

บทที่ 3

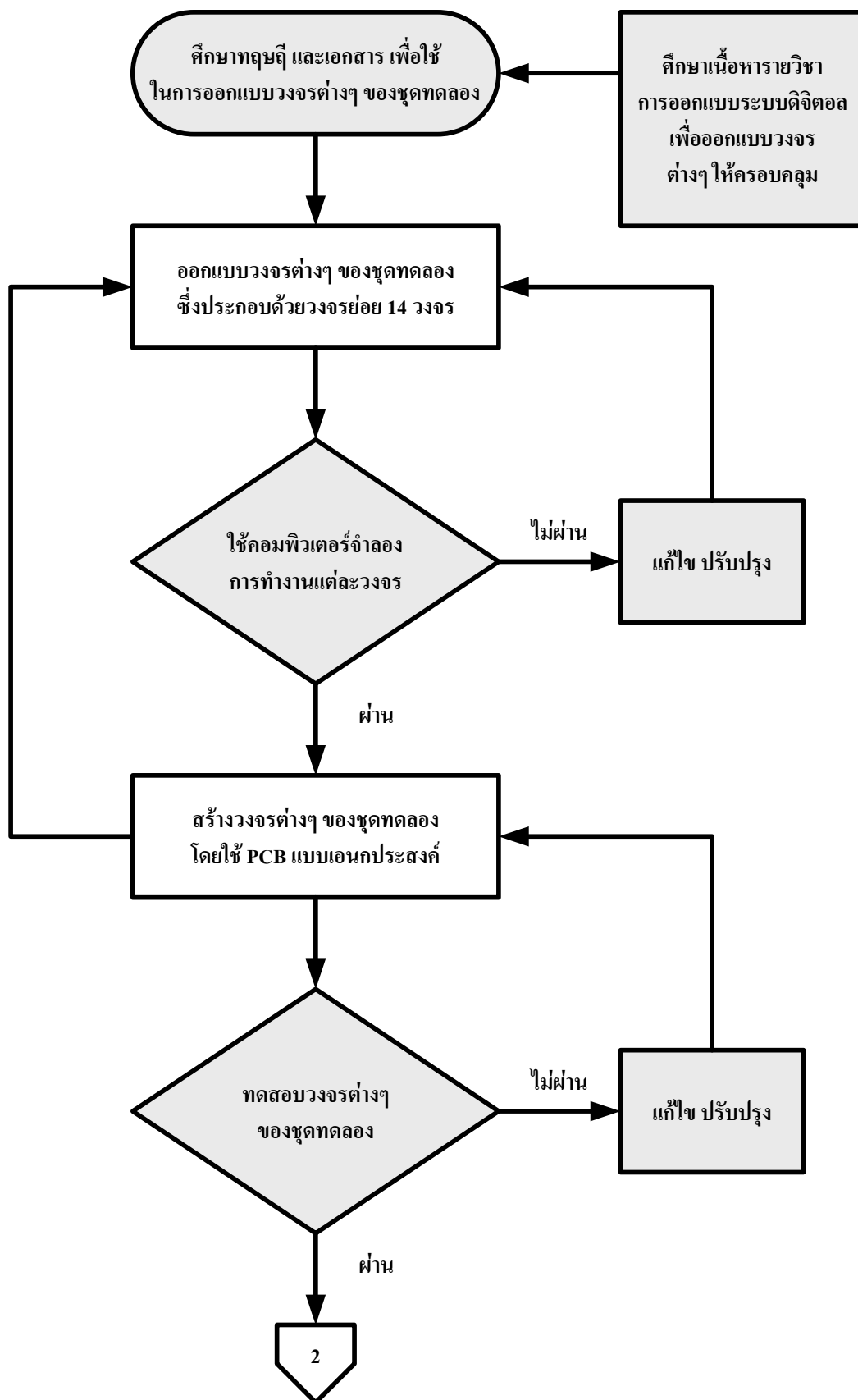
วิธีดำเนินการวิจัย

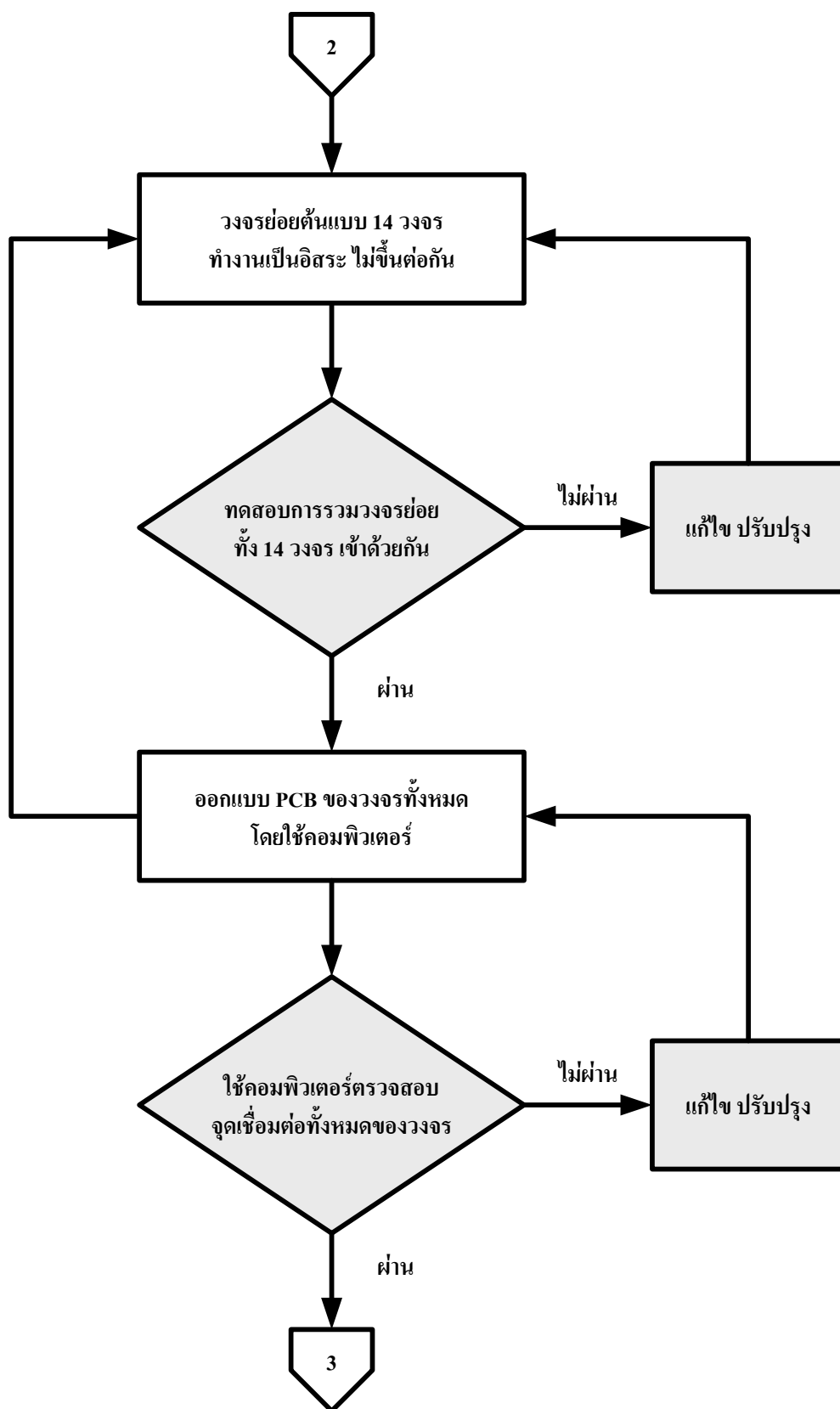
การพัฒนาชุดทดลองออกแบบวงจรรวมดิจิทัล

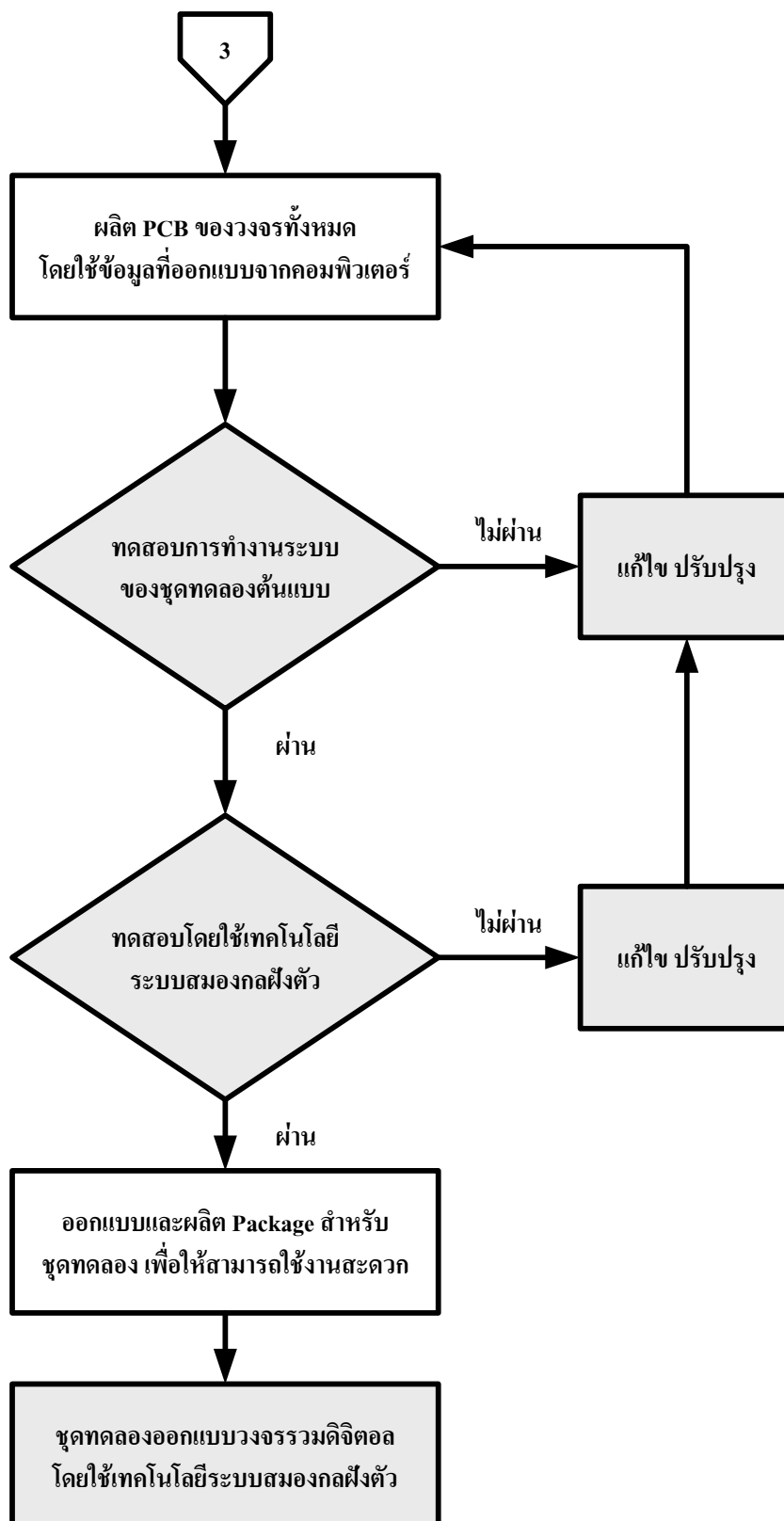
โครงการวิจัยนี้เป็นการวิจัยเชิงทดลอง เพื่อใช้ในการพัฒนาชุดทดลองออกแบบวงจรรวมดิจิทัลโดยใช้เทคโนโลยีระบบสมองกลฝังตัว ซึ่งมีขั้นตอนดำเนินการวิจัยดังนี้

1. ศึกษาทฤษฎี และเอกสาร เพื่อใช้ในการออกแบบวงจรต่างๆ ของชุดทดลอง
 2. ศึกษาเนื้อหารายวิชาการออกแบบระบบดิจิทัล เพื่อออกแบบวงจรต่างๆ ให้ครอบคลุม
 3. ออกแบบวงจรต่างๆ ของชุดทดลอง ซึ่งประกอบด้วยวงจรย่อย 14 วงจร
 4. ใช้คอมพิวเตอร์จำลองการทำงานแต่ละวงจร
