไฟเลี้ยง ถ้าหากแรงดันไฟเลี้ยง (VCC) ตกลงมาต่ำกว่า 1.25 × VBAT โวลต์ ซึ่ง VBAT คือไฟเลี้ยงที่ได้จากแบตเตอรี่ วงจรรวมฐานเวลาจริงจะเข้าสู่การทำงานแบบกินกระแสต่ำโดยจะใช้พลังงานจากแบตเตอรี่ ซึ่งจะไม่สามารถอ่านเขียนข้อมูลลงไปได้แต่วงจรภายในยังคงทำงานอยู่ และถ้าหากจ่ายไฟที่ขาไฟเลี้ยง (VCC) มีค่ามากกว่า VBAT + 0.2V โวลต์ วงจรรวมจะกลับมารับพลังงานจากแหล่งจ่ายไฟของระบบและถ้าแรงดันไฟเลี้ยง (VCC) มากกว่า 1.25 × VBAT โวลต์ ก็จะสามารถอ่านเขียนข้อมูลติดต่อกับวงจรรวมได้อีกครั้ง

วงจรรวมฐานเวลาจริง DS1338 เชื่อมต่อกับไมโครคอนโทรเลอร์แบบ I^2C ดังนั้นการอ่านเขียนข้อมูลสัญญาณต่างๆ จะต้องเป็นไปตามเงื่อนไขที่กำหนดไว้ในระบบบัสแบบ I^2C นี้เพื่ออ่านและเขียนข้อมูลต่างๆ จากโครงสร้างหน่วยความจำภายในของวงจรรวม ในภาพที่ 23 ได้แสดงการต่อวงจรรวม DS1338 กับไมโครคอนโทรเลอร์แบบ I^2C ไว้ดังนี้



ภาพที่ 23 การเชื่อมต่อทางฮาร์ดแวร์ของวงจรรวมฐานเวลาจริง DS1338C

ที่มา : Maxim. (2004)

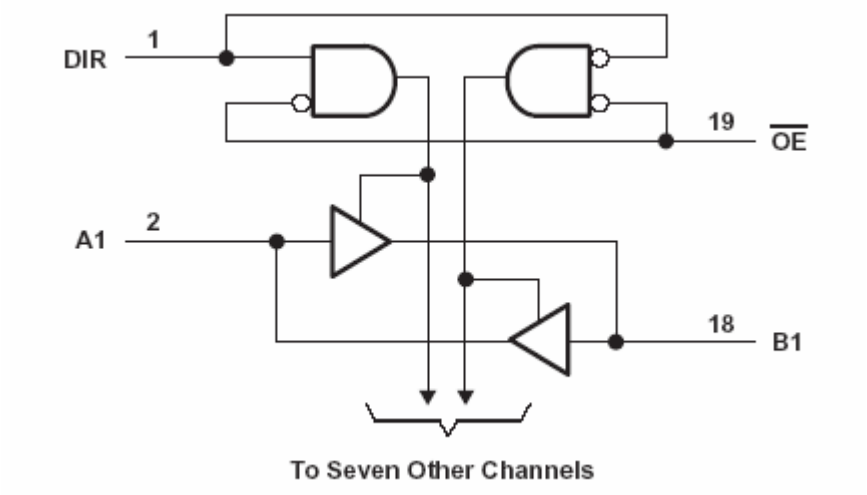
4.8 SN74LVTH2245 / SN74LVTH162245

เป็นบัสบัฟเฟอร์ (Bus-Buffer) ขนาด 8 บิต และ 16 บิต ตามลำดับ ทำงานด้วยไฟเลี้ยง 3.3 โวลต์ แต่สามารถรับแรงดันอินพุตได้ถึง 5 โวลต์ ในกรณีต่อกับอุปกรณ์ TTL (Transistor-Transistor Logic) วงจรรวมชนิดนี้ถูกออกแบบมาเพื่อใช้เป็นบัฟเฟอร์ (Buffer) ระหว่างบัสของวงจรแบบอะซิงโครนัสโดยที่วงจรรวม LVTH2245 และ LVTH162245 ประกอบด้วยตัวขับสัญญาณเอาพุตแบบสามสถานะ (3-state) ทั้งหมด 8 ชุด และ 16 ชุด ตามลำดับ สามารถกำหนดทิศทางการส่งสัญญาณระหว่าง บัสA และ บัสB ได้ โดยสั่งที่ขาควบคุมทิศทาง (DIR; Direction-Control) และสามารถกำหนดเอาต์พุตให้มีอิมพีแดนซ์สูง (High-Impedance) ได้โดยสั่งที่ขาควบคุมเอาต์พุต (OE; Output-Enable) วงจรรวมชนิดนี้มีลักษณะพิเศษคือที่เอาต์พุตพอร์ต B มีตัวต้านทานไฟฟ้าเสมือนขนาด 22 โอห์ม ต่อแบบอนุกรมอยู่ เพื่อลดกระแสไฟฟ้าที่พุ่งเกินขณะที่ลอจิกมีการเปลี่ยนแปลงที่เอาต์พุต ฟังก์ชันการทำงานและวงจรสมมูลของวงจรรวม LVTH2245 และ LVTH162245 ได้แสดงไว้ในตารางที่ 9 และภาพที่ 24 ดังต่อไปนี้

ตารางที่ 9 โหมดการทำงานของวงจรรวม SN74LVTH2245 / SN74LVTH162245

INPUTS		OPERATION
$\overline{OE}$	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

ที่มา : Texas Instruments. (2003)



ภาพที่ 24 โครงสร้างภายในของวงจรรวม SN74LVTH2245 / SN74LVTH162245

ที่มา : Texas Instruments. (2003)

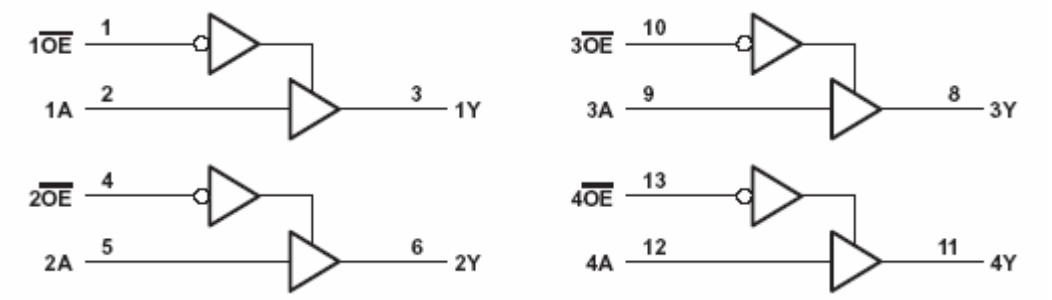
4.9 SN74LVTH125

เป็นบัสบัฟเฟอร์ขนาด 4 บิต ทำงานด้วยไปเลี้ยง 3.3 โวลต์ แต่สามารถรับแรงดันอินพุตได้ถึง 5 โวลต์ ในกรณีต่อกับอุปกรณ์ TTL วงจรรวม LVTH125 ประกอบด้วยตัวขับสัญญาณเอาต์พุตแบบสามสถานะทั้งหมด 4 ชุด ซึ่งทำงานเป็นอิสระต่อกัน ที่เอาต์พุตกำหนดให้มีอิมพีแดนซ์สูงได้โดยสั่งที่ขา Output-Enable (OE) ฟังก์ชันการทำงานและวงจรสมมูลของวงจรรวม LVTH125 ได้แสดงไว้ในตารางที่ 10 และรูปภาพที่ 25 ดังต่อไปนี้

ตารางที่ 10 โหมดการทำงานของวลจรรวม SN74LVTH125

INPUTS		OUTPUT
$\overline{OE}$	DIR	Y
L	L	H
L	H	L
H	X	Z

ที่มา : Texas Instruments. (2003)



ภาพที่ 25 โครงสร้างภายในของวงจรรวม SN74LVTH125

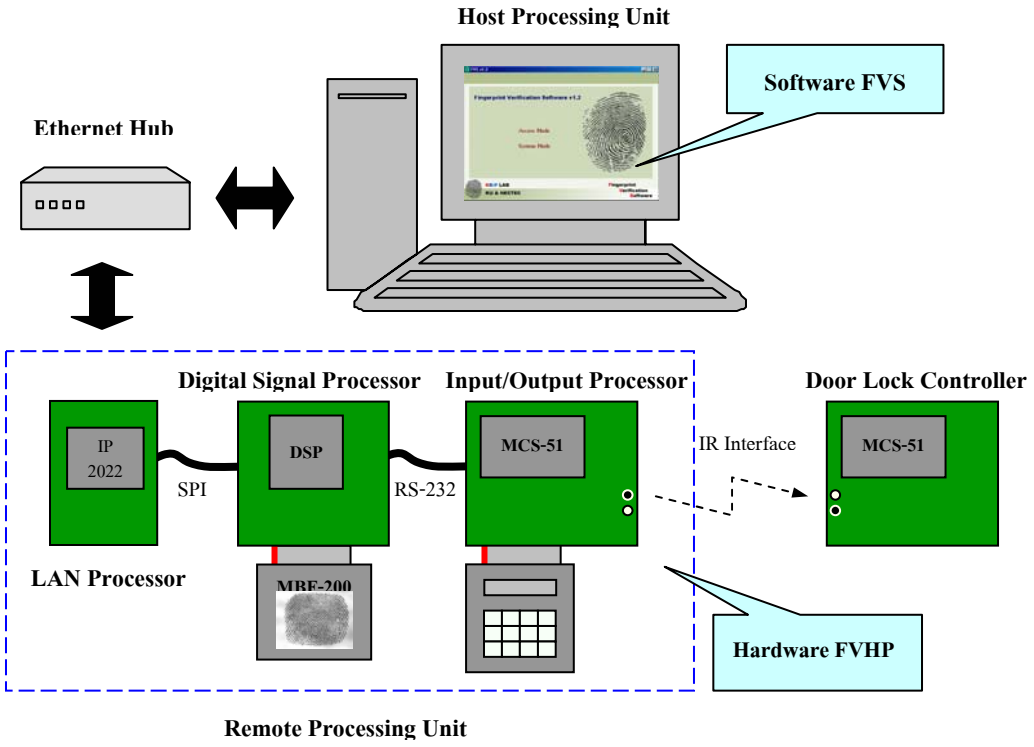
ที่มา : Texas Instruments. (2003)

เนื้อหาที่กล่าวมาข้างต้นเป็นเพียงการอธิบายองค์ประกอบหลักของวงจรรวมที่ใช้ในฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ในส่วนของรายละเอียดที่สมบูรณ์ของ การเขียนแผนภาพวงจร (Schematic Diagram) และ การออกแบบลายวงจร (PCB Design) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลจะถูกแสดงไว้ในภาคผนวก ก. และภาคผนวก ข.

การออกแบบระบบตรวจสอบลายนิ้วมือ

ระบบตรวจสอบลายนิ้วมือที่ออกแบบ และพัฒนาขึ้นมาโดยคณะวิจัย KSIP Lab (Kasetsart Signal and Image Processing Laboratory) นั้น จะแบ่งระบบออกเป็นสองส่วน ได้แก่ ระบบใหญ่ หรือ HPU (Host Processing Unit) ซึ่งทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล และระบบย่อย หรือ RAU (Remote Access Unit) ซึ่งใช้ตัวประมวลผลสัญญาณดิจิทัลเป็นตัวประมวลผล

ในหัวข้อนี้จะกล่าวถึงการออกแบบ และพัฒนาฮาร์ดแวร์ระบบย่อย หรือเรียกอีกชื่อว่า ต้นฮาร์ดแวร์ตรวจสอบลายนิ้วมือ FVHP (Fingerprint Verification Hardware Prototype) โดยใช้ตัวประมวลผลสัญญาณดิจิทัล TMS320C6713 ของบริษัท Texas Instrument ร่วมกับฮาร์ดแวร์ที่ออกแบบและประกอบขึ้นเอง ได้แก่ ฮาร์ดแวร์กราดตรวจลายนิ้วมือ ฮาร์ดแวร์เชื่อมต่อเครือข่าย ฮาร์ดแวร์แป้นพิมพ์และแสดงผล ฮาร์ดแวร์ควบคุมการเปิดปิดประตู เมื่อนำมารวมกับการวิจัยขั้นตอนวิธีการตรวจสอบลายนิ้วมือและซอฟต์แวร์ FVS (Fingerprint Verification Software) ที่เขียนขึ้นโดยคนไทยทั้งหมด จะทำให้ได้ผลิตภัณฑ์ที่พึ่งพาตนเองมากที่สุดของคนไทยและราคาต่ำที่สุดเท่าที่เป็นไปได้



ภาพที่ 26 องค์ประกอบต่างๆของระบบตรวจสอบลายนิ้วมือ

ฮาร์ดแวร์ FVHP เป็นฮาร์ดแวร์ที่สามารถประมวลผลพิสูจน์ลายนิ้วมือได้ด้วยตัวเอง โดยมีตัวกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ (Capacitive Fingerprint Sensor) เป็นตัวรับภาพลายนิ้วมือแล้วส่งให้ตัวประมวลสัญญาณดิจิทัล (DSP Processor) โดยระบบมีความสามารถในการทำงานอิสระด้วยตัวเอง (Standalone) หรือทำงานร่วมกับคอมพิวเตอร์ส่วนบุคคลที่ใช้โปรแกรม FVS ผ่าน RS-232 หรือ LAN ลักษณะโครงสร้างบล็อกของการเชื่อมต่อระหว่างฮาร์ดแวร์จะเป็นดังภาพที่ 26

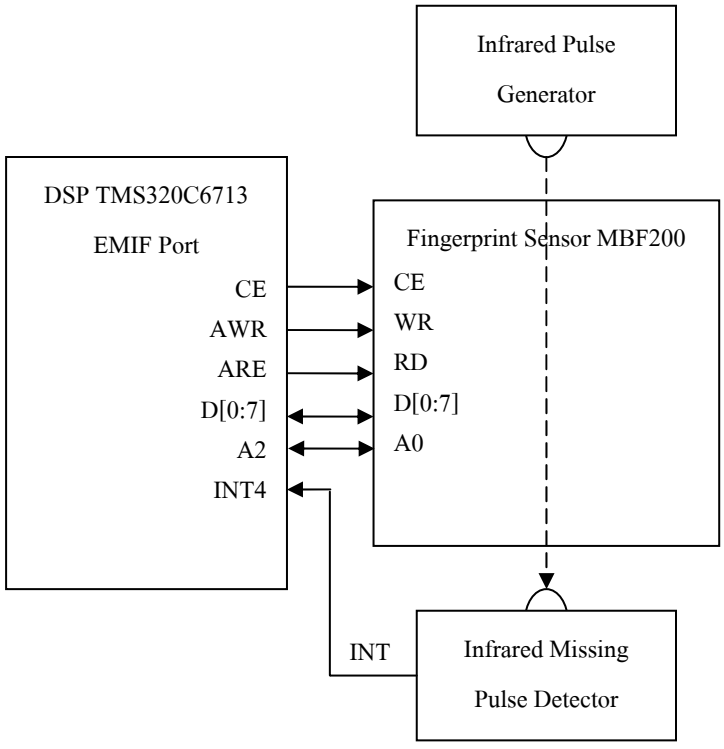
จะเห็นว่าเมื่อขยายฮาร์ดแวร์ของส่วนระบบย่อย (RAU; Remote Access Unit) หรือ FVHP ซึ่งเป็นฮาร์ดแวร์ที่มีตัวสแกนลายนิ้วมือและประมวลผลได้ในตัวเองออกมา จะสามารถแบ่งทางฮาร์ดแวร์ออกเป็น 5 ส่วน คือ

1. DSP Board เป็นส่วนประมวลผลลายนิ้วมือโดยเชื่อมต่อกับคอมพิวเตอร์ที่ใช้โปรแกรม FVS ผ่าน LAN หรือ RS-232 และเชื่อมต่อกับ IOP Board ผ่าน RS-232
2. Fingerprint Capture Board (FC Board) มีหน้าที่เป็นตัวรับภาพลายนิ้วมือแล้วส่งให้ DSP Board
3. I/O Processor Board (IOP Board) เป็นส่วนป้อนข้อมูลและแสดงผลให้ระบบ FVHP ในกรณีที่ไม่ได้ต่อกับคอมพิวเตอร์ โดยมี Keypad และ LCD Display เป็น Input และ Output รวมทั้งมีการเข้ารหัสและสั่งควบคุมให้ DLC Board ทำงาน
4. Door Lock Control Board (DLC Board) จะมีหน้าที่รับคำสั่งที่เข้ารหัส ถอดรหัส และทำการควบคุมการเข้าออกของประตูหรือระบบ
5. LAN Processor Board (LAN Board) เป็นส่วนที่ใช้ในการเชื่อมต่อระหว่าง DSP Board กับเครือข่ายภายนอก

ซึ่งจะกล่าวในรายละเอียดของฮาร์ดแวร์ที่ใช้ร่วมกับบอร์ดประมวลผลสัญญาณดิจิทัล (DSP Board) ในระบบตรวจสอบลายนิ้วมือ ดังต่อไปนี้

1. <u>การออกแบบฮาร์ดแวร์กราดตรวจลายนิ้วมือ</u>

ฮาร์ดแวร์กราดตรวจลายนิ้วมือ สามารถแบ่งการทำงานได้เป็น 2 ส่วนหลัก ๆ คือ ส่วนของวงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ เบอร์ MBF200 ของ Fujitsu และวงจรตรวจสอบการวางนิ้วเชิงแสง เพื่อใช้ในการตรวจสอบการวางนิ้วมือแบบอัตโนมัติ (Auto Finger Detection) ในการเชื่อมต่อทางด้านฮาร์ดแวร์ของชุดกราดตรวจลายนิ้วมือมีการเข้าถึงข้อมูลแบบขนาน สามารถเชื่อมต่อกับมอดูล EMIF (External Memory Interface) ของตัวประมวลผลสัญญาณดิจิทัลตระกูล TMS3206x โดยใช้วิธีการเข้าถึงแบบ ASRAM (Asynchronous Static Random Accress Memory) ซึ่งเป็นลักษณะการเข้าถึงข้อมูลแบบไม่ต้องรอการเข้าจังหวะ สามารถเขียนแผนภาพกรอบ (Block Diagram) ได้ ดังภาพที่ 27



<u>ภาพที่ 27</u> โครงสร้างของฮาร์ดแวร์กราดตรวจลายนิ้วมือ

ฮาร์ดแวร์กราดตรวจลายนิ้วมือนี้จะเริ่มทำงานเมื่อวางนิ้วมือลงบนวงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ จะทำให้วงจรตรวจสอบพัลส์อินฟราเรดที่ขาดหายไป (Infrared Missing Pulse Detector) ตรวจจับได้ว่าไม่มีพัลส์ส่งมาจากวงจรกำเนิดพัลส์อินฟราเรด (Infrared Pulse Generator) วงจรตรวจสอบพัลส์นี้จึงทำการส่งสัญญาณขัดจังหวะ (Interrupt Signal) ไปที่

หน่วยประมวลผลสัญญาณดิจิทัล เพื่อให้รู้ว่ากำลังจะมีข้อมูลเข้ามาที่มอดูล EMIF ซึ่งมีขาสัญญาณข้อมูล 8 บิตในการรับและส่งข้อมูล ส่วนขาสัญญาณ A2 จะใช้ในการบอกให้ตัวกราดตรวจลายนิ้วมือให้รับรู้ว่าเป็นการเขียนคำสั่งหรือข้อมูล (ใช้เฉพาะในกรณีของตัวกราดตรวจลายนิ้วมือ) ส่วนสัญญาณ CE2 AWE และ ARE ใช้สำหรับควบคุมการอ่าน หรือเขียนข้อมูลบนตัวกราดตรวจลายนิ้วมือ เพื่อรับภาพลายนิ้วมือเข้าไปทำการประมวลผลที่หน่วยประมวลผลสัญญาณดิจิทัลต่อไป

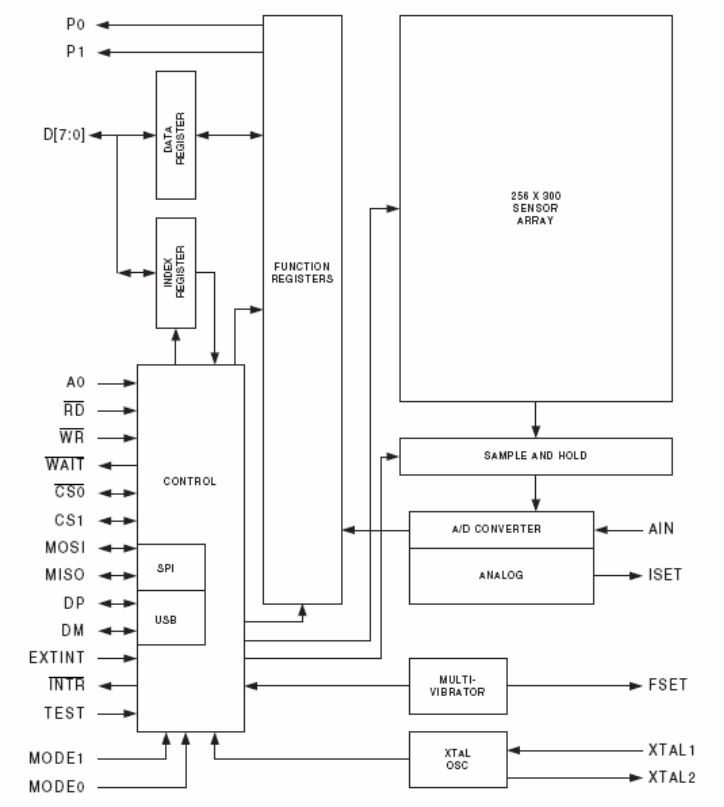
อุปกรณ์ที่ถูกใช้บนแผงวงจรมี 2 ชนิดที่ต้องทำการศึกษา คือ วงจรรวมกราดตรวจลายนิ้วมือ เบอร์ MBF200 ของบริษัท Fujitsu และวงจรรวม TLC555 ของบริษัท Texas Instruments

1.1 Fujitsu Fingerprint Sensor MBF200

วงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ เบอร์ MBF200 เป็นเซนเซอร์รับภาพลายนิ้วมือที่มีประสิทธิภาพสูง ใช้พลังงานไฟฟ้าต่ำ และมีตัวตรวจวัดค่าความจุไฟฟ้าเรียงกันเป็นลักษณะแถวลำดับ (Array) ทั้งแนวตั้ง และแนวนอนอยู่ภายใน วงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุนี้ถูกผลิตบนมาตรฐานเทคโนโลยี CMOS (Complementary Metal Oxide Semiconductor) โดยภาพจะมีขนาด 256 × 300 จุดภาพ และมีระยะห่างระหว่างจุดภาพ (Pixel Pitch) เท่ากับ 50 ไมโครเมตร (μm) ซึ่งทำให้ภาพมีความละเอียดระดับ 500 จุดต่อ 1 นิ้ว (Dot Per Inch, dpi) และบนหน้าสัมผัสของวงจรรวมรับภาพลายนิ้วมือนี้ จะมีสารเคมีเคลือบไว้เพื่อป้องกันรอยขีดข่วนที่อาจเกิดขึ้นจากการใช้งาน รายละเอียดชื่อ และฟังก์ชันการทำงานของขาต่างๆ ดูได้ที่ (Fujitsu, 2004) ลักษณะเด่นของวงจรรวม MBF200 มีดังนี้

- พื้นที่ผิวหน้าสัมผัสมีขนาด 1.28 × 1.50 เซนติเมตร
- สามารถทำงานได้ในช่วงไฟเลี้ยง 3.3 ถึง 5 โวลท์
- ใช้พลังงานไฟฟ้าน้อยกว่า 70 มิลลิวัตต์ ขณะทำงานที่ไฟเลี้ยง 5 โวลท์
- มีวงจรแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (Analog to Digital Converter) ขนาด 8 บิต อยู่ภายใน
- สามารถเชื่อมต่อได้แบบไมโครโพรเซสเซอร์ (Microprocessor) ขนาด 8 บิต แบบ USB (Universal Serial Bus) ความเร็วเต็มที่ (Full-Speed) และแบบ SPI (Serial Peripheral Interface)
- มีฟังก์ชันตรวจสอบการวางนิ้วมือแบบอัตโนมัติ

การทำงานของวงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ เบอร์ MBF200 แสดงดังภาพที่ 28 โดยเมื่อวางนิ้วมือลงบนหน้าสัมผัสของวงจร ภาพที่รับเข้ามาจะผ่านวงจรชักตัวอย่างและคงค่า (Sample-and-Hold Circuits) และวงจรแปลงสัญญาณแอนาลอกเป็นดิจิทัลตามลำดับ หลังจากได้ข้อมูลเป็นดิจิทัลแล้ว ข้อมูลจะถูกส่งไปเก็บที่ฟังก์ชันรีจิสเตอร์ (Function Register) และหน่วยประมวลผลจะส่งสัญญาณออกมา เพื่อทำการรับข้อมูลจากวงจรรวมกราดตรวจลายนิ้วมือนี้

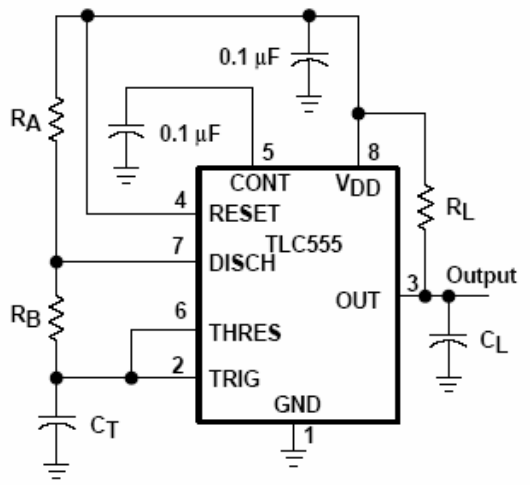


ภาพที่ 28 สถาปัตกรรมภายในของวงจรรวมรับภาพลายนิ้วมือ

ที่มา : Fujitsu Semiconductor. (2004)

1.2 TLC555

วงจรรวม TLC555 เป็นส่วนหนึ่งของวงจรตรวจสอบการวางนิ้วเชิงแสง ซึ่งจะมีอยู่ทั้งวงจรกำเนิดพัลส์อินฟราเรด และวงจรตรวจสอบพัลส์อินฟราเรดที่ขาดหายไป ในส่วนของวงจรกำเนิดพัลส์นั้นได้ใช้วงจรประยุกต์ของวงจรรวม TLC555 ที่เรียกว่าวงจรกำเนิดความถี่ (ไม่เสถียร หรือ Astable) ดังภาพที่ 29



ภาพที่ 29 การต่อวงจรแบบกำเนิดความถี่แบบ Astable

ที่มา : Texas Instruments. (2005)

โดยที่ความถี่ และความกว้างของพัลส์นั้นสามารถปรับเปลี่ยนได้ โดยการเปลี่ยนค่าความต้านทาน R_A, R_B และค่าตัวเก็บประจุ C_T ตามสูตรคณิตศาสตร์ดังสมการ (1) ถึง (4)

$$t_{c(H)} \approx C_T (R_A + R_B) \ln 2 \tag{1}$$

$$t_{c(T)} \approx C_T R_B \ln 2 \tag{2}$$

$$Period = t_{c(H)} + t_{c(L)} \approx C_T (R_A + 2R_B) \ln 2 \tag{3}$$

โดยที่ $t_{c(H)}$ คือ ช่วงเวลาที่พัลส์มีค่าสูง

$t_{c(L)}$ คือ ช่วงเวลาที่พัลส์มีค่าต่ำ

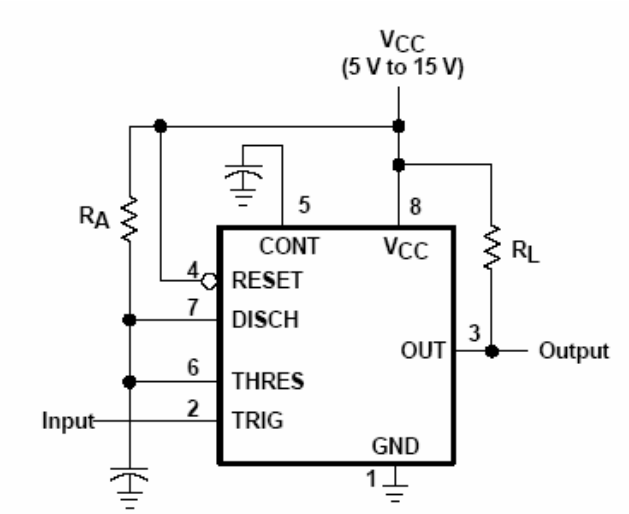
$Period$ คือ ช่วงเวลาของหนึ่งคลื่นสัญญาณพัลส์

ดังนั้นความถี่ของพัลส์ที่ส่งออกมาจากวงจรกำเนิดความถี่แบบ Astable นี้คือ

$$f = \frac{1}{Period} \tag{4}$$

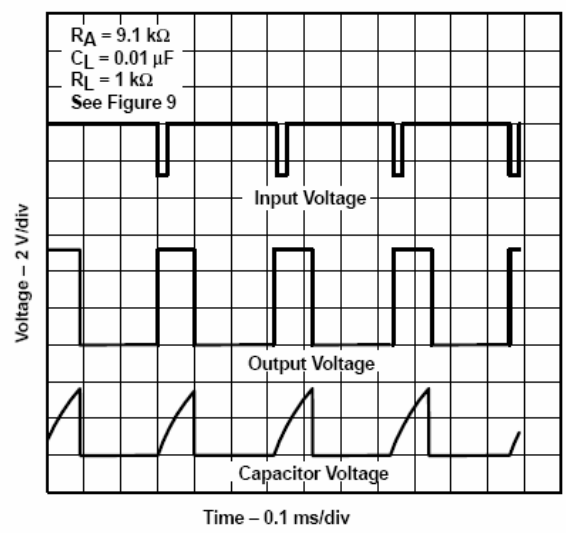
จากสมการที่ (1), (2), (3) และ (4) ถ้าหากเลือก $R_A = 5k\Omega$, $R_B = 3k\Omega$ และ $C_T = 0.15\mu F$ แล้วจะได้ $t_{c(H)} \approx 0.83ms$, $t_{c(L)} \approx 0.31ms$, $Period \approx 1.14ms$ และ $f = 877.19Hz$

ในส่วนของภาครับได้ดัดแปลงมาจากวงจรเอกเสถียร (Monostable) ดังภาพที่ 30 ซึ่งมีรูปแบบสัญญาณ (Waveform) ที่ออกจากวงจรดังภาพที่ 31



ภาพที่ 30 การต่อวงจรเอกเสถียร (Monostable)

ที่มา : Texas Instruments. (2005)



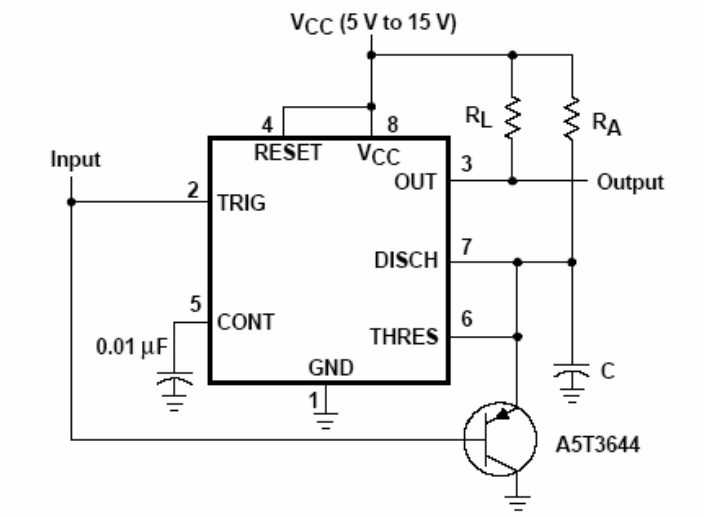
ภาพที่ 31 แผนภาพเวลาของวงจรเอกเสถียร (Monostable)

ที่มา : Texas Instruments. (2005)

จากภาพที่ 31 จะเห็นว่ารูปแบบของสัญญาณที่ออกจากวงจรเอกเสถียรจะเป็นการกลับรูปสัญญาณ และทำการขยายความกว้างของพัลส์ออก ซึ่งเป็นตามสมการ (5)

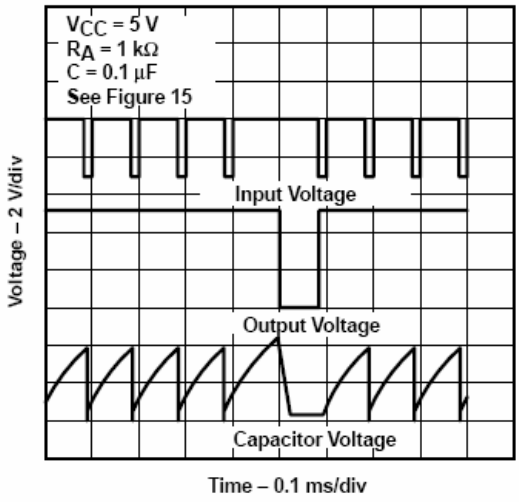
$$t_W = 1.1 R_A C_L \tag{5}$$

แต่วงจรเอกเสถียรนี้ยังไม่สามารถนำมาใช้เป็นวงจรตรวจสอบพัลส์ที่ขาดหายไปได้ เนื่องจากความกว้างของสัญญาณพัลส์ออกนั้นไม่สามารถขยายให้ชนกันเป็นสัญญาณที่มีค่าสูงอย่างต่อเนื่องได้ และความกว้างจะยังมีขอบของสัญญาณเกิดขึ้นอยู่ ซึ่งขอบของสัญญาณที่เกิดขึ้นนี้จะทำให้ไม่สามารถนำสัญญาณที่ออกจากวงจรนี้ไปใช้เป็นสัญญาณขัดจังหวะได้ ดังนั้นจึงต้องดัดแปลงวงจรนี้โดยการเพิ่ม พี-เอ็น-พี ทรานซิสเตอร์ (PNP Transistor) เข้ามาในวงจรเอกเสถียร ทำให้ได้วงจรและรูปแบบสัญญาณใหม่ดังภาพที่ 32 และ 33 ตามลำดับ



ภาพที่ 32 การต่อวงจรตรวจสอบพัลส์

ที่มา : Texas Instruments. (2005)



ภาพที่ 33 แผนภาพเวลาของวงจรตรวจสอบพัลส์

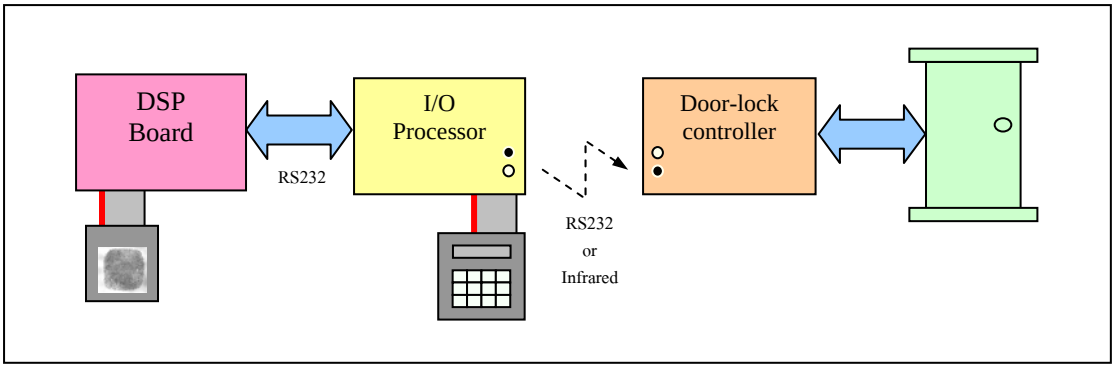
ที่มา : Texas Instruments. (2005)

ในรายละเอียดที่สมบูรณ์ของ การเขียนแผนภาพวงจร (Schematic Diagram) และการออกแบบลายวงจร (PCB Design) ของฮาร์ดแวร์กราดตรวจลายนิ้วมือจะถูกแสดงไว้ในภาคผนวก ค. และภาคผนวก ง.

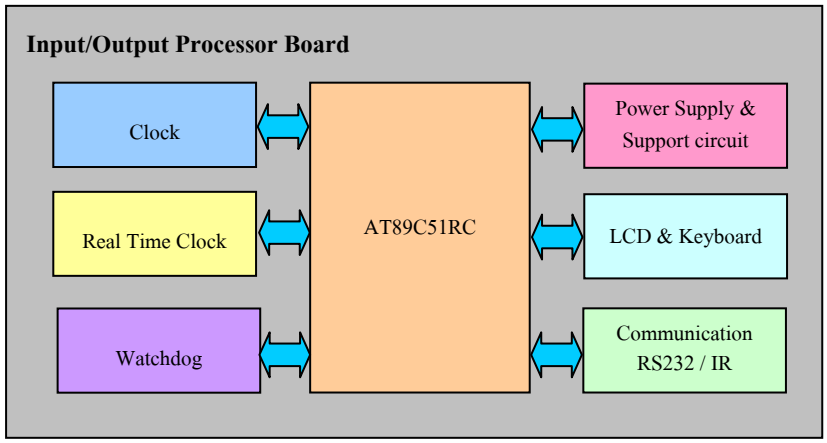
2. <u>การออกแบบฮาร์ดแวร์แป้นพิมพ์และแสดงผล</u>

ฮาร์ดแวร์ต้นแบบวงจรแป้นพิมพ์และแสดงผล (Input/Output Processor Board) นั้นประกอบด้วยไมโครคอนโทรลเลอร์ AT89C51RC ของบริษัท Atmel เชื่อมต่ออยู่กับวงจรรวมป้องกันการค้าง (Watchdog) และวงจรรวมฐานเวลาจริง (Real Time Clock) โดยมีจุดประสงค์หลักเพื่อเป็นวงจรควบคุมแป้นพิมพ์และแสดงผล วงจรสร้างฐานเวลาจริง และ วงจรสำหรับการติดต่อสื่อสาร ทั้งแบบใช้สายผ่านการสื่อสารพอร์ตอนุกรม (RS232) และแบบไร้สายด้วยอุปกรณ์อินฟราเรด (Infrared) ดังแสดงไว้ในภาพที่ 34 และ 35 ฮาร์ดแวร์แป้นพิมพ์และแสดงผล มีหน้าที่สำหรับให้ผู้ใช้ติดต่อกับระบบตรวจสอบลายนิ้วมือ โดยฮาร์ดแวร์แป้นพิมพ์และแสดงผลจะรับคำสั่งจากผู้ใช้ผ่านคีย์แพตและแสดงผลทางหน้าจอแอลซีดี จากนั้นจะส่งคำสั่งที่ได้ไปยังฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ซึ่งเชื่อมต่อผ่านทางพอร์ตอนุกรม และอีกทางหนึ่งเชื่อมต่อกับฮาร์ดแวร์วงจรควบคุมการเปิดปิดประตูโดยผ่านสายหรือไร้สาย แต่สัญญาณต้องมีการเข้ารหัสเพื่อป้องกันการแทรกแทรงการควบคุมจากแหล่งอื่น นอกจากนั้นฮาร์ดแวร์วงจรแป้นพิมพ์และแสดงผลยังมีวงจรนาฬิกาในเวลาจริงเพื่อบันทึกเวลาเข้าออกประตูให้กับระบบตรวจสอบลายนิ้วมืออีกด้วย สำหรับวงจรรวมและอุปกรณ์ที่ใช้ในแผงวงจร มีดังนี้

1) AT89C51RC 8-bit CMOS MCU, 32KB Flash, 512B RAM, 32 I/O
2) DS1232LP Low power Micromonitor chip
3) MTCS20200XFGHSA LCD 20 Chars × 2 Lines
4) ETT - Key Board 4 × 4
5) DS1307 Real Time Clock, 2-wire(serial)
6) MAX3100 UART (IrDA, Use 3 Bit port)
7) MAX232CPE Dual RS-232 Transmitter/Receiver
8) NE555 Single Timer
9) TSOP4838 Receiver module 38 KHz
10) TSAL7400 Infrared emitting diode



ภาพที่ 34 การทำงานของฮาร์ดแวร์แป้นพิมพ์และแสดงผล



ภาพที่ 35 ส่วนประกอบต่างๆของฮาร์ดแวร์แป้นพิมพ์และแสดงผล

2.1 AT89C51RC

ไมโครคอนโทรลเลอร์ AT89C51RC ที่เลือกใช้สำหรับฮาร์ดแวร์แป้นพิมพ์และแสดงผลนี้ เป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (Flash Memory) ของบริษัท Atmel เหตุผลที่เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายประการดังนี้

1) หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรเลอร์ชิปเดียวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ

2) ต้นทุนและเวลาในการพัฒนาระบบไมโครคอนโทรลเลอร์ลดลงอย่างมาก เนื่องจากไม่ต้องใช้เครื่องมือจำพวกอีมูเลเตอร์ (Emulator) และเครื่องโปรแกรมอีพรอม (EPROM)

3) บริษัทผู้ผลิตได้ทำการผลิตไมโครคอนโทรลเลอร์ตระกูลนี้ออกมาหลายเบอร์ และมีความสามารถแตกต่างกันไป ทำให้มีทางเลือกในการใช้งานได้หลากหลาย

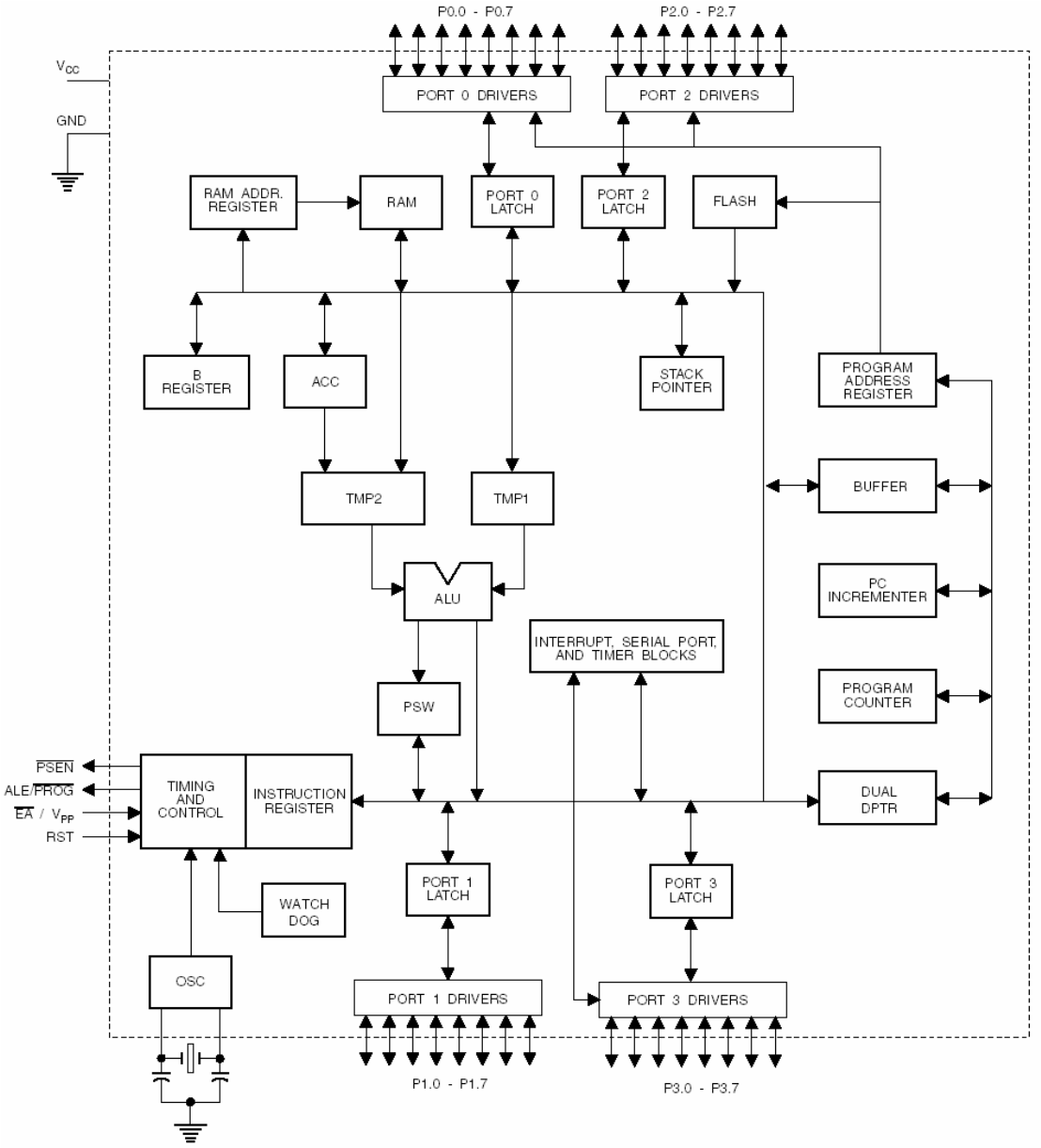
4) ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้เป็นอย่างดี

5) ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่น ไม่ว่าจะเป็น Intel, Siemens หรือ Dallas

คุณสมบัติของไมโครคอนโทรลเลอร์ ตะกูล MCS-51 เบอร์ AT89C51RC มีดังนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมแบบแฟลชสามารถลบและเขียนใหม่ได้นับพันครั้ง ขนาด 32K ไบต์
- หน่วยความจำข้อมูลเป็นหน่วยความจำแบบแรม ขนาด 512 ไบต์
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต 32 ขา
- มีวงจรสื่อสารอนุกรมแบบ (UART) สื่อสารสองทางเต็มอัตรา (Full-duplex)
- วงจรเวลา/ตัวนับ ขนาด 16 บิต 3 ตัว
- สามารถรองรับการขัดจังหวะได้ 8 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64K ไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในวงจรรวม

ในภาพที่ 36 แสดงถึงสถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์เบอร์ AT89C51RC

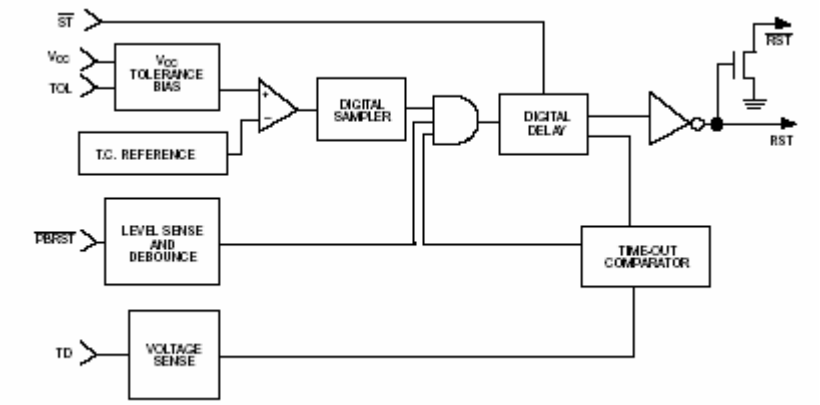


<u>ภาพที่ 36</u> สถาปัตยกรรมภายในของตัวประมวลผล AT89C51RC

ที่มา : Atmel Corporation. (2005)

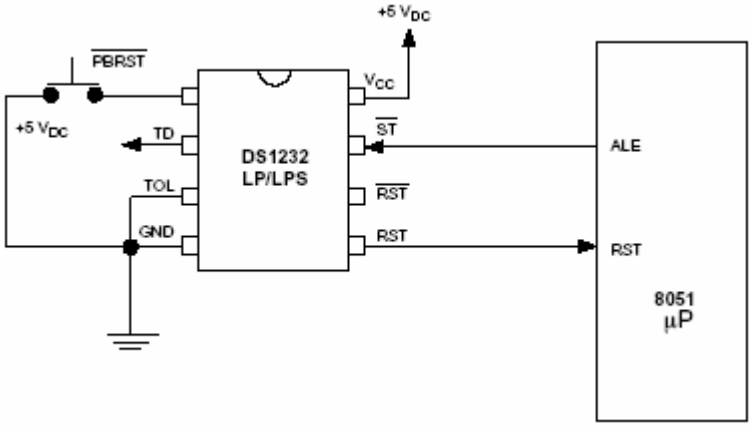
2.2 DS1232LP

วงจรรวม DS1232LP เป็นวงจรรวมสำหรับการตั้งระบบใหม่ (Reset) และป้องกันการค้าง (Watchdog) สำหรับไมโครคอนโทรลเลอร์ มีหน้าที่หลัก 3 อย่างด้วยกัน หน้าที่อย่างแรกคือ ตรวจสอบความเสถียรของแหล่งจ่ายแรงดัน สัญญาณรีเซต (Reset) จะไม่ทำงานจนกว่าระดับแรงดันจะอยู่ในเกณฑ์ปกติ จากนั้นจะหน่วงเวลาไว้อีก 250 มิลลิวินาที แล้วจึงให้สัญญาณรีเซตเป็นลอจิกสูง และมีโครงสร้างไว้สำหรับต่อกับสวิตซ์เพื่อกดสัญญาณรีเซตได้ โดยจะหน่วงเวลาไว้ 250 มิลลิวินาที เช่นเดียวกัน และ หน้าที่สุดท้ายคือ มีตัวนับเวลา (Timer) ซึ่งทำหน้าที่เป็นวงจรป้องกันการค้างสำหรับการตั้งค่าระบบใหม่กรณีไมโครคอนโทรลเลอร์ทำงานค้าง สำหรับในภาพที่ 37 เป็นโครงสร้างพื้นฐานของวงจรรวม DS1232LP และภาพที่ 38 เป็นการเชื่อมต่อวงจรรวม DS1232LP เข้ากับไมโครคอนโทรเลอร์



ภาพที่ 37 โครงสร้างภายในของวงจรรวม DS1232LP

ที่มา : Maxim. (2004)

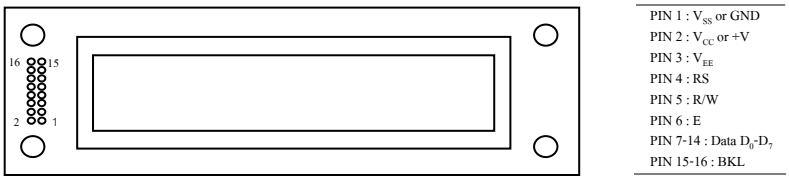


ภาพที่ 38 การเชื่อมต่อทางฮาร์ดแวร์ของวงจรรวม DS1232LP

ที่มา : Maxim. (2004)

2.3 LCD Module

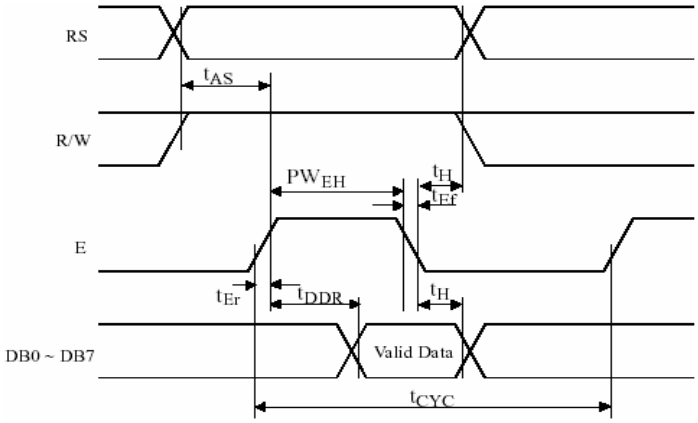
สำหรับโมดูลแอลซีดีที่ใช้เป็นแบบแสดงผลตัวอักขระ 20×2 แสดงผลได้สองบรรทัดแต่ละบรรทัดแสดงผล 20 ตัวอักษร มีแสงพื้นหลัง (Back light) ในตัว ดังแสดงในภาพที่ 39



ภาพที่ 39 แอลซีดีมอดูลและการเชื่อมต่อ

- ขา V_{DD} และ V_{SS} จะใช้ต่อกับไฟเลี้ยง 5 โวลต์ และลงดิน (Ground) ส่วน V_D จะใช้เป็นขาปรับแรงดันไฟ เพื่อควบคุมความสว่างของการแสดงผล
- ขา RS (Register Select) เป็นขาเลือกรีจิสเตอร์เนื่องจากภายในแอลซีดีโมดูลจะมีรีจิสเตอร์ที่สำคัญภายในสองตัวคือ รีจิสเตอร์เก็บคำสั่งและเก็บข้อมูล ขา RS นี้จะเป็นตัวเลือกว่าข้อมูลที่ส่งเข้าไปจะเป็นคำสั่งหรือข้อมูล โดยถ้า RS = 0 หมายความว่าข้อมูลที่เข้ามาเป็นคำสั่ง และถ้า RS = 1 หมายความว่าข้อมูลที่เข้ามาเป็นค่าข้อมูลที่จะแสดงผลบนหน้าจอแอลซีดี
- ขา R/W (Read/Write) เป็นขาอินพุต ถ้าขานี้เป็นลอจิก “1” จะเป็นการอ่านข้อมูลจากแอลซีดี ถ้าขานี้เป็นลอจิก “0” จะเป็นการเขียนข้อมูลลงในแอลซีดี
- ขา E (Enable) เป็นขาเปิดทางให้แอลซีดีทำงานโดยป้อนสัญญาณพัลส์เข้าไปหนึ่งลูก โดยพัลส์นี้ต้องมีความกว้างอย่างน้อย 450 นาโนวินาที
- ขา D_0-D_7 เป็นขาที่รับส่งข้อมูลระหว่างแอลซีดีมอดูลกับอุปกรณ์ภายนอก

ในการทำให้แอลซีดีมอดูลทำการแสดงผลเริ่มต้นจะต้องเขียนคำสั่งควบคุมแอลซีดีลงไปก่อน ซึ่งจะเป็นการกำหนดให้แสดงผลในลักษณะต่างๆ จากนั้นก็ส่งรหัส ASCII (American Standard Code for Interchange) ของตัวอักขระต่างๆที่จะแสดงผลออกไปทางแอลซีดี สำหรับรหัสควบคุมต่างๆแสดงไว้ดังตารางที่ 11 การเขียนคำสั่งต่างๆลงในแอลซีดีนั้นจะต้องตรวจสอบก่อนว่าแอลซีดีนั้นพร้อมที่จะรับคำสั่งหรือไม่โดยอ่านจากบิตแฟลกความพร้อมใช้ (Busy Flag) ซึ่งเป็นบิต D_7 การอ่านข้อมูลนั้นทำได้โดยให้ขา R/W เป็นลอจิก “1” แสดงว่าแอลซีดีมอดูลกำลังทำกระบวนการภายในของมันอยู่ไม่สามารถรับข้อมูลได้ แต่ถ้าบิตนี้เป็นลอจิก “0” แสดงว่าแอลซีดีมอดูลสามารถรับข้อมูลได้



ภาพที่ 40 แผนภาพเวลาในการอ่านค่าจากแอลซีดี

ที่มา : Microtips. (2000)