 5. สร้างวงจรต่างๆ ของชุดทดลอง โดยใช้ PCB แบบเอนกประสงค์
 6. ทดสอบวงจรต่างๆ ของชุดทดลอง
 7. ได้วงจรย่อย 14 วงจร ทำงานเป็นอิสระ ไม่ขึ้นต่อกัน
 8. ทดสอบการรวมวงจรย่อยทั้ง 14 วงจร เข้าด้วยกัน
 9. ออกแบบ PCB ของวงจรทั้งหมดโดยใช้คอมพิวเตอร์
 10. ใช้คอมพิวเตอร์ตรวจสอบจุดเชื่อมต่อทั้งหมดของวงจร
 11. ผลิต PCB ของวงจรทั้งหมด โดยใช้ข้อมูลที่ออกแบบจากคอมพิวเตอร์
 12. ประกอบอุปกรณ์ลง PCB และทดสอบการทำงานระบบของชุดทดลองต้นแบบ
 13. ทดสอบการทำงานชุดทดลองต้นแบบโดยใช้เทคโนโลยีระบบสมองกลฝังตัว
 14. ออกแบบและผลิต Package เพื่อใช้กับชุดทดลองออกแบบวงจรรวมดิจิทัล
- สรุปขั้นตอนโดยละเอียดของการสร้างชุดทดลองออกแบบวงจรรวมดิจิทัล ดังแสดงใน

ภาพที่ 15







ภาพที่ 15 ขั้นตอนการสร้างชุดทดลองออกแบบวงจรรวมดิจิทัล

รายละเอียดของแต่ละขั้นตอนมีดังต่อไปนี้

1. ศึกษาทฤษฎี เอกสาร ข้อมูลจากบริษัทผู้ผลิตและนำเข้าอุปกรณ์ทางด้าน CPLD, FPGA และ I/O เพื่อใช้ในการออกแบบวงจรต่างๆ ของชุดทดลอง ซึ่งข้อมูลส่วนใหญ่ที่ได้จะมาจาก <http://www.aiologicttechnology.com> ซึ่งเป็น Website ที่เกี่ยวข้องกับชิพ CPLD ตระกูล XC9500 ที่ใช้ในการทดลอง <http://www.astronlogic.com> ซึ่งเป็น Website ที่เกี่ยวข้องกับชิพ CPLD ตระกูล MAX7000 ที่ใช้ในการทดลอง อุปกรณ์ I/O หาจาก <http://www.thaiio.com>, <http://www.etteam.com> <http://www.inex.co.th> และ <http://www.micro-research.co.th> ซึ่งเป็น Website ที่มี Project ทางด้าน ไมโครคอนโทรลเลอร์, CPLD, FPGA และการประยุกต์ใช้อุปกรณ์ I/O

2. ศึกษาเนื้อหาภาคปฏิบัติรายวิชาการออกแบบระบบดิจิทัล เพื่อออกแบบวงจรต่างๆ ให้ครอบคลุมตามเนื้อหาวิชา ผลจากการศึกษาขั้นตอนที่ 1 และ 2 ทำให้ได้วงจรย่อยทั้งหมด 14 วงจร ตามรายละเอียดดังนี้

- วงจร XC9500 and MAX7000 CPLD
- วงจร ISP (In-System Programming) Download
- วงจร Power Supply
- วงจร 16 Bit Logic Monitor
- วงจร 5x7 Dot Matrix
- วงจร BCD to 7 Segment
- วงจร Direct 7 Segment
- วงจร 8 Bit Input Switch
- วงจร 8 Bit DIP Switch
- วงจร 4x4 Matrix Switch
- วงจร Clock Generator
- วงจร Buzzer Generator
- วงจร Pulse Switch Generator
- วงจร 840 Point Photo Board

วงจรย่อยทั้ง 14 วงจรที่สร้างขึ้น สามารถรองรับการออกแบบวงจรรวมในภาคปฏิบัติของ รายวิชาการออกแบบระบบดิจิทัลได้ครบทุกใบงาน ซึ่งมีรายละเอียดดังนี้

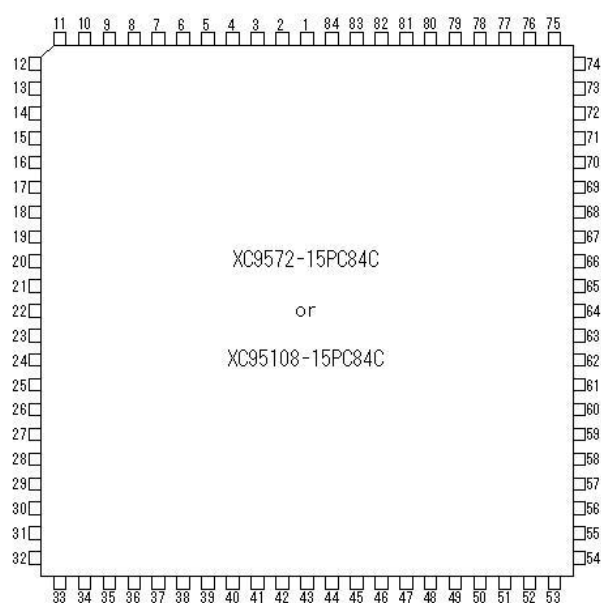
- การทดลองออกแบบวงจร Combination and Logic
- การทดลองออกแบบวงจร Arithmetic and Logical Unit
- การทดลองออกแบบวงจร Multiplex and Demultiplex

- การทดลองออกแบบวงจร Decoder and Encoder
- การทดลองออกแบบวงจร Latch and Flip-Flop
- การทดลองออกแบบวงจร Synchronous and Asynchronous Counter

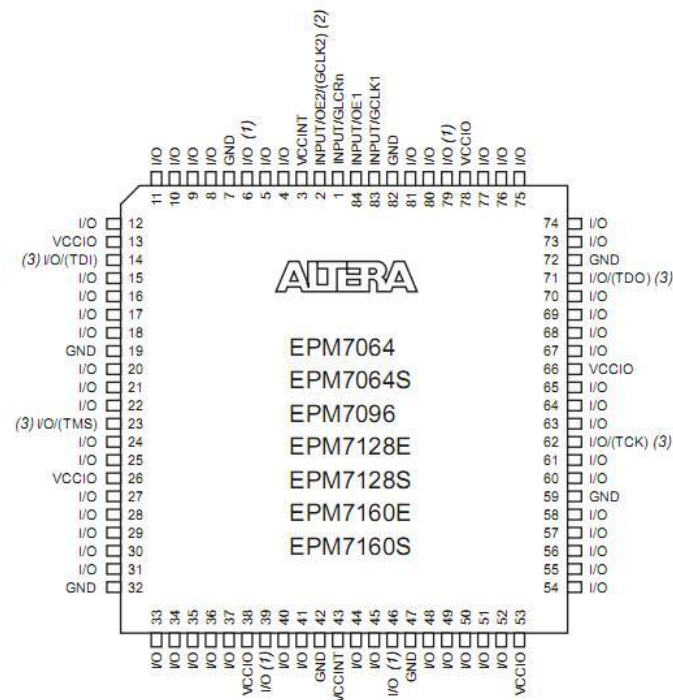
3. ออกแบบวงจรต่างๆ ของชุดทดลองทั้ง 14 วงจร โดยใช้คอมพิวเตอร์ในการออกแบบ ซึ่งมีรายละเอียดของการออกแบบแต่ละวงจร ดังนี้

วงจร XC9500 and MAX7000 CPLD

อุปกรณ์หลักที่ใช้ในการทดลอง คือ CPLD ตระกูล XC9500 เบอร์ XC95108 PC84 -15C ของ Xilinx Inc. และ MAX7000 เบอร์ EPM7128S LC84 -15C ของ Altera Co. มีความจุเกต 2,400 และ 2,500 เกต ตามลำดับ สามารถโปรแกรมซ้ำได้ประมาณ 10,000 ครั้ง ตัวถังเป็นแบบ PLCC จำนวน 84 ขา มี Terminal สำหรับต่อวงจร I/O ของชิพกับอุปกรณ์ภายนอกจำนวน 68 และ 69 I/O ตามลำดับ เพียงพอต่อการทดลองออกแบบระบบดิจิทัล รายละเอียดแสดงในภาพที่ 16 และ 17