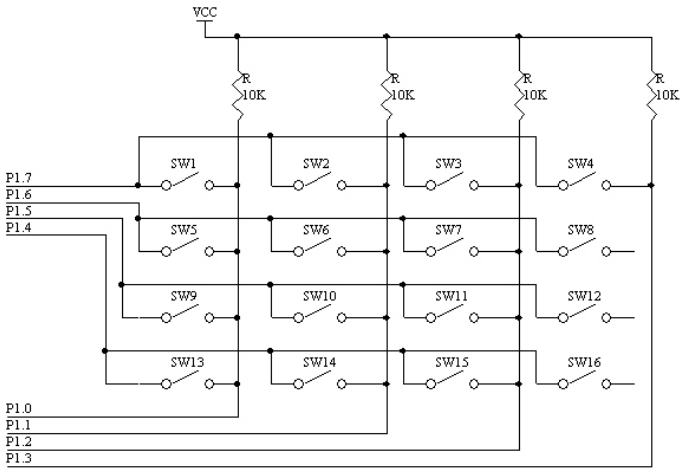
ตารางที่ 11 คำสั่งควบคุมการทำงานของแอลซีดี

คำสั่ง (HEX)	**การทำงาน**
01	ให้เคลียร์หน่วยแสดงผล
02	ให้เคอร์เซอร์กลับสู่ตำแหน่งซ้ายสุด (Home)
04	แสดงผลโดยเลื่อนเคอร์เซอร์ไปทางซ้าย
05	เลื่อนไปทางขวา
06	แสดงผลโดยเลื่อนเคอร์เซอร์ไปทางขวา
07	เลื่อนไปทางซ้าย
08	ปิดการแสดงผล ไม่แสดงเคอร์เซอร์
0A	ปิดการแสดงผล แต่แสดงเคอร์เซอร์
0C	แสดงผล แต่ไม่แสดงเคอร์เซอร์
0E	แสดงผล และแสดงเคอร์เซอร์
0F	แสดงผล และแสดงเคอร์เซอร์กะพริบ
10	เลื่อนเคอร์เซอร์ไปทางซ้าย
14	เลื่อนเคอร์เซอร์ไปทางขวา
18	เลื่อนตัวอักษรใหม่ไปทางซ้าย
1C	เลื่อนตัวอักษรใหม่ไปทางขวา
80	ตำแหน่งเริ่มต้นบรรทัดแรก
C0	ตำแหน่งเริ่มต้นบรรทัดที่สอง
38	เป็นแบบ 2 บรรทัด ขนาดตัวอักษร 5×7 จุด

ที่มา : Microtips. (2000)

2.4 Keypad

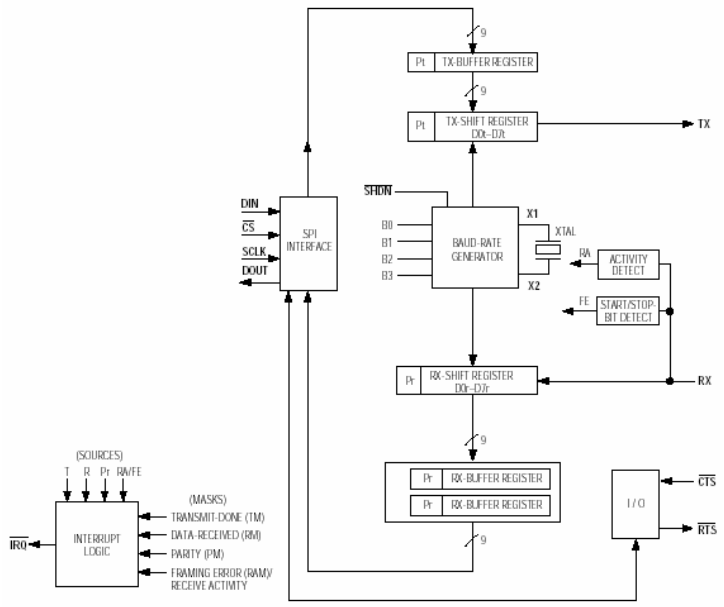
สำหรับแผงวลจรพิมพ์นั้นใช้คีย์สวิตซ์แบบเมตริกขนาด 4×4 แถวจำนวน 16 ตัว ดังภาพที่ 41 ต่อเข้ากับพอร์ต 1 ของไมโครคอนโทรลเลอร์ ซึ่งจะถูกโปรแกรมให้พอร์ต 1 ล่าง (P1.0-P1.3) เป็นอินพุต และพอร์ต 1 บน (P1.4-P1.7) เป็นเอาท์พุต โดยพอร์ต 1 ล่างจะมีตัวต้านทานต่อเข้ากับแรงดันไฟฟ้า +5 โวลต์อยู่ (Pullup) ซึ่งจะทำให้ข้อมูลที่เข้ามาทางพอร์ต 1 ล่างทุกบิตเป็นลอจิก “1” อยู่ตลอดเวลา ในการตรวจสอบการกดคีย์นั้นจะส่งลอจิก “0” ไปทางพอร์ต 1 ล่างครั้งละบิต ส่วนบิตอื่น ๆ เป็นลอจิก “1” เริ่มต้นจากบิต P1.4 ก่อนและอ่านข้อมูลเข้ามาทางพอร์ต 1 บน ต่อมาให้ P1.5 เป็นลอจิก “0” และอ่านข้อมูลเข้ามาทางพอร์ต 1 บนอีกและทำไปเรื่อย ๆ จนครบ สำหรับข้อมูลที่อ่านเข้ามาทางพอร์ต 1 บน ถ้าหากทุกบิตมีลอจิกเป็น “1” ทั้งหมดหรือมีค่าเลขฐาน 16 เป็น F แสดงว่าไม่มีการกดคีย์ แต่ถ้าค่าที่อ่านเข้ามาไม่เท่ากับ F แสดงว่ามีการกดคีย์



ภาพที่ 41 โครงสร้างทางฮาร์ดแวร์ของแผงวงจรพิมพ์

2.5 MAX3100

วงจรรวม MAX3100 (Maxim Inc, 2001) เป็นวงจรรวมที่ทำหน้าที่รับและส่งข้อมูลแบบ Universal Asynchronous (UART) สำหรับใช้กับไมโครคอนโทรลเลอร์ขนาดเล็ก ซึ่งใช้การติดต่อสื่อสารแบบ Serial Port Interface (SPI) กับไมโครคอนโทรลเลอร์ การส่งข้อมูลแบบอะซิงโครนัสนั้นสามารถใช้ได้กับระบบ RS-232, RS-485, อินฟราเรด แต่สำหรับแผงวงจรนี้จะเลือกใช้กับการรับส่งข้อมูลแบบ RS-232 เท่านั้น



<u>ภาพที่ 42</u> โครงสร้างภายในของวงจรรวม MAX3100

ที่มา : Maxim. (2001)

<u>ตารางที่ 12</u> รูปแบบในการอ่านและเขียนข้อมูลของวงจรรวม MAX3100

Table 1. Write Configuration (D15, D14 = 1, 1)

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN	1	1	$\overline{FEN}$	SHDNi	$\overline{TM}$	$\overline{RM}$	$\overline{PM}$	$\overline{RAM}$	IR	ST	PE	L	B3	B2	B1	B0
DOUT	R	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2. Read Configuration (D15, D14 = 0, 1)

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	TEST
DOUT	R	T	$\overline{FEN}$	SHDNo	$\overline{TM}$	$\overline{RM}$	$\overline{PM}$	$\overline{RAM}$	IR	ST	PE	L	B3	B2	B1	B0

Table 3. Write Data (D15, D14 = 1, 0)

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN	1	0	0	0	0	$\overline{TE}$	RTS	Pt	D7t	D6t	D5t	D4t	D3t	D2t	D1t	D0t
DOUT	R	T	0	0	0	RA/FE	CTS	Pr	D7r	D6r	D5r	D4r	D3r	D2r	D1r	D0r

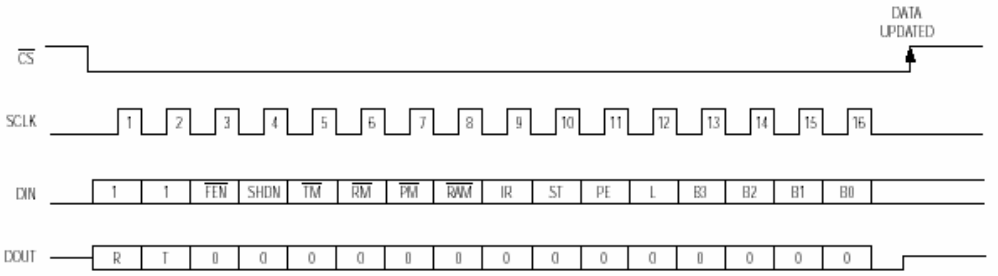
Table 4. Read Data (D15, D14 = 0, 0)

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DOUT	R	T	0	0	0	RA/FE	CTS	Pr	D7r	D6r	D5r	D4r	D3r	D2r	D1r	D0r

ที่มา : Maxim. (2001)

จากโครงสร้างภายในจะเห็นว่าวงจรรวม MAX3100 นั้นมีระบบ UART อยู่ภายในรวมทั้งยังสามารถปรับอัตราการส่งข้อมูลในตัวผ่านทาง Serial Port Interface (SPI) และขาขัดจังหวะ (IRQ; Interrupt Request) การกำหนดค่าโครงแบบต่างๆสามารถเขียนลงไปในรีจิสเตอร์ขนาด 16 บิต คือ รีจิสเตอร์ Write-configuration ยกตัวอย่างเช่น ความเร็วในการส่งข้อมูลหรืออัตราบอต (Baud rate), ความยาวคำ (Data-word length), การเชคสภาวะคู่หรือคี่ (Parity Enable), การเปิดทางให้ไฟโฟแบบ 8 บิต (8-word Receive Fist-in/First-out (FIFO)) รวมถึงการเลือกโหมดเวลาระหว่าง UART หรืออินฟราเรด โดยสามารถดูรายละเอียดได้จากตารางรีจิสเตอร์ Write Configuration ในตารางที่ 12

ในส่วนของ SPI ข้อมูลที่ส่งผ่านขา Din และ Dout ประกอบไปด้วย 16 บิต ซึ่งแสดงไว้ในการทำงานของ MAX3100 ดังภาพที่ 43 โดยที่สัญญาณ Dout จะเปลี่ยนค่าเมื่อขอบลงของสัญญาณ SCLK และสัญญาณ Din จะค้างเมื่อขอบขึ้นของสัญญาณ SCLK ในขณะที่สองบิตแรกจะเป็นตัวบอกว่าเป็นการ เขียนข้อมูล อ่านข้อมูล เขียนคำสั่ง หรืออ่านคำสั่ง ส่วนบิต B_0-B_3 ของรีจิสเตอร์ Write Configuration นั้นมีไว้สำหรับกำหนดอัตราส่งข้อมูล จากตารางที่ 13 แสดงให้เห็นว่าอัตราส่งข้อมูล (Baud-Rate) เป็นค่าที่ได้จากการหารความถี่ของผลึก (Crystal) โดยสามารถใช้ค่าความถี่ได้สองค่าคือ 1.832 MHz และ 3.6864 MHz โดยที่ BAUD = Crystal Frequency / 16 × Division Ratio



ภาพที่ 43 แผนภาพเวลาของวงจรรวม MAX3100

ที่มา : Maxim. (2001)

ดังนั้นบนฮาร์ดแวร์ควบคุมแป้นพิมพ์และแสดงผลจึงใช้การติดต่อสื่อสารรับและส่งข้อมูลแบบ UART ได้สองทาง ทางหนึ่งสำหรับติดต่อฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล (DSP Board) ผ่านพอร์ตอนุกรม อีกทางหนึ่งติดต่อกับฮาร์ดแวร์ควบคุมการเปิดปิดประตูผ่านวงจรรวม MAX3100 เพื่อติดต่อกับฮาร์ดแวร์ควบคุมการเปิดปิดประตู

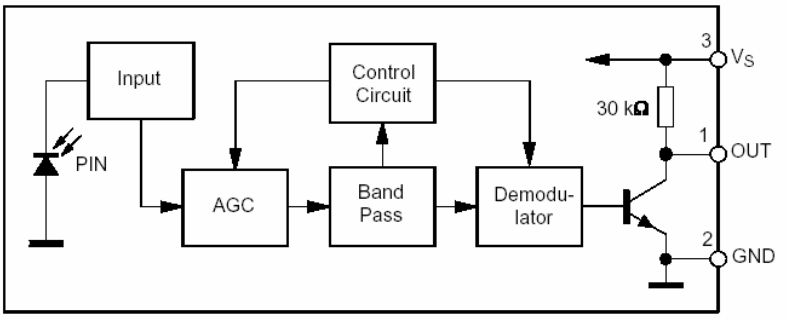
ตารางที่ 13 การกำหนดค่าอัตราบอตของวงจรรวม MAX3100

BAUD				DIVISION	BAUD RATE	BAUD RATE
B3	B2	B1	B0	RATIO	(fosc = 1.8432MHz)	(fosc = 3.6864MHz)
0	0	0	0	1	115.2K	230.4K
0	0	0	1	2	57.6K	115.2K
0	0	1	0	4	28.8K	57.6K
0	0	1	1	8	14.4K	28.8K
0	1	0	0	16	7200	14.4K
0	1	0	1	32	3600	7200
0	1	1	0	64	1800	3600
0	1	1	1	128	900	1800
1	0	0	0	3	38.4K	76.8K
1	0	0	1	6	19.2K	38.4K
1	0	1	0	12	9600	19.2K
1	0	1	1	24	4800	9600
1	1	0	0	48	2400	4800
1	1	0	1	96	1200	2400
1	1	1	0	192	600	1200
1	1	1	1	384	300	600

ที่มา : Maxim. (2001)

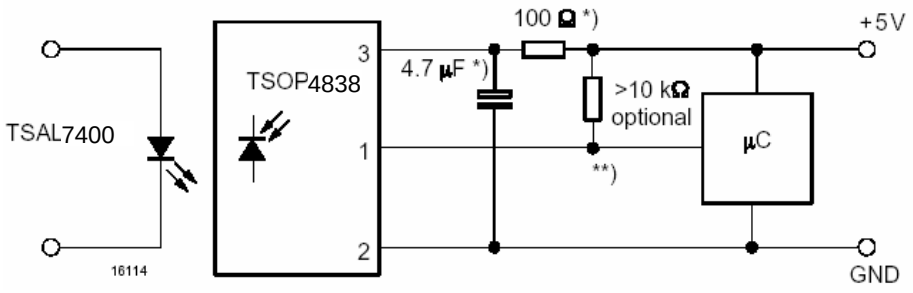
2.6 Infrared Module

สำหรับวงจรรับส่งสัญญาณแบบอินฟราเรดนั้นประกอบไปด้วยมอดูลเชิงแสงสำหรับระบบควบคุมระยะไกล วงจรรวม TSOP4838 ของบริษัท Vishay Telefunken ซึ่งเป็นตัวรับสัญญาณ อินฟราเรดสำหรับการควบคุมระยะไกล มีหลักการทำงานคือ สัญญาณอินพุตจะต้องถูกมอดูเลตด้วยค่าความถี่ 38 KHz ลักษณะการทำงานของ TSOP4838 แสดงไว้ในภาพที่ 44 และวงจรสำหรับภาครับแสดงไว้ในภาพที่ 45



ภาพที่ 44 โครงสร้างภายในของวงจรรวม TSOP4838

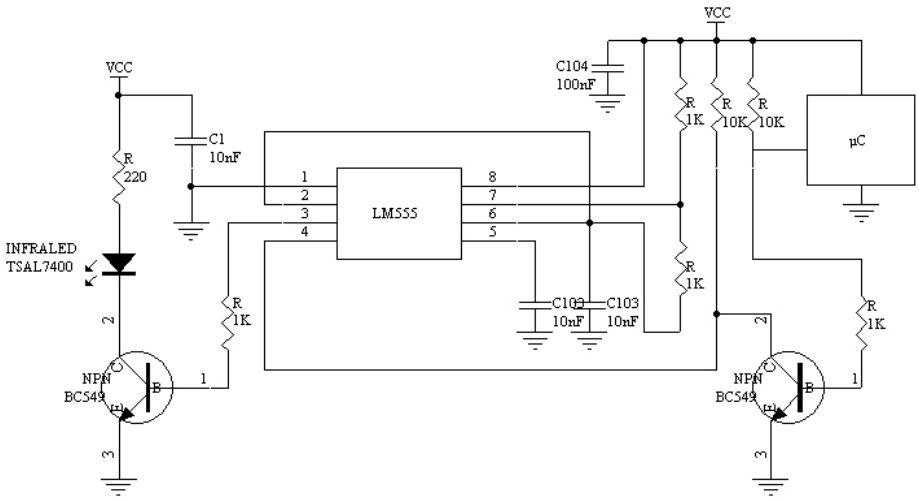
ที่มา : Vishay. (2001)



ภาพที่ 45 การเชื่อมต่อทางฮาร์ดแวร์ของวงจรรวม TSOP4838

ที่มา : Vishay. (2001)

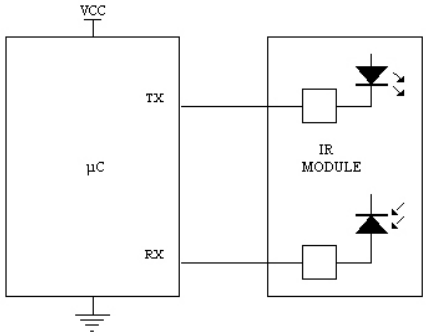
สำหรับวงจรภาคส่งนั้นใช้ไดโอดแสงเบอร์ TSAL7400 ของบริษัท Vishay Telefunken โดยที่จะต้องใช้ร่วมกับวงจรทรานซิสเตอร์สำหรับขับสัญญาณส่ง และวงจรกำเนิดความถี่เบอร์ LM555 ของบริษัท National Semiconductor เพื่อมอดูเลตสัญญาณด้วยความถี่ 38 KHz วงจรสำหรับภาคส่งแสดงไว้ในภาพที่ 46



ภาพที่ 46 วงจรสำหรับมอดูลภาคส่งของวงจรรวม TSOP4838

ที่มา : Vishay. (2001)

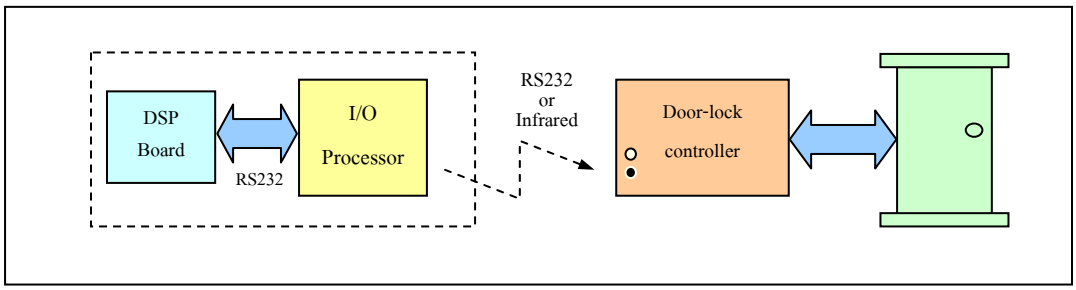
ลักษณะของการส่งข้อมูลจะเป็นการรับส่งข้อมูลแบบอนุกรมเช่นเดียวกับในระบบ UART เพียงแต่เพิ่มการมอดูเลตสัญญาณก่อนส่ง และดีมอดูเลตสัญญาณในภาครับ โดยมีข้อกำหนดว่าอัตราบอดที่ใช้ต้องมีค่าสูงสุดไม่เกิน 2400 บิตต่อวินาที (Bit-per-Second, bps) ดังนั้นจึงสามารถนำมอดูลตัวรับและตัวส่งต่อกับสัญญาณในระบบ UART ได้โดยตรงดังภาพที่ 47



ภาพที่ 47 การเชื่อมต่อมอดูลอินฟราเรดกับไมโครคอนโทลเลอร์

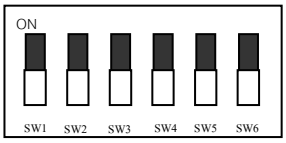
ที่มา : Vishay. (2001)

ดังนั้นฮาร์ดแวร์ควบคุมบนแป้นพิมพ์และแสดงผลจึงใช้การติดต่อสื่อสารรับและส่งข้อมูลกับ ฮาร์ดแวร์ควบคุมการเปิดปิดประตูได้สองทางคือ ผ่านทาง RS232 และอีกทางผ่านมอดูลอินฟราเรดได้อีกด้วยดังแสดงในภาพที่ 48 โดยบนแผงวงจรจะมีสวิตช์ (DIP Switch) ไว้สำหรับเลือกโหมดการติดต่อกับอุปกรณ์ภายนอก โหมดการทำงานนั้นได้แสดงไว้ดังตารางที่ 14



ภาพที่ 48 การทำงานของมอดูลอินฟราเรต

ตารางที่ 14 การเลือกวิธีการเชื่อมต่อของฮาร์ดแวร์แป้นพิมพ์และแสดงผล



SW DIP-6

$SW_1 - SW_6$	การทำงาน
10XXXX	เชื่อมต่อกับ DSP ผ่านตัวแปลงระดับสัญญาณ
01XXXX	เชื่อมต่อกับ DSP ผ่าน RS232
XX1100	เชื่อมต่อกับ Door-lock control ผ่าน RS-232
XX0011	เชื่อมต่อกับ Door-lock control ผ่าน Infrared

ในส่วนของวงจรรวมฐานเวลาจริง (DS1307), วงจรรวมนับเวลา (NE555) และวงจรรวม MAX232CPE ได้อธิบายรายละเอียดการใช้งานไปแล้วจึงไม่ขอกล่าวไว้ในหัวข้อนี้ ในส่วนของรายละเอียดที่สมบูรณ์ของ การเขียนแผนภาพวงจร (Schematic Diagram) และการออกแบบลายวงจร (PCB Design) ของฮาร์ดแวร์ควบคุมบนแป้นพิมพ์และแสดงผลจะถูกแสดงไว้ในภาคผนวก จ. และภาคผนวก ฉ.

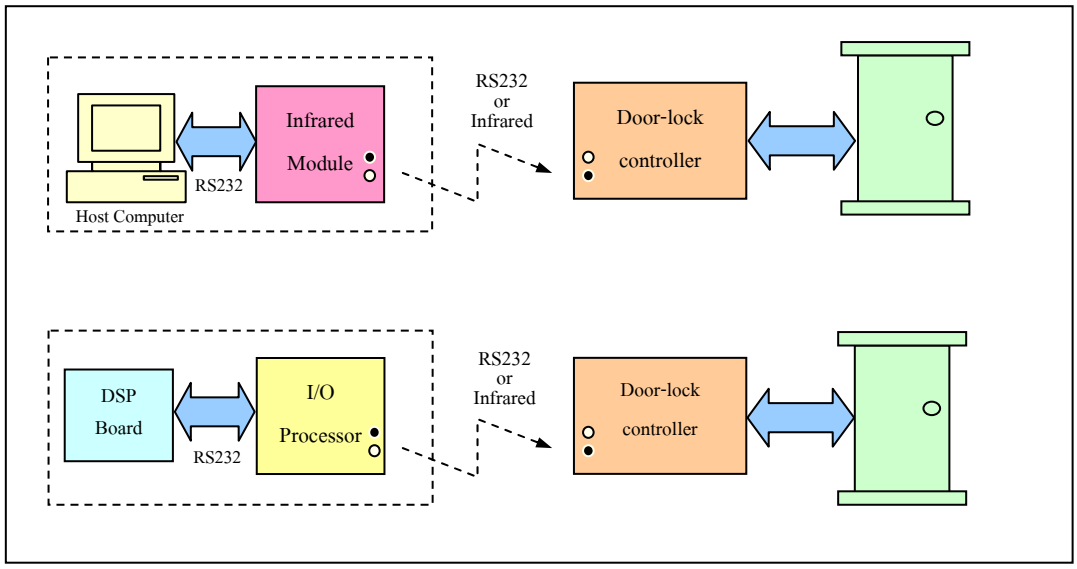
3. <u>การออกแบบฮาร์ดแวร์ควบคุมการเปิดปิดประตู</u>

ฮาร์ดแวร์ต้นแบบวงจรควบคุมการเปิดปิดประตู (Door-lock Control Board) นั้นประกอบด้วยไมครคอนโทรลเลอร์ AT89C4051 ของบริษัท Atmel เชื่อมต่ออยู่กับวงจรป้องกันการค้าง (Watchdog) และวงจรสำหรับควบคุมประตูผ่านตัวล็อกกลอนไฟฟ้า (Electric Lock) โดยมีจุดประสงค์หลักเพื่อเป็นวงจรสั่งควบคุมการเปิดประตูแบบอัตโนมัติ โดยมีวงจรสำหรับการติดต่อสื่อสาร ทั้งแบบใช้สายผ่านมาตรฐานการสื่อสารพอร์ตอนุกรม (RS232) และแบบไร้สายด้วยอุปกรณ์อินฟราเรด (Infrared) ดังแสดงไว้ในภาพที่ 49

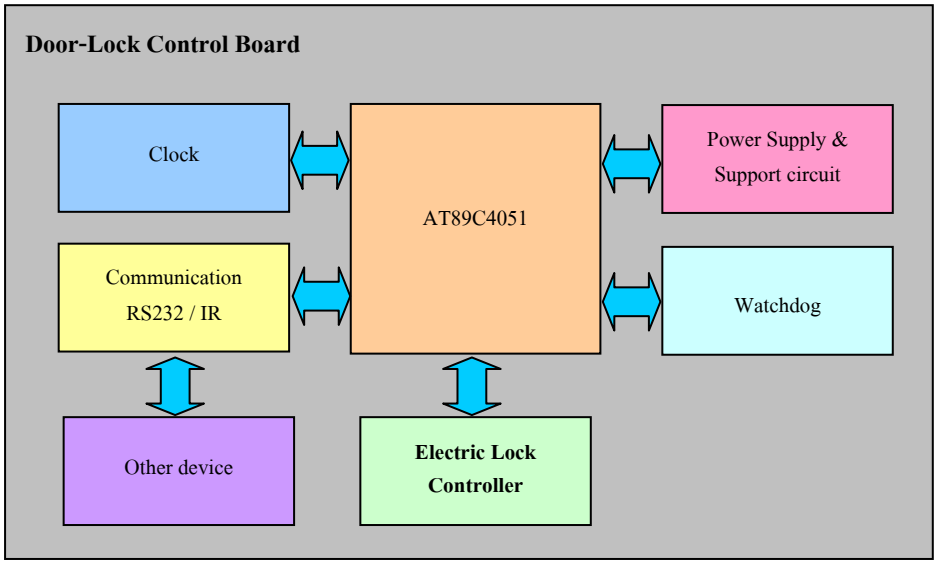
ฮาร์ดแวร์ควบคุมการเปิดปิดประตูสามารถเชื่อมต่อกับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล หรือเครื่องคอมพิวเตอร์ส่วนบุคคล โดยเชื่อมต่อผ่านสาย (RS232) หรือไร้สาย (Infrared) แต่สัญญาณต้องมีการเข้ารหัสเพื่อป้องกันการแทรกแซงการควบคุมจากแหล่งอื่น วงจรจะทำการถอดรหัสสัญญาณที่ส่งจากฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล หรือเครื่องคอมพิวเตอร์ส่วนบุคคล

สำหรับไอซีที่ใช้ในแผงวงจร มีดังนี้

1) AT89C51RC 8-bit CMOS MCU, 4KB Flash, 64B RAM, 16 I/O
2) DS1232LP Low power Micromonitor chip
3) MAX232CPE Dual RS-232 Transmitter/Receiver
4) NE555 Single Timer
5) TSOP4838 Receiver module 38 KHz
6) TSAL7400 Infrared emitting diode



ภาพที่ 49 การทำงานของฮาร์ดแวร์ควบคุมการเปิดปิดประตู



ภาพที่ 50 ส่วนประกอบต่างๆของฮาร์ดแวร์ควบคุมการเปิดปิดประตู

ฮาร์ดแวร์ควบคุมการเปิดปิดประตูได้เลือกใช้ไมโครคอนโทรลเลอร์ AT89C4051 ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (Flash Memory) ของบริษัท Atmel เหตุผลที่เลือกใช้ไมโครคอนโทรลเลอร์เบอร์นี้ เพราะเป็นไมโครคอนโทลเลอร์ขนาดเล็ก เหมาะสมกับใช้ในกับฮาร์ดแวร์ควบคุมการเปิดปิดประตู

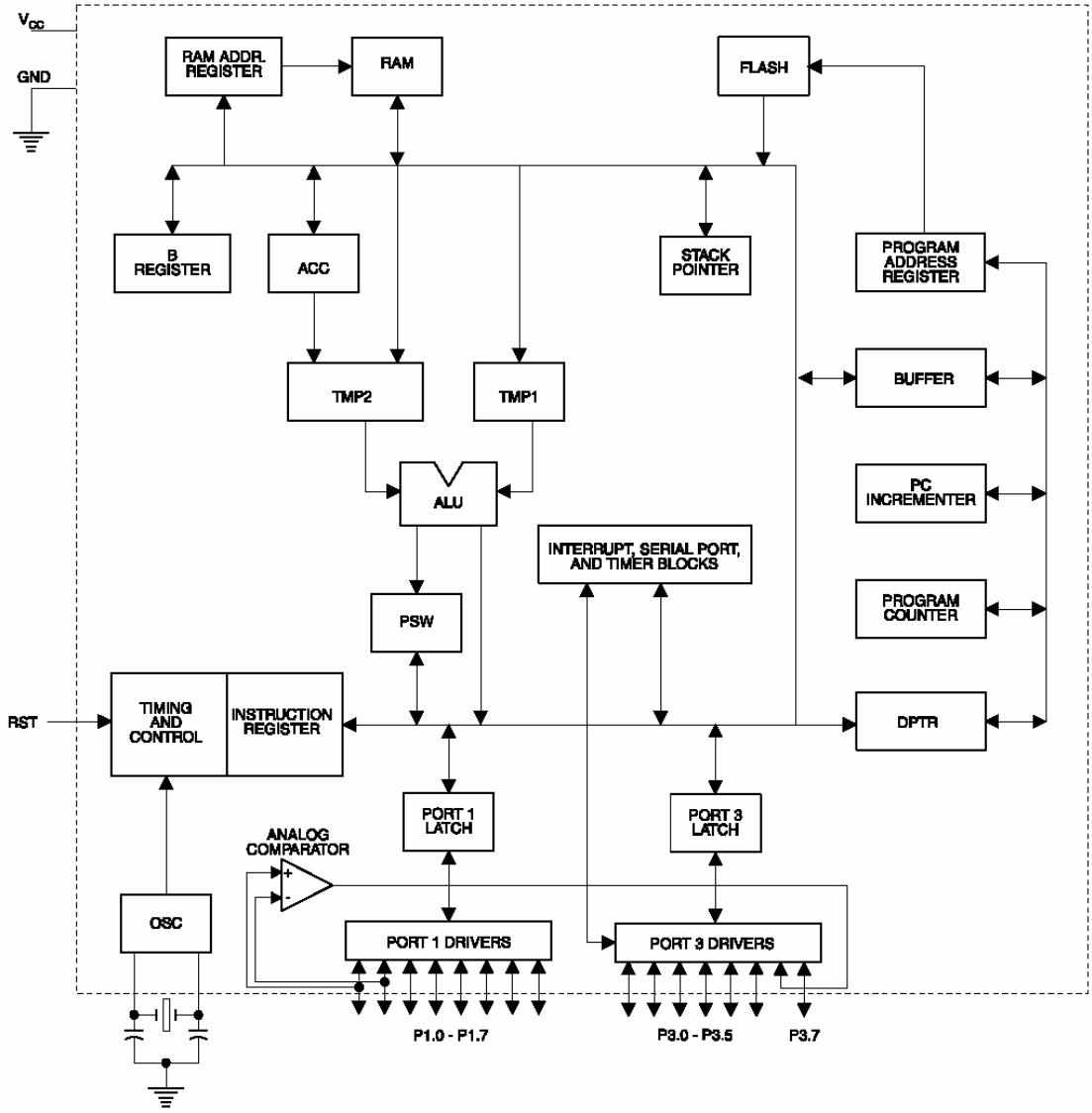
ซึ่งทำหน้าที่แค่เพียง รับคำสั่งที่เข้ารหัส ถอดรหัส และทำการควบคุมการเข้าออกของประตูหรือระบบ ซึ่งไม่มีความจำเป็นต้องใช้ หน่วยความจำโปรแกรมหรือใช้พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์เป็นจำนวนมาก

คุณสมบัติของไมโครคอนโทรลเลอร์ ตะกูล MCS-51 เบอร์ AT89C51RC มีดังนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมแบบแฟลชสามารถลบและเขียนใหม่ได้นับพันครั้ง ขนาด 4K ไบต์
- หน่วยความจำข้อมูลเป็นหน่วยความจำแบบแรม ขนาด 256 ไบต์
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต 16 ขา
- มีวงจรสื่อสารอนุกรมแบบ (UART) สื่อสารสองทางเต็มอัตรา (Full-duplex)
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 2 ตัว
- สามารถรองรับการขัดจังหวะได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64K ไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในวงจรรวม

สำหรับในภาพที่ 51 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ AT89C4051

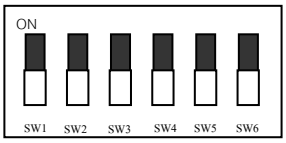
ฮาร์ดแวร์ควบคุมการเปิดปิดประตูสามารถติดต่อสื่อสารรับและส่งข้อมูลได้สองทาง คือ ผ่าน RS232 และอีกทางผ่านมอดูลอินฟราเรด นอกจากนั้นฮาร์ดแวร์ควบคุมการเปิดปิดประตูยังสามารถปรับเป็นวงจรสำหรับเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคลโดยตรงเพื่อใช้ส่งสัญญาณอินฟราเรตได้อีกด้วย โดยบนแผงวงจรจะมีสวิตซ์ (DIP Switch) ไว้สำหรับเลือกโหมดการติดต่อกับอุปกรณ์ภายนอก โหมดการทำงานนั้นได้แสดงไว้ดังตารางที่ 15



ภาพที่ 51 สถาปัตยกรรมภายในของตัวประมวลผลAT89C4051

ที่มา : Atmel Corporation. (2005)

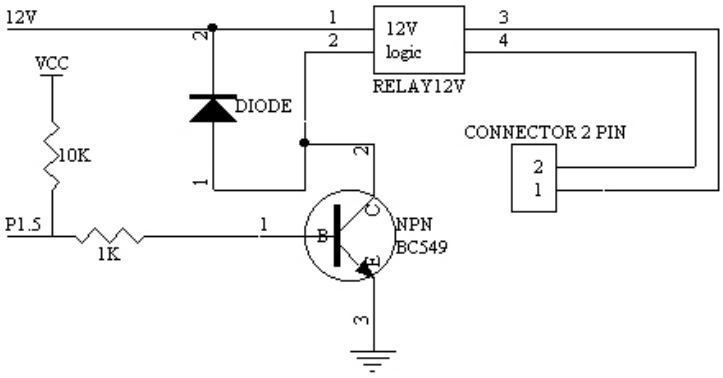
<u>ตารางที่ 15</u> การเลือกวิธีการเชื่อมต่อของฮาร์ดแวร์ควบคุมการเปิดปิดประตู



SW DIP-6

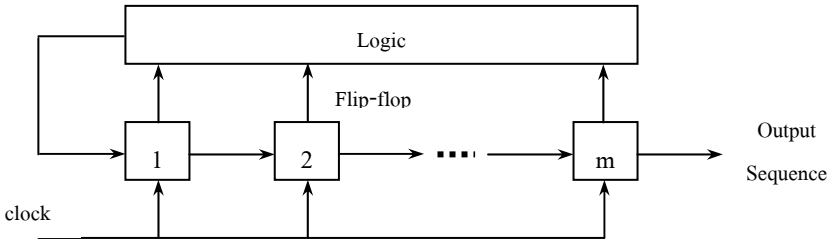
$SW_1 - SW_6$	การทำงาน
110000	แปลงสัญญาณ RS-232 เป็น Infrared
001100	เชื่อมต่อ Door-lock control ผ่านทาง RS-232
000011	เชื่อมต่อ Door-lock control ผ่านทาง Infrared

สำหรับการออกแบบวงจรชุดควบคุมล็อกกลอนไฟฟ้านั้นจะใช้กับตัวล็อกแม่เหล็กไฟฟ้าที่ควบคุมด้วยไฟฟ้ากระแสตรง 12 โวลต์ และกินกระแสขณะทำงานไม่เกิน 1 แอมแปร์ โดยอุปกรณ์บนฮาร์ดแวร์ควบคุมการเปิดปิดประตูจะประกอบด้วยรีเลย์ (Relay) 12 โวลต์ เบอร์ ST-12-K ของบริษัท Fujitsu และวงจรทรานซิสเตอร์สำหรับขับรีเลย์ดังแสดงไว้ในภาพที่ 52 โดยจะเชื่อมผ่านจุดเชื่อมต่อบนแผงวงจรไปยังชุดจ่ายกระแสไฟฟ้าและชุดตัวล็อกแม่เหล็กไฟฟ้า ซึ่งต่ออนุกรมกัน



ภาพที่ 52 การเชื่อมต่อทางฮาร์ดแวร์กับตัวล็อกแม่เหล็กไฟฟ้า

การติดต่อสื่อสารของฮาร์ดแวร์ควบคุมการเปิดปิดประตูนั้น สัญญาณต้องถูกเข้ารหัสเพื่อป้องกันการแทรกแซงการควบคุมจากแหล่งอื่นในที่นี้เลือกการเข้ารหัสด้วยวิธี Pseudo-noise sequence หรือ PN-code โดยใช้หลักการทำงานของรีจิสเตอร์เลื่อนป้อนกลับ (Feedback Shift Register) แล้วนำค่าในรีจิสเตอร์ของแต่ละสถานะมาทำการมอดูโลสอง (modulo-2) ดังภาพที่ 53



ภาพที่ 53 การสร้างลำดับรหัสของ PN-Code

จากรูปจะเห็นว่าการสร้างลำดับ (Sequence Generator) นี้ประกอบด้วยสองส่วนสำคัญในการสร้างรหัส นั่นคือส่วนของการกำหนดสภาวะเริ่มต้น (Seed) และโครงสร้างของวงจรภายใน (Wire) ดังนั้นจะต้องมีการกำหนด Seed และ Wire ให้กับทั้งสองฝ่ายที่จะติดต่อกัน ก่อนจะเริ่มต้นการทำงาน เพื่อให้รหัสของทั้งสองฝั่งตรงกัน โดยรหัสที่ใช้นั้นกำหนดเอาไว้ *m* บิตหรือต้องเลื่อนทั้งหมด *m* ครั้งเพื่อสร้างรหัส (PN-code) ที่ใช้กับคำสั่ง 1 คำสั่ง

สำหรับโปรแกรมควบคุมการทำงานของฮาร์ดแวร์ควบคุมบนแป้นพิมพ์และแสดงผล และ ฮาร์ดแวร์ควบคุมการเปิดปิดประตูนั้น เขียนโดยใช้ภาษาแอสแซมบลี (Assembly) ทั้งหมดโดยเขียนโปรแกรมบนคอมพิวเตอร์ด้วยเครื่องมือสำหรับเขียนโปรแกรม Keil μ Vision2 (V2.20a) และไฟล์นั้นจะถูกส่งไปยังไมโครคอนโทรลเลอร์ด้วยชุดฮาร์ดแวร์ Microcontroller Programmer Version2.2 (SILA PRO-100 v2.2) ของบริษัท Sila Research Co. Ltd. เป็นเครื่องโปรแกรมสำหรับตัวประมวลผลของบริษัท ATMEL โดยเฉพาะสามารถใช้ได้ถึง 13 เบอร์ มีการทำงานได้ด้วยตัวเอง (Standalone) ซึ่งเป็นเครื่องมือช่วยในการพัฒนาได้เป็นอย่างดี

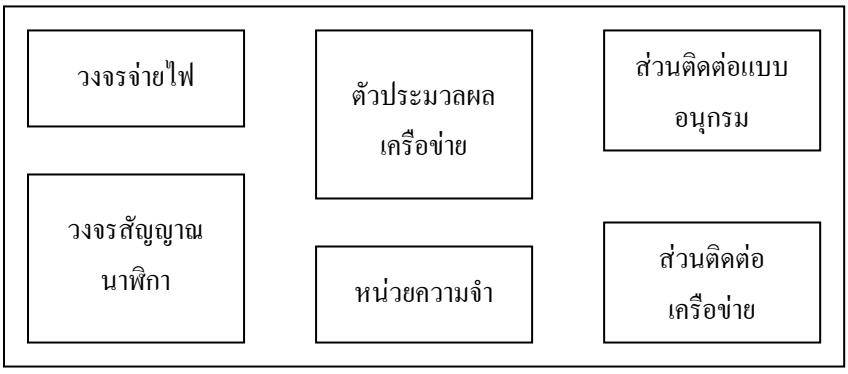
ในรายละเอียดที่สมบูรณ์ของ การเขียนแผนภาพวงจร (Schematic Diagram) และการออกแบบลายวงจร (PCB Design) ของฮาร์ดแวร์ควบคุมการเปิดปิดประตูนั้นจะถูกแสดงไว้ใน ภาคผนวก ช. และภาคผนวก ซ.

4. <u>การออกแบบฮาร์ดแวร์เชื่อมต่อเครือข่าย</u>

ฮาร์ดแวร์เชื่อมต่อเครือข่ายเป็นฮาร์ดแวร์ที่ใช้ในการเชื่อมต่อระหว่างฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลกับเครือข่ายภายนอก ซึ่งลักษณะการเชื่อมต่อของฮาร์ดแวร์ทั้งหมดสามารถแสดงได้ดังภาพที่ 54 โดยจะทำการแปลงรูปแบบการติดต่อแบบเครือข่ายให้เป็นการรับส่งแบบอนุกรมเนื่องจากส่วนติดต่ออุปกรณ์ภายนอกของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลเป็นแบบพอร์ตอนุกรม

<u>ภาพที่ 54</u> โครงสร้างการเชื่อมต่อของฮาร์ดแวร์เชื่อมต่อเครือข่าย

ฮาร์ดแวร์เชื่อมต่อเครือข่าย ได้ถูกออกแบบและพัฒนาขึ้นโดยคณะวิจัย KSIP Lab ซึ่งถูกออกแบบลักษณะเหมือนบอร์ดประมวลผลทั่วไป ซึ่งมีส่วนประกอบต่าง ๆ ได้แก่ ตัวประมวลผล หน่วยความจำ พอร์ตติดต่อกับอุปกรณ์ภายนอก วงจรจ่ายไฟ วงจรสัญญาณนาฬิกา เป็นต้น ดังภาพที่ 55

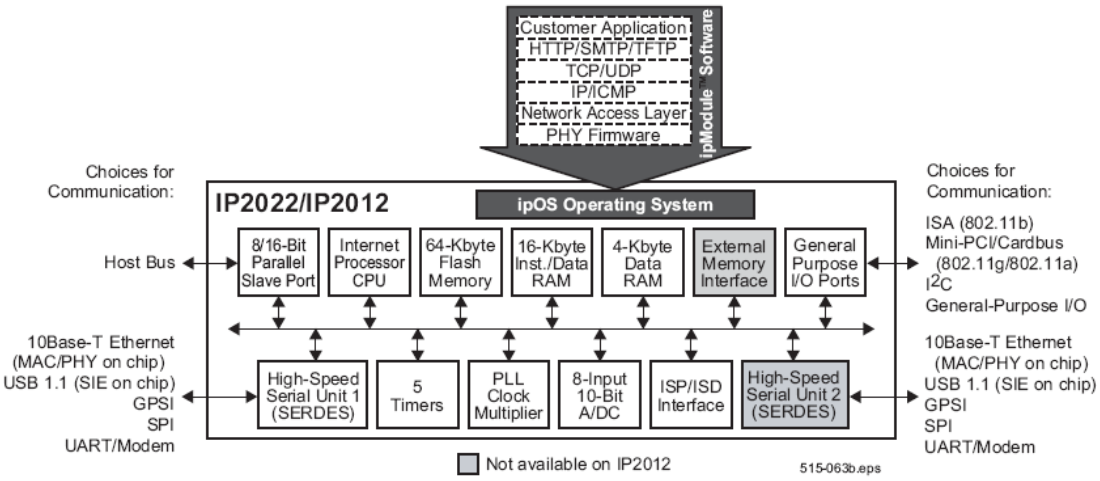


<u>ภาพที่ 55</u> ส่วนประกอบต่างๆของฮาร์ดแวร์เชื่อมต่อเครือข่าย

จากภาพที่ 55 ตัวประมวลผลเครือข่ายจะทำหน้าที่แปลงข้อมูลระหว่างการรับส่งข้อมูลแบบอนุกรมและการรับส่งข้อมูลผ่านเครือข่ายผ่านทางส่วนติดต่อแบบอนุกรม (Serial Interface) และส่วนติดต่อเครือข่าย (Network Interface) ตามลำดับ ซึ่งในการประมวลผลจะใช้หน่วยความจำ

มาเก็บข้อมูลชั่วคราวด้วย วงจรสัญญาณนาฬิกาทำหน้าที่ส่งสัญญาณนาฬิกาให้กับตัวประมวลผลเครือข่าย และส่วนวงจรจ่ายไฟทำหน้าที่ป้อนกระแสไฟให้กับอุปกรณ์ทุกตัวที่อยู่ในแผงวงจร

ฮาร์ดแวร์เชื่อมต่อเครือข่ายนี้ได้เลือกตัวประมวลผลเครือข่าย IP2022 ของบริษัท Ubicom เนื่องจากใช้งานง่าย คือ ไม่ต้องการตัวควบคุมอีเธอร์เน็ตเพิ่มทำให้ไม่ต้องเพิ่มฮาร์ดแวร์ในส่วนจัดการอีเธอร์เน็ต และมีชุดโพรโทคอล (Protocol) TCP/IP อยู่ภายในทำให้สามารถพัฒนาโปรแกรมได้เร็ว ตัวประมวลผลเครือข่าย IP2022 เป็นตัวประมวลผลแบบ RISC (Reduced Instruction Set Computer) ประสิทธิภาพ 120 MIPS (Millon Instructions Per Second) มีพอร์ตอนุกรม 2 พอร์ตสามารถเลือกติดต่อแบบต่าง ๆ ได้ ดังภาพที่ 56



ภาพที่ 56 สถาปัตยกรรมภายในของวงจรรวม IP2022
ที่มา : Ubicom Inc. (2001)

จากภาพที่ 56 พอร์ตอนุกรมสามารถเลือกการทำงานได้หลายประเภท เช่น 10Base-T Ethernet, USB1.1, GPSI, SPI หรือ UART ในการทำงานของตัวประมวลผลจะมีระบบปฏิบัติการ ipOS (ipOS Operating System) ซึ่งจะจัดการทรัพยากรต่าง ๆ ภายใน และในการเขียนโปรแกรมบนตัวประมวลผลเครือข่าย IP2022 จะมีซอฟต์แวร์ ipModule ซึ่งมีลักษณะเป็น TCP/IP Stack ทำให้ง่ายต่อการพัฒนาโปรแกรมบนเครือข่าย

ในรายละเอียดที่สมบูรณ์ของ การเขียนแผนภาพวงจร (Schematic Diagram) และการออกแบบลายวงจร (PCB Design) ของฮาร์ดแวร์เชื่อมต่อเครือข่ายนั้นจะถูกแสดงไว้ในเอกสารอ้างอิง (ธวัชชัย บานใหม่, 2006)

<u>เทคนิคการออกแบบแผงวงจรพิมพ์ความถี่สูง</u>

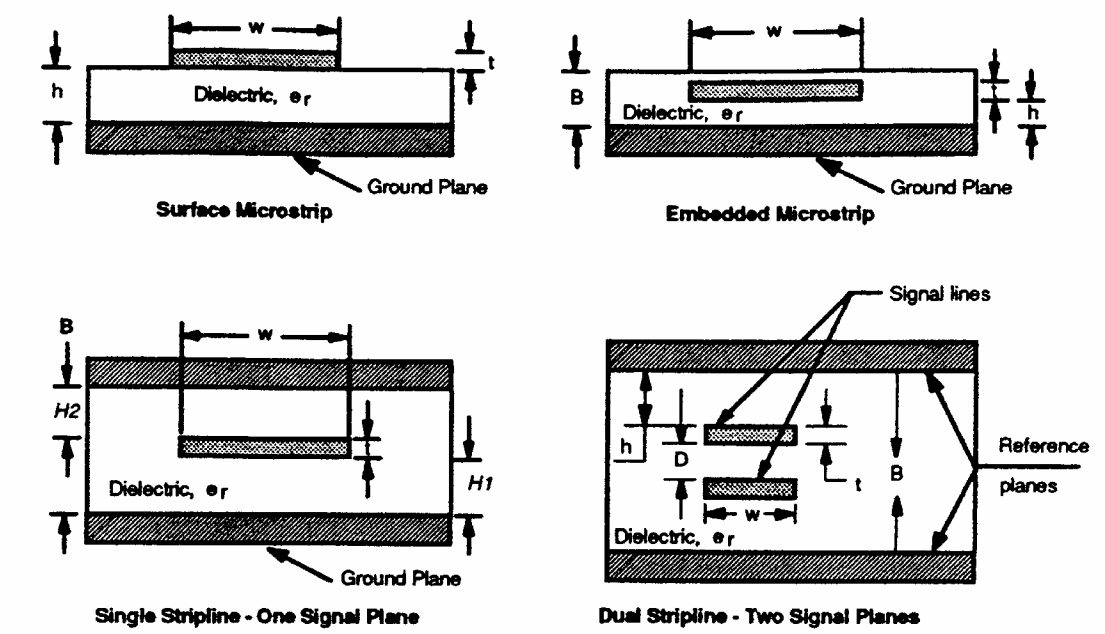
ในการออกแบบวงจรไฟฟ้าหรืออิเลคทรอนิกส์เพื่อนำไปใช้งาน จะต้องนำตัวอุปกรณ์ติดตั้งลงไปบนแผงวงจรพิมพ์ (Printed Circuit Board) หรือแผ่น PCB ซึ่งมีลักษณะเป็นแผ่นสี่เหลี่ยม มีลายวงจรเป็นทองแดงเพื่อใช้เป็นเส้นทางเดินของแหล่งจ่ายไฟ และสัญญาณไฟฟ้าต่างๆ โดยแผงวงจรพิมพ์จะมีขนาดเล็กหรือใหญ่ต่างกันไปตามการออกแบบและลักษณะการใช้งาน แผงวงจรพิมพ์ของอุปกรณ์ไฟฟ้าและอิเลคทรอนิกส์ที่มีราคาแพง ส่วนหนึ่งเกิดจากต้นทุนของแผงวงจรพิมพ์ เนื่องจากอุปกรณ์ไฟฟ้าและอิเลคทรอนิกส์เหล่านี้ถ้ายิ่งขนาดเล็ก และมีระดับชั้นภายใน (Layer) มาก ราคาจะยิ่งสูงตามขึ้นไปด้วย ตัวอย่างเช่น โทรศัพท์มือถือ เครื่องคอมพิวเตอร์โน้ตบุ๊ก เป็นต้น

การออกแบบแผงวงจรพิมพ์ความถี่สูง ควรระวังปัญหาเรื่องการรบกวนทางแม่เหล็กไฟฟ้า (Electromagnetic Interference, EMI) ซึ่งเกิดจากคลื่นแม่เหล็กไฟฟ้าที่มาจากแหล่งกำเนิดต่างกันแล้วเกิดการรบกวนกันขึ้น ไม่ว่าจะเป็นแหล่งกำเนิดจากภายนอกหรือภายในแผงวงจรพิมพ์ ซึ่งทำให้ประสิทธิภาพในการทำงานของแผงวงจรพิมพ์ลดลง (Archambeault, 2002) และปัญหาเรื่องการควบคุมความเข้ากันได้ทางแม่เหล็กไฟฟ้า (Electromagnetic Compatibility, EMC) หรือทำให้เกิดการรบกวนของคลื่นแม่เหล็กไฟฟ้ากับสถาพแวดล้อมภายนอกน้อยที่สุด

การออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ถือเป็นการออกแบบแผงวงจรพิมพ์ความถี่สูง เนื่องจากบัสผ่ายนอกใช้ความถี่สัญญาณนาฬิกาสูงถึง 100 MHz และใช้ตัวประมวลผลสัญญาณดิจิทัล (DSP) TMS320C6713BGDP ซึ่งมีลักษณะเป็น Ball Grid Array (BGA) ซึ่งมีขามากถึง 272 ขา ทำให้ต้องออกแบบแผงวงจรพิมพ์ถึง 6 ระดับชั้น (Six-Layer) ซึ่งถูกออกแบบโดยใช้โปรแกรม PADS Logic 2004 ในการเขียนแผนภาพวงจร (Schematic Diagram) และโปรแกรม PADS Layout 2004 ในการออกแบบลายวงจรของแผงวงจรพิมพ์ (PCB Layout) และโปรแกรม HyperLynx version 7.2 ในการจำลองสัญญาณความถี่สูงของแผงวงจรพิมพ์ ในการออกแบบแผงวงจรพิมพ์นั้นจะอ้างอิงตามหลักการและทฤษฎี ดังต่อไปนี้

1. <u>การกำหนดจำนวนชั้นของแผงวงจร</u>

ในการออกแบบแผงวงจรพิมพ์ สิ่งแรกที่ต้องพิจารณาคือ จำนวนระดับชั้นของสัญญาณ (Signal Layer) ซึ่งก็คือระดับชั้นที่มีเส้นลายวงจรทองแดงของสัญญาณ และระนาบกำลังไฟฟ้า (Power Plane) ที่ต้องการสำหรับใช้งาน โดยที่คำนึงถึงราคาต้นทุนที่เหมาะสมด้วย จำนวนระดับชั้นจะถูกกำหนดโดย ข้อกำหนดในการใช้งาน ความทนทานต่อคลื่นรบกวน การแบ่งแยกหมวดหมู่ของสัญญาณ จำนวนวงจรลายทองแดงของสัญญาณ การควบคุม-ค่าอิมพีแดนซ์ เทคโนโลยีที่ใช้ในการผลิต และสิ่งที่เหมาะสมอื่นๆ สำหรับการใช้งานที่ต้องการกำจัดสัญญาณรบกวนจากคลื่นแม่เหล็กไฟฟ้าในแผงวงจรพิมพ์ ควรเลือกใช้รูปแบบของสายแถบแคบ (Stripline) และไมโครสตริป (Microstrip) ให้เหมาะสม การออกแบบให้ระนาบกำลังไฟฟ้าและระนาบดิน (Ground Plane) ฝังอยู่ในแผงวงจรเป็นวิธีสำคัญที่สุดในการกำจัดสัญญาณรบกวนจากคลื่นแม่เหล็กไฟฟ้า ซึ่งมีข้อดี คือ ระนาบที่อยู่ภายในจะทำให้อิมพิแดนซ์ในการกระจายกำลังของความถี่สูงลดลง



<u>ภาพที่ 57</u> ความแตกต่างของสายแถบแคบ (Stripline) และไมโครสตริป (Microstrip)

ที่มา : Montrose. (1996)

ภาพที่ 57 แสดงถึงความแตกต่างของสายแถบแคบ (Stripline) และไมโครสตริป (Microstrip) ซึ่งจะแสดงให้เห็นทั้ง สองประเภท โดยแต่ละประเภทมีความหมายดังนี้