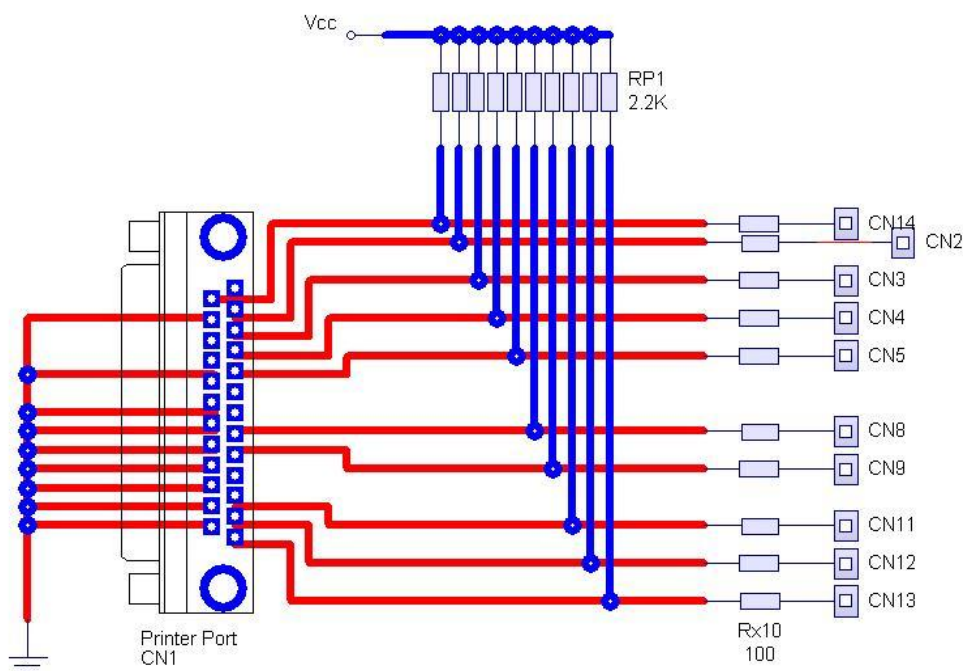
ภาพที่ 16 CPLD ตระกูล XC9500



ภาพที่ 17 CPLD ตระกูล MAX7000

วงจร ISP (In-System Programming) Download

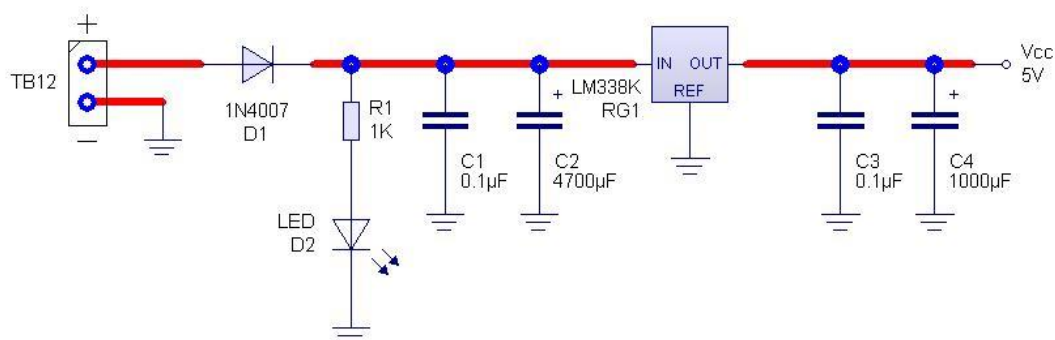
ออกแบบโดยใช้ Printer Port DB25 ตามมาตรฐาน JTAG Boundary Scan (IEEE 1149.1) และมีการต่อตัวต้านทาน $2.2\text{ K}\Omega$ แบบ Pull-up เพื่อรักษาระดับแรงดันให้คงที่ ทำให้อยู่ในสถานะ “1” ตลอดเวลา วงจรจึงมีความปลอดภัยจากสัญญาณรบกวนได้ดี รายละเอียดดังแสดงในภาพที่ 18



ภาพที่ 18 ลายวงจร ISP (In-System Programming) Download

วงจร Power Supply

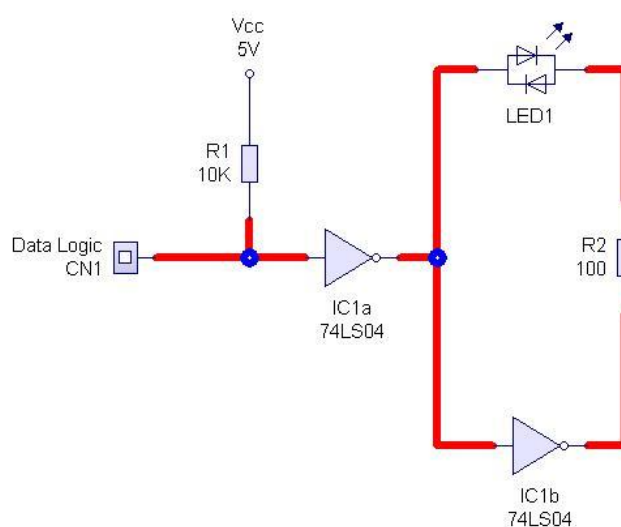
ออกแบบโดยใช้ IC Voltage Regulators เบอร์ LM338K ตัวถังแบบ Steel Can สามารถจ่ายกระแสได้สูงสุด 5A แรงดัน 1.2-32 V เอาต์พุต 5V ใช้สำหรับจ่ายให้วงจรย่อยทั้ง 14 วงจร และจุดต่ออิสระทั้งหมดของวงจร รายละเอียดแสดงในภาพที่ 19



ภาพที่ 19 ลายวงจร Power Supply

วงจร 16 Bit Logic Monitor

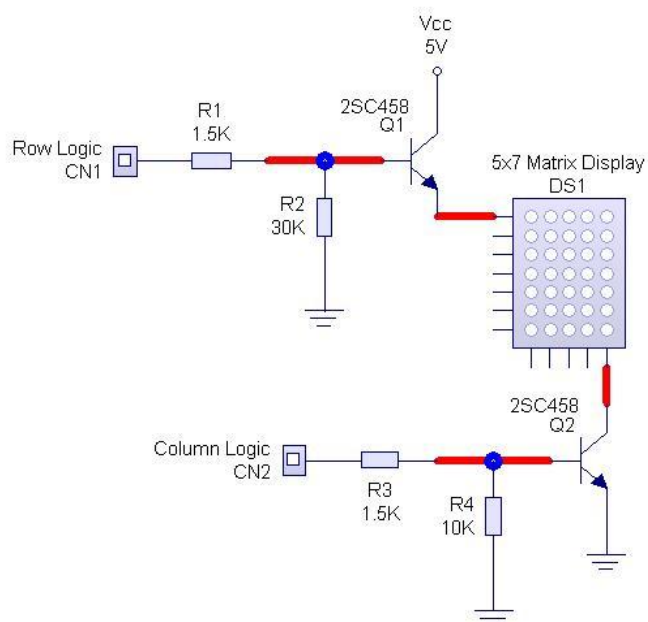
ออกแบบโดยใช้ LED ชนิด Bi Color สามารถแสดงสถานะได้ 2 สถานะ และมีการต่อตัวต้านทาน $10K\Omega$ แบบ Pull-up เพื่อรักษาระดับแรงดันให้คงที่ ทำให้อยู่ในสถานะ “1” ตลอดเวลา วงจรจึงมีความปลอดภัยจากสัญญาณรบกวนได้ดี IC เบอร์ 74LS04 ทำหน้าที่กลับสถานะ Logic และเป็น Buffer ก่อนถึง LED มีทั้งหมด 16 ชุด หรือ 16 Bit รายละเอียดแสดงในภาพที่ 20



ภาพที่ 20 ลายวงจร 16 Bit Logic Monitor

วงจร 5x7 Dot Matrix

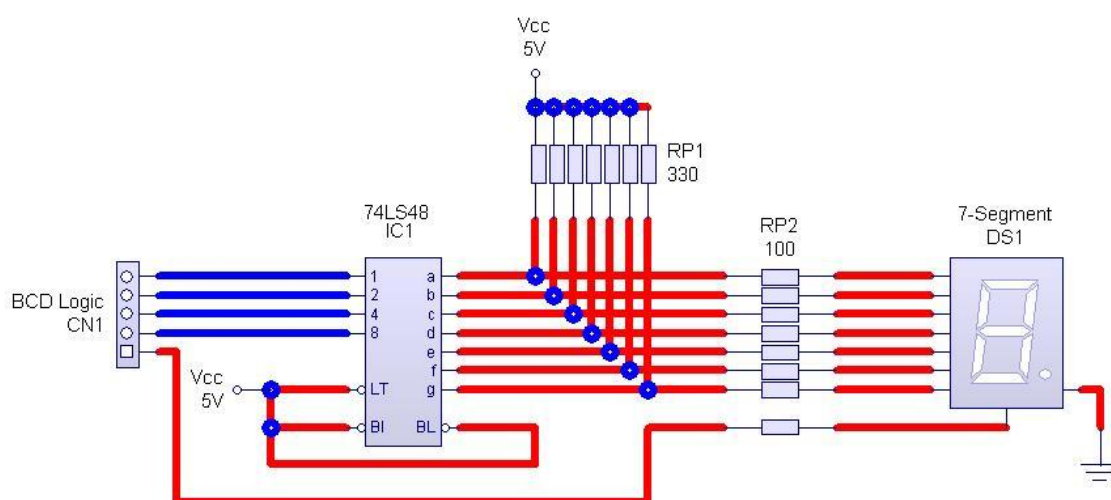
ออกแบบโดยใช้ Dot Matrix Display ขนาด 5x7 จำนวน 2 หลัก โดยมีทรานซิสเตอร์เบอร์ 2SC458 ทำหน้าที่ขยายสัญญาณของ Logic ก่อนเข้าสู่ชุดแสดงผล ตัวต้านทาน R1-4 ทำหน้าที่รักษาเสถียรภาพของวงจร และสามารถแสดงผลแบบ Scan Time ได้พร้อมกันทั้ง 2 ชุด รายละเอียดแสดงในภาพที่ 21



ภาพที่ 21 ลายวงจร 5x7 Dot Matrix

วงจร BCD to 7 Segment

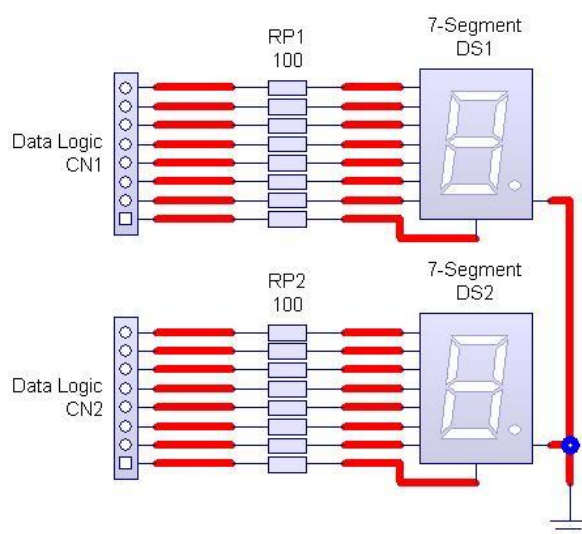
ออกแบบโดยใช้ 7 Segment ชนิด Cathode รวม จำนวน 2 หลัก เชื่อมต่อแบบ Scan Time มี IC เบอร์ 74LS48 ทำหน้าที่ถอดรหัสจาก BCD เป็น Bit Logic เพื่อนำไปแสดงผลในขั้นตอนต่อไป รายละเอียดแสดงในภาพที่ 22



ภาพที่ 22 ลายวงจร BCD to 7 Segment

วงจร Direct 7 Segment

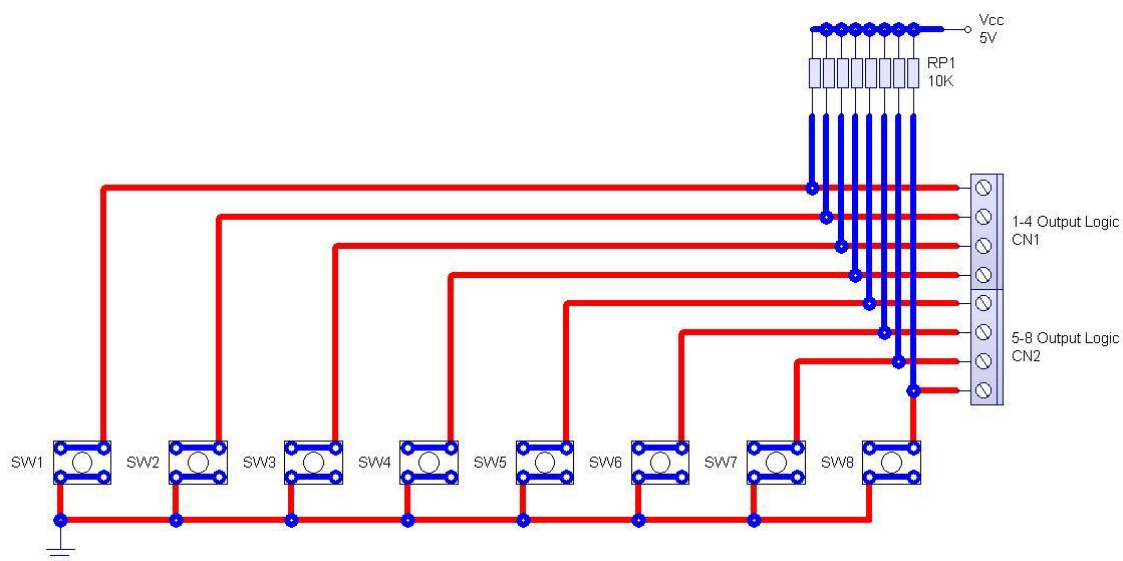
ออกแบบโดยใช้ 7 Segment ชนิด Cathode รวม จำนวน 2 หลัก เชื่อมต่อแบบ Scan Time ต่อตัวต้านทาน 100Ω เพื่อจำกัดกระแสก่อนเข้าสู่แสดงผล Input ของวงจรเป็น Data Logic ที่สามารถนำมาแสดงผลที่ส่วนต่างๆ ของ 7 Segment ได้ทีละส่วนอย่างอิสระ รายละเอียดแสดงในภาพที่ 23