- ไมโครสตริป (Microstrip) หมายถึง ลายวงจรของระดับชั้นด้านนอกของแผงวงจรพิมพ์ ซึ่งแบ่งแยกด้วย ระนาบไดอิเล็กตริก (Dielectric Plane) และระนาบดินหรือระนาบกำลัง เทคนิคไมโครสตริป สามารถใช้กับสัญญาณนาฬิกา (Clock Signal) และสัญญาณลอจิก (Logic Signal) ได้เร็วกว่าแบบสายแถบแคบ (Stripline) เนื่องจากมีการคับปลิงความจุไฟฟ้า (Capacitive Coupling) น้อยกว่าและไม่มีโหลดที่เวลาประวิงการแพร่กระจาย (Propagation Delay) ที่เกิดขึ้นระหว่างทางเดินของระดับชั้นด้านนอกและระนาบดินหรือระนาบกำลัง การที่มีค่าความจุไฟฟ้าระหว่างระนาบแข็ง (Solid Plane) สองระนาบที่มีค่าต่ำ ซึ่งอาจเป็นระนาบกำลังไฟฟ้าหรือระนาบดินก็ได้ จะทำให้ความเร็วในการแพร่กระจายของสัญญาณดีขึ้น แต่ข้อเสียของไมโครสตริป คือ ระดับชั้นด้านนอกของแผงวงจรจะสามารถกระจายคลื่นแม่เหล็กไฟฟ้า ออกไปสู่สภาพแวดล้อมได้ ดังนั้นควรทำการป้องกันระนาบด้านนอกทั้งสองระนาบ เช่นทำการซิลด์ (Shielding) ทั้งด้านบนและด้านล่างของแผงวงจรพิมพ์ เป็นต้น

- สายแถบแคบ (Stripline) หมายถึง การนำระนาบสัญญาณ (Signal Plane) ไปวางไว้ระหว่างระนาบแข็ง ไม่ว่าจะเป็นระนาบดินหรือระนาบกำลังไฟฟ้า สายแถบแคบมีความทนทานต่อคลื่นรบกวนที่เกิดจากการแพร่ของความถี่คลื่นแม่เหล็กไฟฟ้า แต่จะมีความเร็วในการแพร่ต่ำกว่าแบบไมโครสตริป เนื่องจากระนาบสัญญาณตั้งอยู่ระหว่างระนาบแข็ง ซึ่งอาจเป็นระนาบดินหรือระนาบกำลัง หรือทั้งสองอย่าง ทำให้เกิดการคับปลิงความจุไฟฟ้าระหว่างสองระนาบนี้ ซึ่งทำให้สัญญาณความเร็วสูงมีอัตราการเปลี่ยนแปลงของขอบช้าลง โดยทั่วไปแล้วการคับปลิงตัวเก็บประจุของสายแถบแคบ จะมีผลต่อสัญญาณที่มีค่าความเร็วขอบเร็วกว่า 1 นาโนวินาที ประโยชน์หลักของสายแถบแคบ คือ สามารถป้องกันพลังงานของคลื่นแม่เหล็กไฟฟ้าที่เกิดจากเส้นลายทองแดงภายในและกำจัดการแพร่ของสัญญาณแม่เหล็กไฟฟ้าได้

การกำหนดจำนวนชั้นของแผงวงจรจะต้องคำนึงว่าทุกระดับชั้นของสัญญาณจะต้องอยู่ติดกับระนาบแข็ง ตารางที่ 16 เป็นแนวทางการเลือกจำนวนระดับชั้นของแผงวงจรแบบ 2 ถึง 10 ระดับชั้น ซึ่งสามารถเลือกใช้ให้เหมาะสมกับการใช้งานและจำนวนระดับชั้นของสัญญาณที่ต้องการ

ตารางที่ 16 แนวทางการเลือกระดับชั้นของแผงวงจร

จำนวนระดับชั้น	ระดับชั้นที่										คำแนะนำ
	1	2	3	4	5	6	7	8	9	10	
2 ระดับชั้น	S1 G	S2 P									การออกแบบวงจรที่ใช้ความเร็วต่ำ
4 ระดับชั้น (2 routing)	S1	G	P	S2							ยากในการรักษาค่าอิมพิแดนซ์ของสัญญาณให้สูง และค่าอิมพิแดนซ์ของกำลังไฟฟ้าให้ต่ำ
6 ระดับชั้น (4 routing)	S1	G	S2	S3	P	S4					การออกแบบที่มีความเร็วสูง ค่าอิมพิแดนซ์กำลังไฟฟ้าสูง
6 ระดับชั้น (4 routing)	S1	S2	G	P	S3	S4					เฉพาะที่ S2 เท่านั้นที่ใช้สำหรับสัญญาณความเร็วสูง
8 ระดับชั้น (6 routing)	S1	S2	G	S3	S4	P	S5	S6			สัญญาณความเร็วสูงที่ S2 - S3
8 ระดับชั้น (4 routing)	S1	G	S2	G	P	S3	G	S4			ดีที่สุดสำหรับ EMC
10 ระดับชั้น (6 routing)	S1	G	S2	S3	G	P	S4	S5	G	S6	ดีที่สุดสำหรับ EMC แต่ S4 ไวต่อคลื่นรบกวนของกำลังไฟฟ้า

S = ระดับชั้นของสัญญาณ

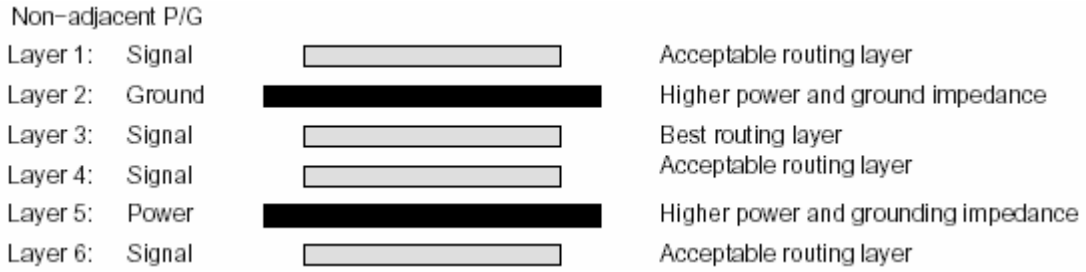
P = ระนาบกำลังไฟฟ้า

G = ระนาบดิน

ที่มา : Montrose. (1996)

สำหรับวงจรแบบ 6 ระดับชั้น มีการจัดวางอยู่ 2 รูปแบบ โดยรูปแบบแรกจะใช้กับอุปกรณ์ที่มีความเร็วสูง ระนาบดินถูกวางในระดับชั้นที่ 2 ระนาบกำลังไฟฟ้าถูกวางในระดับชั้นที่ 5 และ ในระดับชั้นที่ 3 เหมาะสมที่สุดสำหรับสัญญาณนาฬิกาหรือสัญญาณที่มีความไวสูง ขณะที่ระดับชั้นที่ 1, 4 และ 6 ถือว่าจัดวางอยู่ในตำแหน่งที่เหมาะสม ดังแสดงในภาพที่ 58

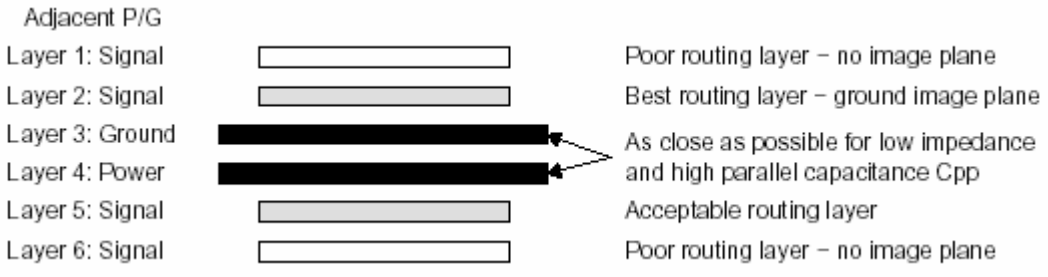
การจัดวางในรูปแบบนี้เป็นการจัดวางที่นิยมใช้กันทั่วไปในปัจจุบัน รวมทั้งถูกใช้ในการออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลด้วย แต่เนื่องจากระนาบแข็งที่อยู่ห่างกัน ทำให้มีค่าอิมพิแดนซ์กำลังไฟฟ้าสูง จำเป็นต้องดีคับปลิงด้วยตัวเก็บประจุไฟฟ้า เพื่อป้องกันคลื่นรบกวนความถี่สูงจากการส่งผ่านพลังงานข้ามระนาบกำลังไฟฟ้าและระนาบดิน



ภาพที่ 58 รูปแบบที่ 1 ของการจัดวางแผงวงจรแบบ 6 ระดับชั้น

ที่มา : Montrose. (1996)

สำหรับการจัดวางในรูปแบบที่ 2 ถูกออกแบบเพื่อการปรับปรุงผลอันเนื่องมาจากการเพิ่มขึ้นของการดีคับปลิง ระหว่างระนาบกำลังไฟฟ้าและระนาบดิน โดยลดระยะห่างระหว่างระนาบทั้งสองลง เพื่อทำให้มีค่าอิมพิแดนซ์กำลังไฟฟ้าต่ำลง มีรูปแบบดังแสดงในภาพที่ 59 เฉพาะระดับชั้นที่ 2 เท่านั้นที่เหมาะสำหรับสัญญาณความเร็วสูง เนื่องจากอยู่ติดกับระนาบดิน ซึ่งไม่เหมาะในการใช้กับตัวประมวลผลสัญญาณดิจิทัล (DSP) ซึ่งต้องการหลายระดับชั้นสำหรับเดินสัญญาณความเร็วสูง

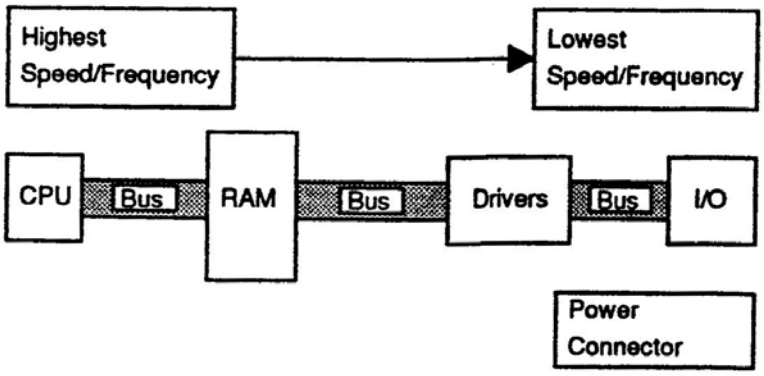


ภาพที่ 59 รูปแบบที่ 2 ของการจัดวางแผงวงจรแบบ 6 ระดับชั้น

ที่มา : Montrose. (1996)

เมื่อเลือกรูปแบบของแผงวงจรได้แล้ว ขั้นตอนต่อไปคือการจัดวางอุปกรณ์ ภาพที่ 60 แสดงถึงการจัดวางอุปกรณ์แบบแพร่กระจายตามรัศมีจากอุปกรณ์ความถี่สูงไปยังอุปกรณ์ความถี่ต่ำ อุปกรณ์ความถี่สูงที่ต้องติดต่อกันควรจะอยู่ใกล้กัน อุปกรณ์ความถี่ต่ำก็ควรอยู่ในกลุ่มพื้นที่ของอุปกรณ์ความถี่ต่ำ และแยกพื้นที่ของวงจรแอนาลอกออกจากวงจรดิจิทัล ยกตัวอย่างเช่น ตัวประมวลผลควรอยู่ใกล้กับหน่วยความจำ ข้อต่อของสัญญาณอินพุต-เอาต์พุตกับภายนอก ก็ควรอยู่ใกล้กับอุปกรณ์ขับสัญญาณอินพุต-เอาต์พุต เป็นต้น เพื่อลดการรบกวนกันทาง

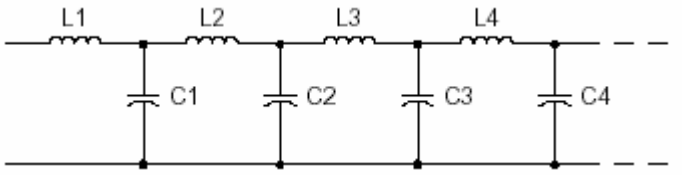
คลื่นแม่เหล็กไฟฟ้า และป้องกันสัญญาณรบกวนเหล่านั้นแพร่กระจายออกไปภายนอก โดยผ่านข้อต่อของสัญญาณอินพุต-เอาต์พุต



ภาพที่ 60 การจัดวางอุปกรณ์แบบแพร่กระจายตามแนวรัศมี
ที่มา : I. Montrose. (1996)

2. ทฤษฎีสายส่งสัญญาณ

ปรากฏการณ์ของสายส่งสัญญาณ (Transmission line, TL) เป็นสาเหตุหนึ่งที่ทำให้เกิดสัญญาณรบกวนขึ้นในแผงวงจรพิมพ์ความถี่สูง ลายวงจรทองแดงซึ่งเป็นเส้นทางเดินของสัญญาณไฟฟ้าจะถูกเรียกเป็นสายส่งสัญญาณ ก็ต่อเมื่อ สัญญาณนั้นมีค่าช่วงเวลาขึ้นของสัญญาณ (Rise Time, Tr) ช้ากว่าค่าประวิงเวลาการแพร่กระจายของสัญญาณ (Propagation Delay, Tp) ในภาพที่ 61 ได้แสดงถึงโมเดลของสายส่งสัญญาณ และพารามิเตอร์ต่างๆที่ใช้ในการกำหนดลักษณะของสายส่งสัญญาณ



speed of the signal, $V_p = \frac{1}{\sqrt{LC}}$,

propagation delay, $T_d = \frac{1}{Vp} = \sqrt{LC}$.

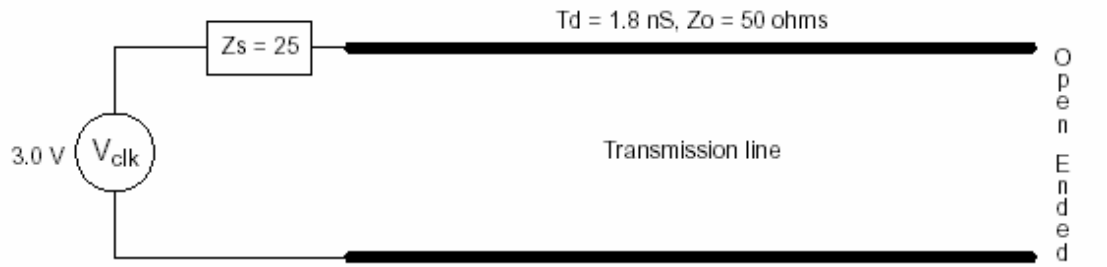
characteristic impedance, $Z_o = \sqrt{\frac{L}{C}}$,

$\Gamma_s = source_reflections = \frac{Z_s - Z_o}{Z_s + Z_o}$,

$\Gamma_s = load_reflections = \frac{Z_L - Z_o}{Z_L + Z_o}$,

ภาพที่ 61 โมเดลของสายส่งสัญญาณ
ที่มา : Texas Instruments. (2005)

ในภาพที่ 62 แสดงวงจรแหล่งกำเนิดสัญญาณนาฬิกาความถี่สูง ที่มีค่าอิมพิแดนซ์ภายใน 25 โอห์ม ส่งสัญญาณผ่านสายส่งสัญญาณ โดยมีโหลดที่ปลายทางเป็นแบบวงจรเปิด และในภาพที่ 63 แสดงภาพของสัญญาณที่เกิดขึ้นในสายส่งสัญญาณ โดยจะเห็นว่าระดับแรงดันมีการพุ่งเกิน (Overshoot) และพุ่งต่ำ (Undershoot) กว่าแรงดันปกติในช่วงแรก ซึ่งปรากฏการณ์ลักษณะนี้อาจทำให้เกิดสัญญาณลอจิกแบบสุ่ม (Random Logic) และเกิดความผิดพลาดในการอ่านค่าของตัวประมวลผลได้ ซึ่งการหาค่าการพุ่งเกินของสัญญาณอย่างง่ายๆ สามารถคำนวณได้ดังนี้



ภาพที่ 62 วงจรการส่งผ่านสัญญาณนาฬิกาความถี่สูง

ที่มา : Texas Instruments. (2005)

$$V_{initial} = V_{clk}\frac{Zo}{Zs + Zo} = 3\frac{50}{25 + 50} = 2V$$

$$\rho_s = \frac{Zs - Zo}{Zs + Zo} = \frac{25 - 50}{25 + 50} = -0.333$$

$$\rho_L = \frac{Z_L - Zo}{Z_L + Zo} = 1$$

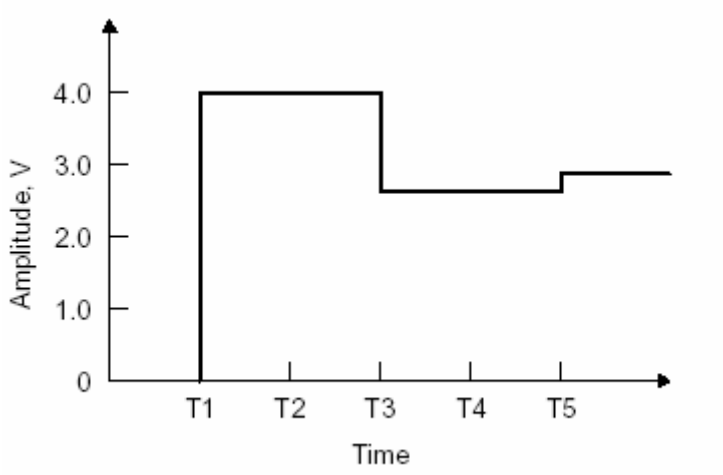
ที่เวลา T1 = 1.8 nS : $V_L = V_I + \rho_L V_I = 2 + 2 = 4.0$ Volt

ที่เวลา T1 = 3.6 nS : $V_S = 4.0 + \rho_S V_I = 4.0 - 0.67 = 3.33$ Volt

ที่เวลา T1 = 5.4 nS : $V_L = 3.33 + \rho_L (\rho_S V_I) = 3.33 - 0.67 = 2.66$ Volt

ที่เวลา T1 = 7.2 nS : $V_S = 2.66 + \rho_S[\rho_L (\rho_S V_I)] = 2.66 + 0.22 = 2.88$ Volt

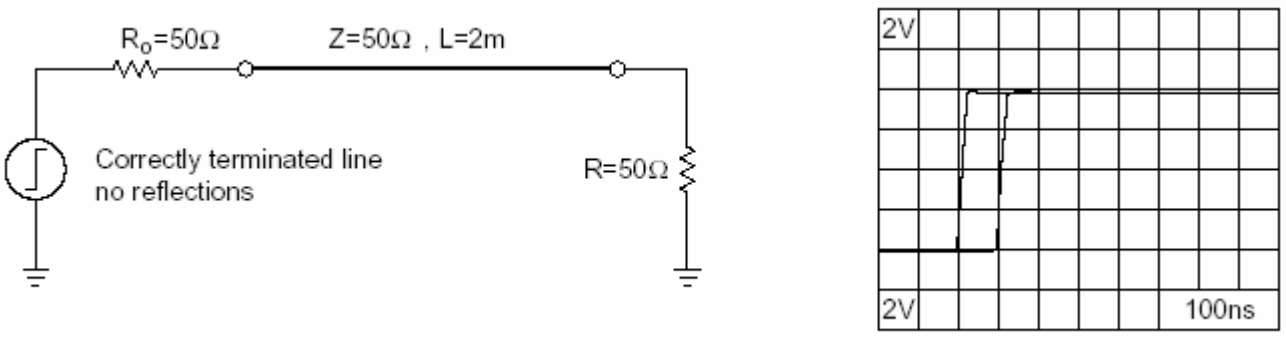
ที่เวลา T5 = 9.0 nS : $V_L = 2.88 + \rho_L (0.22) = 3.1$ Volt



ภาพที่ 63 การตอบสนองชั่วครู่ของสัญญาณนาฬิกาความถี่สูง

ที่มา : Texas Instruments. (2005)

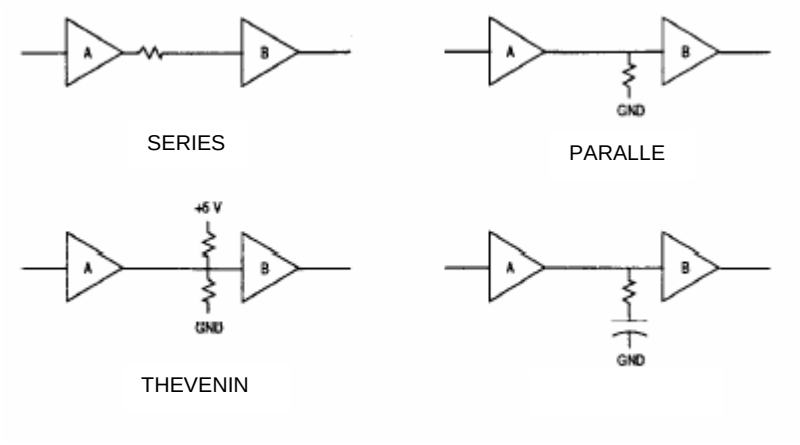
จากตัวอย่างและสมการข้างต้นพิสูจน์ได้ว่าผลของแรงดันพุ่งเกินและแรงดันพุ่งต่ำขึ้นกับความเหมาะสมกันระหว่างเอาต์พุตอิมพิแดนซ์กับโหลดอิมพิแดนซ์ ซึ่งสามารถแก้ไขได้โดยวิธีการเพิ่มโหลด (Termination Technique) เพื่อทำให้ตัวเลขสัมประสิทธิ์ของค่าการสะท้อนมีค่าน้อยที่สุดเท่าที่จะเป็นไปได้ ซึ่งถ้าตัวเลขสัมประสิทธิ์ของค่าการสะท้อนเป็นศูนย์จะทำให้ไม่เกิดการผิดเพี้ยนของสัญญาณเลยดังแสดงในภาพที่ 64



ภาพที่ 64 การเพิ่มโหลดอย่างสมบูรณ์แบบ

ที่มา : Texas Instruments. (2005)

วิธีการเพิ่มโหลดนั้นมีอยู่หลายวิธีดังแสดงในภาพที่ 65 แต่วิธีที่ใช้ง่ายและได้รับความนิยมมากที่สุด คือ การเพิ่มโหลดแบบอนุกรม (Series Termination) ซึ่งจะเพิ่มโหลดในฝั่งของแหล่งกำเนิด

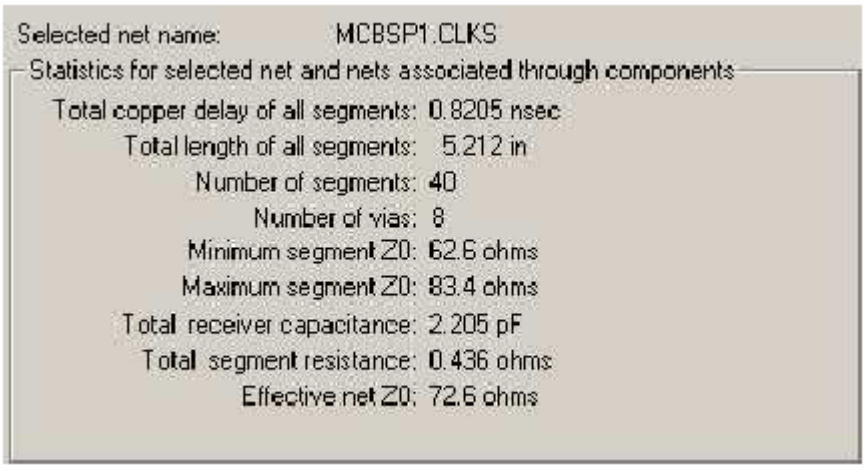


ภาพที่ 65 วิธีการเพิ่มโหลดแบบต่างๆ

ที่มา : Johnson. (1993)

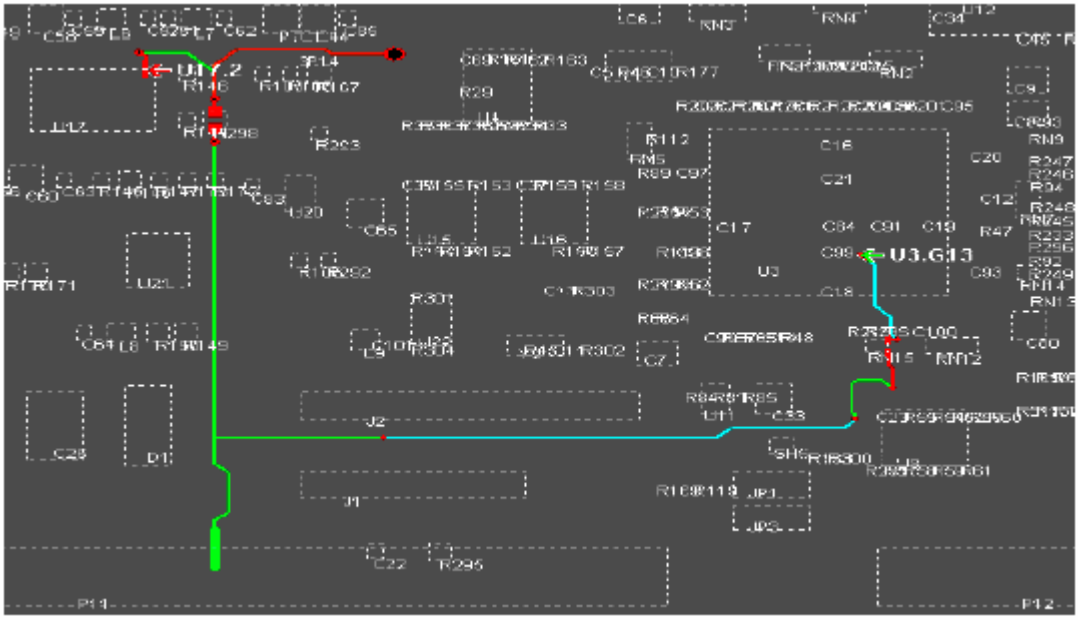
วิธีการเพิ่มโหลดแบบอนุกรม ถูกนำมาใช้ในการออกแบบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล เพื่อทำให้สัญญาณมีความถูกต้องมากขึ้น ยกตัวอย่างเช่น ขาสัญญาณของมอดูล EMIF จะถูกใส่เพิ่มด้วยโหลดตัวต้านทานแบบอนุกรมค่า 33 โอห์ม เข้ากับขาสัญญาณ EDx, EAx, #CEx, #BEx, #AOE, #AWE, #ARE, ECLKOUT ตัวต้านทานเหล่านี้จะต้องต่อใกล้กับตัวประมวลผล DSP มากที่สุด เพื่อความเสถียรของสัญญาณ

โปรแกรม HyperLynx เป็นเครื่องมือช่วยในการจำลองสัญญาณความถี่สูงของแผงวงจรพิมพ์ และช่วยในการหาค่าโหลดที่จะใช้ลดผลของแรงดันพุ่งเกินและแรงดันพุ่งต่ำ ในช่วงการตอบสนองชั่วครู่ (Transient Response) ของสัญญาณความถี่สูง การทำงานจะจำลองโมเดล (Model) การส่งผ่านสัญญาณของทั้งแผงวงจรพิมพ์ โดยการเชื่อมโยงกับโปรแกรม PADS Layout และทำการจำลองโมลเดลต่างๆของอุปกรณ์ ทั้งในภาคส่งและภาครับ ซึ่งนำข้อมูลมาจากไฟล์โมเดลจำลองของอุปกรณ์ (IBIS Model) ซึ่งสามารถดาวน์โหลดได้จากที่อยู่เว็บ (Website) ของผู้ผลิตวงจรรวมชนิดนั้น ตัวอย่างการใช้งานโปรแกรม HyperLynx ได้แสดงไว้ในภาพที่ 66 , 67 และ 68



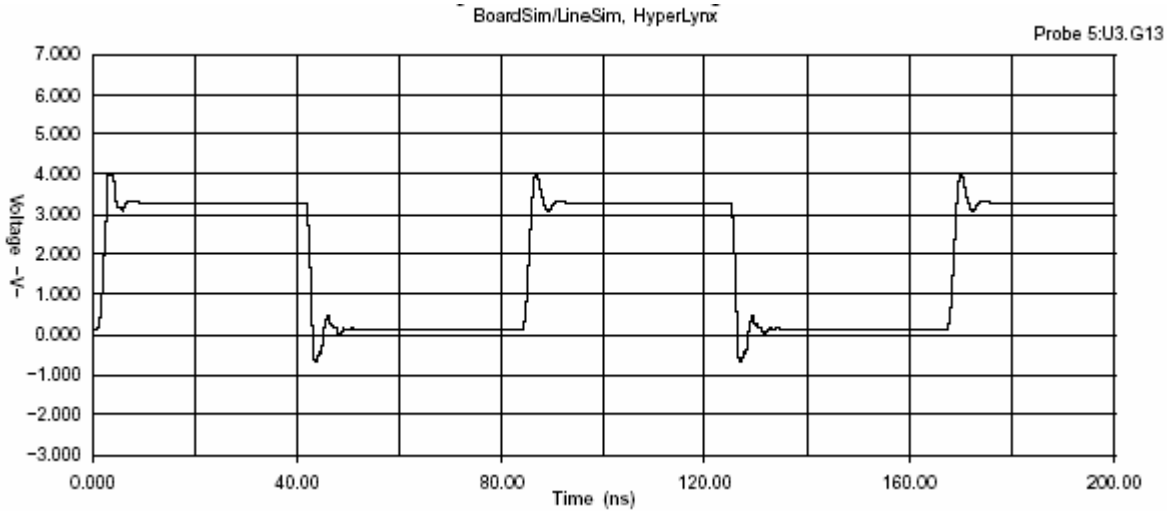
ภาพที่ 66 ค่าพารามิเตอร์ต่างๆในการจำลองสัญญาณด้วยโปรแกรม HyperLynx

ที่มา : Texas Instruments. (2005)



ภาพที่ 67 การจำลองเส้นทางการส่งผ่านสัญญาณด้วยโปรแกรม HyperLynx

ที่มา : Texas Instruments. (2005)



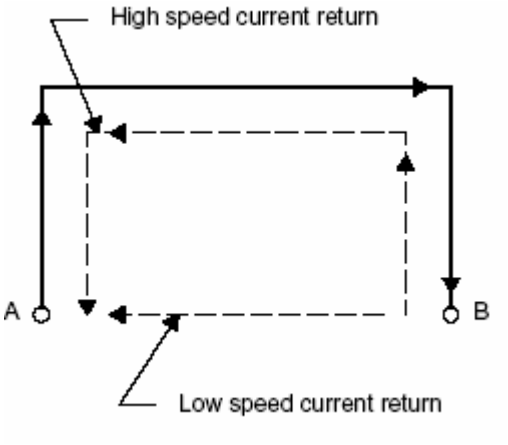
ภาพที่ 68 ผลการจำลองสัญญาณความถี่สูงด้วยโปรแกรม HyperLynx

ที่มา : Texas Instruments. (2005)

3. กระแสย้อนกลับ

ความรู้ความเข้าใจเรื่องกระแสย้อนกลับ (Return Current Parths) มีส่วนช่วยสำคัญในการกำจัดสัญญาณรบกวนทางแม่เหล็กไฟฟ้า (EMI) ได้เป็นอย่างดี การไหลของกระแสไฟฟ้าความถี่สูงจะแตกต่างจากการไหลของไฟฟ้ากระแสไฟฟ้าตรง (Direct Current, DC) โดยทั่วไปแล้วกระแสไฟฟ้าจะเริ่มต้นที่แหล่งกำเนิด เดินทางผ่านสายส่งสัญญาณไปยังภาครับ และย้อนกลับไปยังแหล่งกำเนิดผ่านระนาบดิน การย้อนกลับของสัญญาณไฟฟ้ากระแสตรงหรือสัญญาณไฟฟ้าความถี่ต่ำ สัญญาณจะเลือกทางเดินที่สั้นที่สุดหรือมีความต้านทานต่ำ (Resistance) ที่สุด แต่สำหรับสัญญาณที่มีความถี่เกิน 10 MHz ขึ้นไป สัญญาณจะเลือกทางเดินที่มีความอิมพิแดนซ์ (Impedance) ต่ำที่สุด ซึ่งโดยทั่วไปแล้วจะเป็นบริเวณพื้นที่ผิวด้านบนของระนาบดิน ซึ่งอยู่ด้านล่างของสายส่งสัญญาณ โดยกระแสย้อนกลับจะมีทิศทางตรงข้ามกับสัญญาณส่งเสมอ ความแตกต่างของพฤติกรรมของสัญญาณความถี่สูงและสัญญาณความถี่ต่ำได้แสดงไว้ในภาพที่ 69

ดังนั้นจึงเป็นเหตุผลที่ต้องมีระนาบอ้างอิงอยู่ติดกับระนาบสัญญาณเสมอ การใช้ระนาบอ้างอิงจะทำให้ค่าอิมพีแดนซ์ของเส้นทางเดินของกระแสของสัญญาณความถี่สูงที่ย้อนกลับไปยังแหล่งกำเนิดมีค่าต่ำลง และเพื่อไม่ให้เส้นทางเดินของกระแสย้อนกลับนั้นไปรบกวนสัญญาณอื่นๆ ในวงจร โดยที่ ระนาบอ้างอิง หมายถึง ระดับชั้นของทองแดงที่อยู่ภายในแผงวงจรพิมพ์ อาทิเช่น ระนาบดิน ระนาบกำลัง ตัวถังโลหะ (Chassis) เป็นต้น



ภาพที่ 69 การไหลของกระแสย้อนกลับ

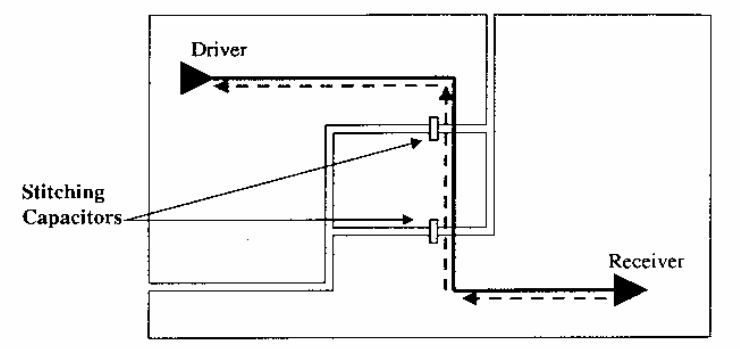
ที่มา : Texas Instruments. (2005)

ในการออกแบบลายวงจรทองแดงโดยไม่ระมัดระวัง บางครั้งอาจเกิดการรบกวนเส้นทางการเดินของกระแสย้อนกลับได้ เช่น ระนาบอ้างอิงไม่ต่อเนื่อง (Split Reference Planes) และ การเปลี่ยนระนาบอ้างอิง (Changing Reference Planes) เป็นต้น

1) ระนาบอ้างอิงไม่ต่อเนื่อง

สิ่งที่ควรจำไว้เสมอเกี่ยวกับกระแสย้อนกลับ คือ กระแสย้อนกลับเป็นผลจากการเหนี่ยวนำระหว่างสายส่งสัญญาณและระนาบอ้างอิง และกระแสย้อนกลับจะเดินอยู่ใต้สายส่งสัญญาณเสมอ และระนาบอ้างอิงนั้นเป็นไปได้ทั้ง ระนาบดินและระนาบกำลังไฟฟ้า โดยปกติแล้วระนาบดินจะต่อเนื่องกันทั้งแผ่น จึงเหมาะสำหรับใช้เป็นทางเดินของกระแสย้อนกลับ แต่ระนาบกำลังไฟฟ้ามักจะถูกแบ่งเป็นส่วน ๆ แต่ละส่วนใช้สำหรับแรงดันไฟฟ้าที่ต่างกัน เมื่อสายส่งสัญญาณวิ่งข้ามรอยแยกที่แบ่งระนาบกำลังไฟฟ้าเหล่านั้น ทำให้กระแสย้อนกลับไม่สามารถผ่านรอยแยกนั้นมาได้ ดังนั้นกระแสย้อนกลับต้องหาเส้นทางอื่นในการย้อนกลับเข้าสู่แหล่งกำเนิด ซึ่งเป็นสาเหตุให้เกิดปัญหาของคลื่นรบกวนความถี่สูงบนแผงวงจรพิมพ์ได้ ดังแสดงในภาพที่ 70

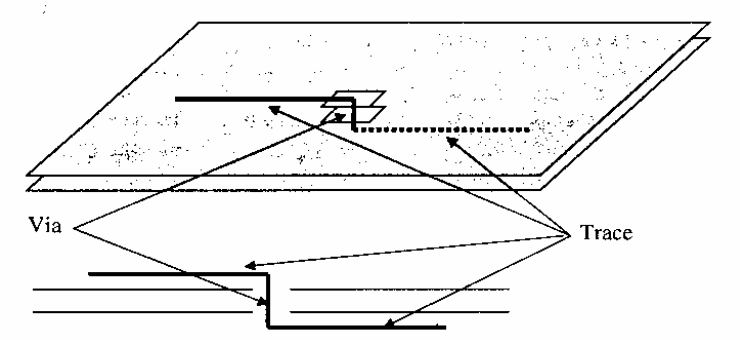
อย่างไรก็ตามผู้ออกแบบไม่สามารถหลีกเลี่ยงการเดินสายส่งสัญญาณผ่านรอยแยกของระนาบอ้างอิงทั้งหมดได้ ตัวเก็บประจุแบบสทิช (Stitching Capacitor) ซึ่งใช้เชื่อมต่อระหว่างรอยแยกระว่างสองแรงดันไฟฟ้า จะสามารถใช้เป็นเส้นทางเดินของกระแสย้อนกลับได้ ค่าอิมพิแดนซ์ของตัวเก็บประจุแบบสทิชจึงเป็นส่วนสำคัญที่ต้องพิจราณา ควรเลือกตัวเก็บประจุแบบสทิชที่มีค่าอิมพิแดนซ์ต่ำที่สุด ณ ที่ความถี่ของกระแสย้อนกลับ และวางอยู่ใกล้กับสายส่งสัญญาณที่สุด



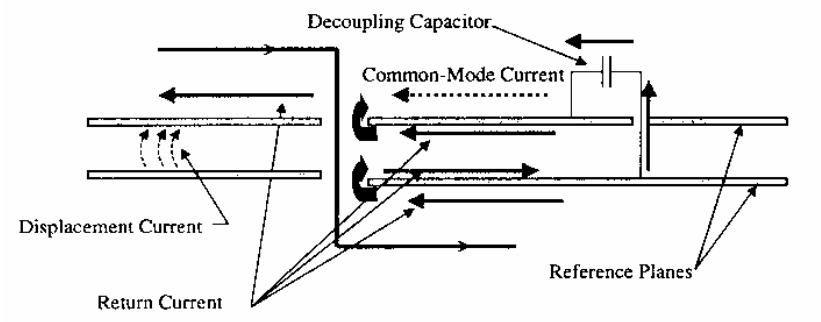
ภาพที่ 70 การเชื่อมต่อตัวเก็บประจุแบบสทิชระหว่างรอยแยกของระนาบอ้างอิง
ที่มา : I. Montrose. (1996)

2) การเปลี่ยนระนาบอ้างอิง

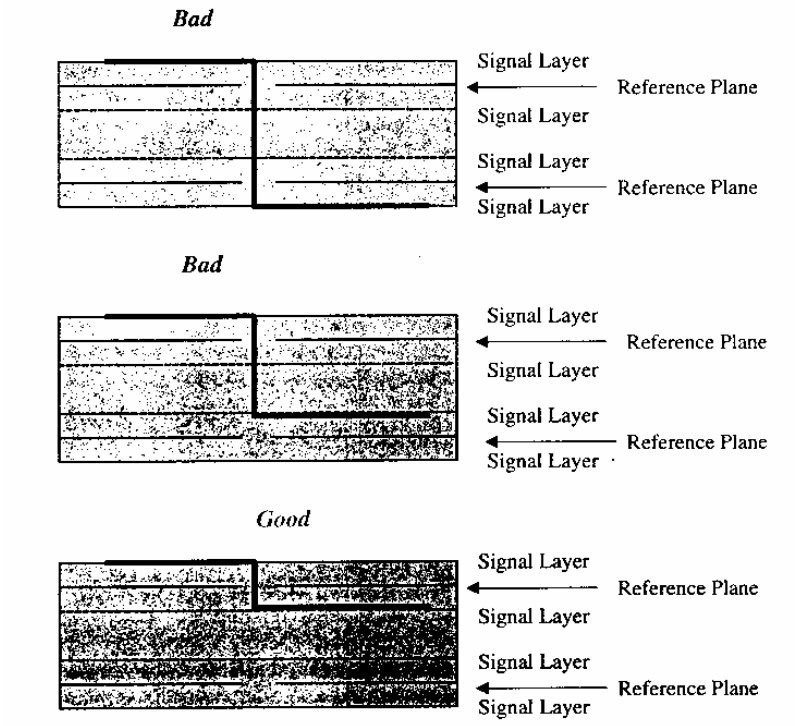
ในการออกแบบลายวงจรทองแดง บ่อยครั้งที่สายส่งสัญญาณมีความจำเป็นต้องเดินผ่านในหลายระดับชั้นจากแหล่งกำเนิดไปยังภาครับ หมายถึงกระแสย้อนกลับจำเป็นต้องเปลี่ยนระนาบอ้างอิงเช่นเดียวกัน ภาพที่ 71 เป็นตัวอย่างแสดงการเปลี่ยนระดับชั้นของสายส่งสัญญาณผ่านรูทะลุผ่านชั้น (Via) ในตัวอย่างนี้แสดงเป็นแผงวงจรพิมพ์ 4 ระดับชั้น สายส่งสัญญาณเดินผ่านระนาบสัญญาณทั้งด้านบนและด้านล่าง สิ่งที่เกิดขึ้นคือ กระแสย้อนกลับความถี่ต่ำจะวิ่งข้ามระนาบโดยผ่านตัวเก็บประจุแบบดีคับปลิงที่อยู่ใกล้กับรูทะลุผ่านชั้นที่สุด สำหรับกระแสย้อนกลับความถี่สูงจะวิ่งข้ามระนาบโดยผ่านตัวเก็บประจุเสมือน ดังภาพที่ 72



ภาพที่ 71 การเปลี่ยนระดับชั้นของสายส่งสัญญาณผ่านรูทะลุผ่านชั้น (Via)
ที่มา : I. Montrose. (1996)



ภาพที่ 72 ทางเดินของกระแสย้อนกลับเมื่อเปลี่ยนระนาบอ้างอิง
ที่มา : I. Montrose. (1996)



ภาพที่ 73 วิธีทางเดินของกระแสย้อนกลับเมื่อเปลี่ยนระนาบอ้างอิง
ที่มา : Texas Instruments. (2005)

โดยสรุปแล้วคือไม่ควรให้เกิดการเปลี่ยนระนาบอ้างอิง แต่ไม่ได้หมายความว่าจะต้องเดินสายส่งสัญญาณทั้งหมดภายในระนาบสัญญาณเดียว แต่ควรระมัดระวังในเรื่องการการเปลี่ยนระดับชั้นของสายส่งสัญญาณ ภาพที่ 73 แสดงการแผงวงจรพิมพ์ 6 ระดับชั้น ภาพที่ 73(ก) และ 73 (ข) แสดงการเดินสายส่งสัญญาณผ่านสองระนาบสัญญาณและเปลี่ยนระนาบอ้างอิง ซึ่งต้องการ

ตัวเก็บประจุแบบคับปลิง ภาพที่ 73(ค) แสดงการเดินสายส่งสัญญาณผ่านสองระนาบสัญญาณโดยไม่เปลี่ยนระนาบอ้างอิง เพียงแต่กระแสย้อนกลับจะเดินผ่านผิวด้านบนและผิวด้านล่างของระนาบอ้างอิง โดยผ่านรูทะลุผ่านชั้น (Via) ของระนาบอ้างอิง จึงทำให้ไม่เกิดคลื่นรบกวนความถี่สูงขึ้นบนแผงวงจรพิมพ์ ดังนั้นรูปแบบการเดินสายส่งสัญญาณในภาพที่ 73(ค) จึงเหมาะสมและดีที่สุด

4. <u>การบายพาสและดีคับปลิง</u>

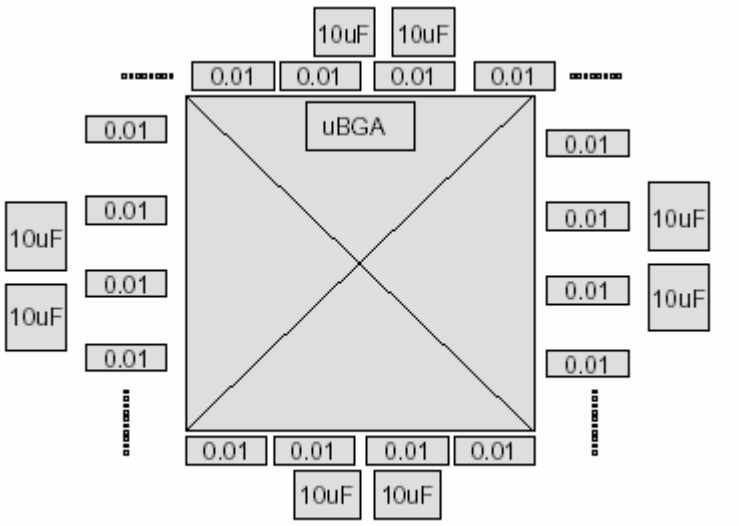
การบายพาสและดีคับปลิงเป็นเทคนิคที่ใช้ในการป้องกันการส่งผ่านพลังงานข้ามระหว่างระนาบกำลังไฟฟ้าและระนาบดิน โดยมีบริเวณของวงจรที่น่าสนใจอยู่ 3 บริเวณ คือ ระนาบกำลังไฟฟ้าและระนาบดิน อุปกรณ์ และการเชื่อมต่อกำลังไฟฟ้าภายใน

การดีคับปลิงที่ดีต้องจ่ายแรงดันและกระแสไฟฟ้ากระแสตรงเพียงพอต่อความเหมาะสมในการทำงานของอุปกรณ์เชื่อมต่อที่อยู่ในช่วงการเปลี่ยนผ่านสัญญาณนาฬิกาหรือข้อมูล ที่เกิดจากการเปลี่ยนแปลงอย่างทันทีทันใด ภายใต้ค่าความจุไฟฟ้าของโหลดสูงสุดการใช้ดีคับปลิงให้มีประสิทธิภาพนั้น ต้องแน่ใจว่ามีความเป็นอิมพิแดนซ์ต่ำของแหล่งจ่ายไฟในระนาบกำลังไฟฟ้าและระนาบดิน คลื่นรบกวนความถี่สูงจะถูกกำจัดออกไปจากเส้นลายทองแดงของสัญญาณอย่างมีประสิทธิภาพ เนื่องจากตัวเก็บประจุจะมีอิมพิแดนซ์ต่ำลงเมื่อความถี่สูงขึ้น การใช้ตัวเก็บประจุแบบบัลค์ (Bulk Capacitor) บายพาส และ ดีคับปลิง จะให้ผลดีที่สุดสำหรับการลดการรบกวนทางแม่เหล็กไฟฟ้า (EMI) ค่าความจุไฟฟ้าทั้งหมดจะต้องถูกคำนวณสำหรับการใช้งานเฉพาะแต่ละงาน และต้องเลือกวัสดุไดอิเล็กตริกของตัวเก็บประจุให้เหมาะสมด้วย คำจำกัดความของตัวเก็บประจุทั้งสามแบบในการกำจัดการรบกวนทางแม่เหล็กไฟฟ้ามีดังต่อไปนี้

- แบบดีคับปลิง : สามารถกำจัดพลังงานคลื่นความถี่สูงบนระนาบกำลัง ซึ่งเกิดจากอุปกรณ์ความถี่สูง ตัวเก็บประจุแบบดีคับปลิงยังเป็นแหล่งเก็บสะสมกำลังไฟฟ้ากระแสตรงสำหรับอุปกรณ์แต่ละตัว และสามารถลดค่าการกระชากกระแส (Surge) ที่เกิดขึ้นในแผงวงจรได้
- แบบบายพาส : สามารถกำจัดคลื่นความถี่ที่ไม่ต้องการออก และยังทำหน้าที่เป็นตัวกรอง (Filter) ได้อีกด้วย
- แบบบัลค์ : ใช้ในการรักษาค่าแรงดันและกระแสไฟฟ้ากระแสตรงที่จ่ายไปยังอุปกรณ์ต่างๆให้คงที่ เมื่อขาสัญญาณทั้งหมดเปลี่ยนแปลงอย่างทันทีทันใดภายใต้ค่าความจุไฟฟ้าของโหลดสูงสุด และยังป้องกันกำลังไฟฟ้าตก เนื่องจากกระแสกระชากที่เกิดจากอุปกรณ์ต่างๆ

ข้อกำหนดในการดีคับปลิงของตัวประมวลผลสัญญาณดิจิทัล (DSP) คือ ใส่ตัวเก็บประจุแบบดีคับปลิงระหว่างทุกขากำลังไฟฟ้าและขาดินให้มากที่สุดเท่าที่จะเป็นไปได้ ที่ด้านล่างของตัวประมวลผล DSP โดยมีระยะห่างระหว่างตัวเก็บประจุถึงขาของตัวประมวลผล DSP ไม่ควรเกิน 1.25 เซนติเมตร และใส่ตัวเก็บประจุแบบบัลก์ อย่างน้อย 8 ตัว สำหรับแหล่งจ่ายอินพุต-เอาต์พุต (IO supplies) 4 ตัว และแหล่งจ่ายภายใน (Core supplies) 4 ตัว โดยอยู่ที่บริเวณแต่ละด้านของตัวประมวลผล DSP ตัวเก็บประจุแบบบัลก์จะทำหน้าที่เป็นตัวกรองความถี่ต่ำ และเก็บรักษาค่าแรงดันและกระแสไฟฟ้าที่จ่ายไปยังตัวเก็บประจุแบบดีคับปลิงต่างๆให้คงที่ แผนภาพการวางตำแหน่งของตัวเก็บประจุแสดงได้ดังภาพที่ 74

การกำหนดชนิดและขนาดของตัวเก็บประจุ ตัวเก็บประจุแบบดีคับปลิงให้ใช้เป็นชนิดเซรามิกค่าความจุ 0.01-0.1 μF โดยมีขนาดเล็กที่สุดเท่าที่เป็นไปได้ เนื่องจากตัวเก็บประจุที่มีขนาดใหญ่จะมีค่าความต้านทานเสมือนมากกว่าตัวเก็บประจุขนาดเล็ก สำหรับตัวเก็บประจุแบบบัลก์ให้ใช้เป็นชนิดแทนทาลัมค่าความจุ 10 μF และมีขนาดใหญ่ เนื่องจากตัวเก็บประจุแบบแทนทาลัมจะมีค่าความจุและขนาดใหญ่กว่าแบบเซรามิก ที่สำคัญตัวเก็บประจุทั้งสองชนิดจะต้องเป็นประเภทที่มีค่าตัวนำเสมือน (Equivalent Series Inductance, ESL) และค่าตัวต้านทานเสมือน (Equivalent Series Resistance, ESR) ต่ำ เพื่อป้องกันสัญญาณรบกวนที่เกิดขึ้นขณะความถี่สูง



ภาพที่ 74 แผนภาพการวางตำแหน่งตัวเก็บประจุแบบดีคับปลิง

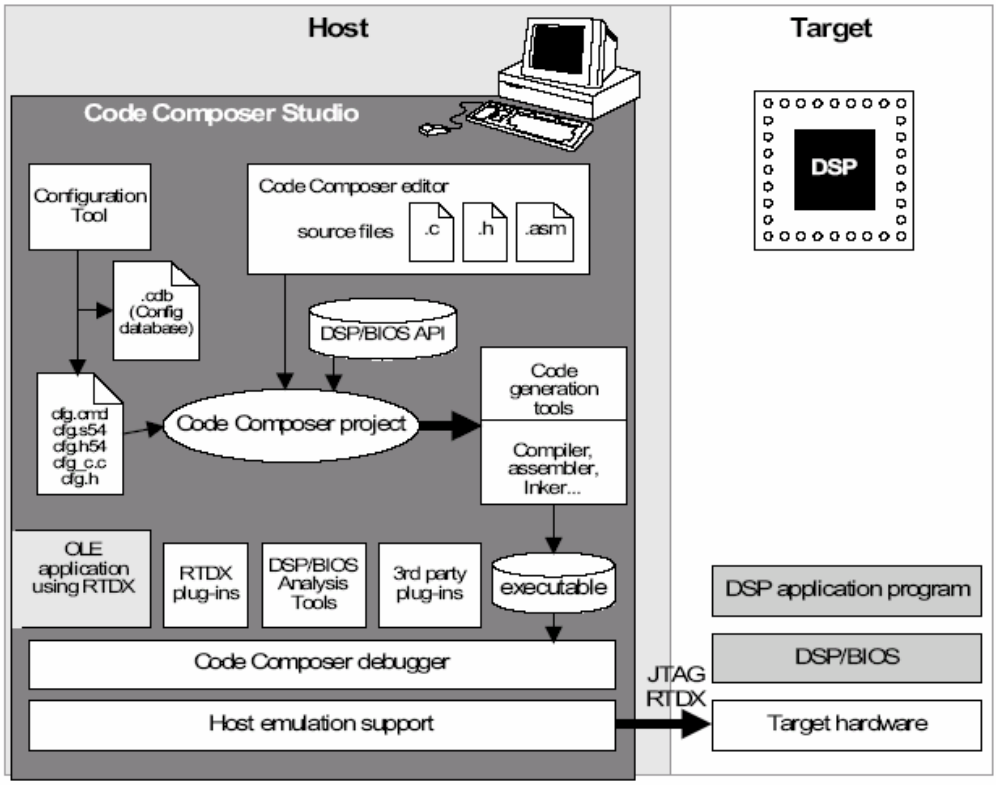
ที่มา : Texas Instruments. (2005)

การทดสอบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

การเขียนโปรแกรมควบคุมและทดสอบ เป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งนอกเหนือจากขั้นตอนการออกแบบ เนื่องจากเป็นการทดสอบว่าฮาร์ดแวร์จะทำงานได้ตามที่ออกแบบไว้หรือไม่ ชุดเครื่องมือรวมสำหรับเขียนโปรแกรม (Code Composer Studio) หรือ CCS ของบริษัท TI เป็นเครื่องมือช่วยสำคัญที่ใช้ในการเขียน และทดสอบโปรแกรมบนตัวประมวลผลสัญญาณดิจิทัล (DSP) ของ TI หรือเรียกได้ว่าเป็นการพัฒนาแบบบูรณาการ (Integrated Development, IDE) ซึ่งทำงานอยู่บนตัวปฏิบัติการวินโดวส์ ซึ่งโปรแกรม CCS ได้ถูกใช้เป็นเครื่องมือช่วยในการสร้างและเขียนโปรแกรม ยกตัวอย่างเช่น ตัวแปลโปรแกรมภาษาซี (C Compiler) ตัวแปลโปรแกรมภาษาแอสเซมบลี (Assembler) และเป็นโปรแกรมที่ใช้ในการเชื่อมโยงโปรแกรมหลาย ๆ โปรแกรมให้เป็นโปรแกรมเดียว (Linker) นอกจากนี้แล้ว โปรแกรม CCS ยังมีเครื่องมือช่วยในการวาดกราฟและแก้ไขจุดบกพร่องได้ในเวลาจริง (Real-Time Debugging) ซึ่งทำให้ง่ายต่อการใช้งานในการสร้าง (Build) และแก้ไขข้อผิดพลาดของโปรแกรม (Debug)

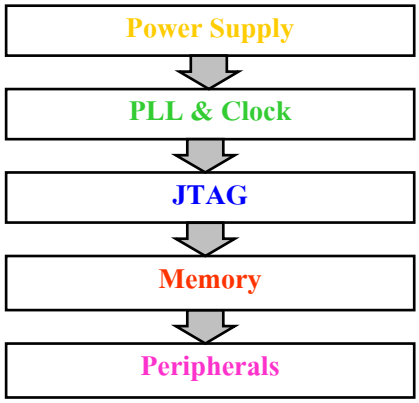
การใช้งานโปรแกรม CCS เริ่มด้วยการสร้างโปรแกรมพัฒนาโครงงาน (Application Project) จากนั้นเพิ่มเติมไฟล์ที่จำเป็นต่อการใช้งานลงในโครงงาน แล้วจึงทำการแปล (Compile) และเชื่อมโยง (Link) ซึ่งมีขั้นตอนดังต่อไปนี้ คือเริ่มต้นจากตัวแปลโปรแกรมภาษาซีจะทำหน้าที่แปลงโปรแกรมต้นฉบับภาษาซี (.c) ใช้เป็นโปรแกรมภาษาแอสแซมบลี (.asm) จากนั้นตัวแปลโปรแกรมภาษาแอสเซมบลีจะแปลภาษาแอสแซมบลีให้เป็นภาษาเครื่อง (.obj) และตัวเชื่อมโยง (Linker) จะนำเอาไฟล์จุดหมาย (Object File) และไฟล์อื่น ๆ ในคลัง (Libraries File) มารวมเข้าด้วยกันเพื่อสร้างเป็นไฟล์ที่พร้อมสำหรับใช้งาน (Executable File) หรือเรียกอีกอย่างว่ารูปแบบไฟล์จุดหมายร่วมกัน (Common Object File Format, COFF) ซึ่งจะอยู่ในรูปไฟล์เอาต์พุต (.out) จึงสามารถนำไปใช้โหลดลงใน TMS320C6713 DSP ได้โดยตรง เพื่อทำการตรวจสอบและแก้ไขจุดบกพร่องของโปรแกรม โดยมีลักษณะเด่นคือ สามารถกำหนดจุดหยุดเพื่อตรวจสอบ (Breakpoint) ในขณะที่โปรแกรมกำลังทำงานได้เพื่ออ่านค่าตัวแปร (Variable) หน่วยความจำ (Memory) และรีจิสเตอร์ (Register) ในขณะนั้น นอกจากนั้นแล้วยังสามารถตรวจสอบโปรแกรมโดยให้ทำงานทีละบรรทัด (Step Over) เข้าไปยังฟังก์ชัน (Step Into) หรือออกจากฟังก์ชัน (Step Out) ได้ และที่สำคัญคือ สามารถแสดงผลโดยการวาดกราฟเพื่อตรวจสอบการทำงานและเวลาที่ถูกใช้ (Execution Time) ในขณะเวลาจริงได้อีกด้วย

การประมวลผลในเวลาจริงนั้นทำงานโดยการแลกเปลี่ยนข้อมูลเวลาจริง (Real-Time Data Exchange, RTDX) ซึ่งเชื่อมโยงกับ DSP/BIOS โดย RTDX จะมีหน้าที่ดูแลการแลกเปลี่ยนข้อมูลกันระหว่างคอมพิวเตอร์แม่ (Host) และบอร์ดประมวลผล (Target) เพื่อดึงข้อมูลจากตัวประมวลผลออกมาแสดงผลยังเครื่องคอมพิวเตอร์ โดยไม่ให้มีผลกระทบต่อการทำงานใดๆของการประมวลผลในเวลาจริง ด้วยสาเหตุนี้เองจึงทำให้สามารถตรวจสอบการทำงานของบอร์ดประมวลผลได้อย่างถูกต้องและทันทีในเวลาจริง การสื่อสารที่เกิดขึ้นนี้ตั้งอยู่บนมาตรฐานของ Joint Test Action Group (JTAG) โดยเชื่อมผ่านทางตัวจำลองการทำงานจริง (JTAG Emulator) องค์ประกอบที่กล่าวมาทั้งหมดนี้ทำงานร่วมกันได้ดังแสดงในภาพที่ 75



<u>ภาพที่ 75</u> องค์ประกอบการทำงานของโปรแกรม Code Composer Studio (CCS)
ที่มา : Texas Instruments. (2002)

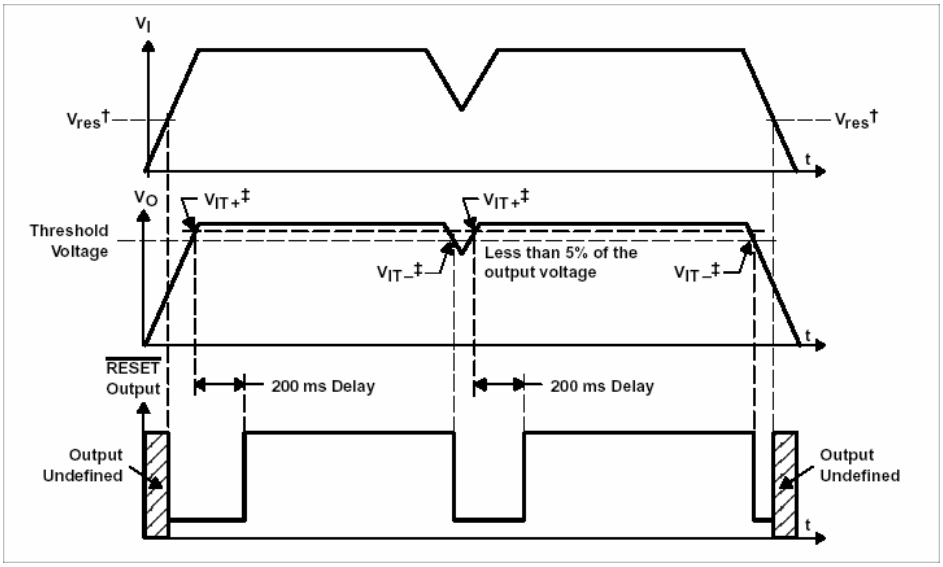
โดยทั่วไปแล้วการทดสอบจะเริ่มจาก การประกอบวงจร การทดสอบจุดเชื่อมต่อ จากนั้นจึงทำการทดสอบด้วยโปรแกรมทดสอบ โดยเริ่มจากวงจรเล็ก ๆ ขยายไปสู่วงจรที่ใหญ่ขึ้นจนครบทั้งแผงวงจร การทดสอบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล มีขั้นตอนตามภาพที่ 76 ดังต่อไปนี้



ภาพที่ 76 ขั้นตอนการทดสอบฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

1. Power Supply

วงจรรวมแปลงแรงดัน (Voltage Regulator) เบอร์ TPS70302PWP มีหน้าที่แปลงจากแรงดัน 5 โวลต์ เป็นแรงดัน 3.3 โวลต์ เพื่อใช้เป็นแหล่งจ่ายไฟให้อุปกรณ์ต่าง ๆ ภายในบอร์ดและแรงดัน 1.26 โวลต์ เพื่อใช้เป็นแหล่งจ่ายแรงดันไฟฟ้าหลักของ DSP การทดสอบเริ่มด้วยการป้อนไฟผ่านให้กับวงจรรวมแปลงแรงดัน แล้ววัดสัญญาณที่ขาแรงดันขาออก (Voltage Output) และขารีเซต (Reset) สัญญาณที่ได้จะเป็นดังภาพที่ 77 จากนั้นจึงทำการวัดว่าแรงดันขาเข้า (Voltage Input) และสัญญาณรีเซต ว่าเข้าสู่อุปกรณ์ต่าง ๆ บนแผงวงจรถูกต้องตามที่ออกแบบไว้หรือไม่

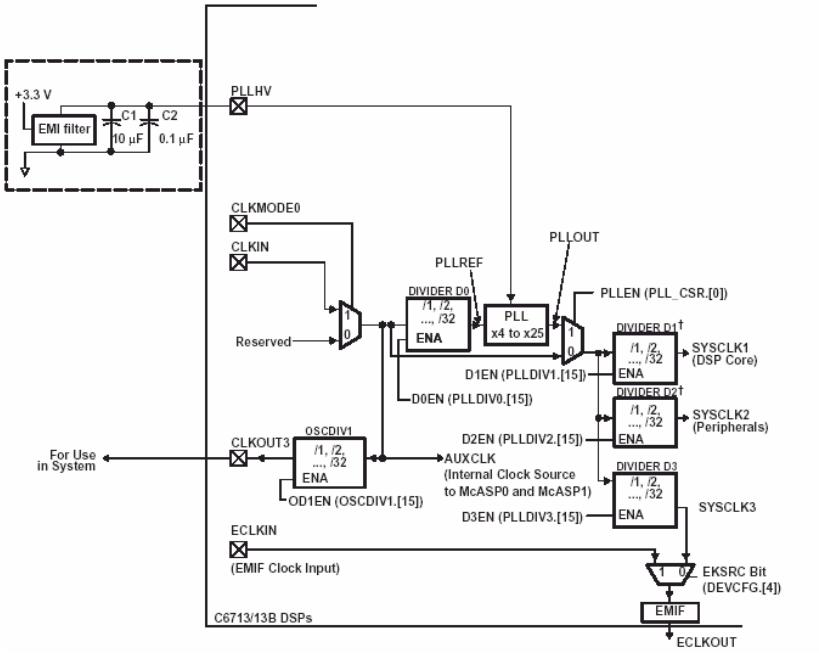


ภาพที่ 77 แผนภาพเวลาในการรีเซตของ TPS70302

ที่มา : Texas Instruments. (2002)

2. PLL &Clock

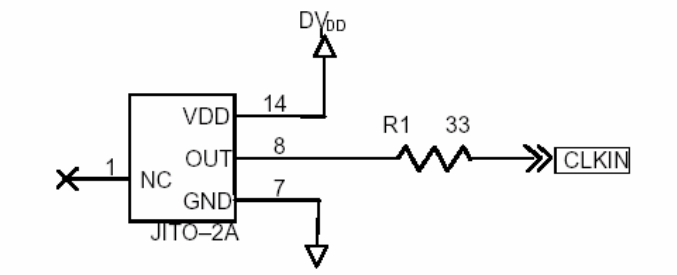
เฟสล็อกลูป (Phase Lock Loop, PLL) มีหน้าที่ในการสร้างสัญญาณนาฬิกาเพื่อนำไปใช้สำหรับมอดูลต่าง ๆ เช่น DSP Core, EMIF, McBSP, McASP, I2C และ Timer เป็นต้น ทำงานโดยการกำหนดค่าควบคุมให้กับรีจิสเตอร์ PLL เพื่อเพิ่มหรือลดค่าความถี่ของสัญญาณนาฬิกาอินพุตและนำสัญญาณนาฬิกาเอาต์พุตที่ได้ไปใช้งาน



ภาพที่ 78 โครงสร้างภายในของการสร้างสัญญาณนาฬิกาและ PLL

ที่มา : Texas Instruments. (2005)

- CLKIN เป็นสัญญาณอินพุตที่ได้รับมาจากวงจรออสซิลเลเตอร์ สามารถเชื่อมต่อกันได้โดยตรงดังภาพที่ 79



ภาพที่ 79 การต่อวงจรออสซิลเลเตอร์

ที่มา : Texas Instruments. (2001)

- CLKOUT3 เป็นสัญญาณเอาต์พุตที่ได้จากการหารค่าความถี่ CLKIN ด้วยตัวหาร OSCDIV 1 ซึ่ง CLKOUT3 จะถูกนำออกไปใช้เป็นสัญญาณนาฬิกาของอุปกรณ์ภายนอก

- SYSCLK1 เป็นสัญญาณนาฬิกาเอาต์พุตภายใน ที่ได้จากการหารค่าความถี่ PLLOUT ด้วยตัวหาร D1 ซึ่ง SYSCLK1 จะถูกนำไปใช้กับสัญญาณนาฬิกาหลักของ DSP

- SYSCLK2 เป็นสัญญาณนาฬิกาเอาต์พุตภายใน ที่ได้จากการหารค่าความถี่ PLLOUT ด้วยตัวหาร D2 ซึ่ง SYSCLK2 จะต้องมีความถี่เป็นครึ่งหนึ่งของ SYSCLK1 เสมอ ส่วนมอดูลที่ SYSCLK2 ถูกนำไปใช้ได้แก่ IC2, McASP, McBSP, HPI และ Timers

- SYSCLK3 เป็นสัญญาณนาฬิกาเอาต์พุตภายใน ที่ได้จากการหารค่าความถี่ PLLOUT ด้วยตัวหาร D3 ซึ่ง SYSCLK3 จะถูกนำไปใช้กับส่วนเชื่อมต่อ EMIF

ตารางที่ 17 รีจีสเตอร์ควบคุมของ PLL

Acronym	**Register Name**
PLLPID	PLL controller peripheral identification register
PLLCSR	PLL control/status register
PLLM	PLL multiplier control register
PLLDIV0-3	PLL controller divider registers
OSCDIV1	Oscillator divider 1 register

ที่มา : Texas Instruments. (2005)

หลังจากที่ตรวจสอบแหล่งกำลังจ่ายไฟแล้ว จากนั้นจึงเริ่มประกอบตัวประมวลผลสัญญาณดิจิทัล TMS3206713 กับออสซิลเลเตอร์ความถี่ 25 MHZ และตัวกรองเฟสล็อกลูป (PLL Filter) ลงบนแผงวงจร แล้วทำการตรวจสอบสัญญาณที่ขา CLKIN, CLKOUT2 และ CLKOUT3 เพื่อทำการทดสอบการทำงานในส่วนแหล่งกำเนิดสัญญาณนาฬิกา (Clock Generator) ของ DSP ในส่วนโปรแกรมสำหรับปรับค่าความถี่ที่ขา CLKOUT ทำได้โดยกำหนดค่าการทำงานให้กับตัวควบคุมเฟสล็อกลูป (PLL Controller) ซึ่งจะต้องโปรแกรมด้วยซอฟต์แวร์ทันทีหลังการเริ่มทำงานใหม่ (Reset) ของ DSP และควรจะอยู่ในส่วนเริ่มต้นของโปรแกรมก่อนการทำงานในส่วนติดต่ออุปกรณ์ภายนอกอื่น ๆ การกำหนดค่าการทำงานให้กับตัวควบคุมเฟสล็อกลูปสามารถทำได้ตามขั้นตอนดังนี้

1) กำหนดค่า PLLEN = 0 ในรีจิสเตอร์ PLLCSR เพื่อให้ตัวควบคุมเฟสล็อกลูปอยู่ในโหมดต่อตรง

2) รออย่างน้อย 4 รอบนาฬิกา (Clock Cycles) เพื่อรอสัญญาณ PLLOUT

3) กำหนดค่า PLLRST = 1 ในรีจิสเตอร์ PLLCSR เพื่อให้ตัวควบคุมเฟสล็อกลูปหยุดการทำงานชั่วคราว

4) กำหนดค่าให้กับรีจิสเตอร์ PLLDIV0 รีจิสเตอร์ PLLM และรีจิสเตอร์ OSCDIV1 เพื่อกำหนดค่าคูณและหารความถี่ให้กับสัญญาณ PLLOUT และ CLKOUT3

5) กำหนดค่าให้กับรีจิสเตอร์ PLLDIV1-3 เพื่อกำหนดค่าความถี่ให้กับสัญญาณ SYSCLK1-3 เพื่อกำหนดค่าความถี่ให้กับรีจิสเตอร์แต่ละตัว และรออย่างน้อย 8 รอบนาฬิกา

6) กำหนดค่า PLLRST = 0 ในรีจิสเตอร์ PLLCSR เพื่อกำหนดการเริ่มทำงานใหม่ให้ตัวควบคุมเฟสล็อกลูป

7) รอจนกว่า PLL จะเสถียร

8) กำหนดค่า PLLEN = 1 ในรีจิสเตอร์ PLLCSR เพื่อเริ่มใช้งาน PLL

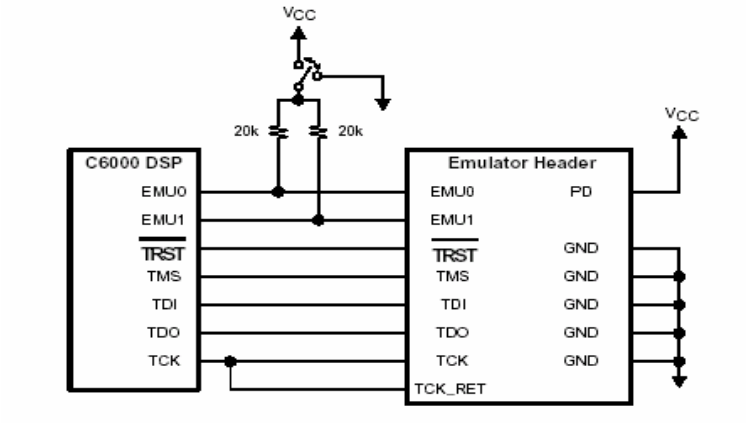
```
// Setup PLL to 200MHz frequency
void FVHPv2_setPLLto200MHz()
{
    // Put PLL in bypass mode before any setup
    PLL_bypass();
    plldelay(20);                                   // Wait at least 4 cycles

    // Reset the PLL
    PLL_reset();
    plldelay(20);

    // Set main clock multiplier/divisor ratio
    PLL_RSET(PLLDIV0, PLL_PLLDIV0_RMK(1, 0));       // 25MHz / 1  = 25MHz (must be between 12-100MHz)
    PLL_RSET(PLLM, 16);                             // 25MHz x 16 = 400MHz (must be between 140-600MHz)
    PLL_RSET(OSCDIV1, PLL_OSCDIV1_RMK(1, 9));       // 400MHz / 10 = 40Mhz
    plldelay(20);                                   // Wait at least 8 cycles

    // Set sysclk2 for peripheral
    PLL_RSET(PLLDIV2, PLL_PLLDIV2_RMK(1, 3));       // 400MHz / 4 = 100MHz (must be exactly half of DSP
    plldelay(20);                                   // Wait at least 8 cycles

    // Set sysclk1 for DSP core
    PLL_RSET(PLLDIV1, PLL_PLLDIV1_RMK(1, 1));       // 400MHz / 2 = 200MHz
    plldelay(20);                                   // Wait at least 8 cycles

    // Set sysclk3 for EMIF
    PLL_RSET(PLLDIV3, PLL_PLLDIV3_RMK(1, 3));       // 400MHz / 4 = 100MHz (must be less than 100MHz)
    plldelay(20);                                   // Wait at least 8 cycles

    // Bring PLL out of reset
    PLL_deassert();
    plldelay(1500);                                 // Wait for PLL to lock

    // Swich to PLL mode
    PLL_enable();
    plldelay(20);
}
```

ภาพที่ 80 โปรแกรมตัวอย่างของการควบคุม PLL

3. Joint Test Action Group (JTAG)

JTAG หมายถึง ข้อกำหนดที่ร่วมกันออกแบบขึ้นมาเพื่อใช้ในการทดสอบ (Testing) โปรแกรม (Programming) และตรวจสอบแก้ไขจุดบกพร่อง (Debugging) กับอุปกรณ์ต่าง ๆ การเชื่อมต่ออุปกรณ์เข้ากับ JTAG ทำให้สามารถดึงตัวอย่างสัญญาณและค่าในรีจิสเตอร์ภายในตัวประมวลผลผ่านทางตัวจำลองการทำงานจริง (JTAG Emulator) เพื่อแสดงผลบนหน้าจอคอมพิวเตอร์ได้ ตัวจำลองการทำงานจริงรุ่น XDS510 USB ของบริษัท Spectrum Digital มีคุณสมบัติคือสามารถเชื่อมต่อกับตัวประมวลผลสัญญาณดิจิทัลของบริษัท TI ได้ โดยเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคลผ่าน USB และเชื่อมต่อทางซอฟต์แวร์กับโปรแกรม Code Composer Studio (CCS) ของ TI ซึ่งสามารถใช้ได้กับตัวปฏิบัติการ Windows 98 SE 2000 และ XP ของบริษัท Microsoft

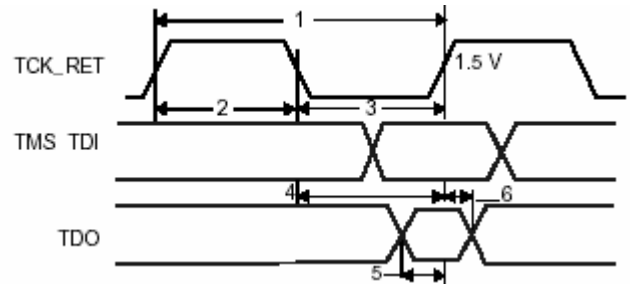
ตัวประมวลผลสัญญาณดิจิทัล TMS320C6713 นั้นรองรับมาตรฐานของ IEEE Standard 1149.1 (JTAG) Emulation โดยติดต่อสื่อสารผ่านขา TMS, TDO, TDI, TCK และ TRST จากจุดเชื่อมต่อ 14 ขา บนแผงวงจร ซึ่งมีไว้เชื่อมต่อกับตัวจำลองการทำงานจริง ดังภาพที่ 81



ภาพที่ 81 การเชื่อมต่อทางฮาร์ดแวร์ของตัวจำลองการทำงานจริง (JTAG Emulator)
ที่มา : Spectrum Digital. (2002)

การตรวจสอบการทำงานทำได้โดยสังเกตจากแผนภาพเวลาของการติดต่อสื่อสาร คือจะต้องมีสัญญาณนาฬิกาความถี่ 12 MHz เกิดขึ้นที่ขา TCK เมื่อมีการต่อตัวจำลองการทำงานจริงเข้ากับอุปกรณ์ และ เครื่องคอมพิวเตอร์ทาง USB หลังจากนั้นเมื่อมีการติดต่อสื่อสารกับโปรแกรม CCS สัญญาณที่เข้าสู่ขา TMS และ TDI จะเกิดตรงกับขอบขาขึ้นของสัญญาณ TCK และสัญญาณที่

ออกจากขา TDO จะเกิดหลังจากขอบขาลงของสัญญาณ TCK การทดสอบทำได้โดย เปิดโปรแกรม CCS แล้วทำการโหลดโปรแกรม (Code) ลงในหน่วยความจำของ DSP แล้วทำการอ่านค่าจากหน่วยความจำ และรีจิสเตอร์ ขณะทำการโปรแกรมทำงานได้

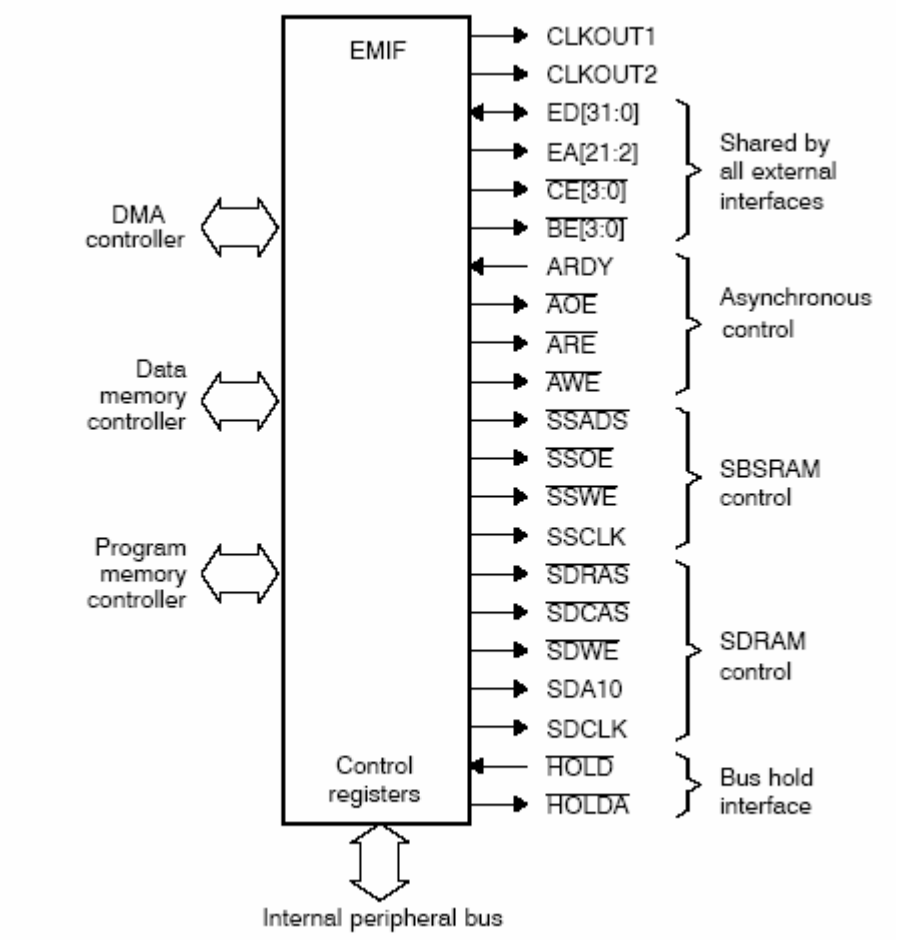


ภาพที่ 82 แผนภาพเวลาของตัวจำลองการทำงานจริง (JTAG Emulator)
ที่มา : Spectrum Digital. (2002)

4. External Memory

การติดต่อกับหน่วยความจำภายนอกแบบขนานของตัวประมวลผลสัญญาณดิจิทัลตระกูล TMS320C6x นั้น สามารถทำได้โดยใช้มอดูล EMIF ซึ่งลักษณะวิธีการสำหรับการติดต่อกับอุปกรณ์ต่าง ๆ นั้นมีอยู่หลายรูปแบบ เช่น การติดต่อแบบ ASRAM (Asynchronous Static RAM), SBSRAM (Synchronous Burst Static RAM) หรือ SDRAM (Synchronous Dynamic RAM) ขึ้นอยู่กับชนิดอุปกรณ์ที่นำมาเชื่อมต่อ ในภาพที่ 83 มอดูล EMIF ได้เตรียมขาสัญญาณต่าง ๆ ไว้เพื่อให้สามารถเข้าถึงอุปกรณ์ได้หลายประเภท ซึ่งได้แก่ ขาสัญญาณข้อมูล (Data Bus, ED[31:0]) ขาสัญญาณตำแหน่งที่อยู่ (Address Bus, EA[21:2], CE[3:0], BE[3:0]) และขาสัญญาณควบคุม (#ARE, #AOE, #AWE and ARDY) เป็นต้น

บนฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลถูกออกแบบให้มีหน่วยความความจำภายนอกอยู่ 2 ชนิด คือ หน่วยความจำแฟลชเบอร์ MBM26LV160BE ของบริษัท Spansion ซึ่งใช้ติดต่อแบบ ASRAM (Asynchronous Static RAM) และหน่วยความจำแบบ SDRAM (Synchronous Dynamic RAM) เบอร์ MT48LC4M32B2 ของบริษัท Micron ในการติดต่อกับหน่วยความจำทั้งสองชนิดนี้ โดยใช้มอดูล EMIF ของตัวประมวลผลสัญญาณดิจิทัล มีความจำเป็นต้องรู้วิธีในการปรับแต่งคุณลักษณะของมอดูล EMIF ให้เหมาะสมกับชนิดของหน่วยความจำเหล่านั้น ดังต่อไปนี้



ภาพที่ 83 แผนภาพของมอดูล EMIF

ที่มา : Texas Instruments. (2003)

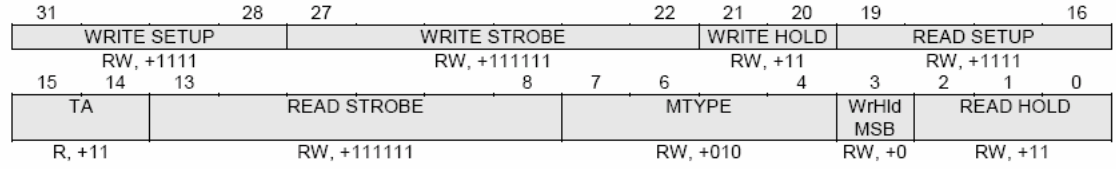
ในการปรับแต่งคุณลักษณะของมอดูล EMIF นั้นตัวประมวลผลสัญญาณดิจิทัลได้เตรียมรีจิสเตอร์ควบคุม (Control Register) ต่าง ๆ ไว้ ซึ่งใช้ในการปรับแต่งความกว้างของสัญญาณที่ออกมา โดยที่ตำแหน่งของรีจิสเตอร์สำหรับติดต่อหน่วยความจำภายนอก นั้น จะแสดงได้ดังตารางที่ 18

จากตารางจะเห็นว่าพื้นที่ของหน่วยความจำสำหรับมอดูล EMIF จะแบ่งเป็น 4 ส่วน คือ ส่วนของ CE0 CE1 CE2 และ CE3 เพื่อเพิ่มความยืดหยุ่นของตัวประมวลผลในกรณีที่มีการต่อกับอุปกรณ์หลายประเภทที่ใช้การเข้าถึงข้อมูลที่แตกต่างกัน สำหรับรีจิสเตอร์ควบคุมครอบคลุม EMIF (Global Control Register) ถูกใช้ในการการควบคุมสัญญาณนาฬิกาและขาสัญญาณอื่น ๆ ที่เหมือนกันโดยทั่วไปของหน่วยความจำทั้ง 4 ส่วน

ตารางที่ 18 รีจิสเตอร์ EMIF ที่ตำแหน่งต่าง ๆ ของหน่วยความจำ

Byte Address	**Name**
0x0180 0000	EMIF global control
0x0180 0004	EMIF CE1 space control
0x0180 0008	EMIF CE0 space control
0x0180 000C	Reserved
0x0180 0010	EMIF CE2 space control
0x0180 0014	EMIF CE3 space control

สำหรับรีจิสเตอร์ CEx โดยที่ x มีค่าเป็น 0 1 2 หรือ 3 นั้นสามารถแสดงได้ดังภาพที่ 84



ภาพที่ 84 รีจิสเตอร์ CEx Space Control สำหรับมอดูล EMIF

ที่มา : Texas Instruments. (2003)

จากภาพที่ 84 ภายในรีจิสเตอร์จะประกอบด้วยเขตข้อมูลควบคุม (Control Field) ต่าง ๆ ที่ใช้กำหนดค่าสำหรับควบคุมคุณลักษณะของมอดูล EMIF ซึ่งเขตข้อมูลควบคุมของรีจิสเตอร์ควบคุมพื้นที่ CEx ประกอบด้วย MTYPE ใช้สำหรับเลือกวิธีการเข้าถึงข้อมูลกับอุปกรณ์แบบต่าง ๆ TA ใช้กำหนดจำนวนสัญญาณนาฬิการะหว่างการอ่านและเขียนข้อมูล หรือระหว่างการอ่านข้อมูล ส่วนเขตข้อมูลควบคุมอื่น ๆ เป็นการกำหนดความกว้างของรูปแบบสัญญาณสำหรับอ่านหรือเขียนข้อมูล โดยกำหนดเทียบกับจำนวนของสัญญาณนาฬิกาของมอดูล EMIF

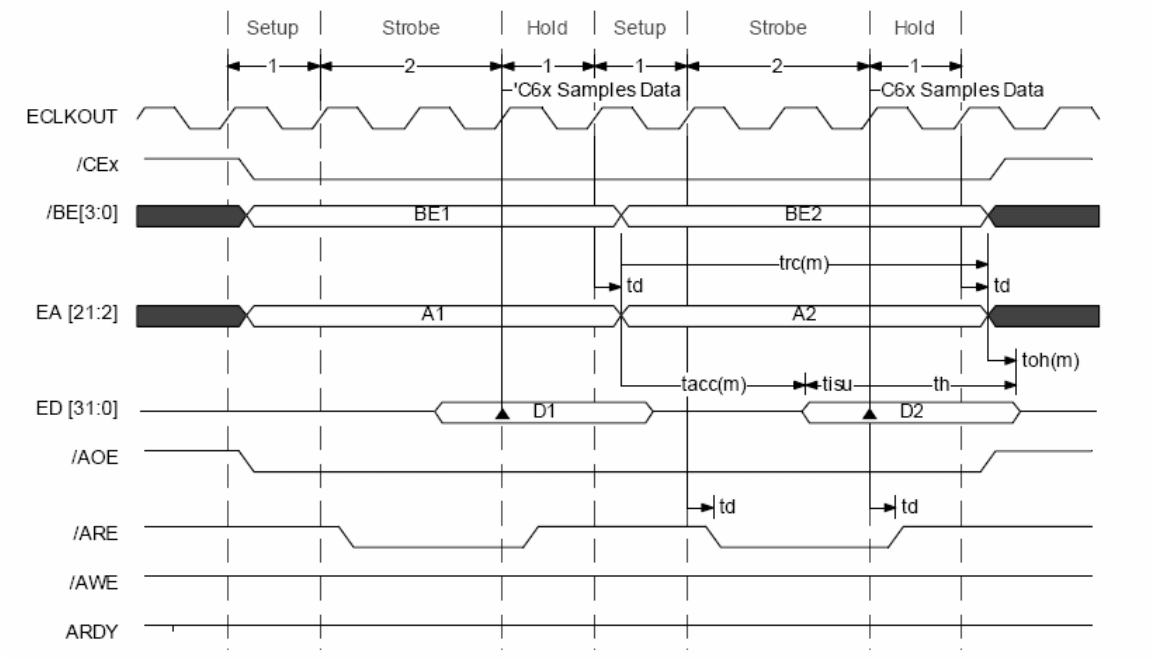
การโปรแกรมค่าพารามิเตอร์สำหรับการเข้าถึงข้อมูลแบบ ASRAM นั้นสามารถทำได้โดยการกำหนดค่า (Configuration) ลงในรีจิสเตอร์ควบคุมในส่วนของมอดูล EMIF ดังที่กล่าวมาแล้วในข้างต้น ซึ่งค่าพารามิเตอร์เหล่านั้นสามารถแบ่งได้เป็น 2 กลุ่ม คือ กลุ่มสำหรับการอ่านและเขียนข้อมูลที่มีพารามิเตอร์เหมือนกัน โดยที่พารามิเตอร์สำหรับการอ่านและเขียนข้อมูลแบบ ASRAM ได้แก่

- Setup เป็นเวลาในช่วงแรกของการเข้าถึงข้อมูลจนถึงสัญญาณอ่านหรือเขียนเริ่มทำงาน

- Strobe เป็นเวลาที่สัญญาณอ่านหรือเขียนเริ่มทำงานจนถึงสิ้นสุด

- Hold เป็นเวลาหลังจากสัญญาณอ่านหรือเขียนทำงานเสร็จแล้วจนกระทั่งสิ้นสุดการเข้าถึงข้อมูล

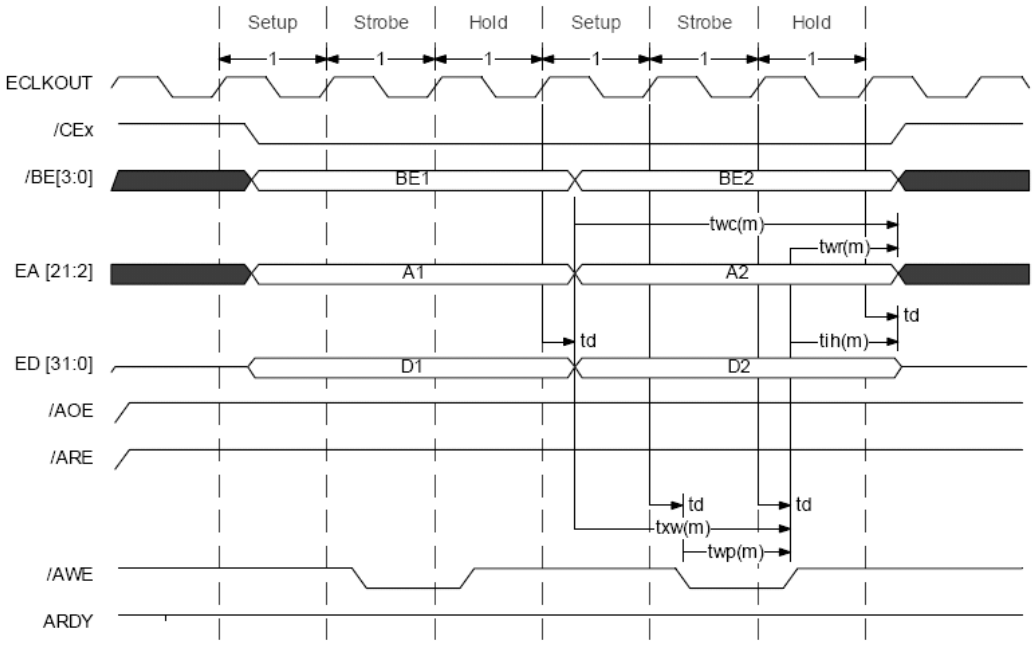
สำหรับตัวประมวลผลสัญญาณดิจิทัลตระกูล TMS320C67x นั้นค่าพารามิเตอร์ที่กำหนดลงไปจะอยู่ในรูปจำนวนเท่าของสัญญาณนาฬิกา (ECLKOUT Cycles) โดยที่พารามิเตอร์ Setup Strobe และ Hold จะใช้ในการเข้าถึงข้อมูล นอกจากนี้ยังมีพารามิเตอร์ TA (Turnaround Cycles) ซึ่งเป็นเวลาระหว่างการอ่านและการเขียนข้อมูล หรือระหว่างการอ่านข้อมูล เพื่อเพิ่มความหน่วงของเข้าถึงข้อมูลได้ โดยที่รูปแบบการอ่านและเขียนข้อมูลของมอดูล EMIF สามารถเขียนแผนภาพเวลาได้ดังภาพที่ 85 และ 86



<u>ภาพที่ 85</u> แผนภาพเวลาในการอ่านข้อมูลแบบไม่เข้าจังหวะบน TMS320C6713 DSP
ที่มา : Texas Instruments. (2003)

จากภาพที่ 85 เป็นตัวอย่างการอ่านข้อมูลของตัวประมวลผลโดยมีการกำหนดค่าของ Setup/Strobe/Hold เป็น 1/2/1 โดยที่ในช่วงแรก (Setup Time) สัญญาณ CE AOE BE และ EA เริ่มมีการทำงาน ต่อมาในช่วงที่สอง (Strobe Time) สัญญาณ ARE จะทำงานจนกระทั่งสิ้นสุดช่วง

ที่ 2 ที่จุดเริ่มต้นในช่วงสุดท้าย (Hold Time) จะมีการอ่านข้อมูล และที่จุดปลายของช่วงสุดท้าย สัญญาณต่าง ๆ ก็จะสิ้นสุดการทำงาน



ภาพที่ 86 แผนภาพเวลาในการเขียนข้อมูลแบบไม่เข้าจังหวะบน TMS320C6713 DSP
ที่มา : Texas Instruments. (2003)

จากภาพที่ 86 เป็นตัวอย่างการเขียนข้อมูลของตัวประมวลผลสัญญาณดิจิทัล โดยมีการกำหนดค่าของ Setup/Strobe/Hold เป็น 1/1/1 โดยที่ในช่วงแรก (Setup Time) สัญญาณ CE BE EA และ ED เริ่มมีการทำงาน ต่อมาในช่วงที่สอง (Strobe Time) สัญญาณ AWE จะทำงาน จนกระทั่งสิ้นสุดช่วงที่ 2 ที่จุดเริ่มต้นในช่วงสุดท้าย (Hold Time) จะมีการอ่านข้อมูล และที่จุดปลายของช่วงสุดท้ายสัญญาณต่าง ๆ ก็จะสิ้นสุดการทำงาน

จากที่กล่าวมาในข้างต้นนั้นเป็นการคำนวณลักษณะของสัญญาณสำหรับอ่านและเขียนข้อมูลของมอดูล EMIF บนตัวประมวลผลสัญญาณดิจิทัลผ่านรีจิสเตอร์ควบคุมในการเข้าถึงข้อมูลแบบ ASRAM เท่านั้น สำหรับการเข้าถึงในแบบ SDRAM นอกจากการปรับแต่งค่าของรีจิสเตอร์ควบคุมพื้นที่ CEx ยังต้องใช้รีจิสเตอร์ควบคุมพิเศษ (SDRAM Control Register) ในการปรับแต่งค่าพารามิเตอร์ต่างๆที่ใช้ในการติดต่อแบบ SDRAM โดยที่ตำแหน่งของรีจิสเตอร์พิเศษสำหรับติดต่อหน่วยความจำภายนอกนั้น จะแสดงได้ดังตารางที่ 19

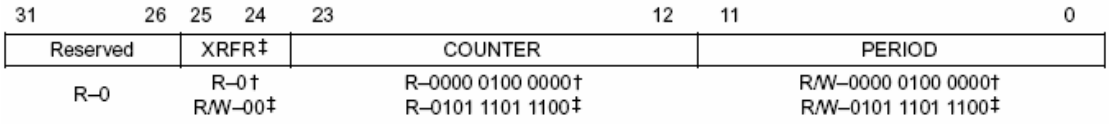
ตารางที่ 19 รีจิสเตอร์ EMIF พิเศษของหน่วยความจำ SDRAM

Byte Address	**Name**
0x0180 0018	EMIF SDRAM control
0x0180 001C	EMIF SDRAM refresh control
0x0180 0020	EMIF SDRAM extension

สำหรับรีจิสเตอร์ควบคุมพิเศษในการปรับแต่งค่าพารามิเตอร์ต่างๆที่ใช้ในการติดต่อแบบ SDRAM นั้นสามารถแสดงได้ดังภาพที่ 87, 88 และ 89

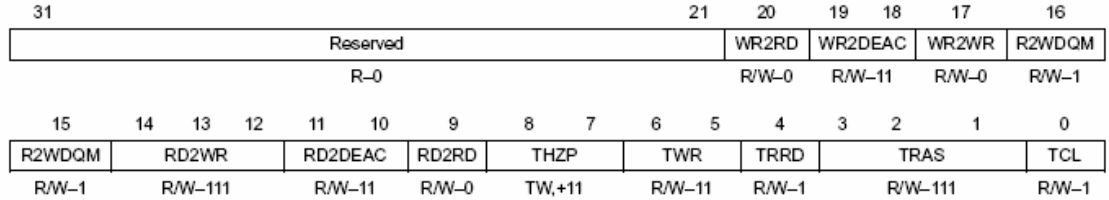
ภาพที่ 87 รีจิสเตอร์ SDRAM Control สำหรับมอดูล EMIF

ที่มา : Texas Instruments. (2004)



ภาพที่ 88 รีจิสเตอร์ SDRAM Timing สำหรับมอดูล EMIF

ที่มา : Texas Instruments. (2004)



ภาพที่ 89 รีจิสเตอร์ SDRAM Extension สำหรับมอดูล EMIF

ที่มา : Texas Instruments. (2004)

รีจิสเตอร์ที่เพิ่มเติมขึ้นมาเพื่อใช้ในการกำหนดค่าพารามิเตอร์ของ SDRAM โดยเฉพาะมีดังนี้

- SDRAM Control Register (SDCTL) ใช้กำหนดค่าตัวแปรต่าง ๆ ทั้งหมดที่ใช้สำหรับควบคุมคุณลักษณะของ SDRAM โดยเฉพาะ เช่น กำหนดค่าเริ่มต้นการทำงาน กำหนดให้โมดูลรีเฟรช (Refresh) ทำงาน (โดยการรีเฟรช คือ การฟื้นฟูข้อมูลให้ยังคงอยู่ซึ่งเป็นลักษณะเฉพาะของ SDRAM) กำหนดจำนวนแถวและจำนวนคอลัมน์ในการเชื่อมต่อทางฮาร์ดแวร์ กำหนดจำนวนจำนวนกลุ่มของหน่วยความจำ เป็นต้น

- SDRAM Timer Register (SDTIM) ใช้กำหนดค่าคาบการฟื้นฟู (Refresh Period) ในการอัดประจุไฟฟ้า

- SDRAM Extension Register (SDEXT) ใช้กำหนดค่าพารามิเตอร์เวลา(Timing Parameters) ที่ถูกใช้ในการเชื่อมต่อ ซึ่งต้องพิจารณาจากสัญญาณที่ออกมาจริง และมีผลมาจากค่าความหน่วง และค่าเฉพาะ (Specification) ของหน่วยความจำที่ใช้

ค่าพารามิเตอร์สำหรับการเข้าถึงข้อมูลแบบ SDRAM ที่สามารถกำหนดค่าลงในรีจิสเตอร์ควบคุมในส่วนของการติดต่อ SDRAM ได้แสดงไว้ในตารางที่ 20 และ 21 แต่สำหรับการเลือกวิธีการเข้าถึงข้อมูลกับอุปกรณ์ ยังคงกำหนดค่าในพารามิเตอร์ MTYPE ของรีจิสเตอร์ควบคุมพื้นที่ CEx เช่นเดิม โดยที่รูปแบบการอ่านและเขียนข้อมูลของมอดูล SDRAM สามารถเขียนแผนภาพเวลาได้ดังภาพที่ 90 และ 91

ตารางที่ 20 พารามิเตอร์ต่างๆ ของหน่วยความจำ SDRAM (1)

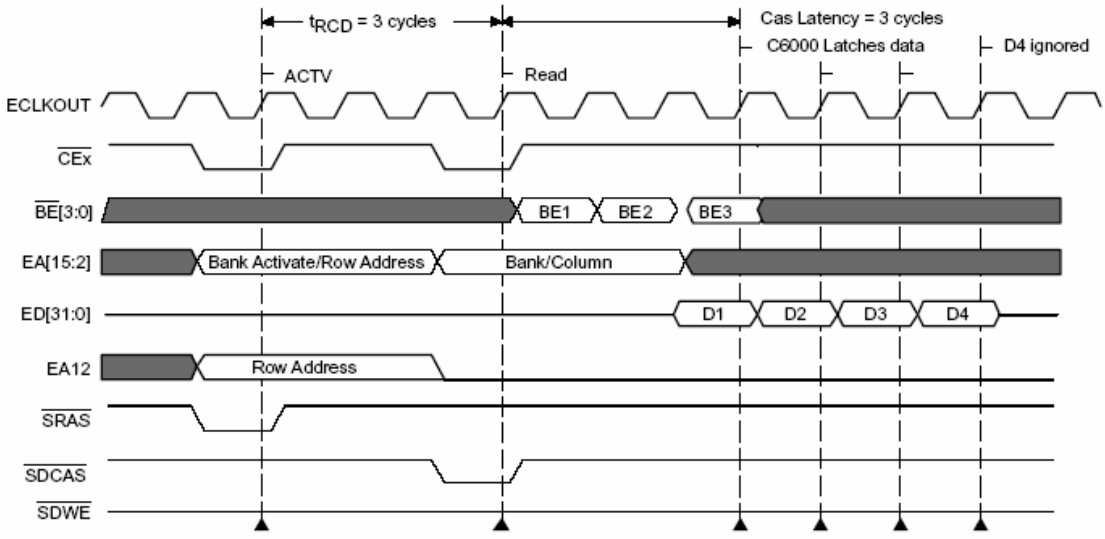
Parameter	Description	Value in EMIF clock cycles†
t_{RC}	REFR command to ACTV, MRS, or subsequent REFR command	TRC + 1
t_{RCD}	ACTV command to READ or WRT command	TRCD + 1
t_{RP}	DCAB/DEAC command to ACTV, MRS, or REFR command	TRP + 1
t_{CL}	CAS latency of the SDRAM	TCL + 2
t_{RAS}	ACTV command to DEAC/DCAB command	TRAS + 1
t_{RRD}	ACTV bank A to ACTV bank B (same CE space)	TRRD + 2
t_{WR}	Write recovery, time from last data out of C6000 DSP (write data) to DEAC/DCAB command	TWR + 1
t_{HZP}	High Z from precharge, time from DEAC/DCAB to SDRAM outputs (read data) in high Z	THZP + 1

ที่มา : Texas Instruments. (2004)

ตารางที่ 21 พารามิเตอร์ต่างๆ ของหน่วยความจำ SDRAM (2)

Parameter	Description	Value in EMIF clock cycles†	Suggested value for TCL = 0	Suggested value for TCL = 1
READ to READ	READ command to READ command. Used to interrupt a READ burst for random READ addresses.	RD2RD + 1	RD2RD = 0	RD2RD = 0
READ to DEAC	Used in conjunction with t_{HZP}. Specifies the minimum amount of time between READ command and DEAC/DCAB command.	RD2DEAC + 1	RD2DEAC = 1	RD2DEAC = 1
READ to WRITE	READ to WRITE command. The value programmed in this parameter depends on t_{CL}. READ to WRITE should be CAS latency plus 2 cycles (in EMIF clock cycles) to provide 1 turnaround cycle before WRITE command.	RD2WR + 1	RD2WR = 3	RD2WR = 4
BE*n* high before write interrupting read	Specifies the number of cycles that the BE*n* outputs should be high before a write is allowed to interrupt a read. This is related to READ to WRITE parameter.	R2WDQM + 1	R2WDQM = 1	R2WDQM = 2
WRITE to WRITE	Number of cycles between a WRITE interrupting a WRITE. Used for random WRITES.	WR2WR + 1	WR2WR = 0	WR2WR = 0
WRITE to DEAC	Number of cycles between a WRITE command and a DEAC/DCAB command.	WR2DEAC + 1	WR2DEAC = 1	WR2DEAC = 1
WRITE to READ	Number of cycles between a WRITE command and a READ command.	WR2RD + 1	WR2RD = 0	WR2RD = 0

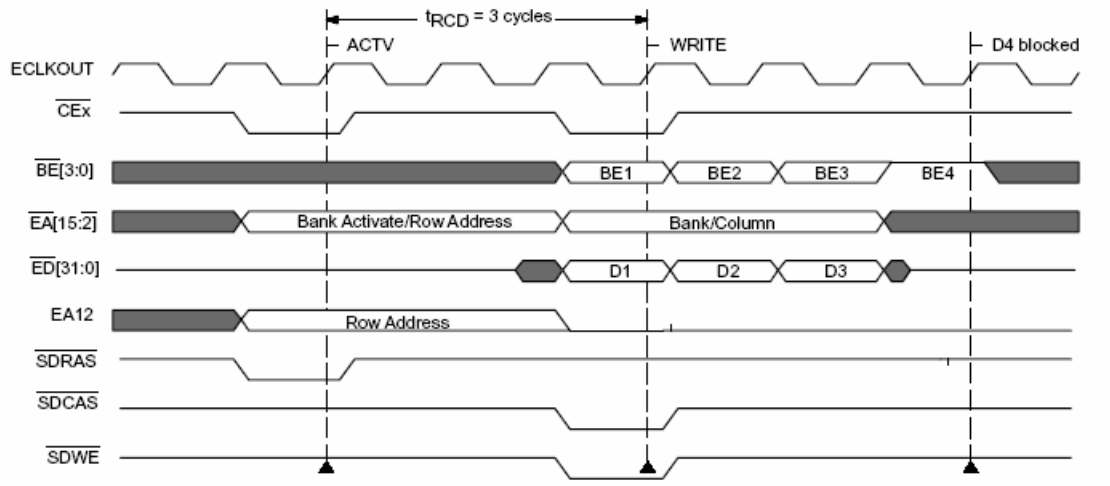
ที่มา : Texas Instruments. (2004)



ภาพที่ 90 แผนภาพเวลาในการอ่านข้อมูลแบบเข้าจังหวะบน TMS320C6713 DSP

ที่มา : Texas Instruments. (2003)

จากภาพที่ 90 เป็นตัวอย่างการอ่านข้อมูลแบบสามคำ (Three-word) ของตัวประมวลผลโดยมีการกำหนดค่าการหน่วงเวลา (CAS Latency) เป็น 3 และ t_{RCD} เป็น 3 โดยที่ในช่วงแรกสัญญาณ CE SDRAS และ EA เริ่มมีการทำงาน โดย EA จะแสดงตำแหน่งที่อยู่ของแถวที่ต้องการอ่าน ต่อมาอีก 3 ช่วงของสัญญาณนาฬิกา สัญญาณ CE, SDCAS, BE, และ EA จะทำงานโดย EA จะแสดงตำแหน่งที่อยู่ของคอลัมน์ที่ต้องการอ่าน พร้อมกับสัญญาณ BE1-BE3 แสดงถึงการอ่านข้อมูลในแต่ละช่วง และอีก 3 ช่วงของสัญญาณนาฬิกาต่อมา สัญญาณเอาต์พุต D1-D3 จะออกมาตามลำดับ ซึ่งจะมีสัญญาณ D4 ออกมาด้วยแต่ไม่ต้องสนใจ เนื่องจากค่ามาตรฐานกำหนดไว้ให้อ่านข้อมูลได้ครั้งละสี่คำ (4-word) เป็นอันสิ้นสุดการทำงาน



ภาพที่ 91 แผนภาพเวลาในการเขียนข้อมูลแบบเข้าจังหวะบน TMS320C6713 DSP
ที่มา : Texas Instruments. (2003)

จากภาพที่ 91 เป็นตัวอย่างการเขียนข้อมูลแบบสามคำ (Three-word) ของตัวประมวลผลโดยมีการกำหนดค่าของ t_{RCD} เป็น 3 แต่ไม่ต้องมีค่าการหน่วงเวลา (Latency) โดยที่ในช่วงแรกสัญญาณ CE SDRAS และ EA เริ่มมีการทำงาน โดย EA จะแสดงตำแหน่งที่อยู่ของแถวที่ต้องการเขียน ต่อมาอีก 3 ช่วงของสัญญาณนาฬิกา สัญญาณ CE, SDCAS, BE, ED และ EA จะการทำงาน โดย EA จะแสดงตำแหน่งที่อยู่ของคอลัมน์ที่ต้องการเขียน พร้อมกับสัญญาณ BE1-BE3 และ ED1-ED3 ซึ่งต้องออกมาพร้อมกันในแต่ละช่วงสัญญาณนาฬิกา เพื่อทำการเขียนข้อมูลในแต่ละช่วงข้อมูล และที่จุดปลายของช่วงสุดท้ายสัญญาณต่าง ๆ ก็จะสิ้นสุดการทำงาน

สำหรับการเขียนโปรแกรมเพื่ออ่านหรือเขียนลงใน SDRAM นั้นสามารถอ้างไปยังตำแหน่งที่อยู่ตรง ๆ ได้เลย ยกตัวอย่างเช่น

*((Uint8 *) 0x80000000) = 0x09 ; เป็นการเขียนข้อมูล (0x09) ซึ่งเป็นเลขฐาน 16 ลงในตำแหน่งที่อยู่ (0x80000000)
Temp = *((Uint8 *) 0x80001000) ; เป็นการอ่านข้อมูล ที่ตำแหน่งที่อยู่ (0x80001000) มาเก็บไว้ในตัวแปร Temp