ภาพที่ 23 ลายวงจร Direct 7 Segment

วงจร 8 Bit Input Switch

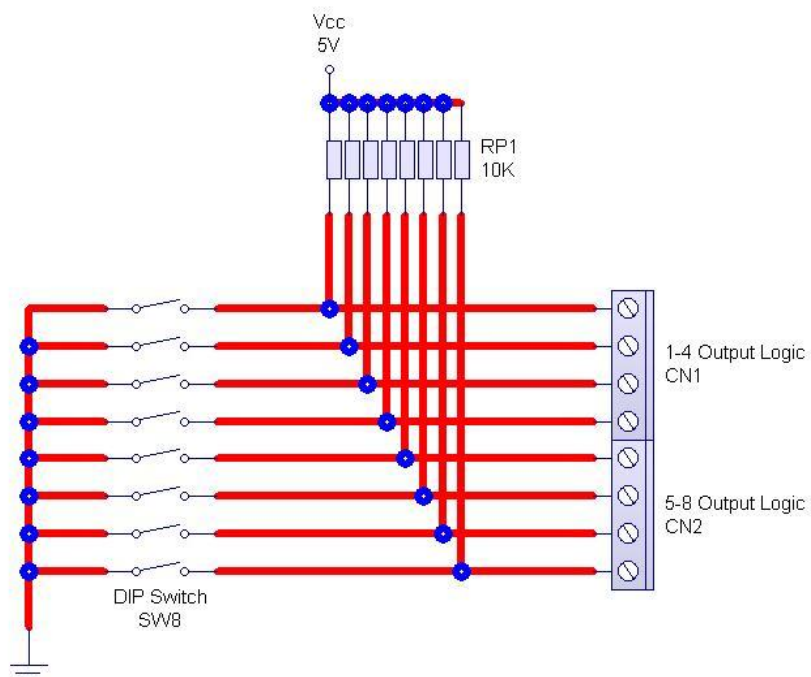
ออกแบบโดยใช้ Switch แบบ Push Button จำนวน 8 Bit มีการต่อตัวต้านทาน $10K\Omega$ แบบ Pull-up เพื่อรักษาระดับแรงดันให้คงที่ ทำให้อยู่ในสถานะ “1” ตลอดเวลาและเมื่อกดสวิตช์จะให้สถานะเป็น “0” เรียกว่า Active Low เพราะกระแสไฟฟ้าจะไหลลง Ground ทันที รายละเอียดแสดงในภาพที่ 24



ภาพที่ 24 ลายวงจร 8 Bit Input Switch

วงจร 8 Bit DIP Switch

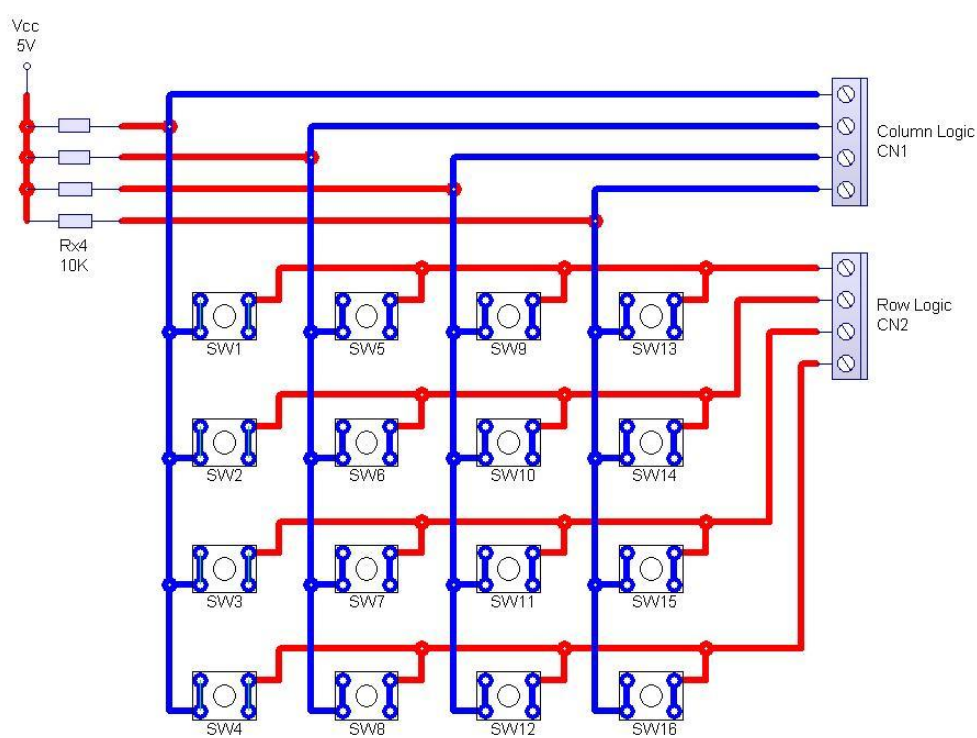
ออกแบบโดยใช้ Switch แบบ DIP ขนาด 8 Bit การทำงานหลักเหมือนกับวงจร 8 Bit Input Switch รายละเอียดแสดงในภาพที่ 25



ภาพที่ 25 ลายวงจร 8 Bit DIP Switch

วงจร 4x4 Matrix Switch

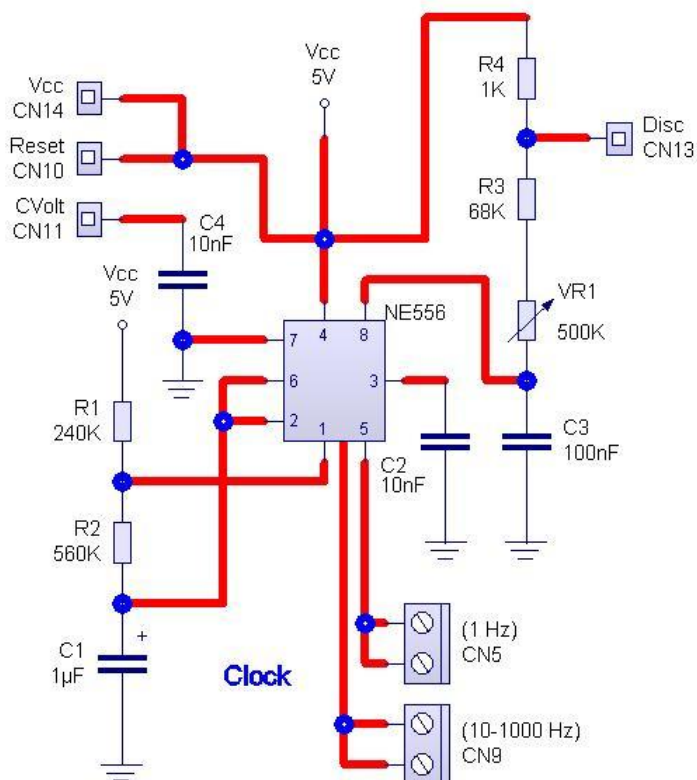
ออกแบบโดยใช้ Switch แบบ Push Button ต่อแบบ Matrix โดยเชื่อมต่อหน้าสัมผัสของ Switch ด้านหนึ่งเข้ากับ Connector ของ Row Logic และเชื่อมต่อหน้าสัมผัสของ Switch อีกด้านหนึ่งเข้ากับ Connector ของ Column Logic โดยมีตัวต้านทาน $10K\Omega$ แบบ Pull-up เพื่อรักษาระดับแรงดันให้คงที่ ทำให้อยู่ในสถานะ “1” ตลอดเวลา การควบคุมการทำงานของ Switch ทั้งหมด ใช้หลักการ Scan Time ในแต่ละ Row และ Column รายละเอียดแสดงในภาพที่ 26