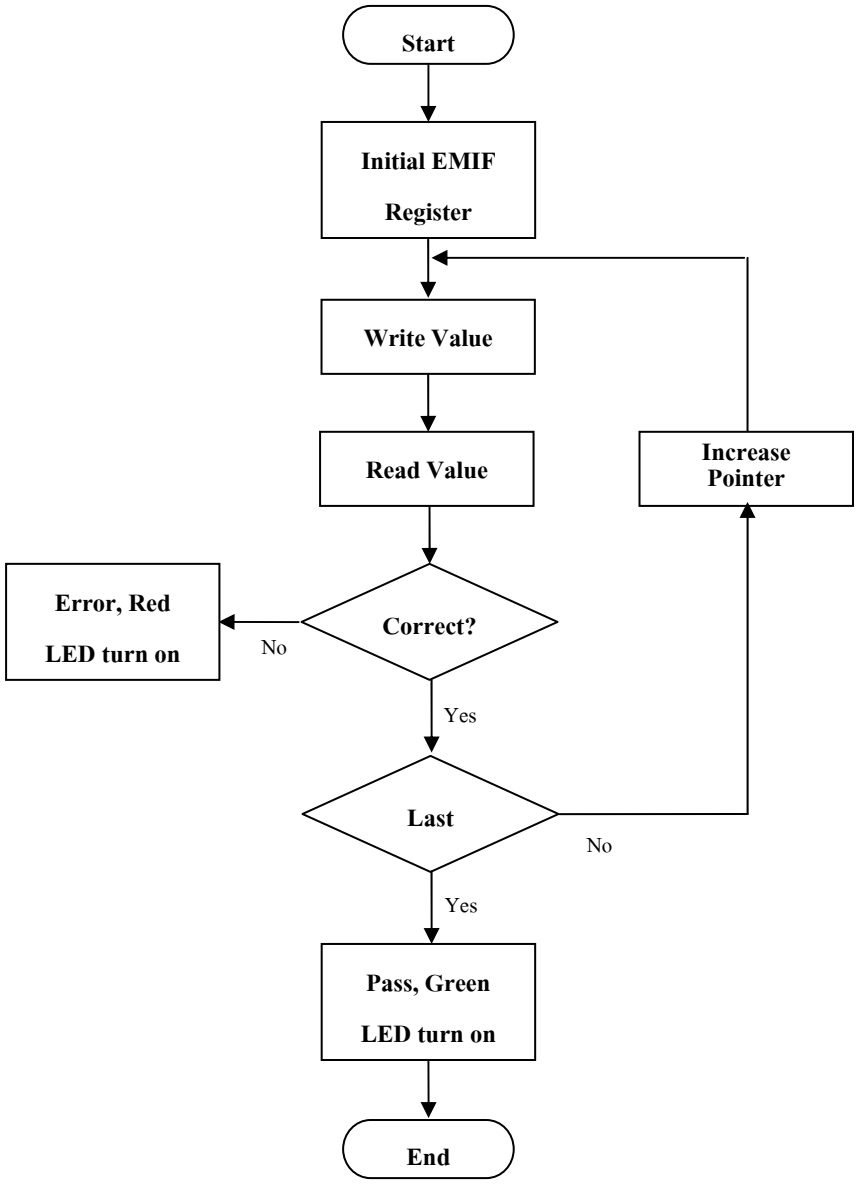
สังเกตได้ว่าเป็นการใช้ลักษณะการอ้างข้อมูลแบบตัวชี้ (Pointer) ซึ่ง (Uint8 *) จะเป็นตัวกำหนดว่าเป็นการอ้างถึงข้อมูลแบบ 8 บิต และนอกจากการอ้างตำแหน่งที่อยู่โดยตรงแล้ว การอ้างผ่านตัวแปรแบบตัวชี้เป็นอีกวิธีที่สามารถใช้ได้เช่นกัน ดังตัวอย่างต่อไปนี้

```
Uint8  sram_ptr = (Uint8 *) 0x80000000 , sdram_ptr = (Uint8 *) 0x80001000 ;
Uint32  i ;
        for  ( i = 0; i < 0x1000; i++ )
            * sdram_ptr++  =  * sram_ptr++ ;
```

เป็นการคัดลอกข้อมูลขนาด 4K ไบต์ จาก SDRAM ที่ตำแหน่งที่อยู่ (0x80000000) ไปเก็บไว้ที่ SDRAM ที่ตำแหน่งที่อยู่ (0x80002000) โดยอ่านและเขียนค่าผ่านตัวแปร 2 ตัว ซึ่งถูกใช้การอ้างข้อมูลแบบ 8 บิต

การทดสอบการทำงานของ SDRAM ทำได้โดยเขียนข้อมูลลงใน SDRAM แล้วทำการอ่านข้อมูลและตรวจสอบความถูกต้อง ส่วนทางด้านกายภาพวัดได้จากแผนภาพเวลาที่ขา CE, AOE, ARE, Address และ Data ว่าถูกต้องตามที่สั่งไปหรือไม่ ภาพที่ 92 แสดงผังงานของการเขียนโปรแกรมตรวจสอบการทำงานของ SRAM โดยโปรแกรมเริ่มจากการกำหนดค่าเริ่มต้นที่ใช้ในการควบคุมให้กับตัวประมวลผล เพื่อทำการติดต่อกับหน่วยความจำภายนอก จากนั้นเขียนค่าตัวแปร (Value) ลงไปโดยเริ่มจากตำแหน่งที่อยู่เริ่มต้น (Start Address) และทำการอ่านค่าของหน่วยความจำที่ตำแหน่งที่อยู่นั้นอีกครั้งเพื่อทำการตรวจสอบความถูกต้อง และทำการทดสอบโดยเพิ่มค่าตำแหน่งขึ้นเรื่อย ๆ จนสิ้นสุดพื้นที่ของ SDRAM ทั้งหมด จากนั้นทำกระบวนการข้างต้นอีกครั้งโดยเปลี่ยนค่าตัวแปร (Value) ต่าง ๆ กัน เช่น 0x0000, 0x55555, 0xAAAA และ 0xFFFF ในรอบสุดท้ายจะทำ

การตรวจสอบตำแหน่งที่อยู่ โดยให้ค่าตัวแปรเป็นค่าเดียวกับค่าตำแหน่งที่อยู่ที่เขียนลงไป เมื่อสิ้นสุดกระบวนการ หากไม่มีข้อผิดพลาดใด ๆ LED สีเขียวจะสว่างขึ้น แต่หากมีข้อผิดพลาดเกิดขึ้น LED สีแดงจะสว่างขึ้น



ภาพที่ 92 ขั้นตอนการทดสอบหน่วยความจำ SDRAM

5. <u>Flash ROM</u>

หน่วยความจำ ROM (Read-Only Memory) หมายถึง หน่วยความจำอ่านอย่างเดียว มีคุณสมบัติคือ แม้ว่าไม่มีการจ่ายแรงดันไฟฟ้าให้กับหน่วยความจำ ข้อมูลที่ถูกจัดเก็บไว้ในหน่วยความจำนี้ก็ยังคงไม่สูญหาย ดังนั้นหน่วยความจำอ่านอย่างเดียวจึงถูกนำไปใช้สำหรับเก็บโปรแกรมคำสั่ง (Source Code) และข้อมูลที่ไม่ต้องการให้สูญหาย ในขณะเริ่มต้นการทำงานตัวประมวลผล DSP จะคัดลอกคำสั่งและข้อมูลที่อยู่ในหน่วยความจำอ่านอย่างเดียว และนำมาเก็บไว้ในหน่วยความจำเข้าถึงแบบสุ่มหรือหน่วยความจำ RAM (Random-Access Memory) ก่อนจากนั้นจึงไปอ่านคำสั่ง (Fetch) จากหน่วยความจำหน่วยความจำเข้าถึงแบบสุ่มเพื่อทำการประมวลผลต่อไป บนฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลได้เลือกใช้หน่วยความจำอ่านอย่างเดียวเป็นแบบแฟลช (Flash ROM) เบอร์ MBM29LV160BE ของ Fujitsu ซึ่งสามารถลบและโปรแกรมซ้ำได้โดยใช้มาตรฐานของ EPROM (Erasable Programmable Read-Only Memory) โดยมีหลักการทำงานดังนี้

การใช้งานหน่วยความจำแฟลชนั้นต้องเขียนเป็นลำดับคำสั่ง (Command Sequence) สั่งผ่านไปยังรีจิสเตอร์คำสั่ง (Command Register) โดยฟังก์ชันหลัก ๆ ของการใช้งานหน่วยความจำแฟลชมีดังนี้

1) Read Command

เป็นการอ่านข้อมูลจากหน่วยความจำแฟลชโดยใช้ 2 คำสั่งต่อเนื่องกัน คำสั่งแรกเป็นรอบการเขียน (Write Cycle) คือการเขียนคำสั่ง F0h ลงไปในแอดเดรสใด ๆ ก็ได้ (Address XXXXXh) เพื่อให้หน่วยความจำแฟลชเข้าสู่โหมดการอ่าน (Read Mode) ซึ่งจำเป็นต้องทำเพียงครั้งเดียวเท่านั้น หลังจากนั้นตามด้วยรอบการอ่าน (Read Cycle) คือการสั่งอ่านข้อมูลที่ตำแหน่งที่อยู่ตามที่ต้องการ โดยสามารถอ่านค่าต่อเนื่องกันได้

2) Auto Select Command

มีไว้เพื่อช่วยในการตรวจสอบอุปกรณ์โดยการอ่านค่า ซึ่งระบุรหัสโรงงาน (Manufacture's Code) และ รหัสอุปกรณ์ (Device Code) ของอุปกรณ์ไว้ กระบวนการตรวจสอบเริ่มต้นด้วยการเขียนลำดับคำสั่งเลือกอัตโนมัติ ("Auto Select" Command) ลงในรีจิสเตอร์คำสั่ง หลังจากนั้นให้ตามด้วยรอบการอ่านที่ตำแหน่งที่อยู่ XX00h เพื่ออ่านรหัสโรงงาน ซึ่งในที่นี้มีค่าเป็น

04h สำหรับอุปกรณ์เบอร์ MBM29LV160BE/TE หรือตามด้วยรอบการอ่าน ที่ตำแหน่งที่อยู่ XX01h สำหรับการเชื่อมต่อแบบ 16 บิต และตำแหน่งที่อยู่ XX02h สำหรับการเชื่อมต่อแบบ 8 บิต เพื่ออ่านรหัสอุปกรณ์ ซึ่ง MBM29LV16TE = C4h และ MBM29LV16BE = 49h สำหรับการเชื่อมต่อ 8 บิต และ MBM29LV16TE = 22C4 และ MBM29LV16BE = 2249 สำหรับการเชื่อมต่อ 16 บิต ขั้นตอนการเขียนลำดับคำสั่งเลือกอัตโนมัติ แสดงไว้ในตารางที่ 23

3) Chip Erase Command

เป็นการลบข้อมูลทั้งหมดที่มีอยู่ในหน่วยความจำแฟลชโดยขั้นตอนนี้มี 6 รอบการเขียน เริ่มจาก 2 รอบการเขียนเพื่อปลดล็อก (“Un-lock” Write Cycle) จากนั้นตามด้วยคำสั่งจัดเตรียม (“Set-up” Command) และ 2 รอบการเขียนเพื่อปลดล็อก และคำสั่งลบทั้งหมด (“Chip Erase” Command) ซึ่งจะทำการเขียนและตรวจสอบพื้นที่หน่วยความจำทั้งหมดให้มีค่าเป็น 1 ดังแสดงรายละเอียดไว้ในตารางที่ 23

4) Sector Erase Command

เป็นการลบข้อมูลเฉพาะส่วน โดยขั้นตอนนี้มี 6 รอบการเขียน เริ่มจาก 2 รอบการเขียนเพื่อปลดล็อก จากนั้นตามด้วยคำสั่งจัดเตรียม และ 2 รอบการเขียนเพื่อปลดล็อก และคำสั่งลบเซกเตอร์ (Sector Erase Command) โดย Sector Address (SA) นั้นหมายถึงค่าตำแหน่งที่อยู่ในแต่ละช่วงของเซกเตอร์ ดังแสดงในตารางที่ 22 ซึ่งเป็นตำแหน่งที่อยู่เซกเตอร์ทั้งหมดของ MBM29LV160BE

5) Byte/Word Programing Command

เป็นการโปรแกรมข้อมูลลงในหน่วยความจำแฟลชสามารถทำได้ทั้งแบบ Byte-by-Byte หรือ “Word-by-Word ด้วย 4 รอบการเขียน เริ่มจาก 2 รอบการเขียนเพื่อปลดล็อก ตามด้วยคำสั่งจัดเตรียม จากนั้นจึงเขียนข้อมูลลงไปยังตำแหน่งที่อยู่ที่ต้องการโปรแกรม ดังแสดงในตารางที่ 23 ข้อควรระวังของการเขียนโปรแกรม คือบิตข้อมูลที่เป็น “0” ไม่สามารถโปรแกรมค่าให้กลับเป็น “1” ได้ ดังนั้นจึงควรลบข้อมูลในส่วนของเซกเตอร์ที่ต้องการโปรแกรมใหม่ทุกครั้งเพื่อให้บิตที่ค่าเป็น “0” เปลี่ยนเป็นค่า “1”

ตารางที่ 22 ตำแหน่งเซกเตอร์แอดเดรสของ MBM29LV160BE

Sector Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	(×8) Address Range	(×16) Address Range
SA0	0	0	0	0	0	0	0	X	00000h to 03FFFh	00000h to 01FFFh
SA1	0	0	0	0	0	0	1	0	04000h to 05FFFh	02000h to 02FFFh
SA2	0	0	0	0	0	0	1	1	06000h to 07FFFh	03000h to 03FFFh
SA3	0	0	0	0	0	1	X	X	08000h to 0FFFFh	04000h to 07FFFh
SA4	0	0	0	0	1	X	X	X	10000h to 1FFFFh	08000h to 0FFFFh
SA5	0	0	0	1	0	X	X	X	20000h to 2FFFFh	10000h to 17FFFh
SA6	0	0	0	1	1	X	X	X	30000h to 3FFFFh	18000h to 1FFFFh
SA7	0	0	1	0	0	X	X	X	40000h to 4FFFFh	20000h to 27FFFh
SA8	0	0	1	0	1	X	X	X	50000h to 5FFFFh	28000h to 2FFFFh
SA9	0	0	1	1	0	X	X	X	60000h to 6FFFFh	30000h to 37FFFh
SA10	0	0	1	1	1	X	X	X	70000h to 7FFFFh	38000h to 3FFFFh
SA11	0	1	0	0	0	X	X	X	80000h to 8FFFFh	40000h to 47FFFh
SA12	0	1	0	0	1	X	X	X	90000h to 9FFFFh	48000h to 4FFFFh
SA13	0	1	0	1	0	X	X	X	A0000h to AFFFFh	50000h to 57FFFh
SA14	0	1	0	1	1	X	X	X	B0000h to BFFFFh	58000h to 5FFFFh
SA15	0	1	1	0	0	X	X	X	C0000h to CFFFFh	60000h to 67FFFh
SA16	0	1	1	0	1	X	X	X	D0000h to DFFFFh	68000h to 6FFFFh
SA17	0	1	1	1	0	X	X	X	E0000h to EFFFFh	70000h to 77FFFh
SA18	0	1	1	1	1	X	X	X	F0000h to FFFFFh	78000h to 7FFFFh
SA19	1	0	0	0	0	X	X	X	100000h to 10FFFFh	80000h to 87FFFh
SA20	1	0	0	0	1	X	X	X	110000h to 11FFFFh	88000h to 8FFFFh
SA21	1	0	0	1	0	X	X	X	120000h to 12FFFFh	90000h to 97FFFh
SA22	1	0	0	1	1	X	X	X	130000h to 13FFFFh	98000h to 9FFFFh
SA23	1	0	1	0	0	X	X	X	140000h to 14FFFFh	A0000h to A7FFFh
SA24	1	0	1	0	1	X	X	X	150000h to 15FFFFh	A8000h to AFFFFh
SA25	1	0	1	1	0	X	X	X	160000h to 16FFFFh	B0000h to B7FFFh
SA26	1	0	1	1	1	X	X	X	170000h to 17FFFFh	B8000h to BFFFFh
SA27	1	1	0	0	0	X	X	X	180000h to 18FFFFh	C0000h to C7FFFh
SA28	1	1	0	0	1	X	X	X	190000h to 19FFFFh	C8000h to CFFFFh
SA29	1	1	0	1	0	X	X	X	1A0000h to 1AFFFFh	D0000h to D7FFFh
SA30	1	1	0	1	1	X	X	X	1B0000h to 1BFFFFh	D8000h to DFFFFh
SA31	1	1	1	0	0	X	X	X	1C0000h to 1CFFFFh	E0000h to E7FFFh
SA32	1	1	1	0	1	X	X	X	1D0000h to 1DFFFFh	E8000h to EFFFFh
SA33	1	1	1	1	0	X	X	X	1E0000h to 1EFFFFh	F0000h to F7FFFh
SA34	1	1	1	1	1	X	X	X	1F0000h to 1FFFFFh	F8000h to FFFFFh

ที่มา : Spansion. (2003)

ในงานหลาย ๆ ด้านที่ใช้ตัวประมวลผลสัญญาณดิจิทัล นั้นมีความจำเป็นต้องเก็บโปรแกรมไว้ที่หน่วยความจำแฟลช ดังนั้นในขณะเริ่มจ่ายไฟ โปรแกรมจะถูกคัดลอกจากหน่วยความจำแฟลชและถูกโหลดเข้าไปยังหน่วยความจำภายในของตัวประมวลผล DSP ในตระกูล TMS320C6x DSP ได้เสนอ วิธีการเริ่มทำงานของระบบ (Boot Configuration) อยู่ 3 วิธี คือ ไม่มีการโหลดโปรแกรมจากภายนอก (No Boot Process) การเริ่มทำงานโดยโหลดโปรแกรมจากหน่วยความจำอ่านอย่างเดียว (ROM Boot Process) และ การเริ่มทำงานโดยโหลดโปรแกรมจาก

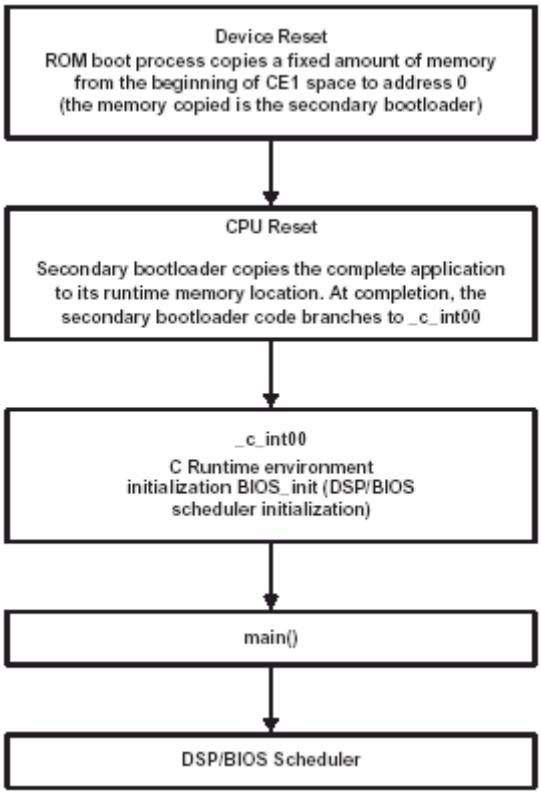
แม่งาน (Host Boot Process) วิธีที่ได้รับความนิยมที่สุดคือ การโหลดโปรแกรมจากหน่วยความจำอ่านอย่างเดียว การทำงานจะทำการคัดลอกข้อมูลและคำสั่งจากหน่วยความจำอ่านอย่างเดียวภายนอกที่พื้นที่ CE1 เริ่มตั้งแต่ตำแหน่งที่อยู่ 00h ขนาด 1K ไบต์ ซึ่ง EDMA Controller จะคัดลอกมาไว้ยังหน่วยความจำภายในที่ตำแหน่งที่อยู่ 00h การส่งผ่านข้อมูลเกิดขึ้นหลังจากขอบขึ้นที่ขาสัญญาณรีเซตของหน่วยความจำแฟลช เมื่อการส่งผ่านข้อมูลเสร็จสิ้นตัวประมวลผล DSP จะทำงานโดยเริ่มอ่านคำสั่งจากตำแหน่งที่อยู่ 00h

ตารางที่ 23 คำสั่งควบคุมการทำงานของ MBM29LV160BE

Command Sequence		BUS Write Cycle Req	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Fourth Bus Read/Write Cycle		Fifth Bus Write Cycle		Sixth Bus Weite Cycle	
			Add	Data	Add	Data	Add	Data	Add	Data	Add	Data	Add	Data
Read/Reset	Word	1	XXX	F0h	-	-	-	-	-	-	-	-	-	-
	Byte													
Read/Reset	Word	3	555	AAh	2AA	55h	555	F0h	RA	RD	-	-	-	-
	Byte		AAA		555		AAA							
Autoselect	Word	3	555	AAh	2AA	55h	555	90h	-	-	-	-	-	-
	Byte		AAA		555		AAA							
Program	Word	4	555	AAh	2AA	55h	555	A0h	PA	PD	-	-	-	-
	Byte		AAA		555		AAA							
Chip Ease	Word	6	555	AAh	2AA	55h	555	80h	555	AAh	2AA	55h	555	10h
	Byte		AAA		555		AAA		AAA		555		AAA	
Sector	Word	6	555	AAh	2AA	55h	555	80h	555	AAh	2AA	55h	SA	30h
	Byte		AAA		555		AAA		AAA		555			

ที่มา : Spansion. (2003)

สำหรับงานบางอย่างที่จำเป็นต้องใช้ข้อมูลและคำสั่งในการเริ่มต้นทำงานมากกว่า 1K ไบต์ สามารถทำได้โดยใช้โปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสอง “Secondary Boot Loader” หรือเรียกอีกอย่างว่า “Custom Boot Code” ซึ่งจะถูกเก็บไว้ภายในส่วน 1K ไบต์ แรกของหน่วยความจำแฟลช จากนั้นเมื่อการส่งผ่านข้อมูลมายังหน่วยความจำภายในเสร็จสิ้นตัวประมวลผล DSP จะเริ่มทำงานจากตำแหน่งที่อยู่ 00h และทำงานตามโปรแกรม “Secondary Boot Loader” ซึ่งมีหน้าที่คัดลอกข้อมูลและคำสั่งในส่วนที่เหลือจากหน่วยความจำแฟลชมาไว้ยังหน่วยความจำภายใน ขั้นตอนการทำงานแสดงไว้ในภาพที่ 93 รายละเอียดสามารถดูเพิ่มเติมได้ใน (Spansion, 2003)



ภาพที่ 93 การทำงานของโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสอง
ที่มา : Texas Instruments. (2004)

การทดสอบหน่วยความจำแฟลชเพื่อใช้เป็นพื้นที่สำหรับเก็บโปรแกรมระบบปฏิบัติการ มีขั้นตอนดังต่อไปนี้

1) การสร้างส่วนพื้นที่ของหน่วยความจำ (Create Necessary Memory Segment)

พื้นที่สำหรับเก็บโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำ (BOOT Section) จะถูกสร้างขึ้นเพื่อเป็นพื้นที่สำหรับวางโปรแกรม “Secondary Boot Loader” ซึ่งถูกคัดลอกมาจาก 1K ไบต์ แรกของหน่วยความจำแฟลช การสร้างพื้นที่ในส่วนนี้ทำได้โดยเปิดไฟล์ “DSP/BIOS Configuration” (.DSB) คลิกขวาบน “MEM-Memory Section Manager Object” และเลือก “Insert MEM” จากนั้นสร้างพื้นที่ของหน่วยความจำแฟลช (FLASH Section) พื้นที่ของหน่วยความจำภายใน (IRAM Section) และพื้นที่สำหรับเก็บโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำ (BOOT Section) ต่อไปนี้ดังนี้

```
FLASH   : origin = 0X90000000 , length = 0X200000
BOOT    : origin = 0X0      , length = 0X400
IRAM    : origin = 0X400          , length = 0X2FB00
```

ข้อควรระวังในการสร้าง BOOT Section คือต้องแน่ใจว่าไม่ได้เลือกเครื่องหมายถูกที่ช่อง “Create A Heap in This Memory” และส่วน “Space Attribute” ให้เลือกเป็น “Code/Data”

2) การสร้างไฟล์คำสั่งเชื่อมโยงสำหรับผู้ใช้งาน (Create User Linker Command File)

โปรแกรม “Secondary Boot Loader” นั้นเมื่อแปลงโปรแกรมแล้วจะอยู่ในไฟล์ .boot_load ดังนั้นจึงจำเป็นต้องมีการสร้างไฟล์คำสั่งเพื่อเชื่อมโยงไฟล์ในส่วนนี้ให้เข้ากับไฟล์อื่นๆ ของโปรแกรม ซึ่งสามารถเขียนได้ดังนี้

```
-l app.cmd  /* DSP/BIOS generated cmd file from cdb */
SECTIONS {
        .boot_loader > BOOT
}
```

คำสั่งข้างต้นนี้จะทำให้ส่วนของไฟล์ .boot_load ถูกเรียกใช้งานใน BOOT Section บนหน่วยความจำภายในของตัวประมวลผล DSP ซึ่งเป็นการเขียนเพิ่มเติมลงในส่วนของไฟล์คำสั่งเชื่อมโยงสำหรับผู้ใช้ (User Linker Command File) หรือไฟล์ userlinker.cmd

3) การเขียนโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสอง (Writing The Secondary Boot Loader)

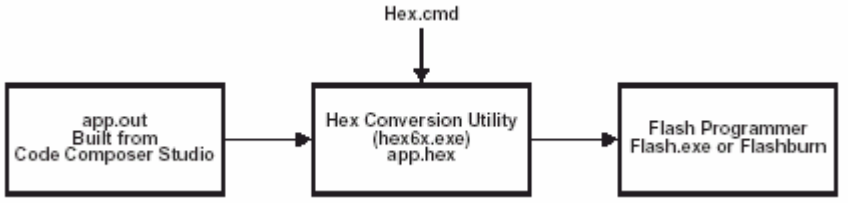
โปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสองนั้น มีความจำเป็นเมื่อขนาดของหน่วยความจำที่ต้องการคัดลอกจากหน่วยความจำแฟลชใหญ่เกินกว่า 1K ไบต์ เพื่อทำหน้าที่คัดลอกข้อมูลสำหรับการตั้งต้นในส่วนที่เหลือ ภาพที่ 94 แสดงตัวอย่างการเขียนของโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสอง หลังจากทำตามขั้นตอนข้างต้นเรียบร้อยแล้ว จากนั้นจึงทำการแปลภาษาและเชื่อมโยงไฟล์ต่างๆ เพื่อสร้างไฟล์เอาต์พุต (app.out) ที่สามารถทำงานบนตัวประมวลผลได้

```
            .title  "Flash bootup utility for 6713 dsk"
            .option D,T
            .length 102
            .width  140

COPY_TABLE    .equ    0x90000400

            .sect ".boot_load"
            .global _boot

_boot:
;*************************************************************************
;* Debug Loop -  Comment out B for Normal Operation
;*************************************************************************

            zero B1
_myloop:  ; [!B1] B _myloop
            nop  5
_myloopend: nop

;*************************************************************************
;* Copy code sections
;*************************************************************************
        mvkl  COPY_TABLE, a3    ; load table pointer
        mvkh  COPY_TABLE, a3

        ldw   *a3++, b1         ; Load entry point

copy_section_top:
        ldw   *a3++, b0         ; byte count
        ldw   *a3++, a4         ; ram start address
        nop   3

 [!b0]  b copy_done             ; have we copied all sections?
        nop   5

copy_loop:
        ldb   *a3++,b5
        sub   b0,1,b0           ; decrement counter
 [ b0]  b     copy_loop         ; setup branch if not done
 [!b0]  b     copy_section_top
        zero  a1
 [!b0]  and   3,a3,a1
        stb   b5,*a4++
 [!b0]  and   -4,a3,a5          ; round address up to next multiple of 4
 [ a1]  add   4,a5,a3           ; round address up to next multiple of 4

;*************************************************************************
;* Jump to entry point
;*************************************************************************
copy_done:
        b     .S2 b1
        nop   5
```

ภาพที่ 94 ตัวอย่างโปรแกรมนำระบบปฏิบัติการสู่หน่วยความจำระดับสอง

4) การโปรแกรมหน่วยความจำแฟลช (Programming Flash Memory)

ชุดเครื่องมือรวมสำหรับเขียนโปรแกรม (Code Composer Studio) นั้นมีเครื่องมือที่ใช้ในการช่วยเขียนข้อมูลลงไปยังหน่วยความจำแฟลช สำหรับในสถาปัตยกรรม TMS320C6X DSP เครื่องมือที่ถูกใช้นั้นได้แก่ เครื่องมือโปรแกรมหน่วยความจำแฟลช (Flash Burn Utility) และเครื่องมือแปลงข้อมูลให้อยู่ในรูปเลขฐานสิบหก (Hex Conversion Utility) โดยไฟล์เอาต์พุต (app.out) ที่ได้มาจะต้องถูกเปลี่ยนเป็นไฟล์เลขฐานสิบหก (app.hex) เพื่อให้สามารถโหลดไฟล์เลขฐานสิบหกนั้นลงในหน่วยความจำแฟลชได้ ดังแสดงขั้นตอนการทำงานในภาพที่ 95



ภาพที่ 95 ขั้นตอนการโปรแกรมหน่วยความจำแฟลช

ที่มา : Texas Instruments. (2002)

4.1) เครื่องมือแปลงข้อมูลให้อยู่ในรูปเลขฐานสิบหก (Hex Conversion Utility)

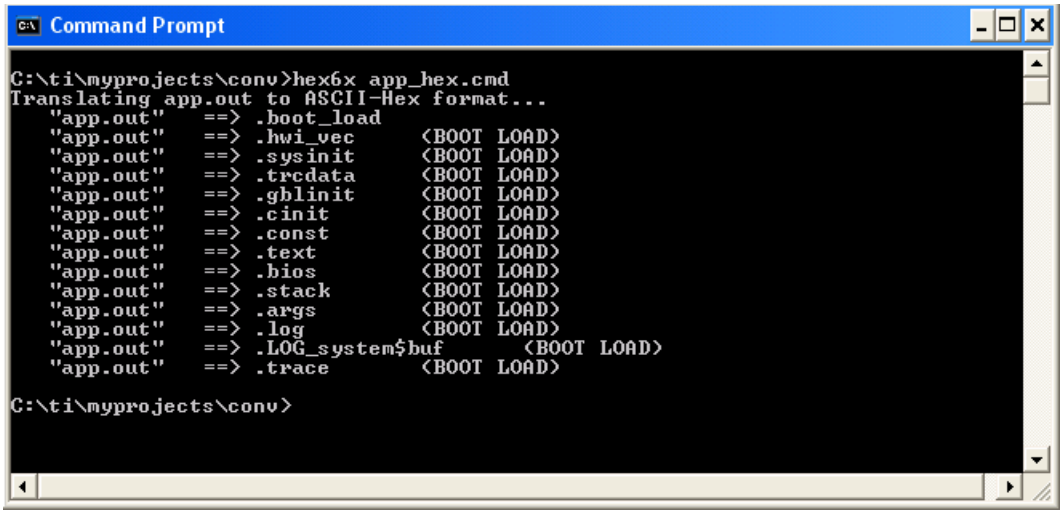
เครื่องมือแปลงข้อมูลให้อยู่ในรูปเลขฐานสิบหกเป็นเครื่องมือซึ่งทำงานโดยรับรูปแบบไฟล์จุดหมายร่วมกัน (Common Object File Format, COFF) หรือไฟล์เอาต์พุต (*.out) เป็นอินพุตไฟล์ และทำการแปลงให้อยู่ในรูปแบบเลขฐานสิบหก (ASCll Hexadecimal) ซึ่งจะต้องใช้ร่วมกับไฟล์คำสั่ง (Command File) ไฟล์คำสั่งจะประกอบไปด้วยข้อมูลที่จำเป็นต่าง ๆ เช่น ขนาด ตำแหน่งที่อยู่ และรูปแบบการติดต่อสื่อสารของหน่วยความจำแฟลช เป็นต้น ซึ่งจะต้องมีการระบุชื่ออินพุตไฟล์ (app.out) และเอาต์พุตไฟล์ (app.hex) ที่ต้องการดังแสดงในภาพที่ 96

```
debug\app.out                          /* input COFF file */
-a                                     /* create ASCII image */
-memwidth  8                           /* width of ROM/Flash memory */
-map post_hex.map                      /* create a hex map file */
-boot                                  /* create a boot table for all initialized sects */
-bootorg 0x90000400                    /* address of the boot/copy-table */
-bootsection .boot_load 0x90000000 /* section containing our asm boot routine */

ROMS
{
    FLASH:  org = 0x90000000, len = 0x10000, romwidth = 8, files = {app.hex}
}
```

ภาพที่ 96 การกำหนดค่าควบคุมต่างๆในไฟล์คำสั่ง

ที่มา : Texas Instruments. (2002)

การใช้งานเครื่องมือแปลงข้อมูลให้อยู่ในรูปเลขฐานสิบหก ทำได้โดยพิมพ์คำสั่ง “hex6x app_hex.cmd” ลงในโปรแกรมประยุกต์ Command Prompt เพื่อเรียกการใช้งาน จากนั้นเครื่องมือจะทำการสร้างไฟล์รูปแบบเลขฐานสิบหกและแสดงรายละเอียดต่างๆ ดังภาพที่ 97



ภาพที่ 97 การแปลงไฟล์จาก COFF (.out) ให้เป็น ASCll Hexadecimal (.hex)

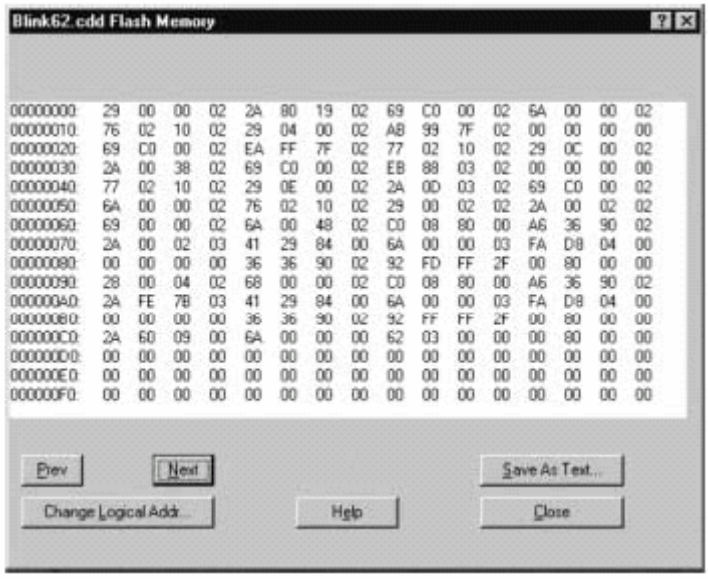
4.2) เครื่องมือโปรแกรมหน่วยความจำแฟลช (Flash Burn Utility)

เมื่อได้ไฟล์รูปแบบเลขฐานสิบหก (*.hex) แล้ว ขั้นต่อไปคือการใช้เครื่องมือโปรแกรมหน่วยความจำแฟลชเพื่อเขียนไฟล์ดังกล่าวลงไปในหน่วยความจำแฟลช เครื่องมือโปรแกรมหน่วยความจำแฟลชเป็นส่วนที่มีมากับชุดเครื่องมือรวมสำหรับการเขียนโปรแกรม (Code Composer Studio, CCS) ถ้าไม่มีสามารถดาวน์โหลดได้จากเว็บ (Website) ของบริษัท TI การทำงานเริ่มจากเปิดโปรแกรมเครื่องมือโปรแกรมหน่วยความจำแฟลชขึ้นมา โดยจะต้องแน่ใจว่าไม่ได้เปิดโปรแกรม CCS ขึ้นมาพร้อมกัน หลังจากนั้นเปิดไฟล์ข้อมูลแสดงรูปลักษณ์ (FlashBurn Config. Details Document File, *.ccd) ซึ่งเป็นไฟล์ที่จะถูกใช้งานร่วมกับเครื่องมือโปรแกรมหน่วยความจำแฟลช เมื่อไฟล์ *.ccd ถูกเปิดขึ้น โปรแกรมจะทำการการตั้งระบบใหม่ (Reset) ให้กับบอร์ดประมวลผล และใช้เวลาอีกเล็กน้อยเพื่อดาวน์โหลดโปรแกรม “Flash Burn Target Component” หรือ FBTC ลงในตัวประมวลผล DSP

โปรแกรม FBTC คือโปรแกรมที่ทำหน้าที่ติดต่อสื่อสารระหว่างเครื่องมือโปรแกรมหน่วยความจำแฟลชและหน่วยความจำแฟลช เนื่องจากเครื่องมือโปรแกรมหน่วยความจำ

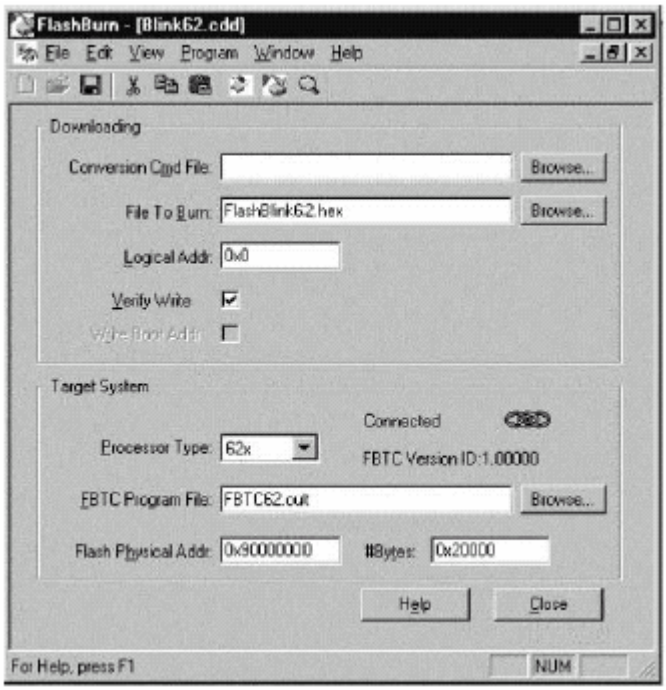
แฟลชนั้น ไม่สามารถลบหรือโปรแกรมหน่วยความจำแฟลชได้โดยตรง แต่โปรแกรม FBTC จะถูกฝังตัวอยู่ในตัวประมวลผล DSP และมีหน้าที่จัดการกับงานเหล่านั้นเอง ภาพที่ 99 แสดงการใช้งานเครื่องมือโปรแกรมหน่วยความจำแฟลช หลังจากเปิดไฟล์ *.ccd ขึ้นมา และทำการติดต่อกับตัวประมวลผล DSP ได้สำเร็จ โปรแกรมจะแสดงสถานะว่าเชื่อมต่อกับ FBTC ได้ โดยห่วงโซ่ทั้ง 3 ห่วงจะคล้องเชื่อมกันอยู่ ถ้ายังไม่สามารถเชื่อมต่อได้ให้เลือกไฟล์ FBTC ให้ถูกต้อง แล้วดาวน์โหลดลงไปใหม่โดยใช้คำสั่ง “Program -> Download FBTC” จากนั้นสั่งลบข้อมูลทั้งหมดในหน่วยความจำแฟลช ด้วยคำสั่ง “Program -> Erase Flash” โปรแกรมจะแสดงสถานะการลบข้อมูลซึ่งจะใช้เวลาเพียง 5-6 วินาที

จากนั้นเลือกไฟล์รูปแบบเลขฐานสิบหก (*.hex) ที่จะเขียนลงไป ตรวจสอบตำแหน่งที่อยู่ที่จะเริ่มเขียน แล้วเลือก “Program -> Program Flash” เพื่อทำการเขียนลงไปยังหน่วยความจำแฟลช โปรแกรมจะแสดงสถานะการเขียนจากเริ่มจนจนสิ้นสุด ส่วนการตรวจสอบข้อมูลที่เขียนลงไปสามารถดูได้จากคำสั่ง “Program -> Show Memory” โดยสามารถระบุตำแหน่งที่อยู่ที่จะขอดูและรูปแบบการแสดงข้อมูลได้ว่าเป็นแบบ 8 บิต 16 บิต หรือ 32 บิต ดังแสดงในภาพที่ 98



ภาพที่ 98 การดูข้อมูลในหน่วยความจำแบบแฟลช

ที่มา : Texas Instruments. (2002)



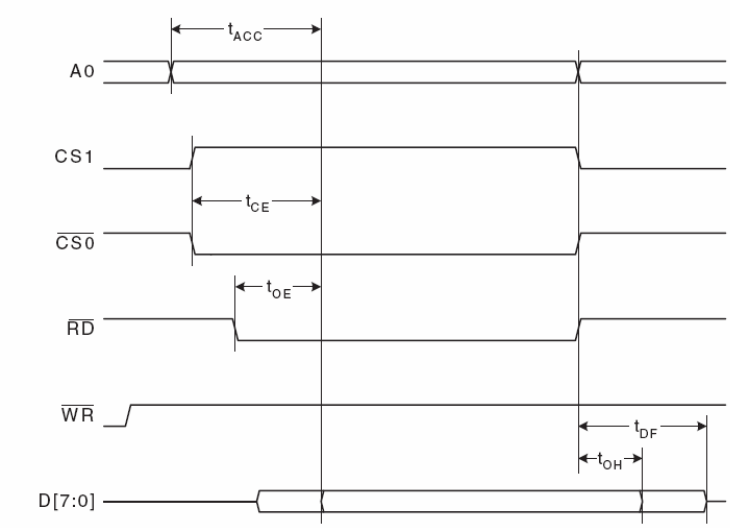
ภาพที่ 99 หน้าต่างของเครื่องมือโปรแกรมหน่วยความจำแฟลช
ที่มา : Texas Instruments. (2002)

6. Fingerprint Sensor

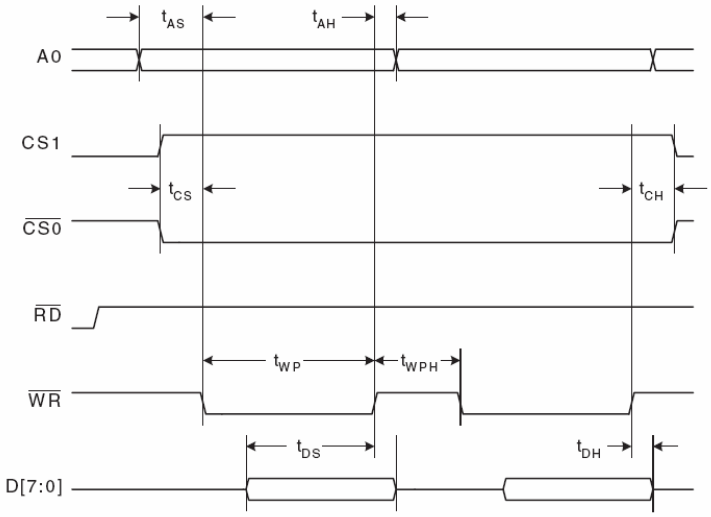
ในหัวข้อนี้จะกล่าวถึงวิธีการเขียนโปรแกรมควบคุมบนตัวประมวลผลสัญญาณดิจิทัล สำหรับการติดต่อกับฮาร์ดแวร์กราดตรวจลายนิ้วมือ ซึ่งสามารถติดต่อผ่านมอดูล EMIF ของตัวประมวลผลสัญญาณดิจิทัล ด้วยวิธีการเข้าถึงแบบ ASRAM (Asynchronous Static Random-Access Memory) โดยมีรายละเอียดของการทำงานดังต่อไปนี้

การรับภาพจากวงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุนี้ ต้องใช้ขาสัญญาณ CS0, CS1, RD, WR, A0 และ D[0:7] ในการอ่านและเขียนข้อมูลของฟังก์ชันรีจิสเตอร์ ซึ่งมีแผนภาพสัญญาณเวลา (Timing Diagram) ดังภาพที่ 100 และภาพที่ 101 ตามลำดับ โดยขา A0 เป็นขาตำแหน่งที่อยู่ (Address) ใช้ในการเลือกระหว่างรีจิสเตอร์ตัวชี้ (Index Register) กับรีจิสเตอร์ข้อมูล (Data Register) ถ้าขา A0 มีค่าต่ำ (Active Low) จะเป็นการเลือกรีจิสเตอร์ตัวชี้ ในทางตรงกันข้ามถ้า A0 มีค่าสูง (Active High) จะเป็นการเข้าถึงฟังก์ชั่นรีจิสเตอร์ (Function Register) ที่ถูกเลือกโดยรีจิสเตอร์ตัวชี้

ในการอ่านข้อมูลของฟังก์ชั่นรีจิสเตอร์ จะให้ขา A0 มีค่าต่ำ และเขียนค่ารีจิสเตอร์ที่ต้องการอ่านค่าผ่านทางขา D[0:7] ตัวอย่างเช่น ต้องการอ่านค่ารีจิสเตอร์ CIDH (Chip Identification High) ณ ตำแหน่งที่อยู่ 0x10 ซึ่งจะบอกสถานะของวงจรรวมตรวจกราดลายนิ้วมือว่าพร้อมทำงานหรือไม่ โดยให้ขา CS0 และ RD มีค่าต่ำ ส่วนขา CS1 และ WR มีค่าสูง และส่งค่า 0x10 ไปยังขา D[0:7] ถ้าวงจรรวมรับภาพลายนิ้วมือนี้ส่งค่ากลับมาทางขา D[0:7] เป็น 0x20 แสดงว่าพร้อมที่จะทำงาน แต่ถ้าเป็นค่าอื่นๆ แสดงว่ายังไม่พร้อมทำงาน

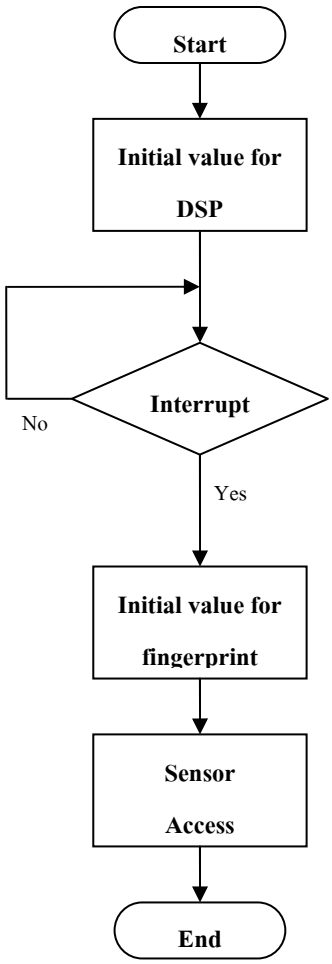


ภาพที่ 100 แผนภาพเวลาในการอ่านข้อมูลของ ตัวกราดตรวจลายนิ้วมือ(MBF200)
ที่มา : Fujitsu Semiconductor. (2004)



ภาพที่ 101 แผนภาพเวลาในการเขียนข้อมูลของตัวกราดตรวจลายนิ้วมือ (MBF200)
ที่มา : Fujitsu Semiconductor. (2004)

ในส่วนของการเขียนข้อมูลจะแบ่งออกเป็น 2 ช่วงด้วยกัน แต่ขา CS1 และ RD จะมีค่าสูง และขา CS0 มีค่าต่ำตลอดการเขียนข้อมูล โดยในช่วงแรกขา A0 และ WR จะมีค่าต่ำ เพื่อเลือกฟังก์ชั่นรีจิสเตอร์ที่ต้องการเขียนข้อมูล หลังจากนั้นจะให้ขา WR มีค่าสูง ในช่วงที่สองจะให้ขา A0 มีค่าสูง และขา WR มีค่าต่ำ เพื่อเขียนค่าที่ต้องการไปยังฟังก์ชั่นรีจิสเตอร์ ตัวอย่างเช่น ต้องการเขียนค่ารีจิสเตอร์ CTRLC (Control Register C) ณ ตำแหน่งที่อยู่ 0x0A ซึ่งเป็นรีจิสเตอร์ที่ควบคุมขา P0 และ P1 ของวงจรรวมกราดตรวจลายนิ้วมือ ถ้าต้องการให้ขา P0 ทำงาน ซึ่งจะปล่อยกระแสออกมา จะต้องเขียนค่า 0x01 ไปยังค่ารีจิสเตอร์นี้ โดยในช่วงแรกจะเขียนค่า 0x0A ไปยังขา D[0:7] และช่วงที่สองจะเขียนค่า 0x01 ไปยังขา D[0:7] ถ้าต้องการให้ขา P1 ทำงานจะต้องเขียนค่า 0x02 ไปยังค่ารีจิสเตอร์นี้



ภาพที่ 102 ขั้นตอนการทดสอบการทำงานของตัวกราดตรวจลายนิ้วมือ

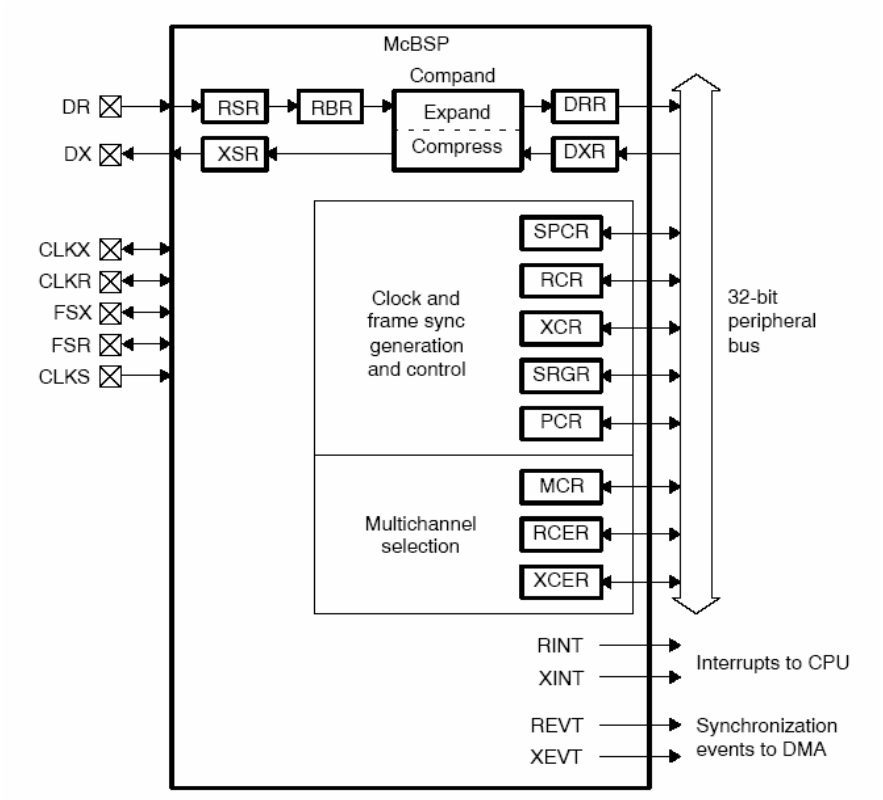
การทดสอบการใช้งานวงจรรวมกราดตรวจลายนิ้วมือแบบตัวเก็บประจุนั้น ติดต่อผ่านมอดูล EMIF ในช่วงพื้นที่ CE2 การทดสอบการทำงาน สามารถทำได้โดยเขียนโปรแกรมบนตัวประมวลผล DSP ซึ่งเป็นโปรแกรมทดสอบการเขียนและอ่านของตัวกราดตรวจลายนิ้วมือ โดยขั้นตอนการเขียนโปรแกรมแสดงดังภาพที่ 102 ซึ่งแสดงถึงผังงานของการเขียนโปรแกรมควบคุมตัวกราดตรวจลายนิ้วมือ โดยโปรแกรมจะเริ่มจากการกำหนดค่าเริ่มต้นของมอดูล EMIF ที่จะใช้ในการควบคุม เช่น กำหนดค่าการอ่านเขียน การเปิดรับการขัดจังหวะสัญญาณ (Interrupt) เป็นต้น หลังจากนั้นจะรอสัญญาณที่มาจากตัวกราดตรวจลายนิ้วมือ ซึ่งเป็นสัญญาณขัดจังหวะเพื่อบอกให้รู้ว่ามีการวางนิ้วมือลงบนตัวกราดตรวจลายนิ้วมือแล้ว ต่อมาจะเริ่มทำการกำหนดค่าต่าง ๆ ให้กับตัวกราดตรวจลายนิ้วมือ เพื่อทำการอ่านข้อมูลดิจิทัลของลายนิ้วมือมาเก็บไว้ในหน่วยความจำเพื่อนำไปประมวลผลต่อไป แล้วจะทำการทดสอบโดยโหลดค่าข้อมูลดิจิทัลของลายนิ้วมือที่อยู่ในหน่วยความจำขึ้นมาเก็บไว้เป็นไฟล์ข้อมูล (*.dat) ผ่านโปรแกรม CCS โดยใช้คำสั่ง “File -> data -> save” โดยระบุขนาดและตำแหน่งที่อยู่เริ่มต้นของข้อมูลให้ถูกต้อง จากนั้นจึงทำการแปลงไฟล์ข้อมูล (*.dat)ให้เป็นไฟล์รูปภาพ (*.bmp) เพื่อใช้ดูผลลัพธ์ของรูปลายนิ้วมือที่ได้

หลังจากทำการประกอบแผงวงจรกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ และได้ทำการทดสอบพบว่า เมื่อเพิ่มวงจรตรวจสอบการวางนิ้วเชิงแสงเข้าไปบนแผงวงจรพิมพ์ แทนฟังก์ชั่นตรวจสอบการวางนิ้วแบบอัตโนมัติที่อยู่ภายในวงจรรวมรับภาพลายนิ้วมือแบบตัวเก็บประจุ ซึ่งทำงานได้ไม่ค่อยเสถียรนัก ทำให้ฮาร์ดแวร์กราดตรวจลายนิ้วมือแบบแบบตัวเก็บประจุสามารถติดต่อสื่อสารกับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลได้ แต่ฮาร์ดแวร์นี้มีอายุการใช้งานขึ้นกับวงจรรวมกราดตรวจลายนิ้วมือ เนื่องจากเมื่อใช้งานเป็นเวลานาน สารเคมีที่เคลือบอยู่บนหน้าสัมผัสของวงจรรวมกราดตรวจลายนิ้วมือ ซึ่งทำหน้าที่ป้องกันรอยขีดข่วน จะลดน้อยลง ทำให้หน้าสัมผัสเกิดรอยขีดข่วนได้ง่าย วงจรรวมกราดตรวจลายนิ้วมือจึงค่อยๆ เสื่อมสภาพลง

7. Multichannel Buffered Serial Port (McBSP)

ในหัวข้อนี้จะกล่าวถึงหลักและวิธีการในการใช้งานมอดูล McBSP (Multi-channel Buffered Serial Port) สำหรับติดต่อกับมาตรฐานการสื่อสาร UART (Universal Asynchronous Receiver/Transmitter) เนื่องจากในการติดต่อสื่อสารกับฮาร์ดแวร์เชื่อมต่อเครือข่าย ฮาร์ดแวร์แป้นพิมพ์และแสดงผล และ ฮาร์ดแวร์ควบคุมการเปิดปิดประตู ล้วนต้องอาศัยการติดต่อผ่านทางพอร์ตอนุกรมของตัวประมวลผลสัญญาณดิจิทัล (McBSP) ทั้งสิ้น ในหัวข้อนี้จะกล่าวถึงวิธีการออกแบบและเขียนโครงสร้างโปรแกรมทางด้านซอฟต์แวร์สำหรับทดสอบการติดต่อทางพอร์ตอนุกรมของบนตัวประมวลผลสัญญาณดิจิทัล ซึ่งจะกล่าวในรายละเอียดของส่วนต่าง ๆ ดังต่อไปนี้

มอดูล McBSP ของตัวประมวลผลสัญญาณดิจิทัลจะประกอบด้วยขาสัญญาณใช้งานและรีจิสเตอร์ภาย ซึ่งแสดงได้ดังภาพที่ 103

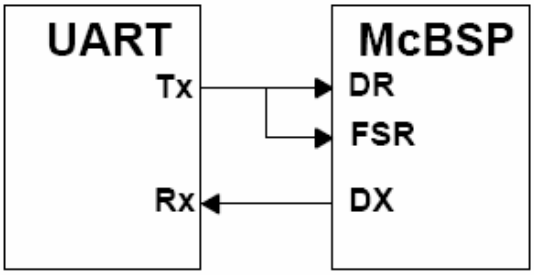


ภาพที่ 103 แผนภาพของมอดูล McBSP

ที่มา : Texas Instruments Inc. (2003)

จากภาพที่ 103 นั้นมอดูล McBSP จะมีขาสัญญาณ 7 ขา โดยแบ่งเป็น 2 กลุ่ม กลุ่มแรกใช้ในการรับและส่งข้อมูล ได้แก่ขา DX และ DR ส่วนกลุ่มที่สองจะใช้ในการเข้าจังหวะสัญญาณ (Synchronization) ได้แก่ขา CLKX CLKR FSX FSR และ CLKS โดยที่ขา CLKX และ CLKR ใช้สำหรับรับส่งสัญญาณนาฬิกาเพื่อทำการเข้าจังหวะบิตสัญญาณระหว่างการส่งและรับตามลำดับ ขา CLKS เป็นสัญญาณนาฬิกาขาเข้าของมอดูล McBSP ส่วนขา FSX และ FSR ใช้สำหรับรับส่งสัญญาณที่ใช้ในการเข้าจังหวะเฟรมของสัญญาณส่งและรับตามลำดับ

ในการติดต่อทางด้านฮาร์ดแวร์ระหว่างมอดูล McBSP บนตัวประมวลผลสัญญาณดิจิทัลกับอุปกรณ์มาตรฐานการสื่อสาร UART นั้น ที่มอดูล McBSP จะใช้เพียงขาสัญญาณ DX DR และ FSR ในการติดต่อกับอุปกรณ์มาตรฐานการสื่อสาร UART ดังภาพที่ 104



ภาพที่ 104 การเชื่อมต่อระหว่างมอดูล McBSP กับอุปกรณ์ UART
ที่มา : Texas Instruments Inc. (2003)

จากภาพที่ 104 ข้อมูลจากมอดูล McBSP ไปยังมอดูล UART จะส่งเข้าขา Rx ผ่านทางขา DX โดยตรง ส่วนข้อมูลจากมอดูล UART ไปยังมอดูล McBSP จะส่งผ่านขา Tx ไปยังขา DR และ FSR พร้อมกัน โดยขา FSR จะทำหน้าที่ตรวจจับแบบไม่เข้าจังหวะ (Asynchronous) ว่ามีข้อมูลถูกส่งมาหรือไม่ ซึ่งจะใช้ซอฟต์แวร์ในการปรับแต่งคุณลักษณะของมอดูล McBSP เพื่อให้สามารถใช้ติดต่อกับ UART ได้

การใช้งานรีจิสเตอร์ควบคุมทำได้โดยสั่งผ่าน 32-bit Peripheral Bus โดยมีรายละเอียดของแต่ละรีจิสเตอร์ดังต่อไปนี้

DRR เป็นรีจิสเตอร์ สำหรับรับข้อมูล
DXR เป็นรีจิสเตอร์สำหรับส่งข้อมูล

SPCR ทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม ซึ่งประกอบไปด้วย บิตที่แสดงสถานะต่าง ๆ ของมอดูล McBSP

PCR ทำหน้าที่กำหนดการทำงานของมอดูล McBSP ให้เป็นพอร์ตอินพุต-เอาต์พุตทั่วไป (General–Purpose I/O Port) ในระหว่างที่ทำการตั้งระบบใหม่ (Reset) และเก็บบิตแสดงสถานะของมอดูล McBSP

RCR ทำหน้าที่กำหนดตัวแปรต่าง ๆ ในการทำงานของภาครับ

XCR ทำหน้าที่กำหนดตัวแปรต่าง ๆ ในการทำงานของภาคส่ง

SRGR ทำหน้าที่ควบคุมและกำหนดตัวแปรที่ใช้สร้างอัตราการชักตัวอย่าง (Samplerate Generator)

MCR ทำหน้าที่ควบคุมการใช้งานแบบหลายช่องสัญญาณ (Multichannel Section Mode) โดยสามารถเลือกใช้งานได้ถึง 128 ช่องสัญญาณ (Channel) ในเวลาเดียวกัน

RCER ทำหน้าที่ควบคุมการทำงานของเฟรมย่อย (Subframe) ต่าง ๆ ของช่องสัญญาณรับ (Receive Channel)

XCER ใช้สำหรับควบคุมการทำงานของเฟรมย่อย (Subframe) ต่าง ๆ ของช่องสัญญาณส่ง (Transmit Channel)

บนบอร์ดฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลจะมีมอดูล McBSP ให้ใช้อยู่ 2 ชุด ได้แก่ McBSP0 และ McBSP1 แต่ละชุดจะมีขาสัญญาณเชื่อมต่อภายนอก ชุดละ 7 ขา ซึ่งประกอบไปด้วยส่วนข้อมูล (Data Path) และส่วนควบคุม (Control Path) โดยแยกส่วนภาครับและภาคส่งออกจากกัน มอดูล McBSP จะส่งข้อมูลไปยังอุปกรณ์ภายนอกโดยผ่านขา DX และรับข้อมูลโดยผ่านขา DR ส่วนขา CLKS, CLKX, CLKR, FSX และ FSR เป็นสัญญาณนาฬิกา (Clock) และสัญญาณเข้าจังหวะ (Frame Synchronization) เพื่อใช้ควบคุมการทำงานของการรับส่งแบบอนุกรม ซึ่งตัวประมวลผลสามารถสั่งงานมอดูล McBSP ได้โดยผ่านรีจิสเตอร์ขนาด 32 บิต โดยอ่านข้อมูลที่ได้รับจากรีจิสเตอร์ DRR และเขียนข้อมูลออกภายนอกโดยผ่านรีจิสเตอร์ DXR ซึ่งข้อมูลที่ถูกเขียนลงรีจิสเตอร์ DXR จะถูกส่งไปยังขา DX ทีละบิต ด้วยการทำงานของรีจิสเตอร์ XSR เช่นเดียวกับการรับข้อมูลผ่านขา DR ข้อมูลจะถูกนำมาเก็บไว้ในรีจิสเตอร์ RSR และถูกคัดลอกไปยังรีจิสเตอร์ RBR ก่อนที่จะถูกส่งไปยังรีจิสเตอร์ DRR เพื่อให้ตัวประมวลผลนำไปประมวลผลต่อไป ซึ่งในภาพที่ 105 แสดงตำแหน่งของรีจิสเตอร์ต่าง ๆ ที่ต้องใช้ในการทดสอบการทำงานฮาร์ดแวร์ในขั้นพื้นฐาน

ในการทดสอบการทำงานฮาร์ดแวร์ในขั้นพื้นฐาน ขา CLKX, FSX, DX, CLKK, FRS, DR และ CLKS จะถูกกำหนดให้ใช้งานในโหมดของพอร์ตอินพุต-เอาต์พุตทั่วไป และทดสอบการรับและส่งข้อมูลของแต่ละขาในระดับบิต ซึ่งทำได้ต่อไปนี้ดังนี้

1) กำหนดให้พอร์ตอนุกรมอยู่ในระหว่างการตั้งระบบใหม่ (Reset) โดยกำหนดบิต (R/X)RST = 0 ในรีจิสเตอร์ SPCR

2) เลือกการใช้งานในโหมดพอร์ตอินพุต-เอาต์พุตทั่วไป โดยกำหนดบิต (R/X)IOEN = 1 ในรีจิสเตอร์ PCR

3) กำหนดบิต CLKXM = 1 ในรีจิสเตอร์ PCR เพื่อกำหนดให้ขา CLKX เป็นเอาต์พุต โดยค่าที่ถูกเขียนลงในบิต CLKXP จะถูกส่งออกไปยังขา CLKX โดยตรง

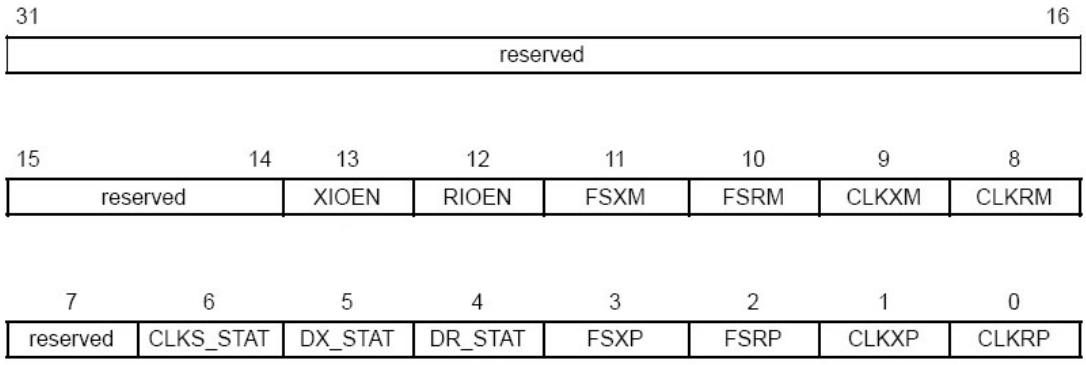
4) กำหนดบิต FSXM = 1 ในรีจิสเตอร์ PCR เพื่อกำหนดให้ขา FSX เป็นเอาต์พุต โดยค่าที่ถูกเขียนลงในบิต FSXP จะถูกส่งออกไปยังขา FSX โดยตรง

5) ขา DX จะถูกกำหนดเป็นเอาต์พุตเสมอ โดยค่าที่ถูกเขียนลงในบิต DXSTAT จะถูกส่งไปยังขา DX โดยตรง

6) กำหนดบิต CLKRM = 0 ในรีจิสเตอร์ PCR เพื่อกำหนดให้ขา CLKR เป็นอินพุต โดยสัญญาณที่ขา CLKR จะถูกอ่านมาเก็บไว้ที่บิต CLKRP

7) กำหนดบิต FSRM = 0 ในรีจิสเตอร์ PRC เพื่อกำหนดให้ขา FSR เป็นอินพุต โดยสัญญาณที่ขา FSR จะถูกอ่านมาเก็บไว้ที่บิต FSRP

8) ขา DR และ CLKS จะถูกกำหนดเป็นอินพุตเสมอ โดยสัญญาณที่ขา DR จะถูกอ่านมาเก็บไว้ที่บิต DRSTAT และสัญญาณที่ขา CLKS จะถูกอ่านมาเก็บไว้ที่บิต CLKSSTAT

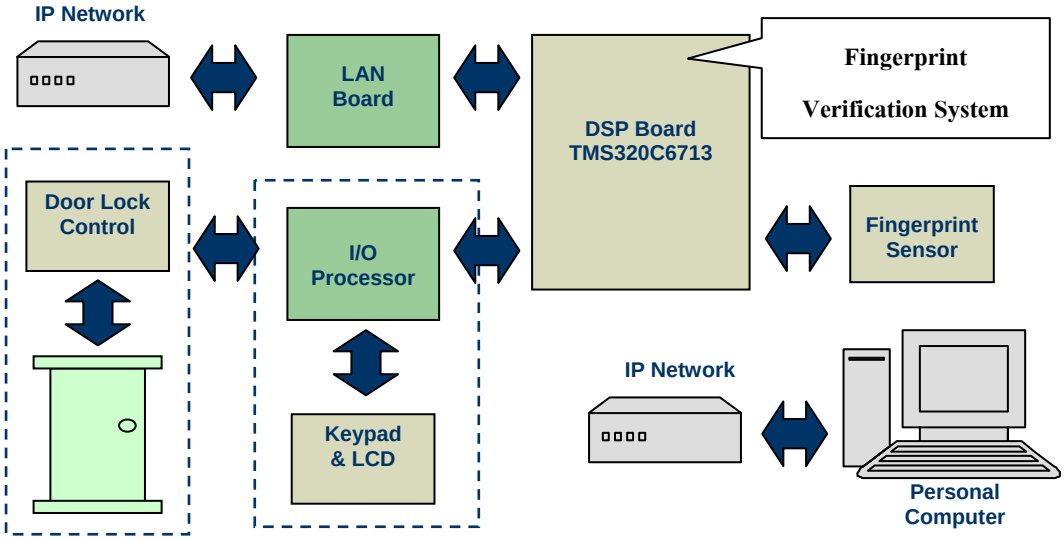


ภาพที่ 105 รีจิสเตอร์ควบคุมขาอินพุต-เอาต์พุตของมอดูล McBSP

ที่มา : Texas Instruments Inc. (2003)

การทดสอบประสิทธิภาพของระบบตรวจสอบลายนิ้วมือ

ในการทดสอบประสิทธิภาพของระบบโดยรวมเป็นขั้นตอนสุดท้ายของการพัฒนาระบบตรวจสอบลายนิ้วมือบนตัวประมวลผลสัญญาณดิจิทัล โดยนำส่วนต่าง ๆ ที่พัฒนาแล้วมาทำงานร่วมกัน ดังภาพที่ 106



ภาพที่ 106 การเชื่อมต่อของระบบตรวจสอบลายนิ้วมือ

จากภาพฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล (DSP Board) ทำหน้าที่ประมวลผลภาพลายนิ้วมือจากตัวกราดตรวจลายนิ้วมือ (Fingerprint Sensor) นอกจากนี้ยังสามารถติดต่อกับชุดควบคุมประตู (Door Lock Controller) และชุดคีย์แพดและแอลซีดี (Keypad & LCD) ผ่านทางการสื่อสารอนุกรมแบบ RS232 และติดต่อกับเครือข่าย (IP Network) ผ่านฮาร์ดแวร์เชื่อมต่อเครือข่าย (LAN Board) โดยฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลจะสามารถติดต่อกับหน่วยประมวลผลแม่ (Host Processing Unit) ผ่านทางพอร์ตอนุกรม RS232 หรือ ทางเครือข่ายได้เช่นกัน

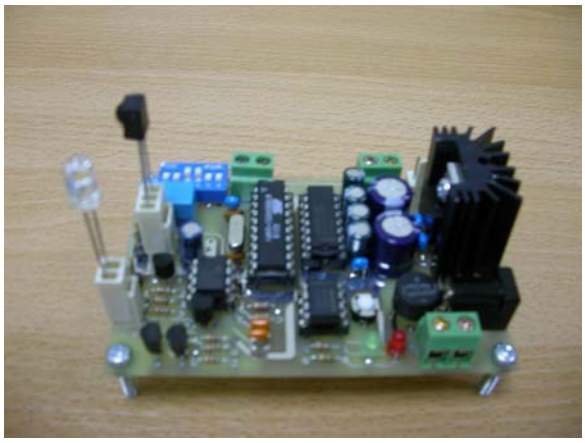
ในการทดสอบได้นำระบบตรวจสอบลายนิ้วมือ ติดตั้งไว้ที่ห้องวิจัยการประมวลสัญญาณและภาพมหาวิทยาลัยเกษตรศาสตร์ (KSIP Lab) เพื่อใช้งานทางด้านการรักษาความปลอดภัยในการเข้าออกสถานที่ โดยมีผู้ร่วมทดสอบเป็นสมาชิกภายในห้องวิจัย โดยระบบมีความสามารถในการทำงานอิสระด้วยตัวเอง (Standalone) ในกรณีไม่ได้เชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล คือ สามารถลงทะเบียนประวัติ และการลงทะเบียนลายนิ้วมือ เมื่อต้องการเพิ่มจำนวนผู้ใช้งานได้ รวมทั้ง

สามารถดูฐานข้อมูลผู้ใช้งานทั้งหมด และดูบันทึกเวลาเข้าระบบของผู้ใช้งานได้ โดยภาพที่ 107 แสดงการทดลองใช้งานระบบตรวจสอบลายนิ้วมืออัตโนมัติ และภาพที่ 108 ถึงภาพที่ 112 แสดงองค์ประกอบทางฮาร์ดแวร์ของระบบตรวจสอบลายนิ้วมือ ซึ่งประกอบด้วย ฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ฮาร์ดแวร์ควบคุมแป้นพิมพ์และแสดงผล ฮาร์ดแวร์ควบคุมการเปิดปิดประตู ฮาร์ดแวร์เชื่อมต่อเครือข่าย และฮาร์ดแวร์กราดตรวจลายนิ้วมือ ตามลำดับดังนี้

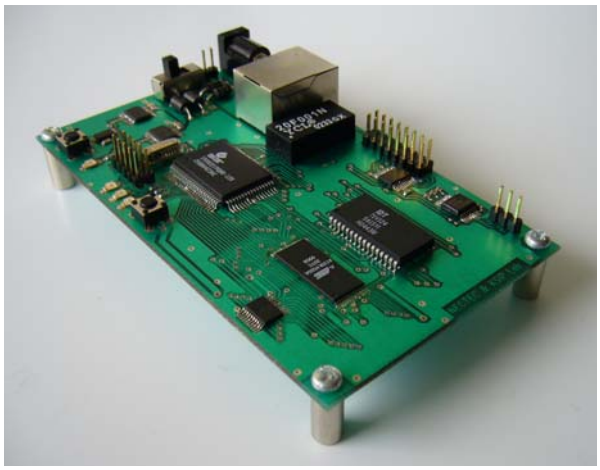
ภาพที่ 107 การใช้งานระบบตรวจสอบลายนิ้วมืออัตโนมัติ

ภาพที่ 108 ฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

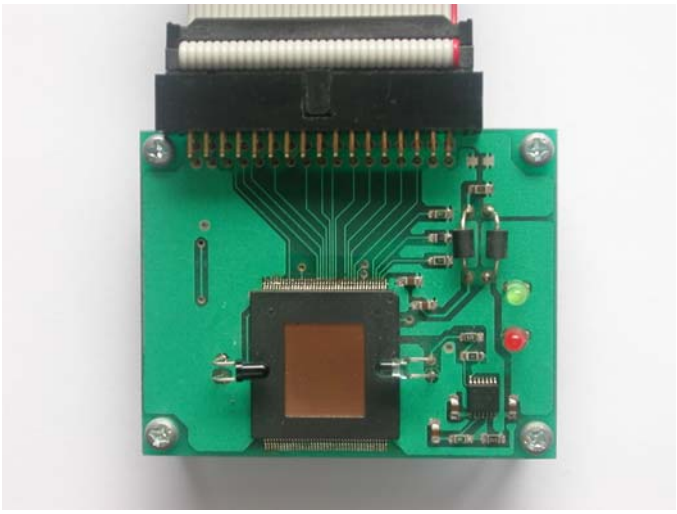
ภาพที่ 109 ฮาร์แวร์ควบคุมแป้นพิมพ์แสดงผล



ภาพที่ 110 ฮาร์ดแวร์ควบคุมการเปิดปิดประตู



ภาพที่ 111 ฮาร์ดแวร์เชื่อมต่อเครือข่าย



ภาพที่ 112 ฮาร์แวร์กราดตรวจลายนิ้วมือ

ผลและวิจารณ์ผลการทดลอง

หลังจากได้ทดลองใช้จริงโดยการติดตั้งระบบตรวจสอบลายนิ้วมือที่ห้องวิจัยการประมวลสัญญาณและภาพมหาวิทยาลัยเกษตรศาสตร์ (KSIP Lab) และพบว่าฮาร์ดแวร์ในแต่ละส่วนทำงานได้อย่างถูกต้องแล้ว ขั้นตอนต่อไปจะเป็นการวัดประสิทธิภาพของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบ โดยฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบดังกล่าวมีคุณลักษณะต่างๆ ดังตารางที่ 24

ตารางที่ 24 คุณลักษณะของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบ

คุณลักษณะของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบ	FVHP
ทีมผู้ผลิตและพัฒนา/ประเทศที่ผลิต	KSIP Lab&NECTEC/ ประเทศไทย
รุ่นของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบ	รุ่นที่ 3
รูปแบบของตัวกราดตรวจลายนิ้วมือ	Capacitive Sensor
ตัวประมวลผลบนฮาร์ดแวร์	DSP TMS320C6713 (TI)
ขนาดของเมนบอร์ด	84 มม. × 95 มม.
ขนาดของเซนเซอร์บอร์ด	53 มม. × 67 มม.
ความเร็วของตัวประมวลผล	225 MHz
ขนาดของหน่วยความจำ (SDRAM)	16 M ไบต์
ขนาดของหน่วยความจำแฟลช (Flash ROM)	2 M ไบต์
จำนวนสูงสุดในการบันทึกลายนิ้วมือ	1,000 แม่แบบ
ขนาดหน่วยความจำสูงสุดต่อหนึ่งแม่แบบ	1,500 ไบต์
แหล่งจ่ายกำลังไฟฟ้าช่วงทำงานสูงสุด	5 โวลต์ ~1 แอมป์
ความเร็วในการส่งผ่านเครือข่าย	38.4 Kbps
ชุดขั้นตอนวิธีการตรวจสอบลายนิ้วมือ (Fingerprint Algorithm Set)	FVS 1.5

จากตารางที่ 24 ได้อธิบายถึงคุณลักษณะของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบของไทยรุ่นที่ 3 (FVHPv3) ซึ่งนี้ได้รับการพัฒนามาจากเวอร์ชันแรกที่ใช้บอร์ด Digital Signal Processing Starter Kit (DSK) ของบริษัท TI และฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบรุ่นที่ 2

(FVHPv2) ซึ่งเป็นฮาร์ดแวร์ที่สร้างโดยคนไทยทั้งหมด ได้รับการออกแบบและพัฒนาจากงานวิจัยระบบสื่อสารแบบใช้สาย (Wireline Communication Section) ซึ่งเป็นหน่วยงานวิจัยของ NECTEC นำทีมโดย ดร.ราชพร เขียนประสิทธิ์ FVHPv2 ได้ถูกออกแบบให้ใช้ตัวประมวลสัญญาณดิจิทัล TMS320C6713PYP ของบริษัท TI ทำงานที่ความเร็ว 200 MHz โดยมีหน่วยความจำแฟลชขนาด 2M ไบต์ และหน่วยความจำ SDRAM ขนาด 8M ไบต์ แต่ FVHPv2 นั้นแผงวงจรมีขนาดใหญ่และใช้เวลาค่อนข้างมากในการประมวลผล

สำหรับ FVHPv3 เป็นฮาร์ดแวร์ที่สร้างขึ้นใหม่ให้มีขนาดเล็ก และเลือกใช้ตัวประมวลสัญญาณดิจิทัล TMS320C6713BGPA ของบริษัท TI ทำงานที่ความเร็ว 225 MHz ซึ่งมีบัสสำหรับติดต่อหน่วยความจำภายนอก 32 บิต มากกว่าตัวประมวลผลสัญญาณดิจิทัล TMS320C6713PYP ซึ่งมีบัสขนาด 16 บิต ทำให้ใช้เวลาประมวลผลเร็วกว่า FVHPv2 ถึงสองเท่า โดยมีหน่วยความจำแฟลชขนาด 2M ไบต์ และหน่วยความจำ SDRAM ขนาด 16M ไบต์ ทำให้สามารถบันทึกลายนิ้วมือได้สูงสุด 1000 แม่แบบ ซึ่งหน่วยความจำสูงสุดต่อหนึ่งแม่แบบมีขนาด 1500 ไบต์ ซึ่งใช้งานร่วมกับตัวกราดตรวจลายนิ้วมือแบบตัวเก็บประจุ (Capacitive Fingerprint Sensor) และฮาร์ดแวร์อื่นๆ อาทิ เช่น ฮาร์ดแวร์แป้นพิมพ์และแสดงผล (I/O Processor Board) ฮาร์ดแวร์ควบคุมการเปิดปิดประตู (Door-lock Control Board) และฮาร์ดแวร์เชื่อมต่อเครือข่าย (LAN Board) ซึ่งสามารถส่งข้อมูลผ่านเครือข่ายที่ความเร็วสูงสุด 38.4K บิตต่อวินาที (bps) โดยต้องการแหล่งจ่ายกำลังไฟฟ้าช่วงทำงานสูงสุดที่ 5 โวลต์ และใช้กระแสไฟฟ้าประมาณ 1 แอมป์

สำหรับขั้นตอนวิธีการตรวจสอบลายนิ้วมือซึ่งใช้ร่วมทดสอบกับฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบ เป็นขั้นตอนวิธีการตรวจสอบลายนิ้วมือและซอฟต์แวร์ FVS (Fingerprint Verification Software) รุ่นที่ 1.5 หรือ FVS 1.5 ที่ทางคณะวิจัย KSIP Lab ได้คิดค้นขึ้น ในการทดสอบประสิทธิภาพการทำงานของขั้นตอนวิธีตรวจสอบลายนิ้วมือ สามารถแบ่งออกเป็นสองคุณลักษณะสำคัญ คือ เวลาที่ใช้ในการตรวจสอบลายนิ้วมือ และค่าเปอร์เซนต์ความผิดพลาดในการตรวจสอบลายนิ้วมือ ดังแสดงไว้ในตารางที่ 25 และ 26 ตามลำดับ

ตารางที่ 25 ผลการทดสอบเวลาในการตรวจสอบลายนิ้วมือ

คุณลักษณะที่ต้องการทดสอบ	ผลการทดสอบ (วินาที)		
	เวลาต่ำสุด	เวลาเฉลี่ย	เวลาสูงสุด
เวลาในการลงทะเบียน	1.152	2.145	2.566
เวลาในการตรวจสอบลายนิ้วมือ	1.170	2.209	2.952
เวลาในการจับคู่เปรียบเทียบ	0.001	0.125	0.491

ตารางที่ 25 เป็นผลการทดสอบที่ได้จากการจับเวลาในการตรวจสอบลายนิ้วมือบนตัวประมวลผลสัญญาณดิจิทัล จำนวน 1000 ครั้ง แล้วทำการหาเวลาเฉลี่ย เวลาต่ำสุด และเวลาสูงสุดของการทดสอบ โดยวัดผลของ เวลาในการลงทะเบียน เวลาในการตรวจสอบลายนิ้วมือ และเวลาในการจับคู่เปรียบเทียบ ซึ่งในหัวข้อการตรวจเอกสารได้กล่าวถึงขั้นตอนการทำงานของระบบตรวจสอบลายนิ้วมือเอาไว้แล้ว และสามารถจำแนกเวลาของการทำงานออกเป็นส่วน ๆ ได้ดังนี้

เวลาในการลงทะเบียน = เวลาในการรับภาพ + เวลาในการหาลักษณะเด่น

เวลาในการตรวจสอบลายนิ้วมือ = เวลาในการรับภาพ + เวลาในการหาลักษณะเด่น + เวลาในการจับคู่เปรียบเทียบ

ในการทดสอบนี้เป็นการตรวจพิสูจน์บุคคล (Verification) ซึ่งมีการจับคู่เปรียบเทียบแบบหนึ่งต่อหนึ่ง (1:1) เท่านั้น ถ้าหากเป็นการทดสอบของการระบุตัวบุคคล (Identification) หรือการจับคู่เปรียบเทียบแบบหนึ่งต่อจำนวนหลายหน่วย (1:N) ระบบจะทำการจับคู่เปรียบเทียบข้อมูลที่ได้รับมากับข้อมูลทั้งหมดในฐานข้อมูล (จำนวน n แม่แบบ) ซึ่งจะต้องใช้เวลาในการจับคู่เปรียบเทียบเพิ่มขึ้นเป็นจำนวน n เท่า ดังนั้นการพัฒนาขั้นตอนวิธีในส่วนของการจับคู่เปรียบเทียบลายนิ้วมือ จึงเป็นส่วนสำคัญที่จะทำให้ระบบตรวจสอบลายนิ้วมือที่มีฐานข้อมูลขนาดใหญ่ สามารถทำการระบุตัวบุคคลได้ในเวลาอันรวดเร็ว

ตารางที่ 26 ผลการทดสอบความผิดพลาดในการตรวจสอบลายนิ้วมือ

คุณลักษณะที่ต้องการทดสอบ	ความผิดพลาด (%)
ค่าผิดพลาดของการยอมรับผู้ใช้งาน (FAR)	0.96
ค่าผิดพลาดของการปฏิเสธผู้ใช้งาน (FRR)	17.17
ค่าผิดพลาดที่เท่ากันของระบบ (EER)	9.24

สำหรับการวัดประสิทธิภาพของขั้นตอนวิธีการตรวจสอบลายนิ้วมือนั้น เพื่อให้มีมาตรฐานเดียวกัน จึงมีการจัดการแข่งขัน Fingerprint Verification Competition (FVC) ขึ้นซึ่งเป็นการแข่งขันในระดับนานาชาติ ซึ่งเปิดโอกาสให้บริษัท สถานศึกษา หรือนักพัฒนาอิสระ ส่งโปรแกรมปฏิบัติการ (Execution Code) เข้าร่วมแข่งขัน โดยแต่ละครั้งในการแข่งขันจะมีฐานข้อมูลลายนิ้วมือให้ทดสอบทั้งหมด 4 ฐานข้อมูลลายนิ้วมือ และเป็นฐานข้อมูลที่เก็บใหม่จากเซนเซอร์ที่เลือกมาจากบริษัทต่าง ๆ แตกต่างกัน ซึ่งแต่ละฐานข้อมูลจะมีจำนวนภาพลายนิ้วมือฐานข้อมูลละ 800 ภาพลายนิ้วมือ ดังนั้นฐานข้อมูลทั้งหมดที่ใช้ในการทดสอบมีรวมกันทั้งสิ้น 3,200 ภาพลายนิ้วมือ

ดังนั้นเพื่อให้เป็นมาตรฐานเดียวกันในการทดสอบความผิดพลาดในการตรวจสอบลายนิ้วมือของขั้นตอนวิธีการตรวจสอบลายนิ้วมือ ซึ่งทางคณะวิจัย KSIP Lab ได้คิดค้นขึ้น จึงได้นำภาพลายนิ้วมือของการแข่งขัน Fingerprint Verification Competition ในปี ค.ศ. 2004 (FVC2004) มาใช้เป็นฐานข้อมูลสำหรับการทดสอบ โดยทำการหาค่า FAR FRR และ EER ของแต่ละฐานข้อมูล แล้วนำค่าที่ได้เฉลี่ยกัน เพื่อหาความผิดพลาดรวมที่เกิดขึ้นจากทั้ง 4 ฐานข้อมูล ซึ่งค่า FAR หมายถึง ค่าของอัตราการหลุดรอดของผู้แปลกปลอมจากการตรวจจับ ค่า FRR หมายถึง ค่าของอัตราการปฏิเสธการผ่านกับผู้ใช้งานที่อยู่ในระบบ และค่า EER หมายถึง ค่าของอัตราความผิดพลาดที่เท่ากันของระบบ (ดังที่ได้กล่าวไว้แล้วในการตรวจเอกสาร) ซึ่งผลการทดสอบได้แสดงไว้ในตารางที่ 26

ตารางที่ 27 ขนาดของซอฟต์แวร์ระบบปฏิบัติการตรวจสอบลายนิ้วมือ

พื้นที่หน่วยความจำ	ขนาดที่ใช้ (ไบต์)
หน่วยความจำภายใน	145,487
หน่วยความจำภายนอก	264,370

ในตารางที่ 27 ได้แสดงถึงขนาดของซอฟต์แวร์และพื้นที่ของหน่วยความจำทั้งหมดบนตัวประมวลสัญญาณดิจิทัล ที่ต้องการใช้สำหรับการเขียนโปรแกรมระบบปฏิบัติการของระบบตรวจสอบลายนิ้วมือ

ในตารางที่ 28 แสดงการเปรียบเทียบประสิทธิภาพของฮาร์ดแวร์ตรวจสอบลายนิ้วมือต้นแบบของไทย กับผลิตภัณฑ์ที่มีขายอยู่ในปัจจุบัน โดยเลือกเปรียบเทียบกับผลิตภัณฑ์ที่เข้าแข่งขันในรายการ FVC2004 ในประเภท Light และเป็นบริษัทที่ติดอันดับ 1 ใน 10 ของผู้ร่วม

เข้าแข่งขันทั้งหมด 26 ราย การแข่งขัน FVC2004 ในประเภท Light นั้นเป็นขั้นตอนตรวจสอบลายนิ้วมือที่ถูกจำกัดพื้นที่หน่วยความจำในการประมวลผลไม่เกิน 2M ไบต์ และขนาดแม่แบบ (Template) ไม่เกิน 2K ไบต์ จำกัดระยะเวลาในการลงทะเบียนไม่เกิน 0.5 วินาที (บนเครื่อง AMD Athlon 1600+ (1.41GHz) Windows XP) และเวลาในการจับคู่เปรียบเทียบแบบหนึ่งต่อหนึ่งไม่เกิน 0.3 วินาที สังเกตได้จากคุณสมบัติดังกล่าวว่าการแข่งขันในแบบ Light เป็นซอฟต์แวร์ที่เหมาะสำหรับใช้งานบนบอร์ดประมวลผลที่มีทรัพยากรหน่วยความจำจำกัด ซึ่งเป็นที่น่าสนใจและเกี่ยวข้องกับโครงงานนี้โดยตรง ในการเปรียบเทียบได้พิจาราณาคุณสมบัติต่าง ๆ ของผลิตภัณฑ์ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ ซึ่งได้แสดงไว้ในตารางที่ 28

ตารางที่ 28 การเปรียบเทียบผลิตภัณฑ์ตรวจสอบลายนิ้วมือ

Type of Comparison	Suprema	Bioscrypt	Testech	Nitgen	IDENCOM	KSIP Lab (FVHPv2)
Country	Korea	Canada	Korea	Korea	Germany	Thailand
Rank in FVC	#1	#4	#5	#6	#9	N/A
Sensor Type	Various	E-field	Optic/Cap	Optic	E-field/ Thermal	Capacitive
Processor	ADSP-BF531	TMS320 C6711	ADSP-BF531	ARM9	TMS320 C6211	TMS320 C6713
Max Flash ROM(bytes)	8M	N/A	2M	1M	N/A	2M(Extendable)
Maximum Template	18,000	4,000(V), 200(I)	5,000	200 (2/1person)	>2,000	1,000
Average byte/Template	384	348(V), 2400 (I)	404-9,052	N/A	N/A	1,500
EER(%) FVC2004 (Light)	3.51%	4.29%	4.33%	4.86%	5.29%	9.24%
Enrollment Time (Sec)	1	1	N/A	N/A	<1	2.145
Verification Time(Sec)	1	1	<1	1.2	<1	2.209

จากการทดสอบพบว่าระบบฮาร์ดแวร์ตรวจสอบลายนิ้วมือที่พัฒนาขึ้น สามารถใช้งานได้จริง แต่เมื่อเปรียบเทียบกับผลิตภัณฑ์ที่มีขายอยู่ในท้องตลาดแล้ว ยังมีข้อด้อยในเรื่องของความเร็วและความถูกต้องของขั้นตอนวิธีตรวจสอบลายนิ้วมือ ซึ่งได้ข้อสรุปในการแก้ไขปัญหาดังนี้

การแก้ปัญหาในเรื่องความเร็วในการทำงานของฮาร์ดแวร์ประมวลผลนั้น ยังสามารถแก้ไขได้โดยการเขียนซอฟต์แวร์ให้ทำงานอย่างสั้นที่สุด (Optimization) ทั้งในส่วนของซอฟต์แวร์จัดการระบบ และซอฟต์แวร์ขั้นตอนวิธี โดยเฉพาะในส่วนซอฟแวร์ขั้นตอนวิธีการตรวจสอบลายนิ้วมือถ้าสามารถพัฒนาอัลกอริทึมในการจับคู่เปรียบเทียบลายนิ้วมือที่ไม่ซับซ้อนมาก และเหมาะสมกับโครงสร้างการทำงานของฮาร์ดแวร์จะช่วยเพิ่มความเร็วในการตรวจสอบลายนิ้วมือได้มาก

ส่วนความถูกต้องของการตรวจสอบลายนิ้วมือยังสามารถพัฒนาประสิทธิภาพของซอฟต์แวร์ขั้นตอนวิธีให้ดียิ่งขึ้นได้ ซึ่งกำลังอยู่ในช่วงพัฒนาซอฟต์แวร์ FVS1.6 ที่เร็วกว่าและดีกว่าเดิม เนื่องจากการออกแบบระบบปฏิบัติการที่ยืดหยุ่นทำให้สามารถพัฒนาทั้งฮาร์ดแวร์ประมวลผลและซอฟต์แวร์ขั้นตอนวิธีขนานกันไปได้

สำหรับการพัฒนาทางฮาร์ดแวร์ในอนาคต ยังสามารถยุบฮาร์ดแวร์แสดงผลและแป้นพิมพ์รวมเข้ากับฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล ซึ่งเตรียมพร้อมสำหรับการติดต่อแอลซีดี และคีย์บอร์ดไว้แล้ว เหลือแต่เพียงส่วนพัฒนาโปรแกรมควบคุม ซึ่งเป็นงานที่ต้องทำต่อไปเพื่อให้ฮาร์ดแวร์มีขนาดเล็กลง และสะดวกในการติดตั้งระบบในสถานที่ต่างๆ ดังแสดงในรูปที่ 113

ภาพที่ 113 ระบบฮาร์ดแวร์ตรวจสอบลายนิ้วมือรุ่นใหม่

สรุป

ในโครงงานวิทยานิพนธ์นี้ ได้ออกแบบและพัฒนาฮาร์ดแวร์สำหรับตรวจสอบลายนิ้วมือด้วยตัวประมวลผลสัญญาณดิจิทัล (DSP) และฮาร์ดแวร์ที่เป็นองค์ประกอบของระบบตรวจสอบลายนิ้วมือ เพื่อใช้งานร่วมกับการพัฒนาขั้นตอนวิธีการตรวจสอบลายนิ้วมือ ซึ่งทางคณะวิจัย KSIP Lab ได้คิดค้นขึ้น และเพื่อเป็นต้นแบบระบบตรวจสอบลายนิ้วมือในเชิงพาณิชย์ที่นำมาใช้ได้จริง โดยผลงานเป็นฝีมือของคนไทยทั้งหมด

การพัฒนาฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัลสำหรับตรวจสอบลายนิ้วมือ ได้ออกแบบโดยใช้ตัวประมวลผลสัญญาณดิจิทัล TMS320C6713 ของบริษัท Texas Instruments เนื่องจากเหมาะกับงานประมวลผลภาพ และมีขนาดเล็ก นอกจากนั้นยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น เครื่องมือ Code Composer Studio และ DSP/BIOS ทำให้ง่ายต่อการพัฒนา

การพัฒนาฮาร์ดแวร์สำหรับระบบตรวจสอบลายนิ้วมืออัตโนมัติ ประกอบด้วย ฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล เชื่อมต่ออยู่กับ ฮาร์ดแวร์แป้นพิมพ์และแสดงผล และ ฮาร์ดแวร์ควบคุมการเปิดปิดประตู ระบบสามารถทำงานเป็นเอกเทศ (Standalone) โดยอัตโนมัติ หรือทำงานเชื่อมโยงข้อมูลเป็นเครือข่าย (Network) ก็ได้ โดยมีคอมพิวเตอร์เป็นแม่ข่าย 1 เครื่อง สามารถต่อพ่วงกับเครื่องตรวจสอบลายนิ้วมือได้หลายตัว (ประมาณ 255 ตัว) โดยการสื่อสารผ่านพอร์ตอนุกรม (RS232) หรือ เครือข่ายอีเธอร์เน็ต (Ethernet) ผ่านทางฮาร์ดแวร์เชื่อมต่อเครือข่าย

จากผลการทดสอบสรุปได้ว่า ระบบตรวจสอบลายนิ้วมือสามารถใช้งานได้จริงแต่ยังไม่สมบูรณ์มากนัก ในเรื่องความเร็วในการประมวลผลของฮาร์ดแวร์ และความถูกต้องของขั้นตอนการตรวจสอบลายนิ้วมือที่ยังมีความแม่นยำไม่มากนัก ซึ่งในส่วนนี้ยังสามารถพัฒนาซอฟต์แวร์ขั้นตอนวิธีให้ดียิ่งขึ้นได้ เพื่อให้มีประสิทธิภาพทัดเทียมของต่างชาติ รวมถึงการพัฒนาฮาร์ดแวร์เพื่อเป็นผลิตภัณฑ์ทางเลือกโดยคนไทย เพื่อการแข่งขันและลดการนำเข้าของสินค้าจากต่างชาติ

เอกสารและสิ่งอ้างอิง

ธวัชชัย บานใหม่. 2546. **ระบบปฏิบัติการไบโอเมตริกบนตัวประมวลผลสัญญาณดิจิทัลผ่านทางเครือข่าย** วิทยานิพนธ์ปริญญาโท, มหาวิทยาลัยเกษตรศาสตร์

ศิริสาร เขตปิยรัตน์. 2546. **เซมิคอนดักเตอร์ อิเล็กทรอนิกส์ ฉบับที่ 241** จัดโดย บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ

Archambeault, Brunce R. 2002. **PCB Design for Real-World EMI Control.** Kluwer Academic Publishers.

Atmel, Corporation. 2003. **AT77C101B Thermal Fingerprint Sensor.** Atmel Corporation.

_____. 2005. **AT89C51RC 8-bit Microcontroller with 32K Bytes Flash.** Atmel Corporation.

_____. 2005. **AT89C4051 8-bit Microcontroller with 4K Bytes Flash.** Atmel Corporation.

BDTImark2000™. 2006. **BDTImark2000™Score for Floating-Point Packaged Processors.** Berkeley Design Technology, Inc. Available Source: http://www.bdti.com/bdtimark/BDTImark2000.htm, February, 2006

Fingerprint Verification Competition 2004. Available Source: http://bias.csr.unibo.it/fvc2004/logon.asp, February, 2006

Fujitsu Semiconductor, Inc. 2004. **MBF200 Solid State Fingerprint Sensor.** Fujitsu Semiconductor, Inc.

International Conference on Biometric Authentication 2006. Available Source: http://www4.comp.polyu.edu.hk/~icba, February, 2006

Johnson, Howard W. 1993. **High-Speed Digital A Handbook of Black Magic.** PTR Prentice-Hall, Inc.

Maltoni, D., D. Maio, A.K. Jain, and S. Pankanti. 2003. **Handbook of Fingerprint Recognition.** Springer, New York.

Maxim Integrated, Inc. 2001. **MAX3100 SPI/Microwire-Compatible UART.** Maxim Integrated, Inc.

_____. 2003. **+5V-Powered Multichannel RS-232 Driver/Receivers.** Maxim Integrated, Inc.

_____. 2004. **DS1338 I^2C RTC with 56-Byte NV RAM.** Maxim Integrated, Inc.

_____. 2004. **DS1232LP Low Power MicroMonitor Chip.** Maxim Integrated, Inc.

Micron Technology, Inc. 2001. **MT48LC4M32B2 128Mb SDRAM×32.** Micron Technology, Inc.

Microtips Technology, Inc. 2000. **MTC-20200X LCD.** Microtips Technology, Inc.

Montrose, Mark I. 1996. **Printed Circuit Board Design Techniques for EMC Compliance.** The Institute of Electrical and Electronics Engineers, Inc

Pankanti, S., S. Prabhakar, and A.K. Jain, On the Individuality of Fingerprints. 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 24. No. 8, pp. 1010-1025.

Spansion, Inc. 2003. **MBM29LV160BE Flash Memory 16M (2M×8/1M×16) Bit.** Spansion, Inc.

Spectrum Digital, Inc. 2002. **XDS510 USB JTAG Emulator Installation Guide.** literature number 506175-0001A.

Texas Instruments, Inc. 2001. **TMS320C6000 Peripherals Reference Guide.** literature number SPRU190D.

_____. 2001. **TMS320C6000 System Clock Circuit Example.** literature number SPRA430A.

_____. 2002. **Dual-Output Low-Dropout Voltage Regulators with Power up Sequencing for Split Voltage DSP System.** literature number SLVS285A.

_____. 2002. **FlashBurn: A DSK Flash Memory Programmer.** literature number SPRA804A.

_____. 2002. **How to Begin Development Today with the TMS320C6713 Floating-point DSP.** literature number SPRU809A.

_____. 2002. **TMS320C6000 Board Design: Considerations for Debug.** literature number SPRA523A.

_____. 2002. **TMS320C6000 EMIF to External Flash Memory.** literature number SPRU568A.

_____. 2002. **TMS320 DSP/BIOS User's Guide.** literature number SPRU423B.

_____. 2003. **DSP for Smart Biometric Solutions.** literature number SPRA894A.

_____. 2003. **SN74LVTH125 3.3-V ABT Quadruple Bus Buffer with 3-State Output.** literature number SCBS703I.

_____. 2003. **SN74LVTH162245 3.3-V ABT 16-Bit Bus Transceivers with 3-State Output.** literature number SCBS260O.

_____. 2003. **SN74LVTH2245 3.3-V ABT Octal Bus Transceivers with 3-State Output.** literature number SCBS707E.

_____. 2003. **TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide.** literature number SPRU266A.

_____. 2003. **Using IBIS Models for Timing Anaysis.** literature number SPRA839A.

_____. 2004. **Creating a Second-Level Bootloader for FLASH Bootloading on TMS320C600 Platform With Code Composer Studio 2.2.** literature number SPRA999.

_____. 2004. **TMS320C6000 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide.** literature number SPRU580C.

_____. 2004. **TMS320C6000 DSP Software-Programmable Phase-Locked Loop (PLL) Controller Reference Guide.** literature number SPRU223B.

_____. 2004. **TMS320C6000 EMIF-to-External SDRAM Interface.** literature number SPRA433D.

_____. 2005. **High-Speed DSP System Design Reference Guide.** literature number SPRU889.

_____. 2005. **TLC555 LinCMOS Timer.** literature number SLFS043F.

_____. 2005. **TMS320C6713 Floating-point Digital Signal Processor.** literature number SPRS186L.

_____. 2006. **TMS320C6000 DSP Platform spec card.** literature number SPRT248F.

Ultra-Scan, Inc. 2006. **Ultrasonic Fingerprinting System.** Ultra-Scan True Identity Biometric Inc.

Vishay Telefunken, Inc. 2001. **TSOP48 Photo Modules for PCM Remote Control System.** Vishay Telefunken Inc.

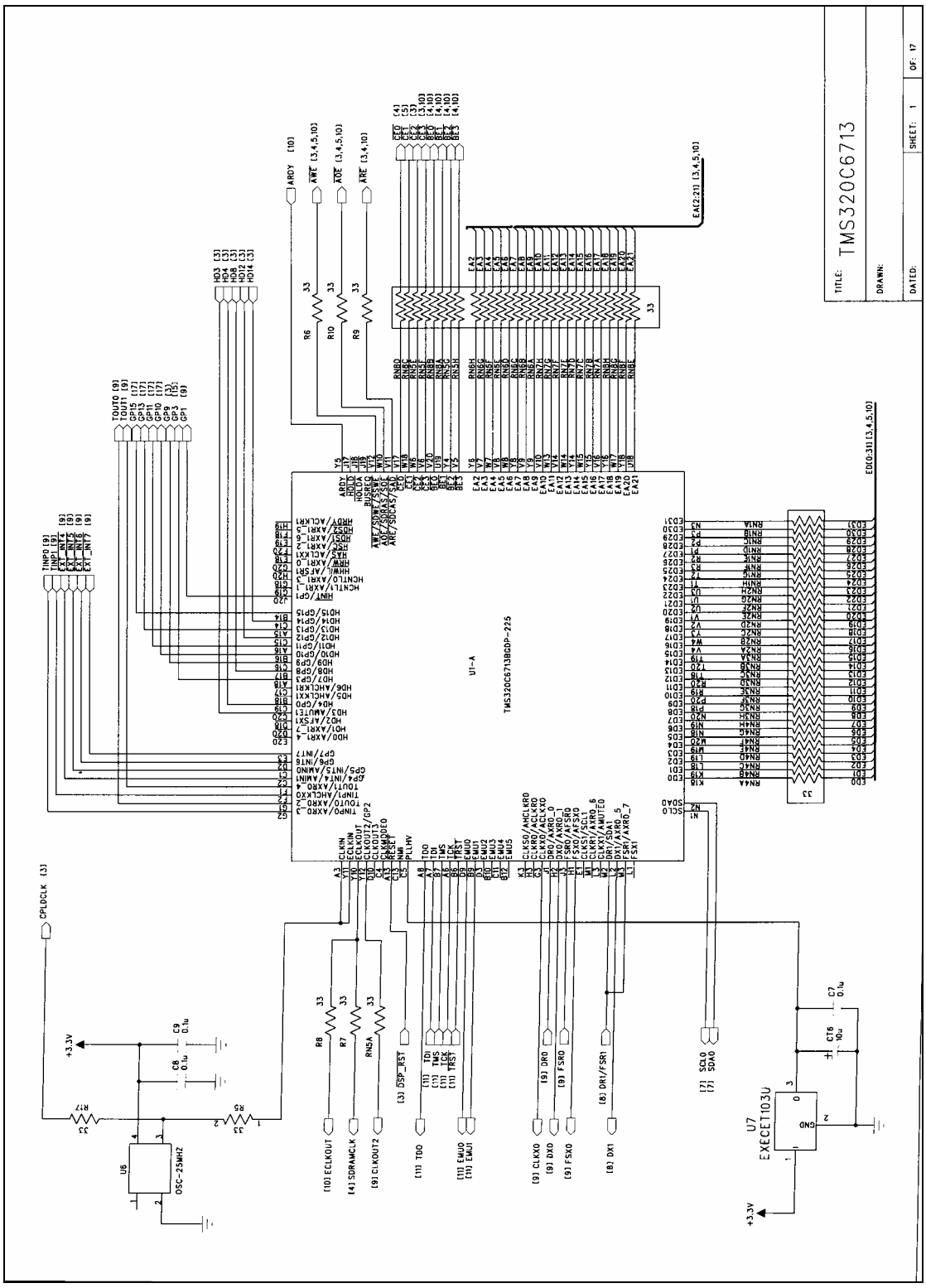
Xilinx, Inc. 2004. **XC95144XL High Performance CPLD.** Xilinx, Inc.