ภาพที่ 26 ลายวงจร 4x4 Matrix Switch

วงจร Clock Generator

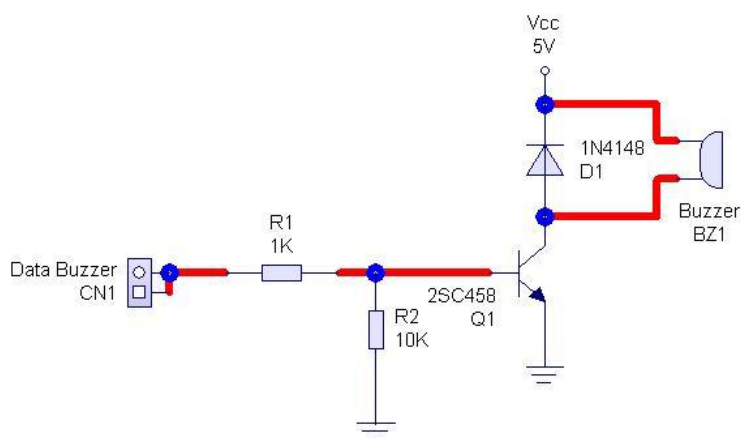
ออกแบบโดยใช้ IC เบอร์ NE555 สามารถสร้างความถี่สูงสุด 500 KHz วงจรรออกแบบให้สามารถใช้ความถี่ได้ 2 แบบ คือ แบบ Fix Frequency ใช้ความถี่ 1 Hz สำหรับสร้างสัญญาณนาฬิกา แบบ Real Time และแบบ Variable Frequency ใช้ความถี่ 10-1000 Hz สำหรับสร้างสัญญาณนาฬิกา แบบต่อเนื่อง สามารถปรับค่าความถี่ได้จาก VR1 รายละเอียดแสดงในภาพที่ 27



ภาพที่ 27 ลายวงจร Clock Generator

วงจร Buzzer Generator

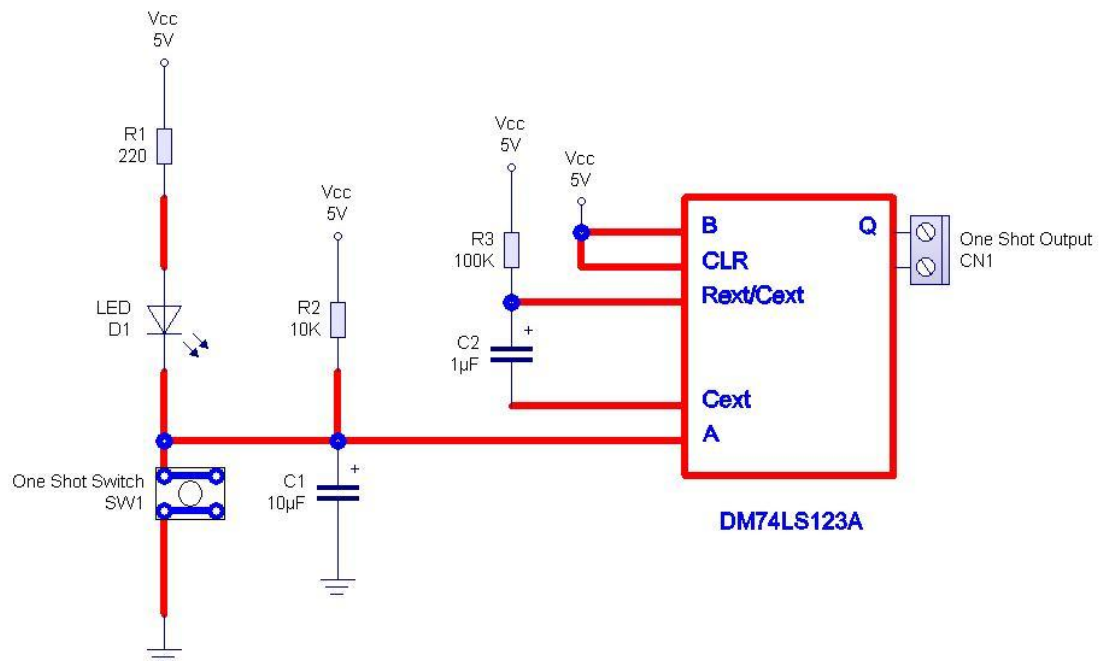
ออกแบบโดยใช้ Buzzer เป็นตัวกำเนิดเสียง โดยมีทรานซิสเตอร์เบอร์ 2SC458 ทำหน้าที่ขยายสัญญาณ รายละเอียดแสดงในภาพที่ 28