Y. Matrix Co., Ltd. 2006. **Fingerscan Time Attendant, Accress Control Standalone.** Y. Matrix Co., Ltd. Available Source: http://www.ymatrix.com/ProductList.html, February, 2006

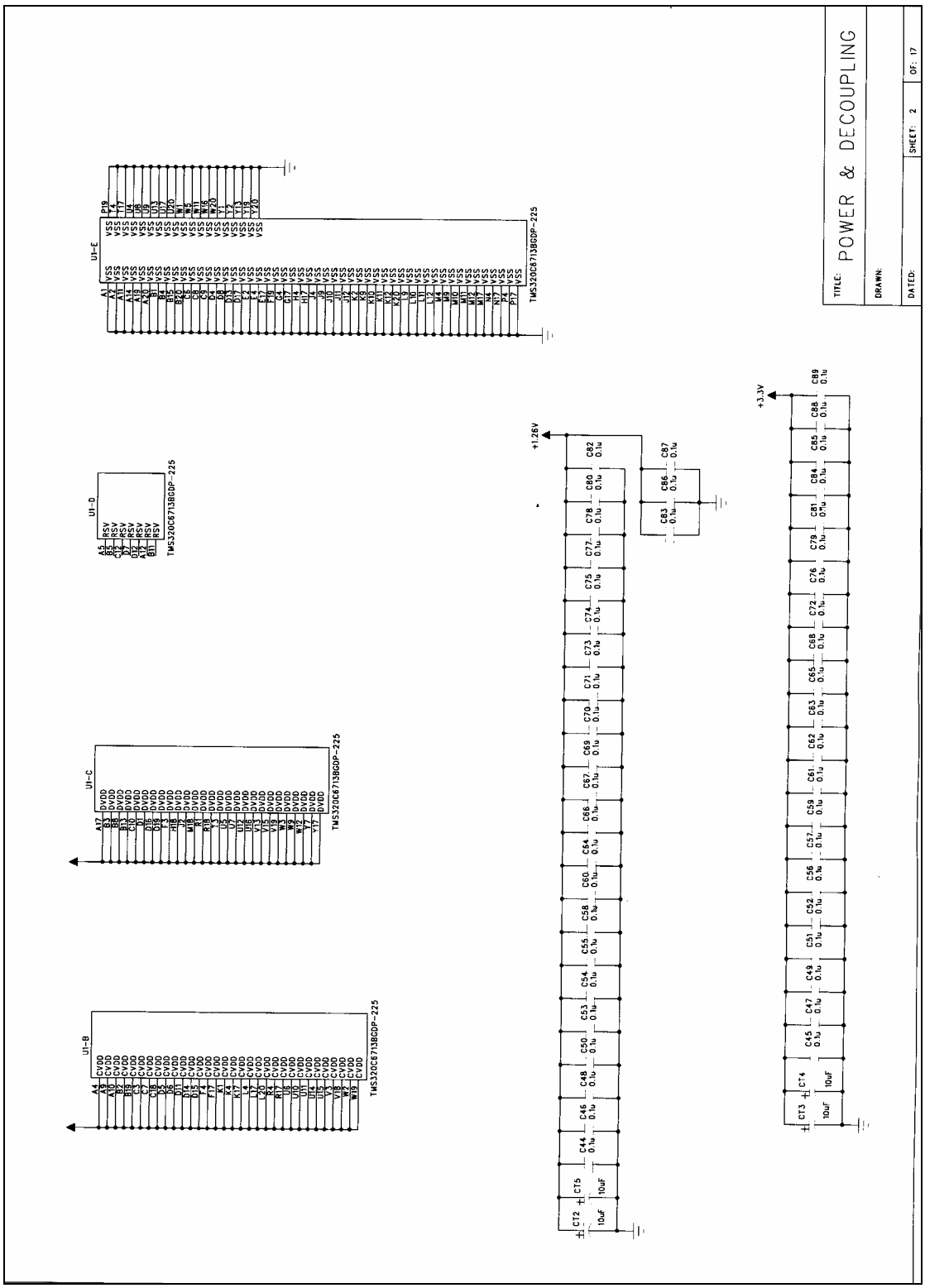
ภาคผนวก

ภาคผนวก ก

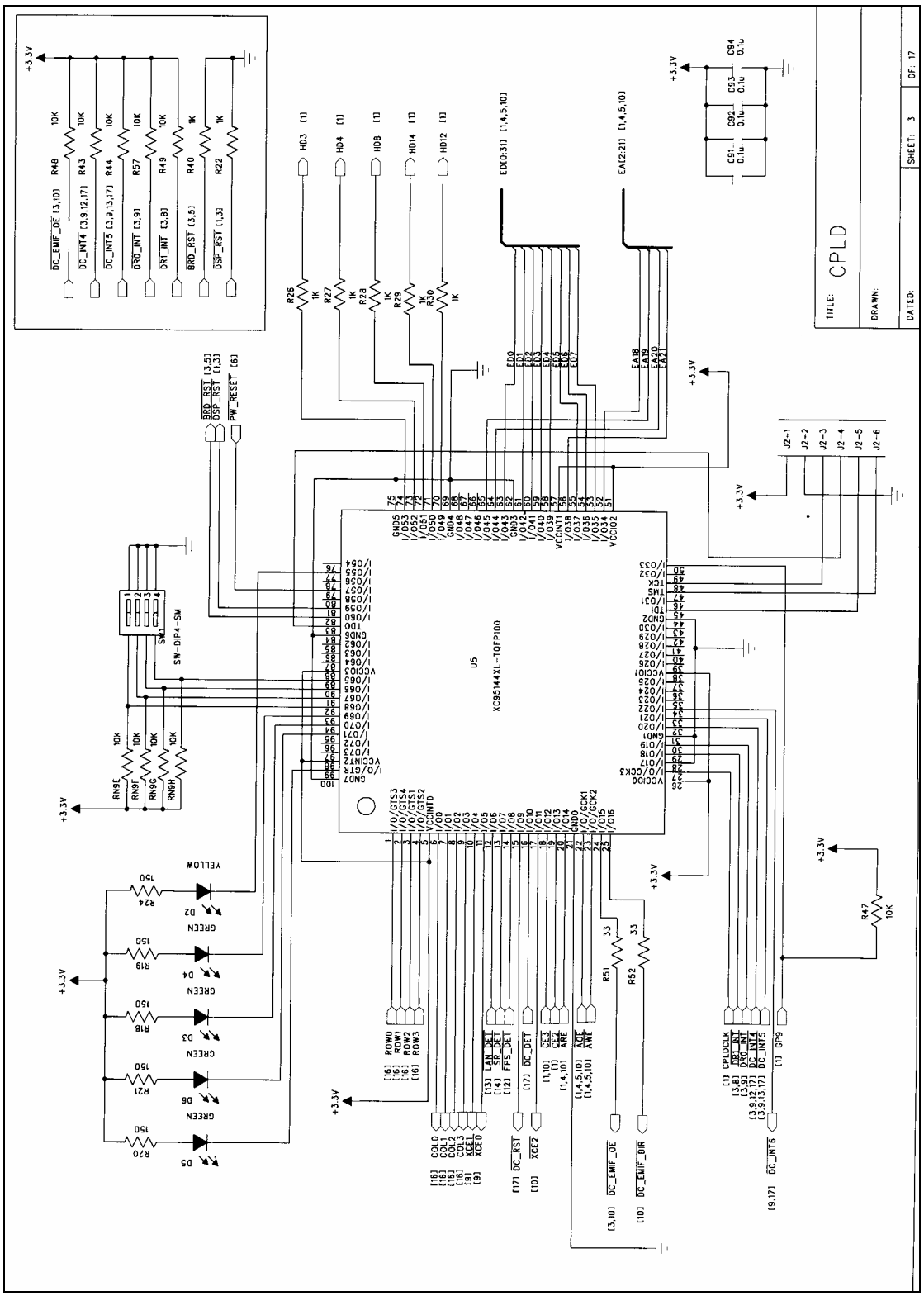
แผนภาพวงจรฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



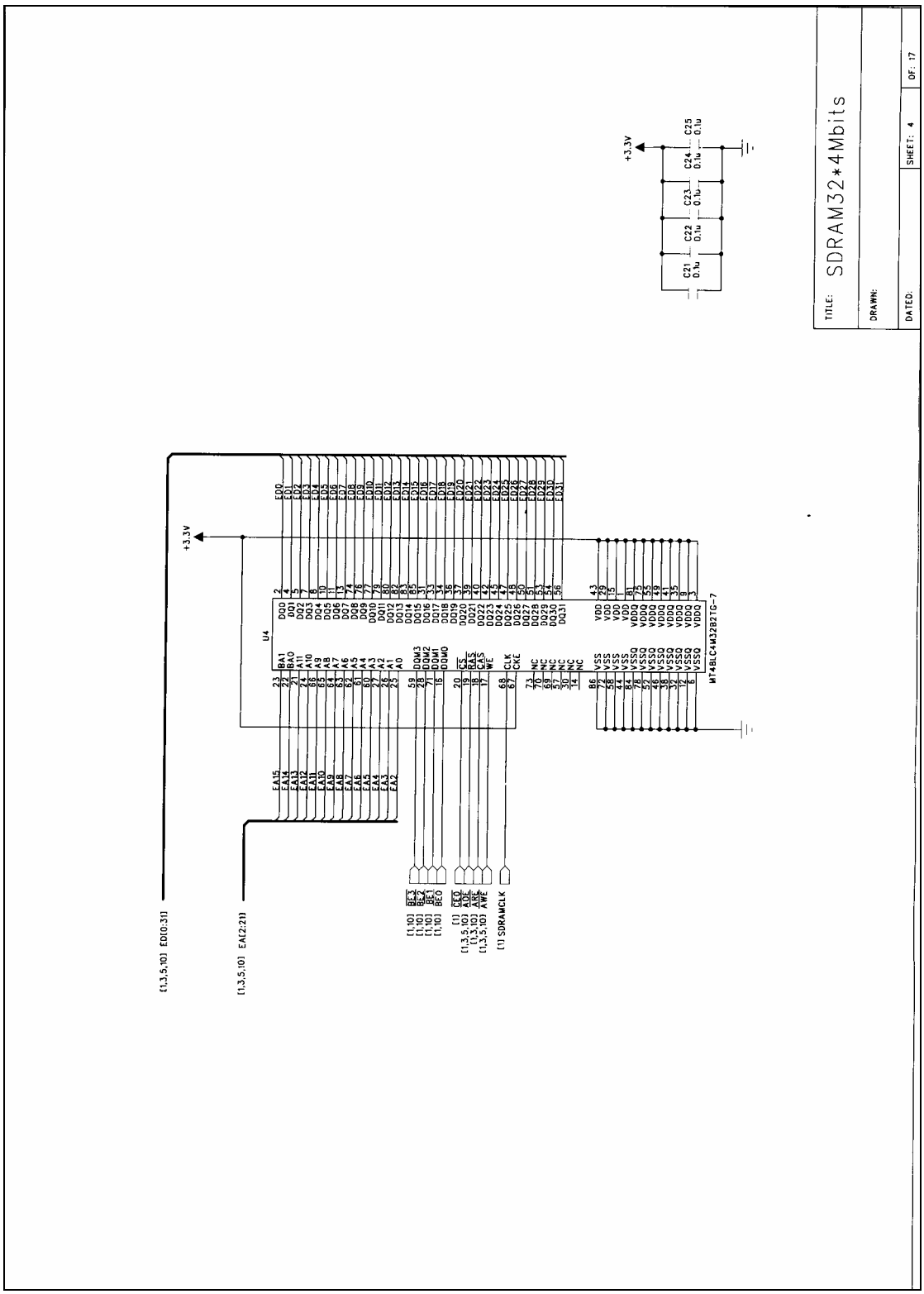
ภาพผนวกที่ 1 ตัวประมวลผล TMS320C6713 DSP



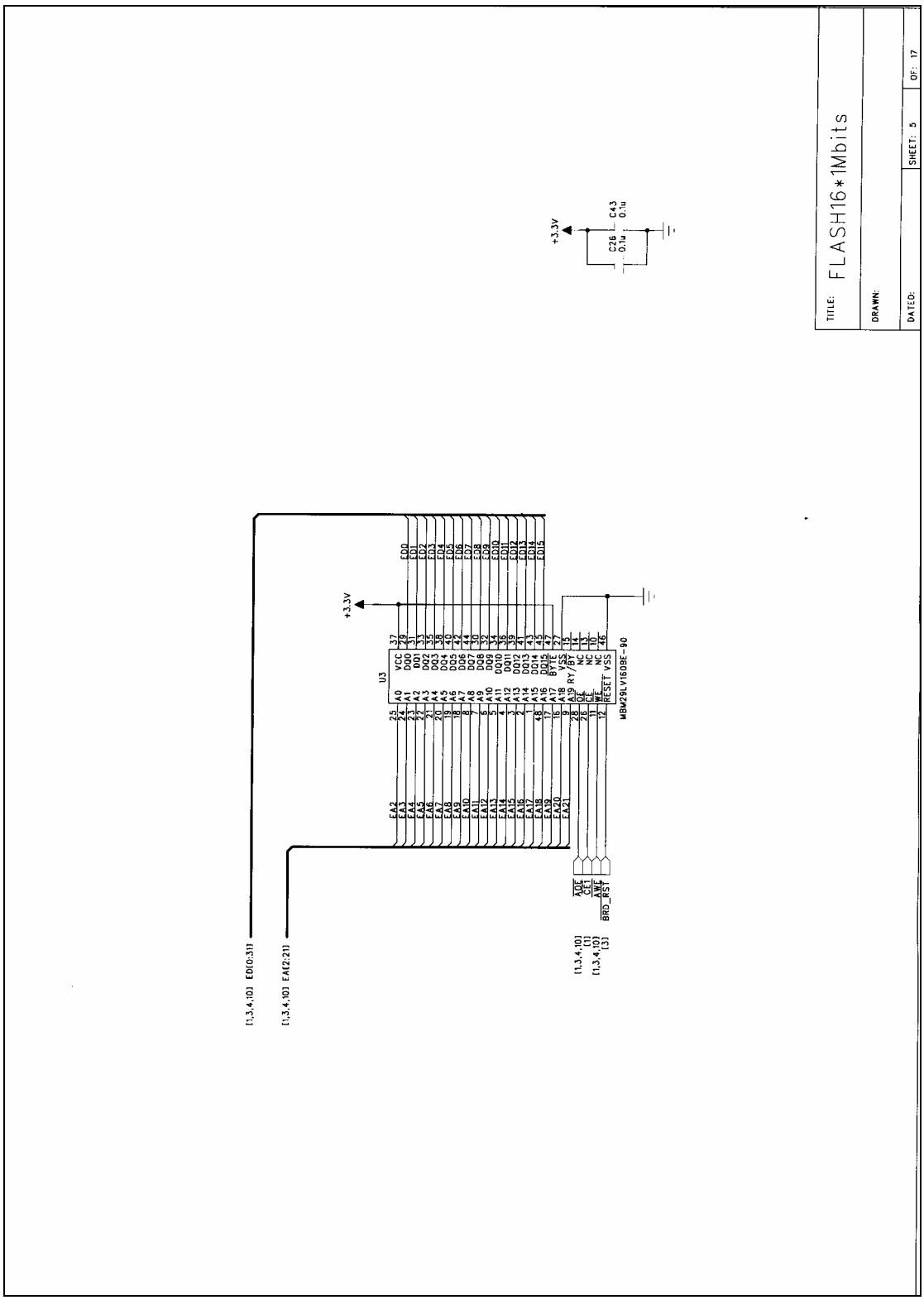
ภาพผนวกที่ 2 วงจรกำลังไฟฟ้าและการดีคับปลิง



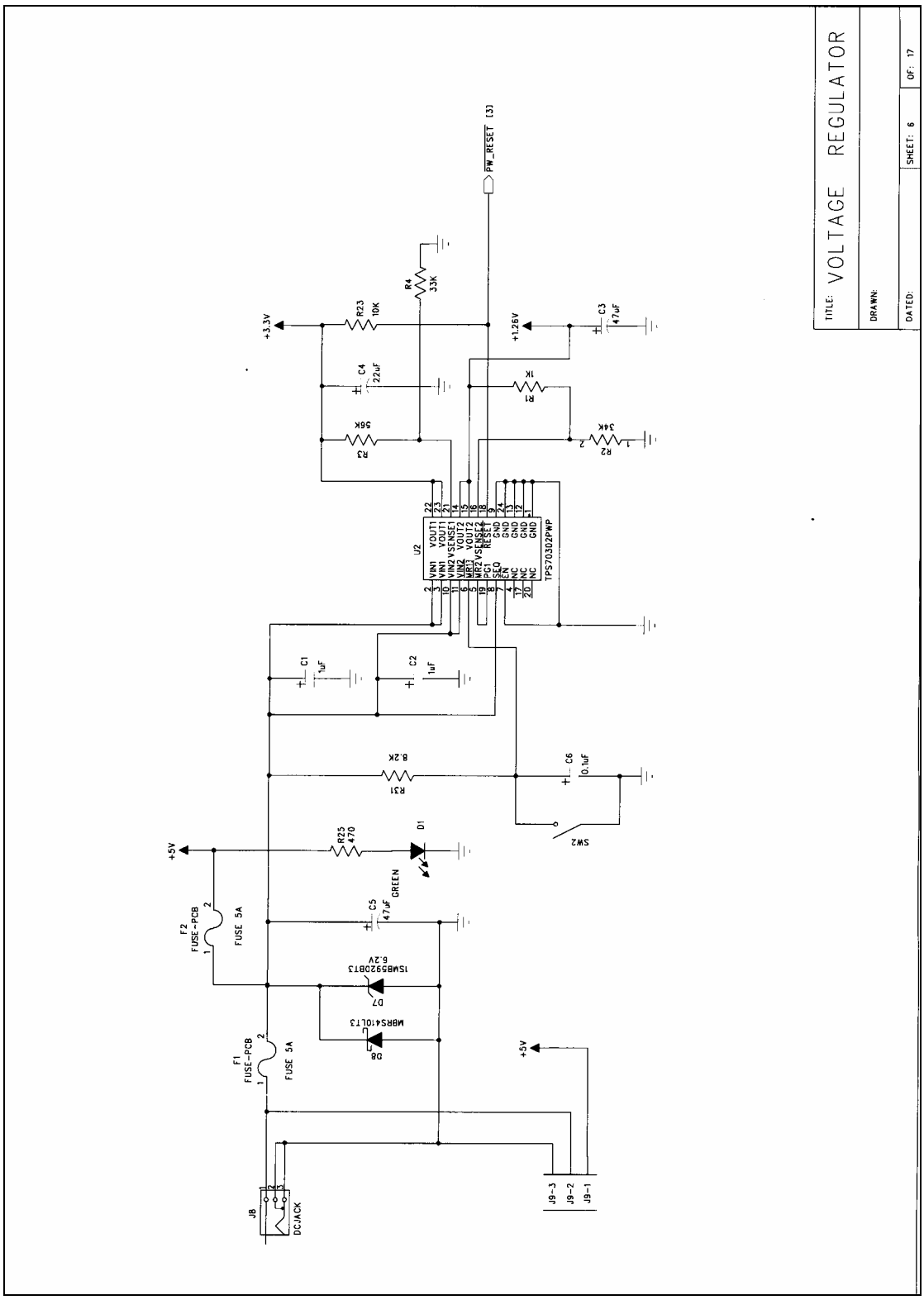
<u>ภาพผนวกที่ 3</u> โปรแกรมลอจิก XC95144XL CPLD



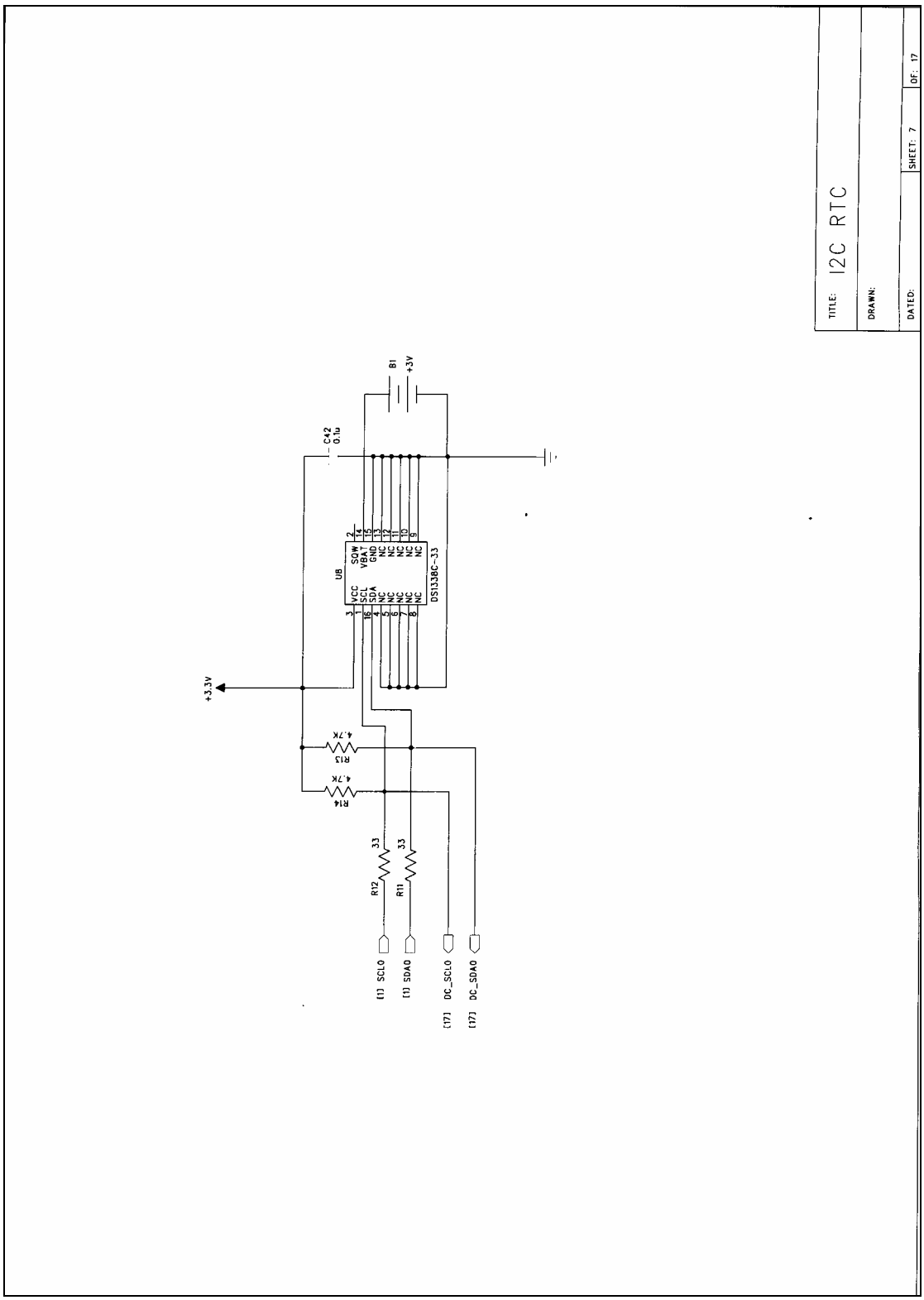
ภาพผนวกที่ 4 หน่วยความจำแรม SDRAM 32×4 Mbits



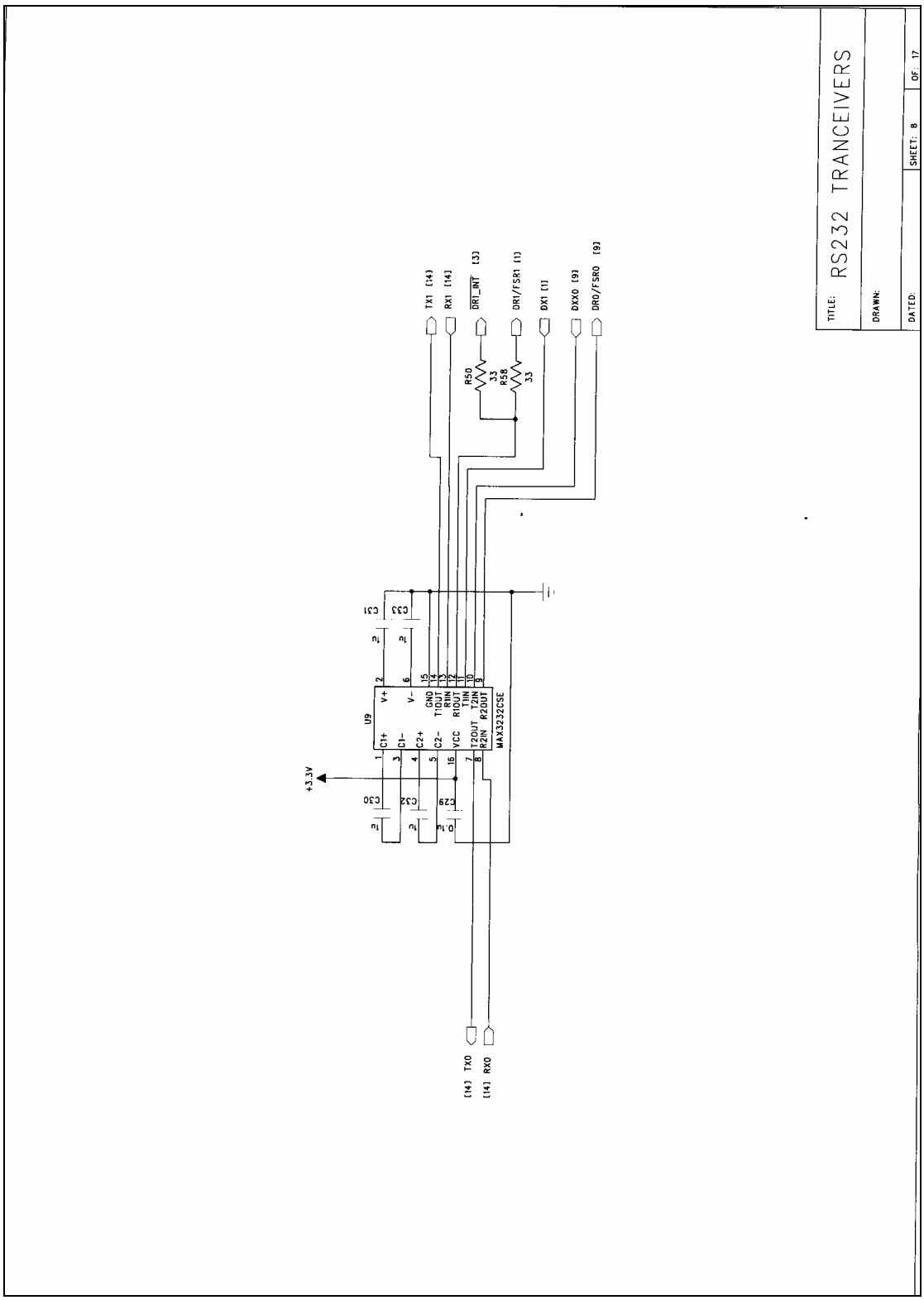
ภาพผนวกที่ 5 หน่วยความจำรอม FLASH 16×1 Mbits



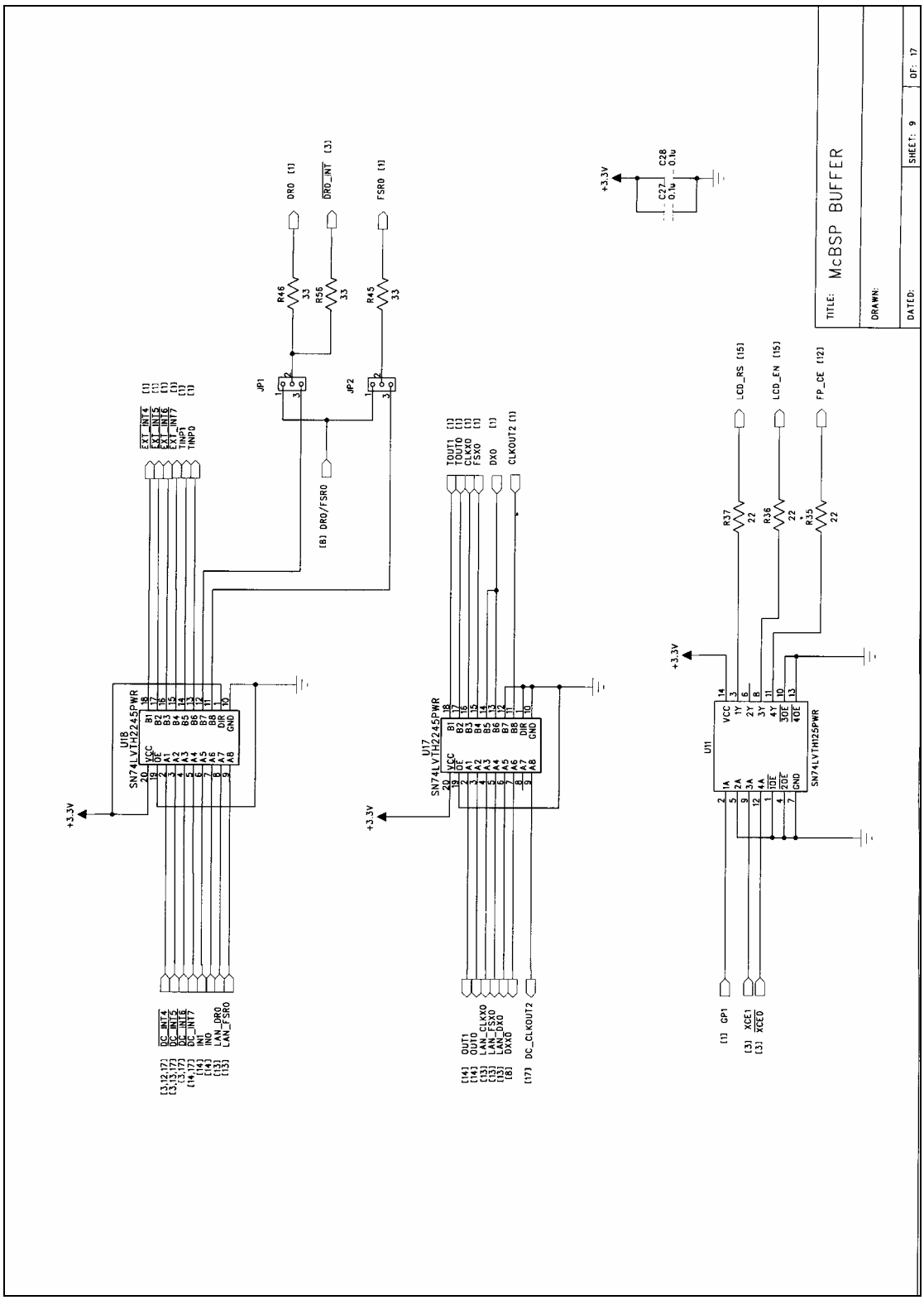
<u>ภาพผนวกที่ 6</u> ตัวคุมค่าแรงดันไฟฟ้า TPS70302



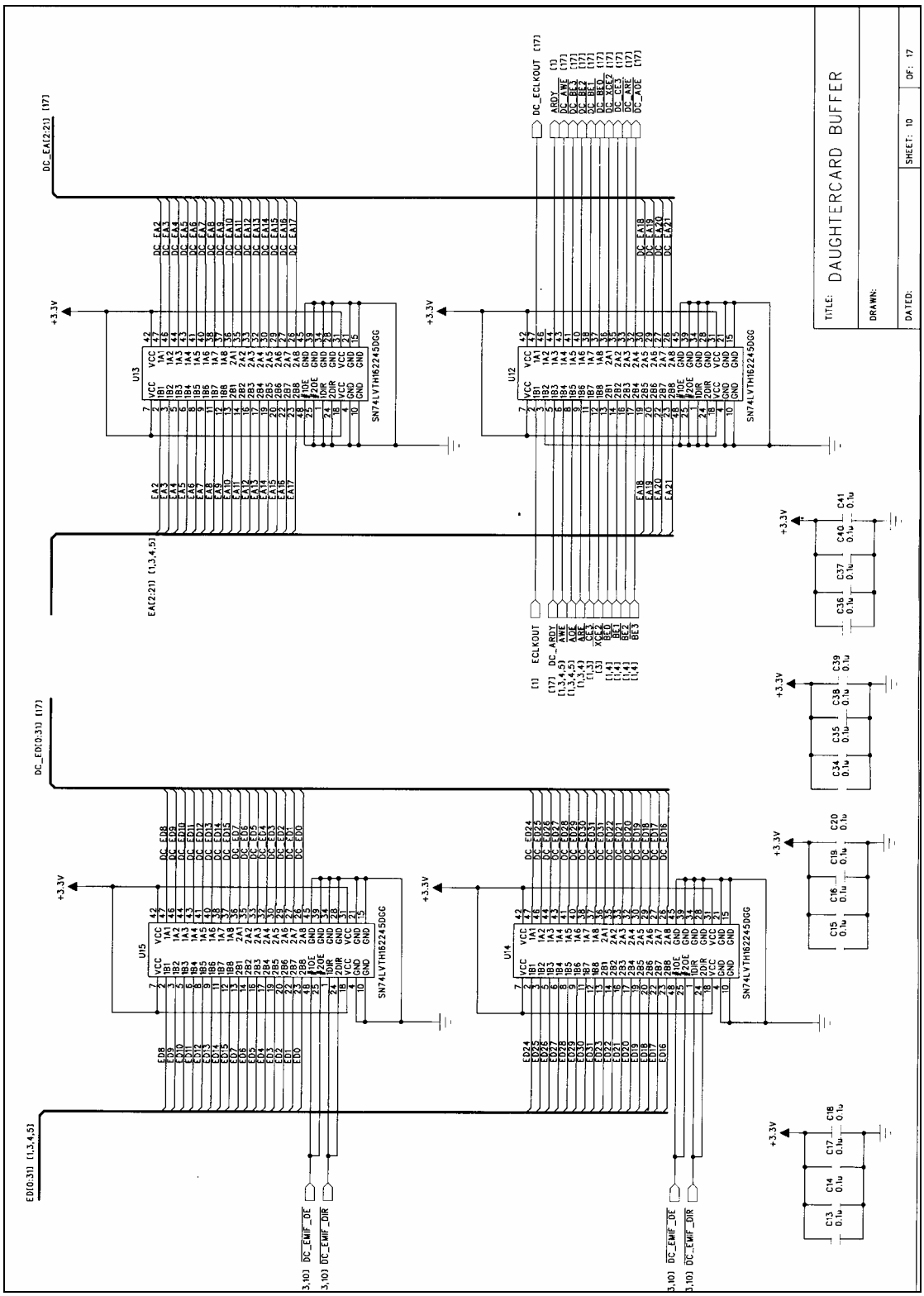
ภาพผนวกที่ 7 วงจรรวมฐานเวลาจริง DS1338C



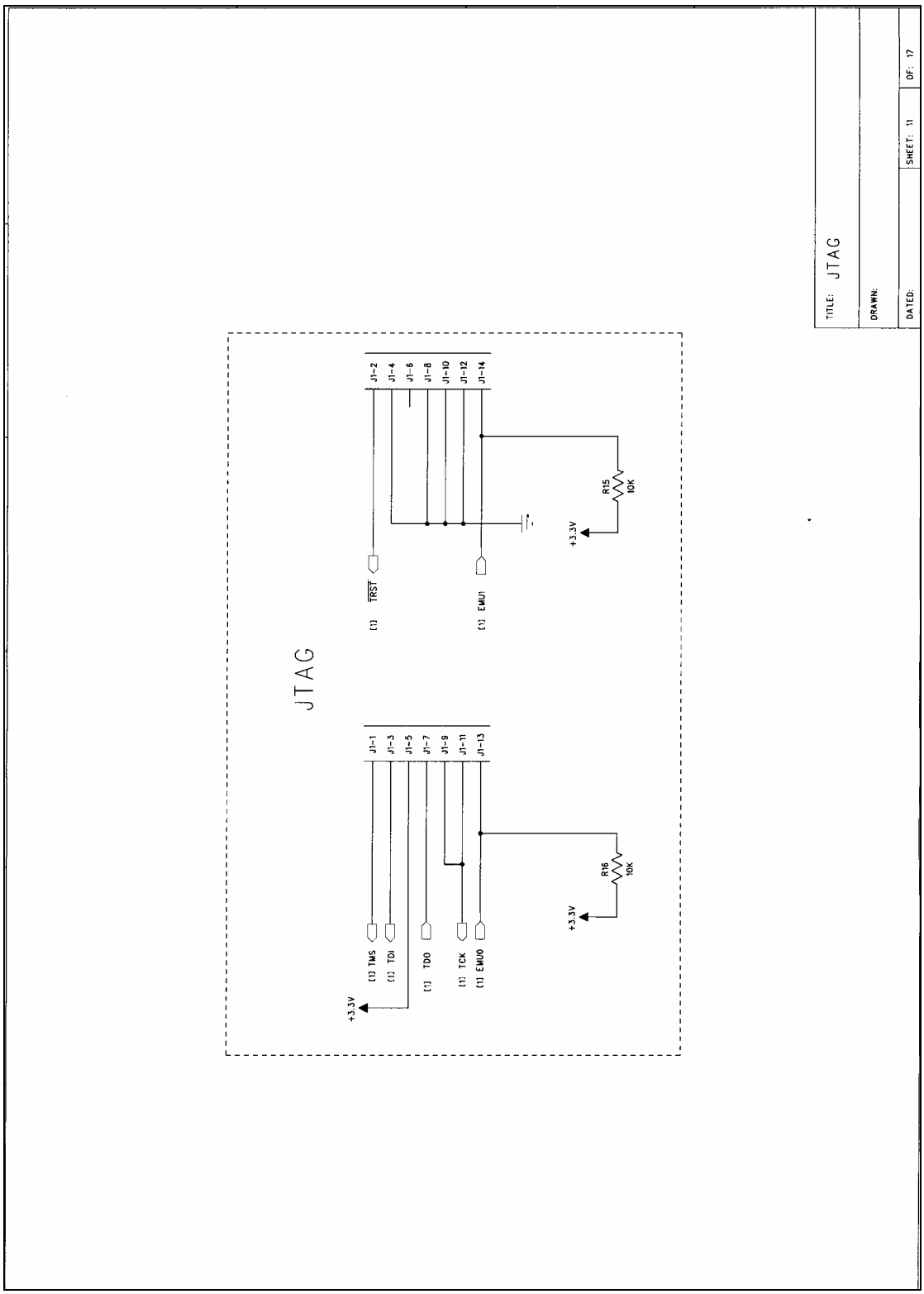
ภาพผนวกที่ 8 ไอซีรับส่งสัญญาณอนุกรม MAX3232



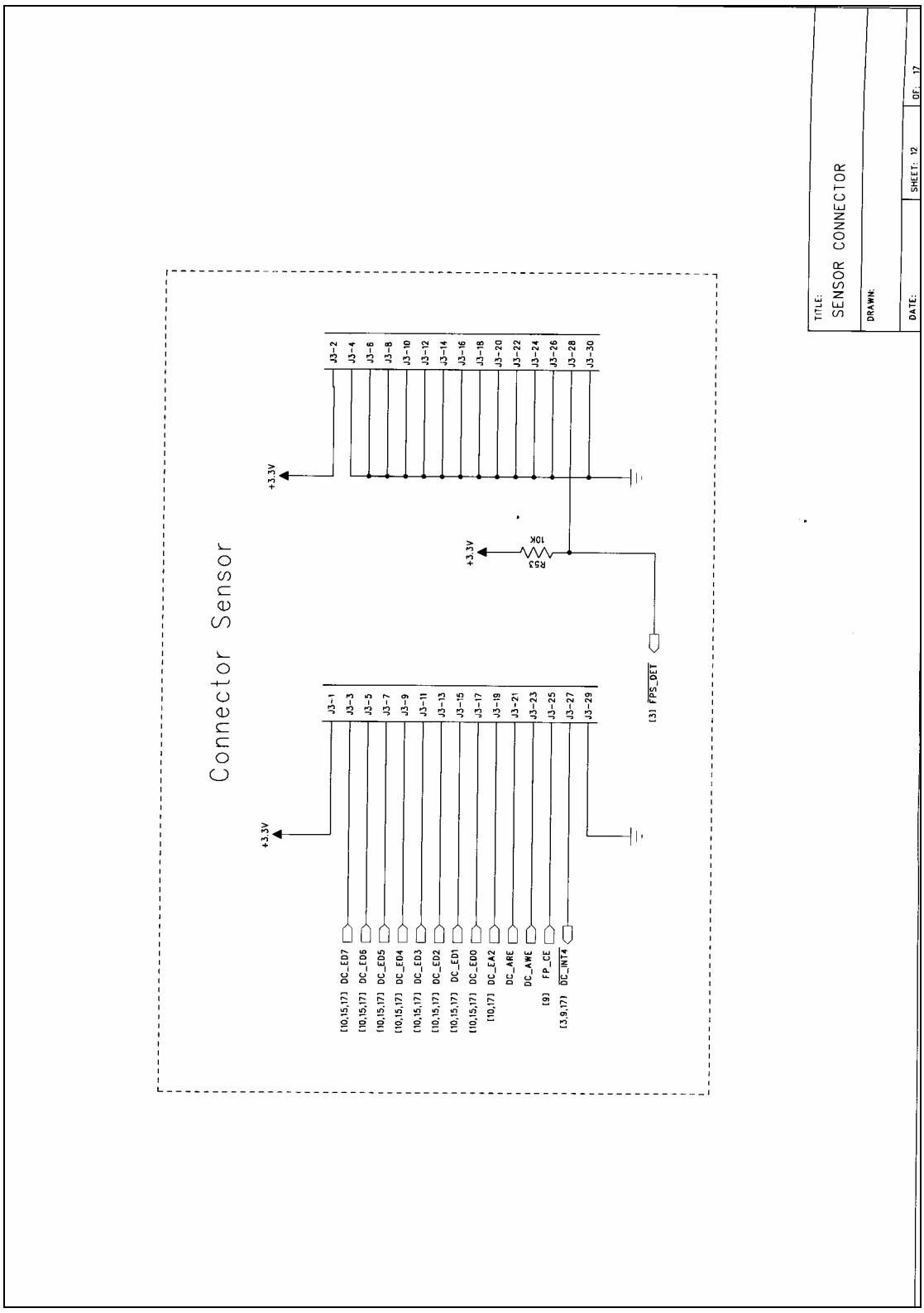
<u>ภาพผนวกที่ 9</u> ตัวรับส่งสัญญาณอนุกรมสำหรับมอดูล McBSP



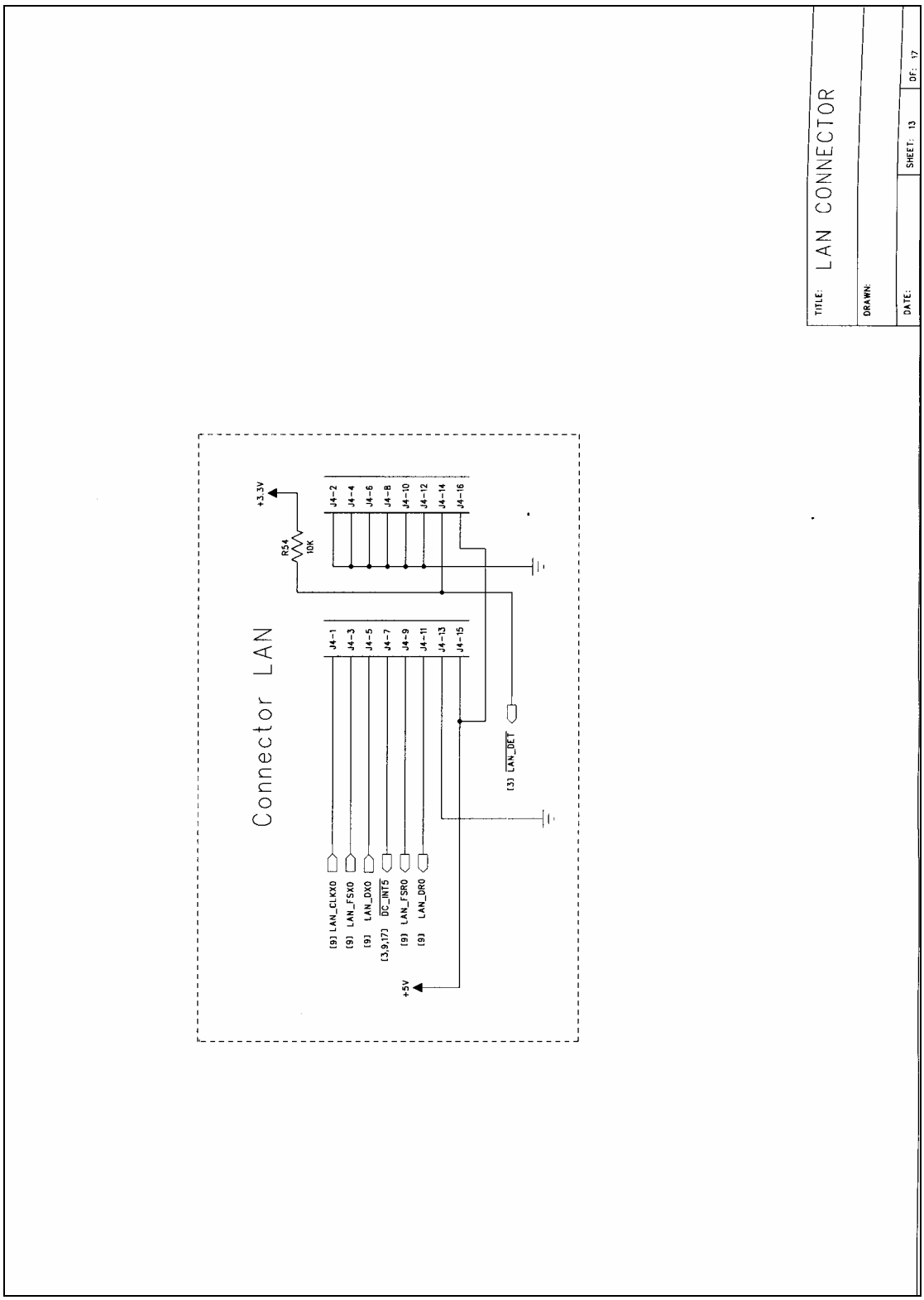
ภาพผนวกที่ 10 ตัวรับส่งบัสขนานสำหรับมอดูล EMIF



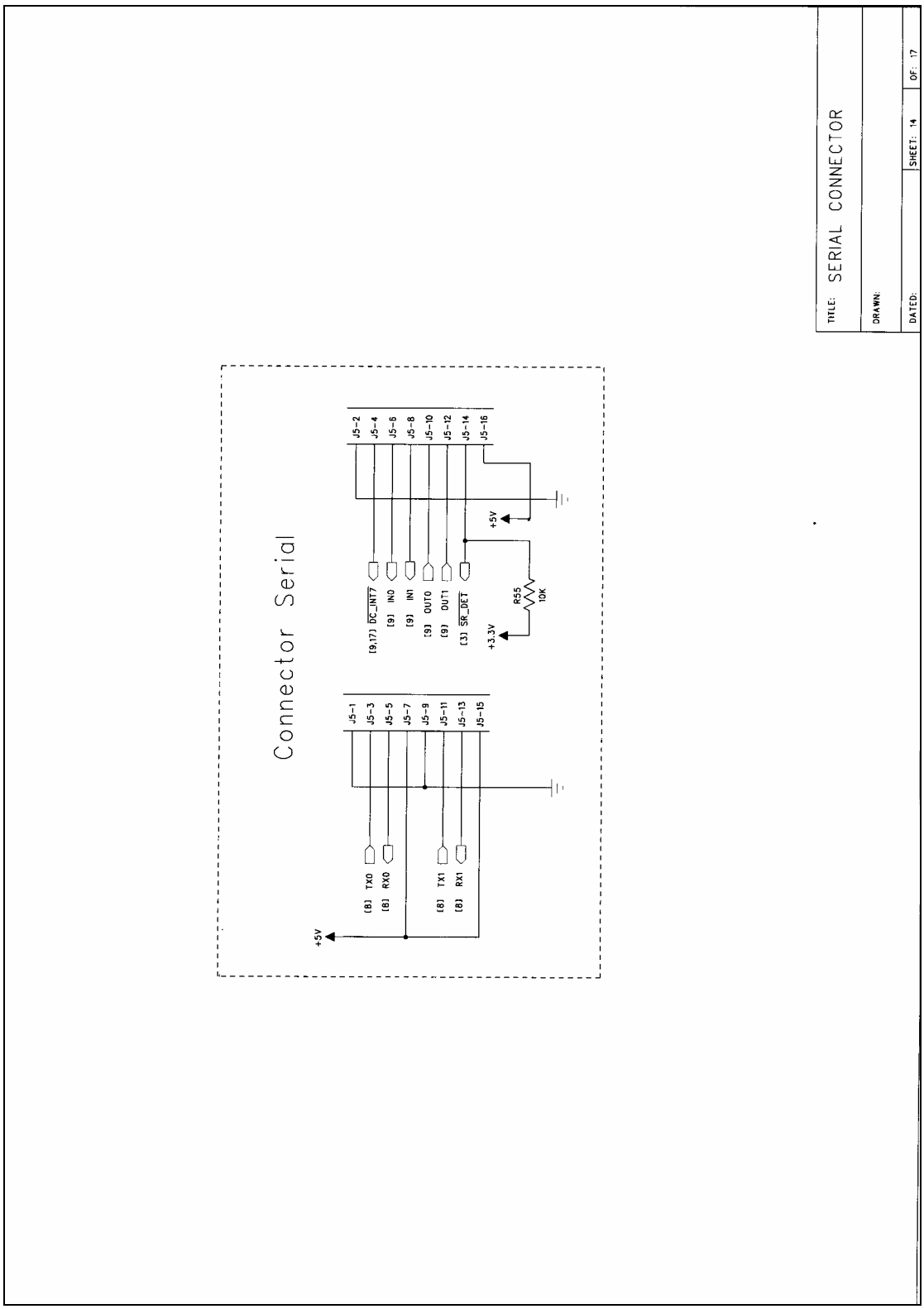
ภาพผนวกที่ 11 พอร์ตเชื่อมต่อ JTAG สำหรับการแก้จุดบกพร่อง (Debug) DSP



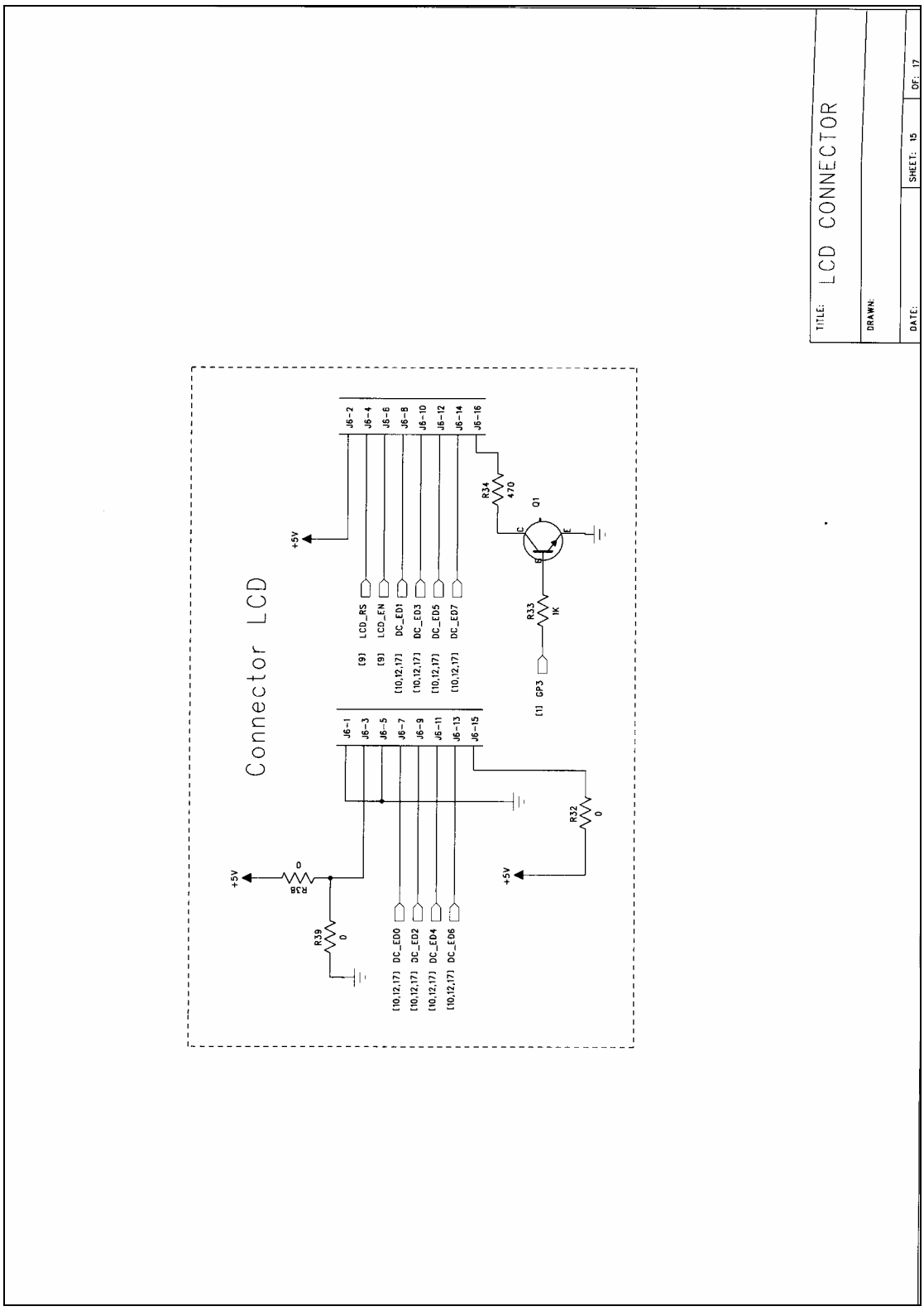
<u>ภาพผนวกที่ 12</u> พอร์ตเชื่อมต่อฮาร์ดแวร์กราดตรวจลายนิ้วมือ



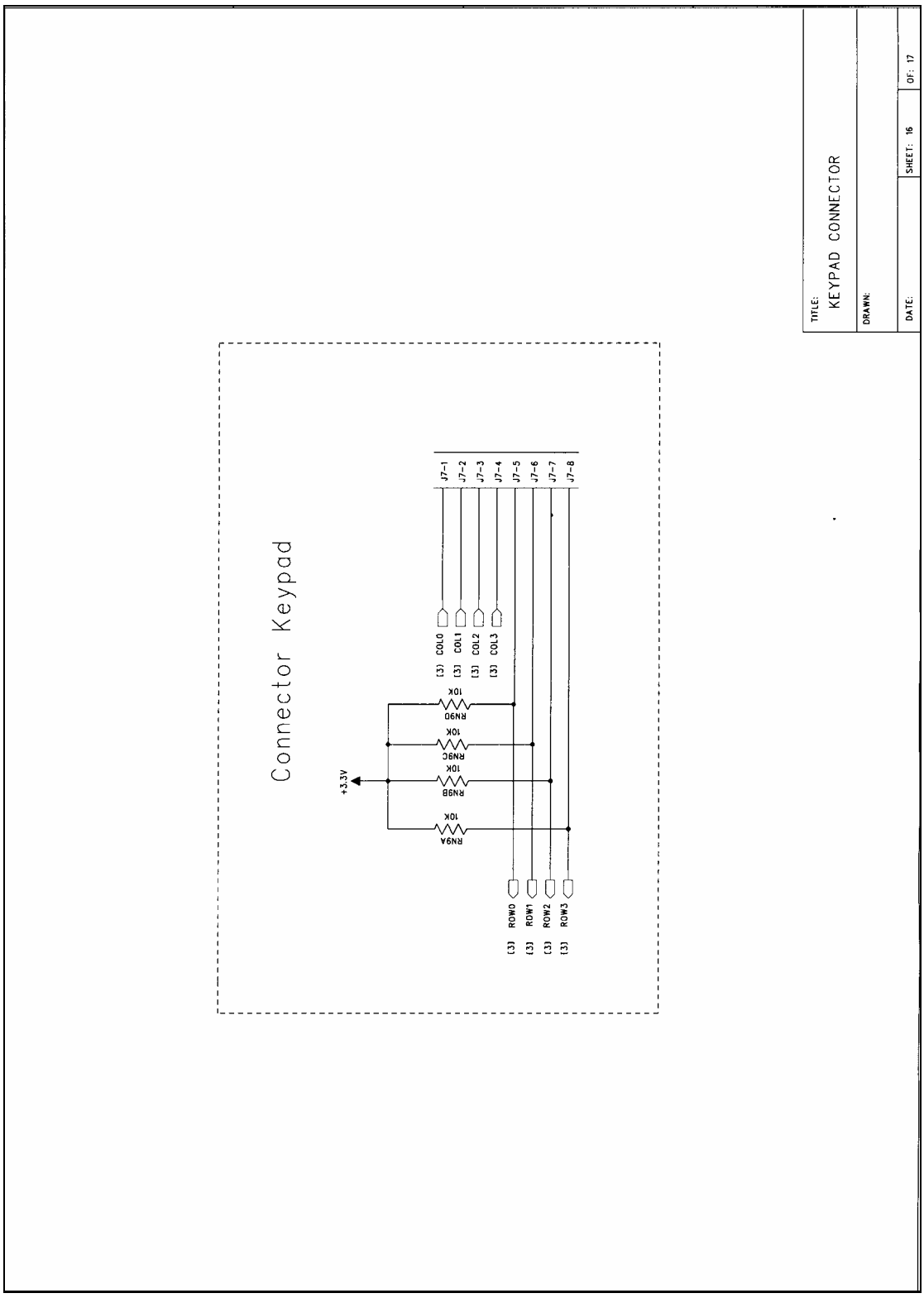
ภาพผนวกที่ 13 พอร์ตเชื่อมต่อฮาร์ดแวร์เชื่อมต่อเครือข่าย LAN



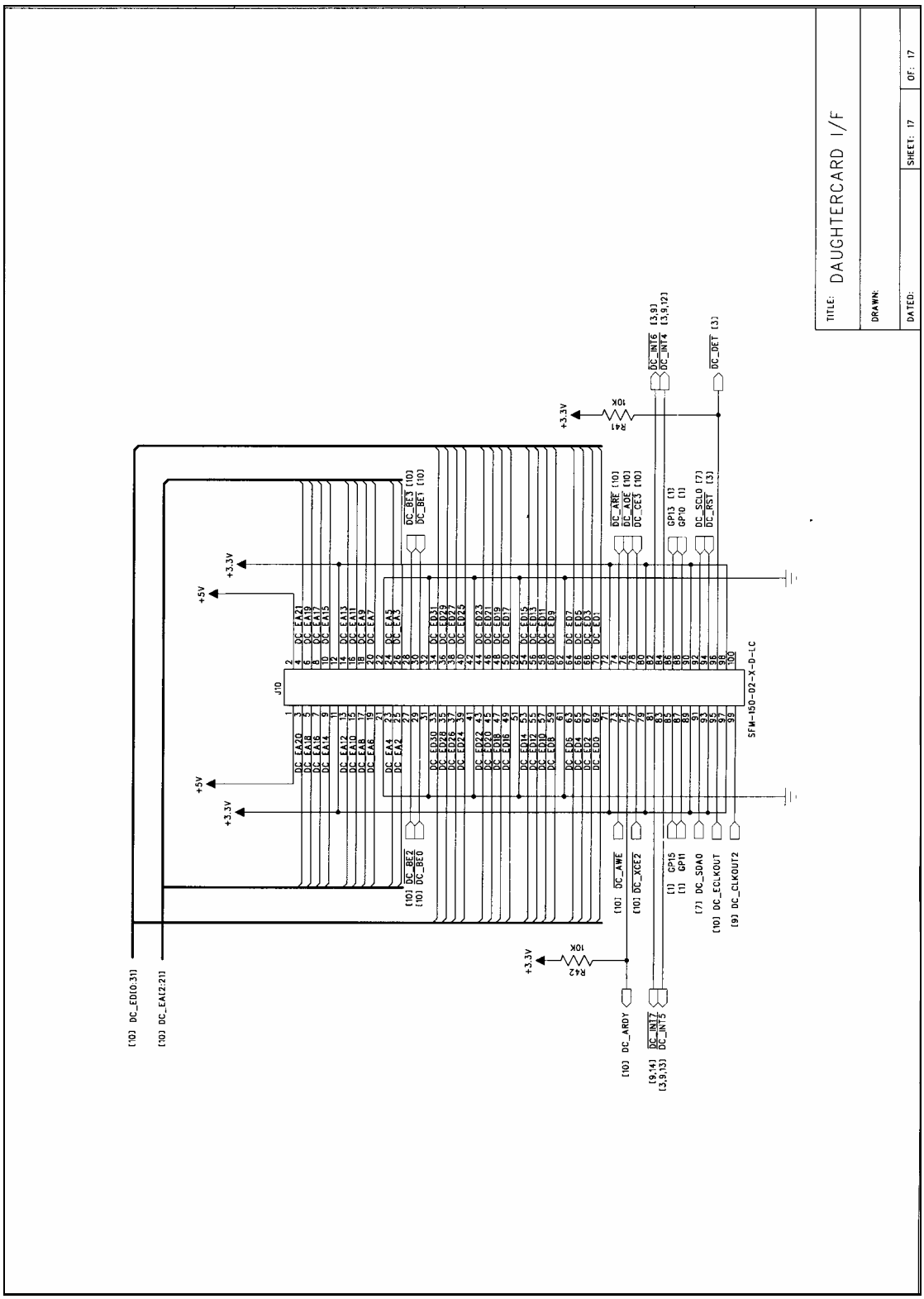
ภาพผนวกที่ 14 พอร์ตเชื่อมต่ออุปกรณ์แบบอนุกรม RS232



ภาพผนวกที่ 15 พอร์ตเชื่อมต่อสำหรับแอลซีดี (LCD)



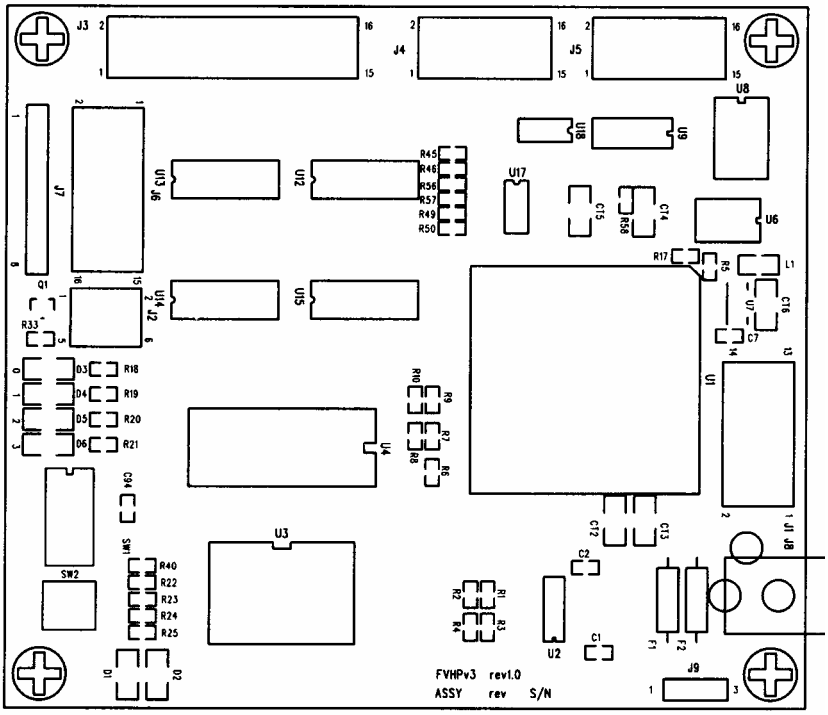
<u>ภาพผนวกที่ 16</u> พอร์ตเชื่อมสำหรับแผงแป้นพิมพ์ (Keypad)



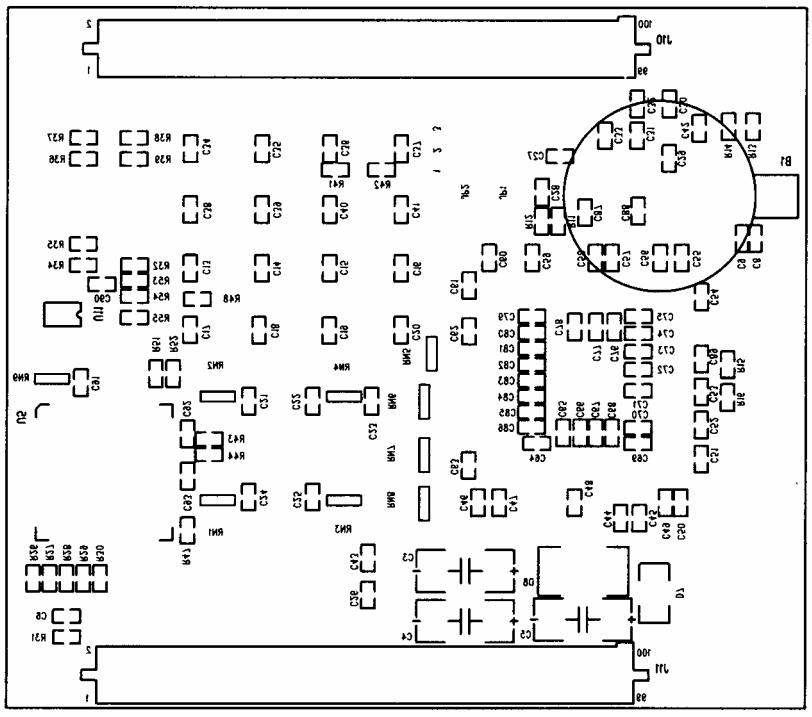
ภาพผนวกที่ 17 พอร์ตเชื่อมต่อสำหรับการขยายแผงวงจร

ภาคผนวก ข

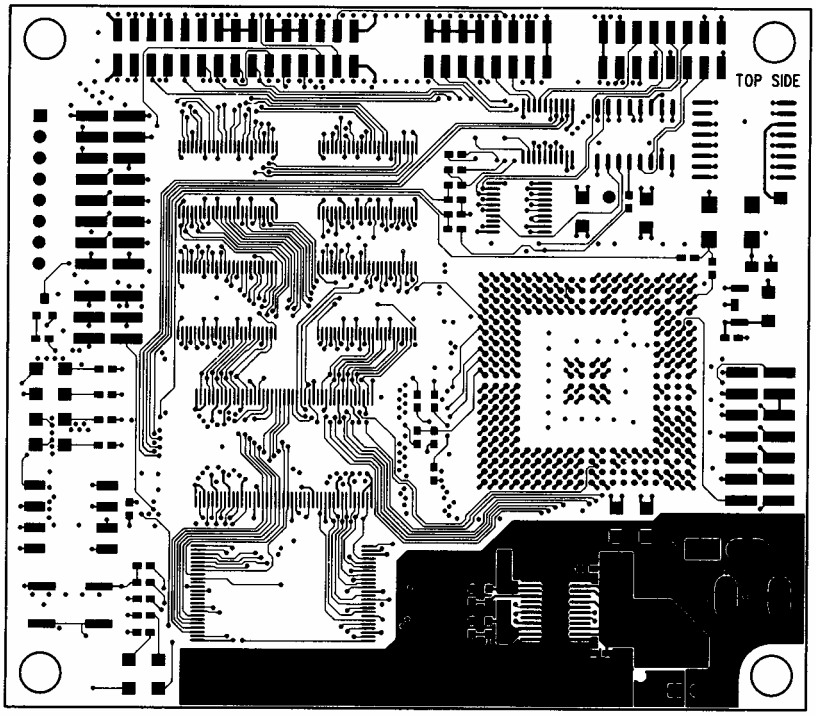
ลายวงจรฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



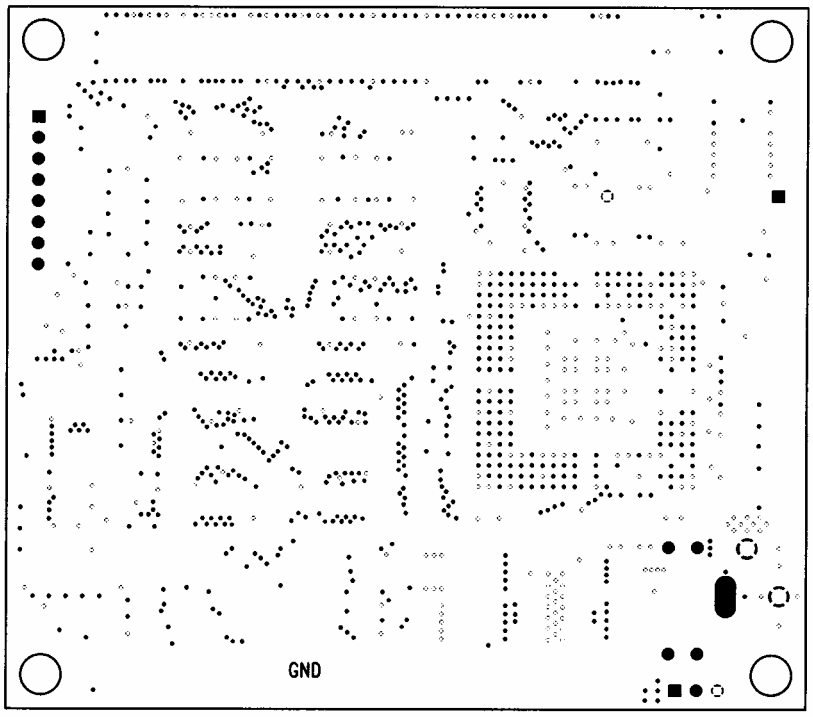
ภาพผนวกที่ 18 การวางอุปกรณ์ด้านบน (Top Layer) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



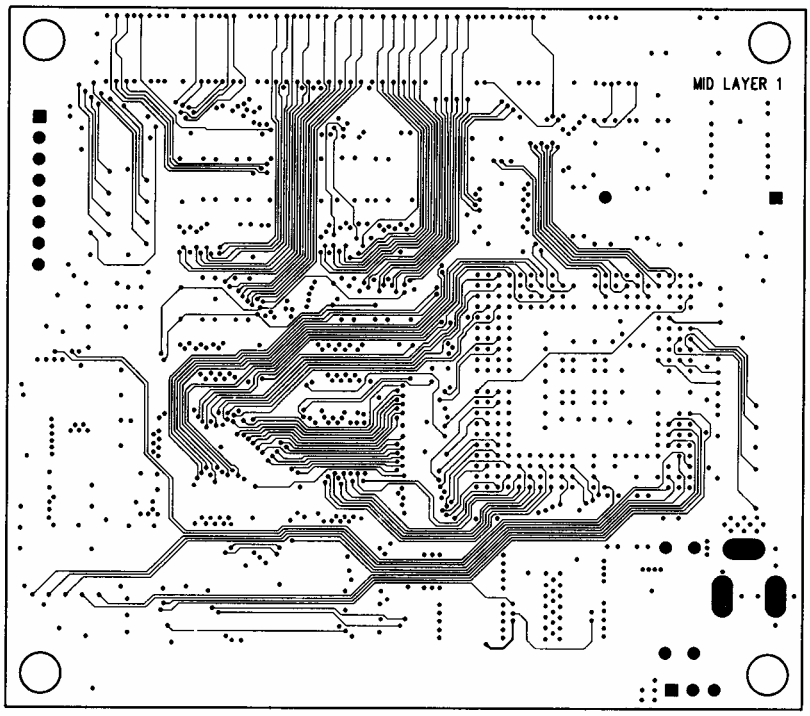
ภาพผนวกที่ 19 การวางอุปกรณ์ด้านล่าง (Bottom Layer) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



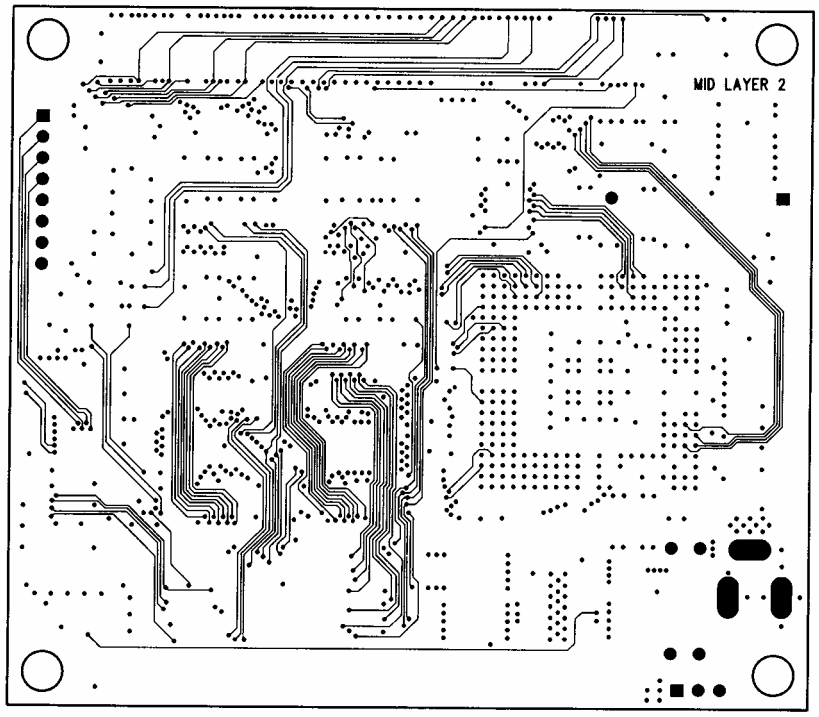
ภาพผนวกที่ 20 ลายวงจรด้านบน (Top Layer) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



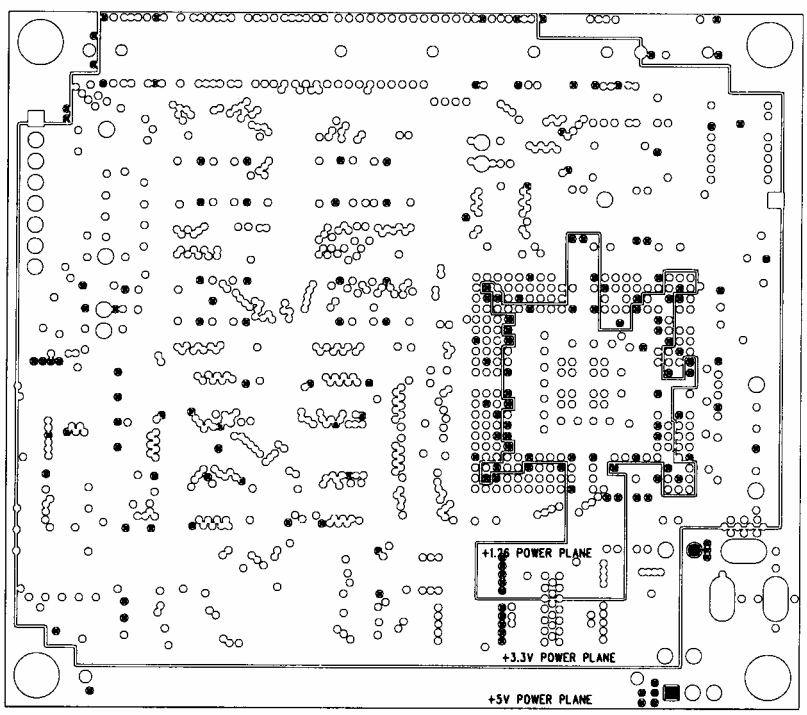
ภาพผนวกที่ 21 ลายวงจรระนาบดิน (Ground Plane) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



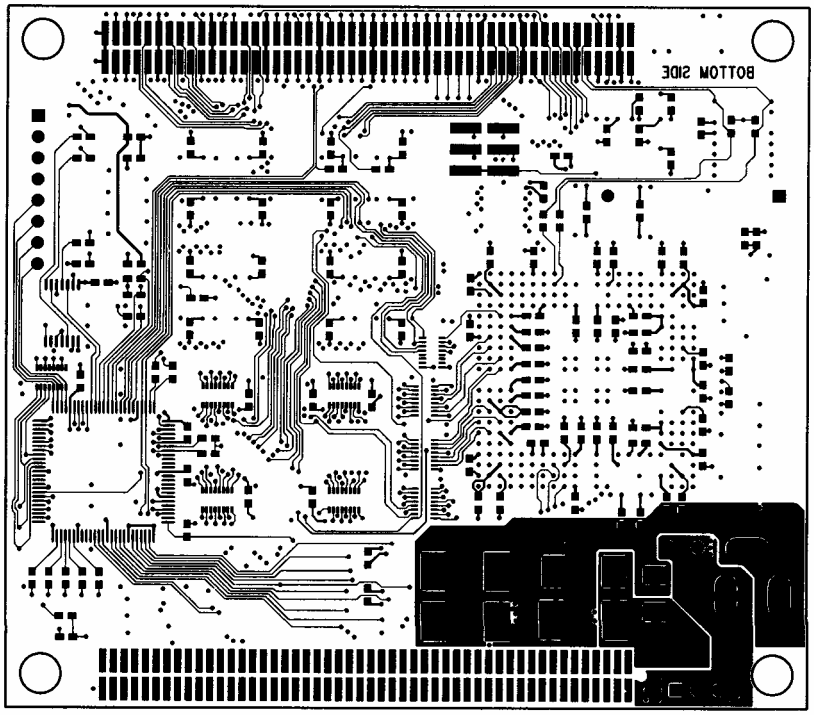
ภาพผนวกที่ 22 ลายวงจรชั้นกลางที่ 1 (Mid Layer 1) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



ภาพผนวกที่ 23 ลายวงจรชั้นกลางที่ 2 (Mid Layer 2) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



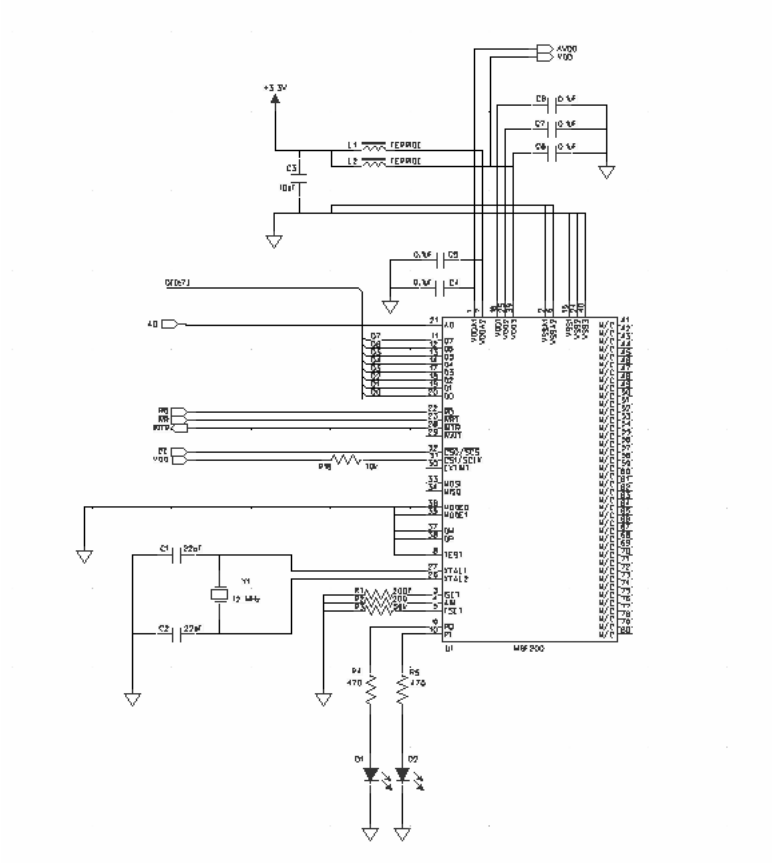
ภาพผนวกที่ 24 ลายวงจรระนาบกำลัง (Power Plane) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล



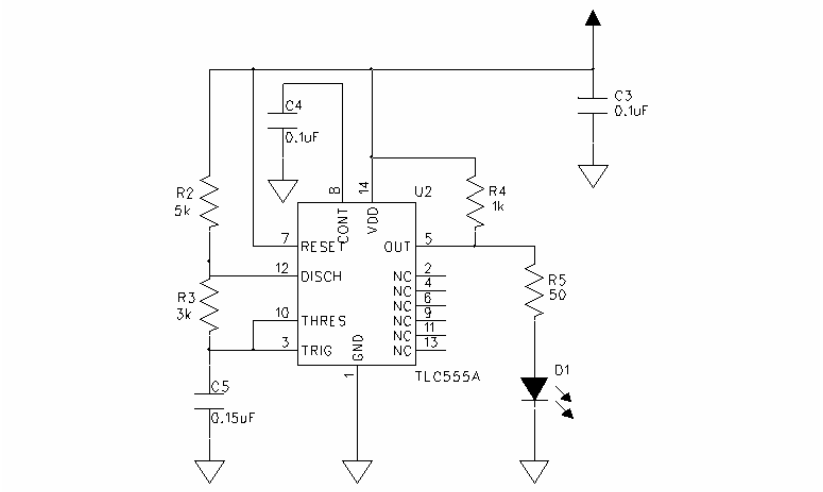
ภาพผนวกที่ 25 ลายวงจรด้านล่าง (Bottom Layer) ของฮาร์ดแวร์ประมวลผลสัญญาณดิจิทัล

ภาคผนวก ค

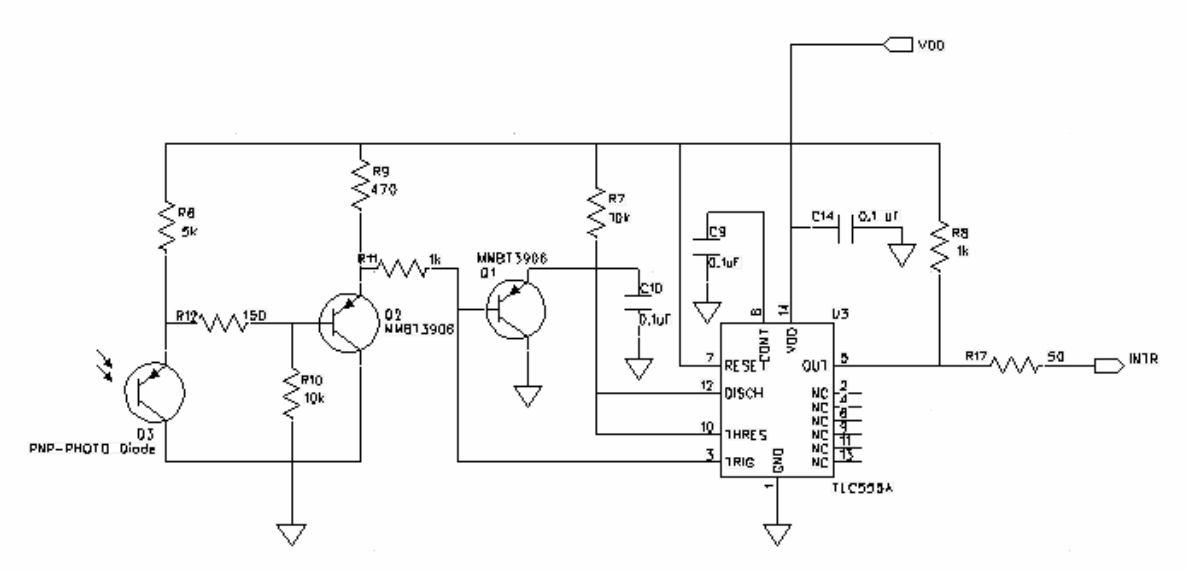
แผนภาพวงจรฮาร์ดแวร์กราดตรวจลายนิ้วมือ



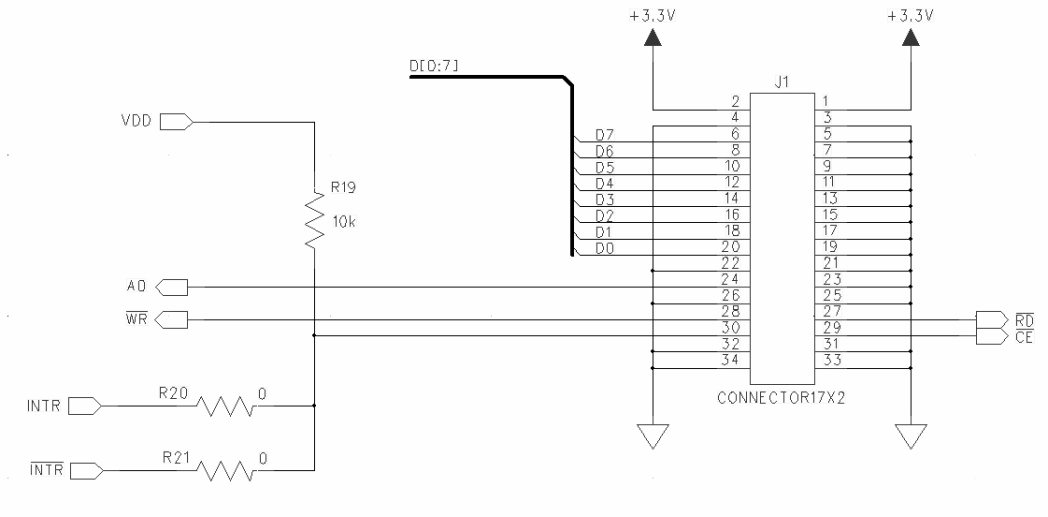
ภาพผนวกที่ 26 วงจรการเชื่อมต่อของวงจรรวมรับภาพลายนิ้วมือ



ภาพผนวกที่ 27 วงจรภาคส่งของอินฟราเรดมอดูล



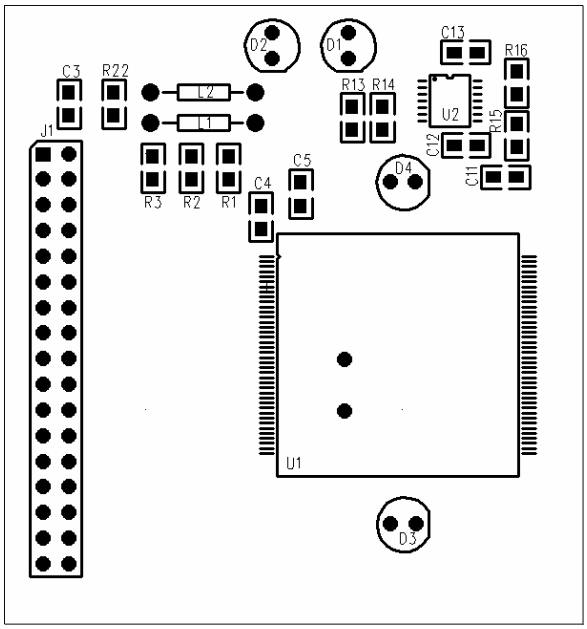
ภาพผนวกที่ 28 วงจรภาครับของอินฟราเรดมอดูล



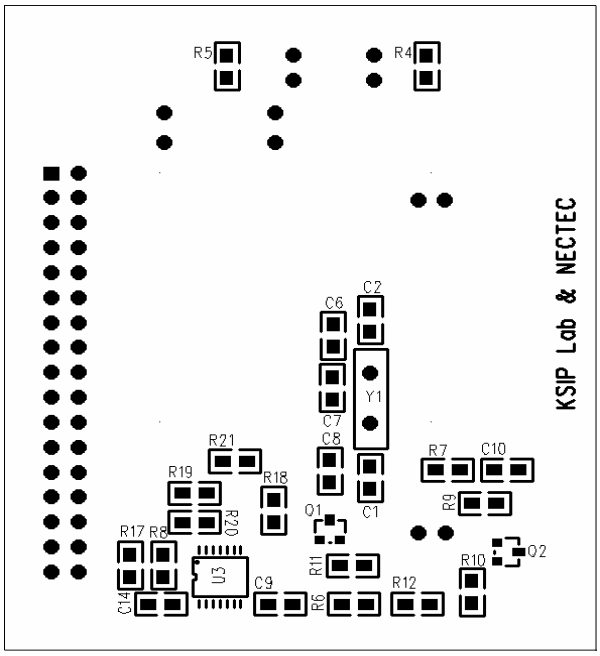
ภาพผนวกที่ 29 พอร์ตเชื่อมต่ออุปกรณ์ของฮาร์ดแวร์กราดตรวจลายนิ้วมือ

ภาคผนวก ง

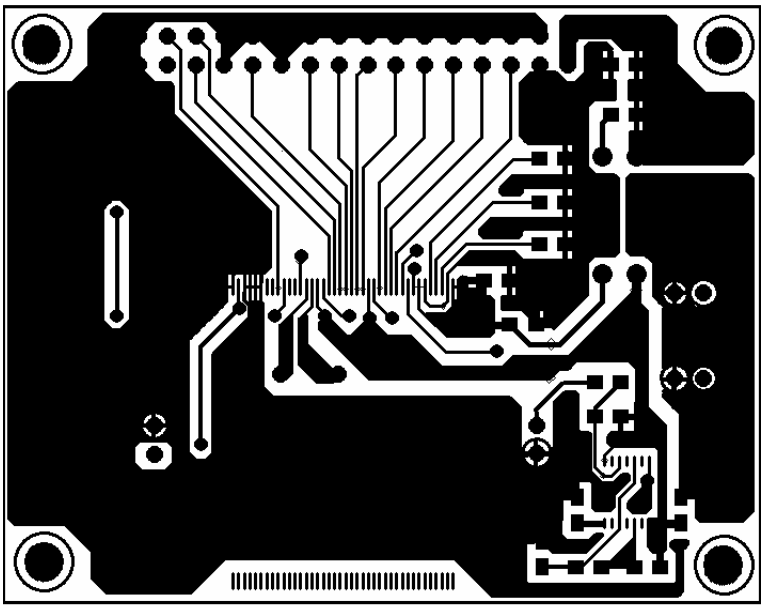
ลายวงจรฮาร์ดแวร์กราดตรวจลายนิ้วมือ



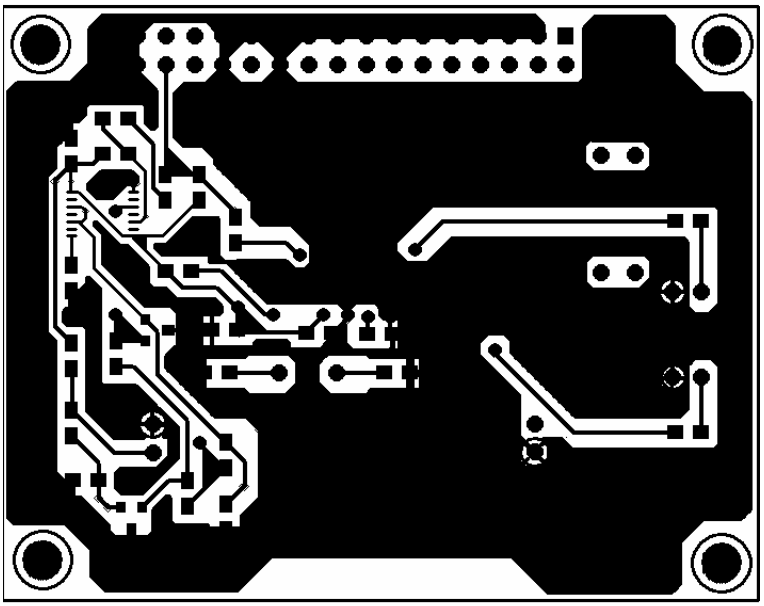
ภาพผนวกที่ 30 การวางอุปกรณ์ด้านบน (Top Layer) ของฮาร์ดแวร์ตรวจกราดลายนิ้วมือ



ภาพผนวกที่ 31 การวางอุปกรณ์ด้านล่าง (Bottom Layer) ของฮาร์ดแวร์ตรวจกราดลายนิ้วมือ



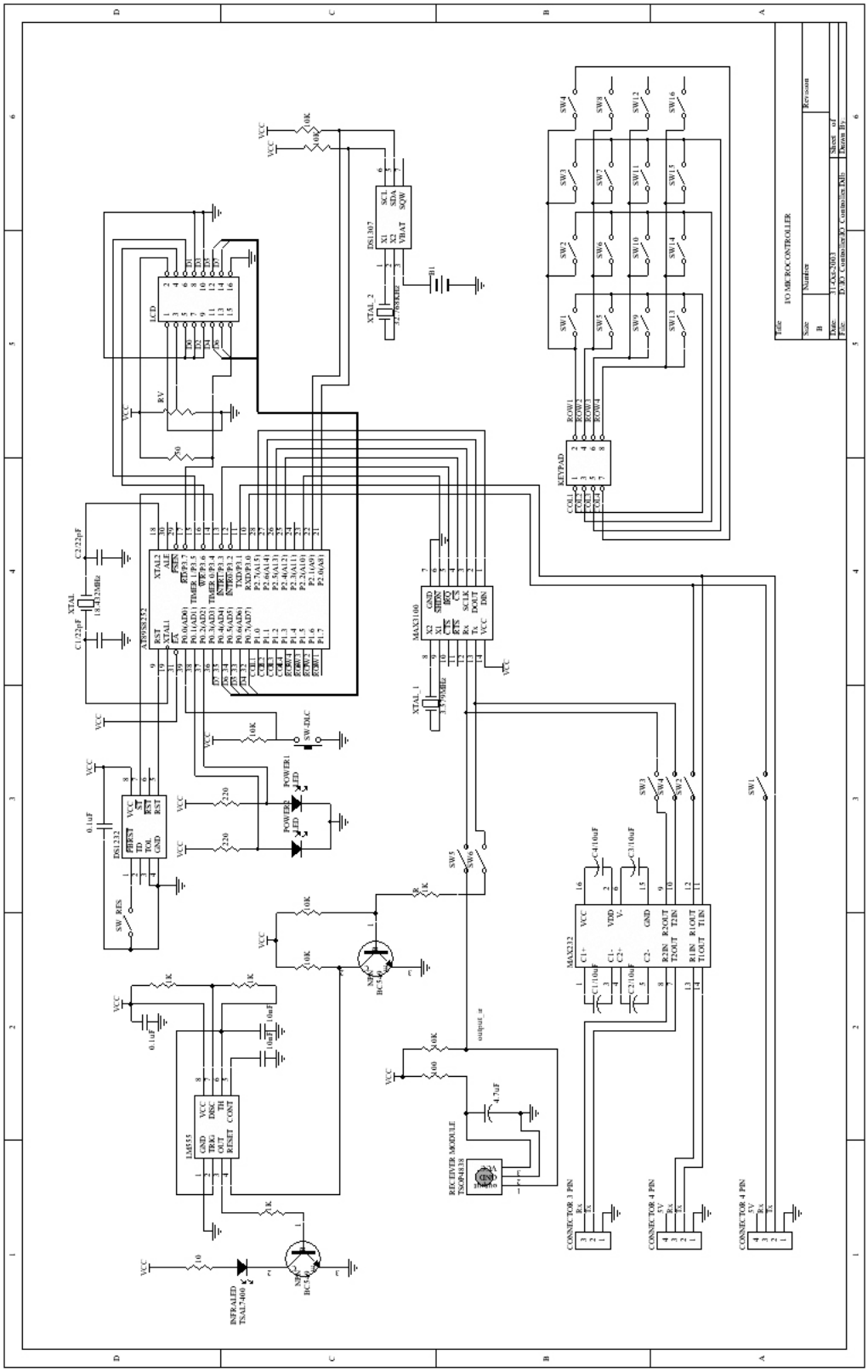
ภาพผนวกที่ 32 ลายวงจรด้านบน (Top Layer) ของฮาร์ดแวร์ตรวจกราดลายนิ้วมือ



ภาพผนวกที่ 33 ลายวงจรด้านล่าง (Botom Layer) ของฮาร์ดแวร์ตรวจกราดลายนิ้วมือ

ภาคผนวก จ

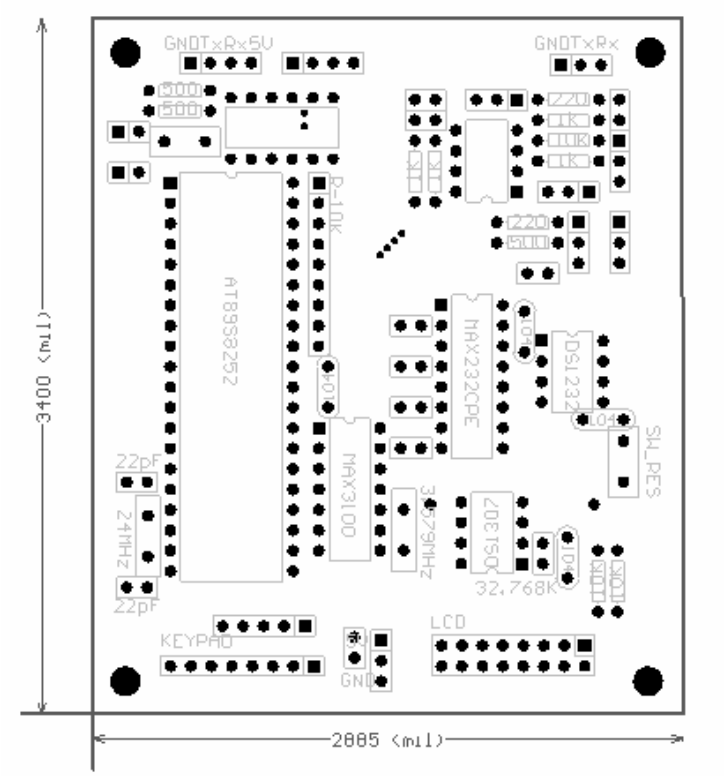
แผนภาพวงจรฮาร์ดแวร์แป้นพิมพ์และแสดงผล



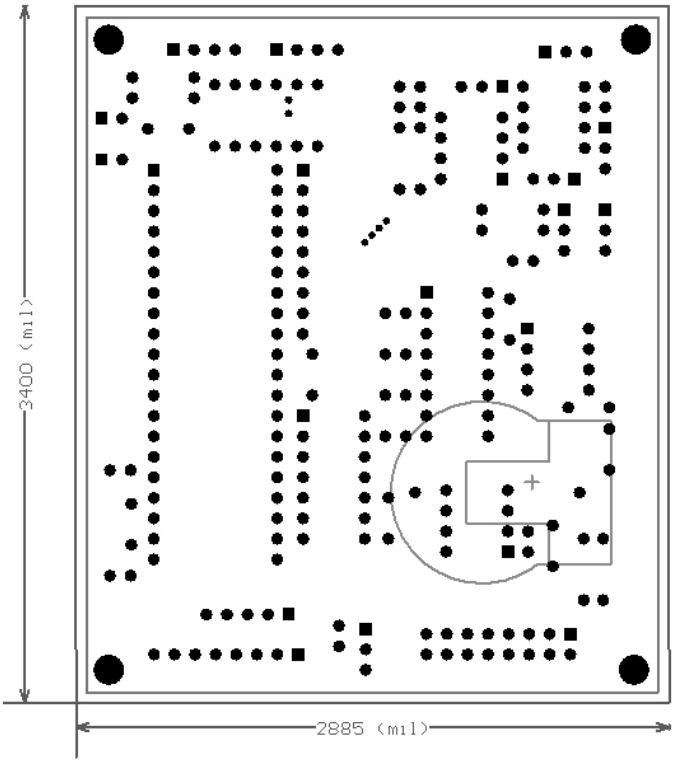
ภาพผนวกที่ 34 แผนภาพวงจรของฮาร์ดแวร์แป้นพิมพ์และแสดงผล

ภาคผนวก ฉ

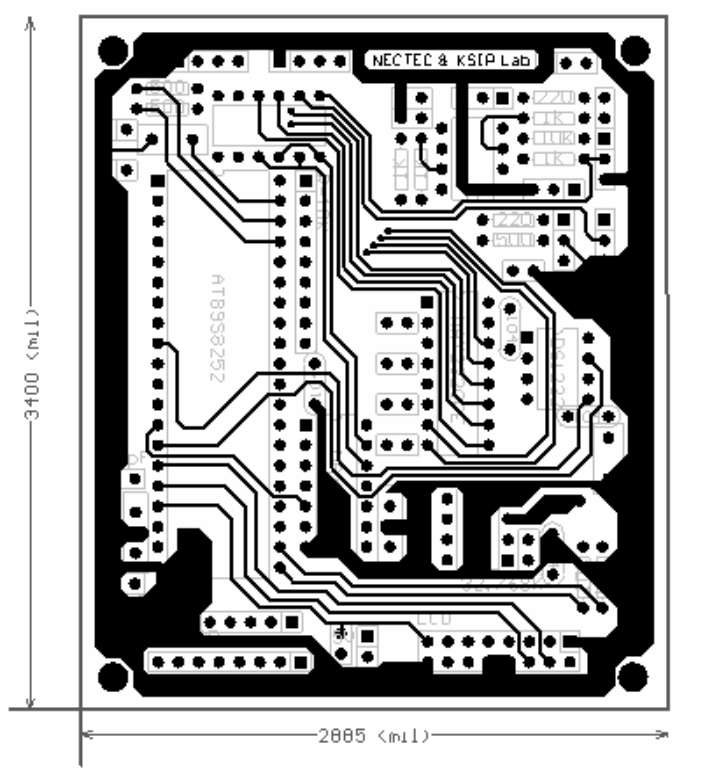
ลายวงจรฮาร์ดแวร์แป้นพิมพ์และแสดงผล



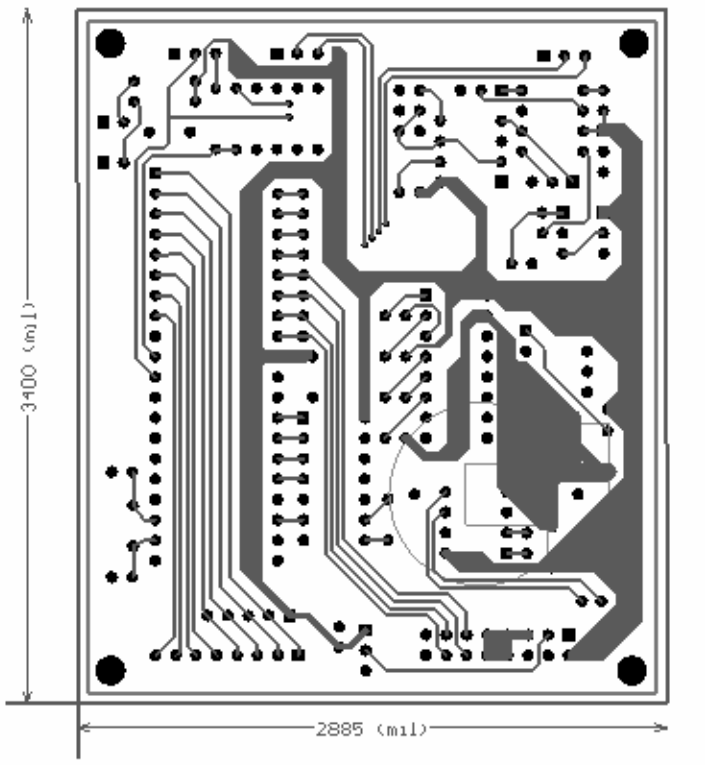
ภาพผนวกที่ 35 การวางอุปกรณ์ด้านบน (Top Layer) ของฮาร์ดแวร์แป้นพิมพ์และแสดงผล



ภาพผนวกที่ 36 การวางอุปกรณ์ด้านล่าง (Bottom Layer) ของฮาร์ดแวร์แป้นพิมพ์และแสดงผล



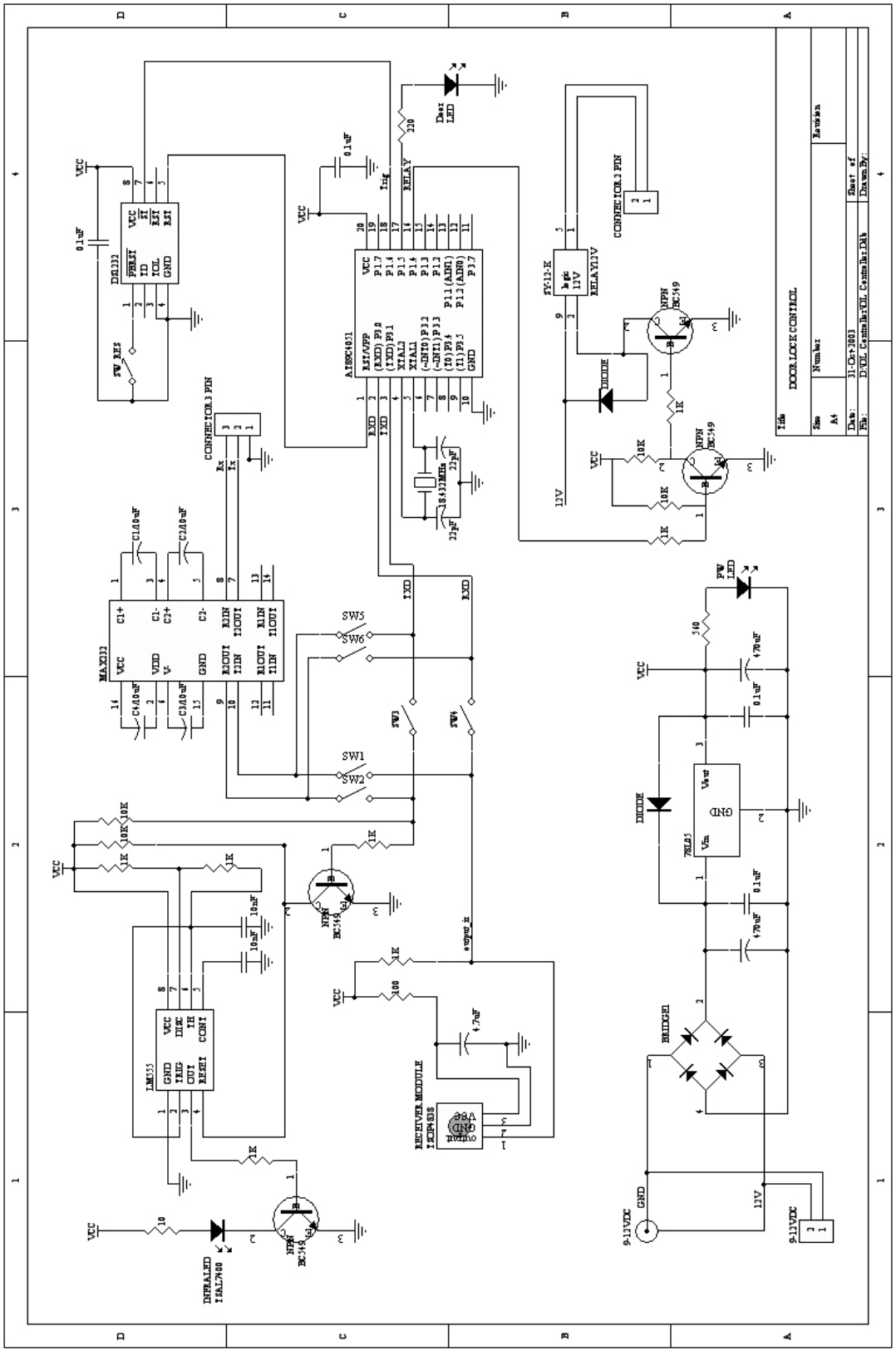
ภาพผนวกที่ 37 ลายวงจรด้านบน (Top Layer) ของฮาร์ดแวร์แป้นพิมพ์และแสดงผล



ภาพผนวกที่ 38 ลายวงจรด้านล่าง (Bottom Layer) ของฮาร์ดแวร์แป้นพิมพ์และแสดงผล

ภาคผนวก ช

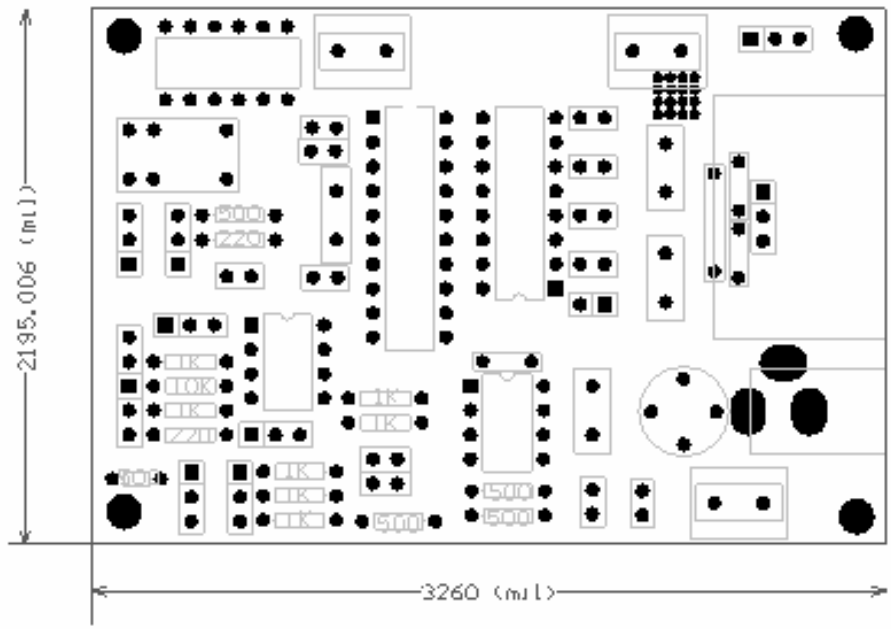
แผนภาพวงจรฮาร์ดแวร์ควบคุมการเปิดปิดประตู



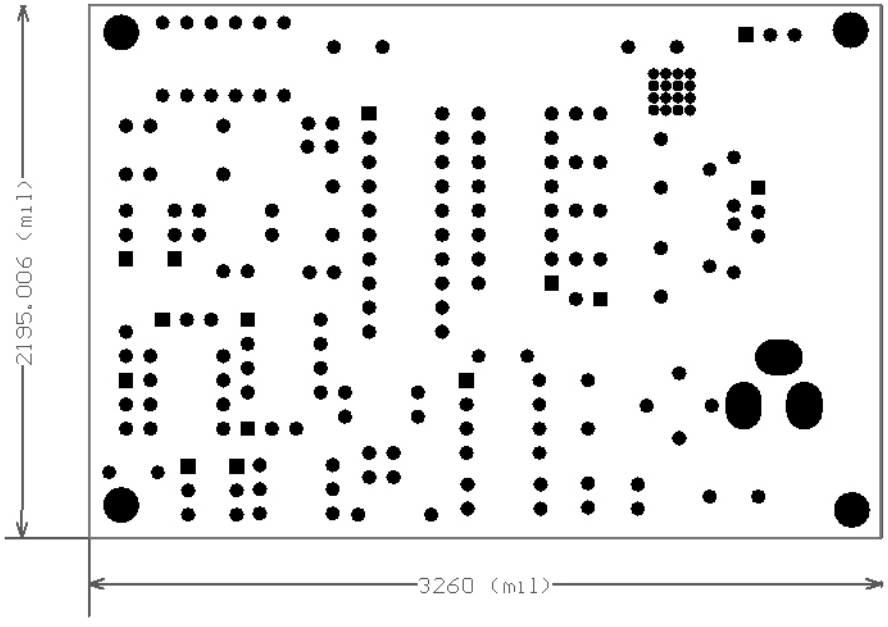
ภาพผนวกที่ 39 แผนภาพวงจรของฮาร์ดแวร์ควบคุมการเปิดปิดประตู

ภาคผนวก ซ

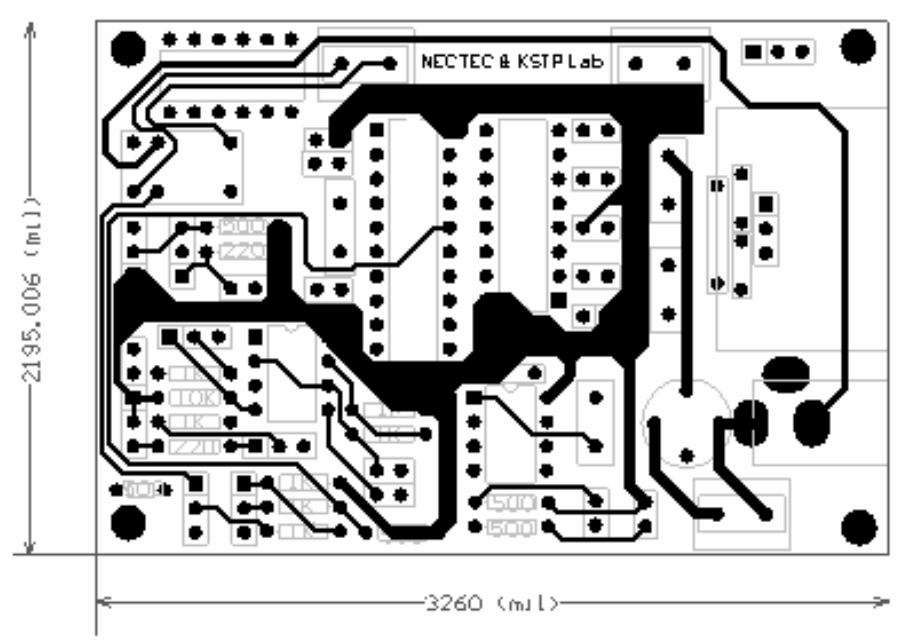
ลายวงจรฮาร์ดแวร์ควบคุมการเปิดปิดประตู



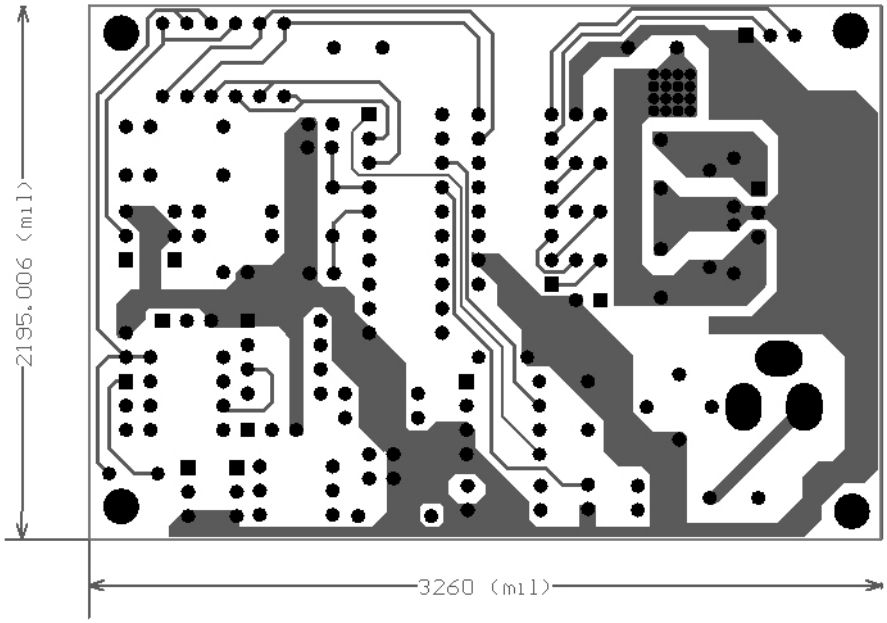
ภาพผนวกที่ 40 การวางอุปกรณ์ด้านบน (Top Layer) ของฮาร์ดแวร์ควบคุมการเปิดปิดประตู



ภาพผนวกที่ 41 การวางอุปกรณ์ด้านล่าง (Bottom Layer) ของฮาร์ดแวร์ควบคุมการเปิดปิดประตู



ภาพผนวกที่ 42 ลายวงจรด้านบน (Top Layer) ของฮาร์ดแวร์ควบคุมการเปิดปิดประตู



ภาพผนวกที่ 43 ลายวงจรด้านล่าง (Bottom Layer) ของฮาร์ดแวร์ควบคุมการเปิดปิดประตู