ภาพที่ 28 ลายวงจร Buzzer Generator

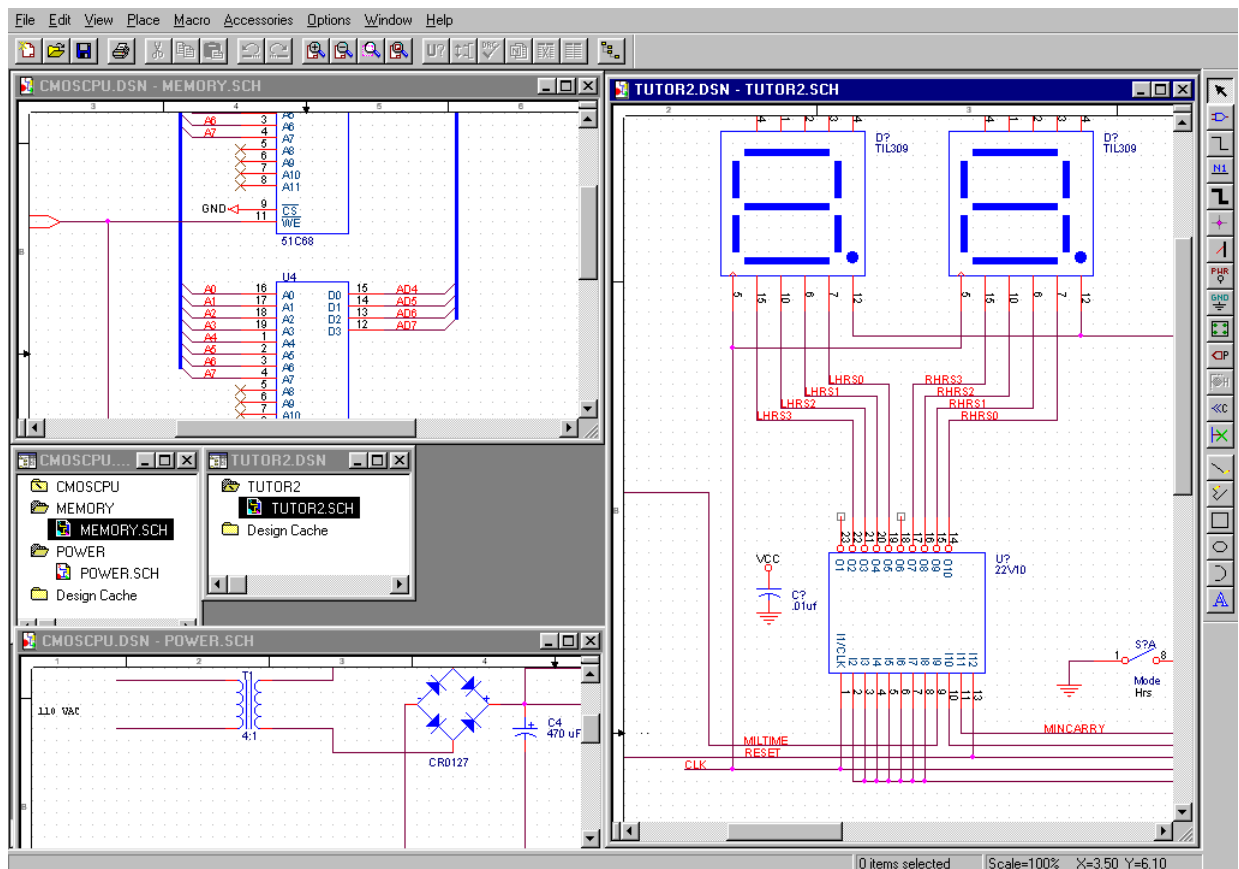
วงจร Pulse Switch Generator

ออกแบบโดยใช้ IC เบอร์ DM74LS123A สำหรับกำเนิดสัญญาณ Pulse แบบ One Shot สามารถ Clear และ Complementary Output ได้ รายละเอียดแสดงในภาพที่ 29

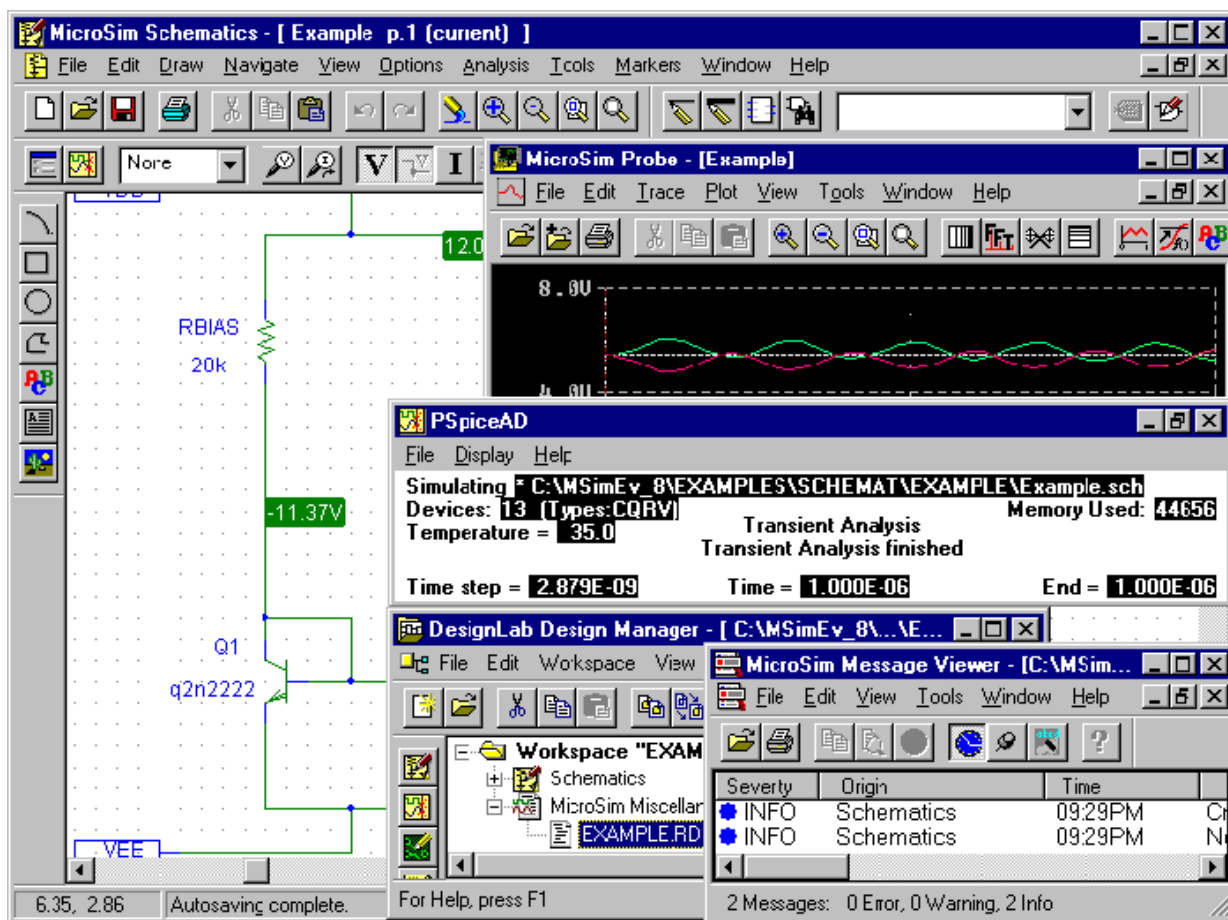


ภาพที่ 29 ลายวงจร Pulse Switch Generator

4. ใช้คอมพิวเตอร์จำลองการทำงานแต่ละวงจรที่ได้ออกแบบไว้ในขั้นตอนที่ 3 ซึ่งถ้าจำลองการทำงานแล้ว เกิดการผิดพลาด จะต้องออกแบบวงจรใหม่ หรือแก้ไข ปรับปรุงข้อผิดพลาดดังกล่าว ซึ่งในที่นี้ผู้วิจัยใช้โปรแกรม OrCAD Capture 7.2 และ MicroSim 8.0 ตัวอย่างโปรแกรมแสดงในภาพที่ 30 และ 31 ตามลำดับ

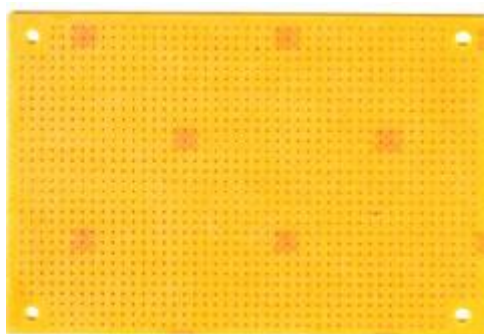


ภาพที่ 30 การออกแบบวงจรโดยใช้โปรแกรม OrCAD Capture 7.2



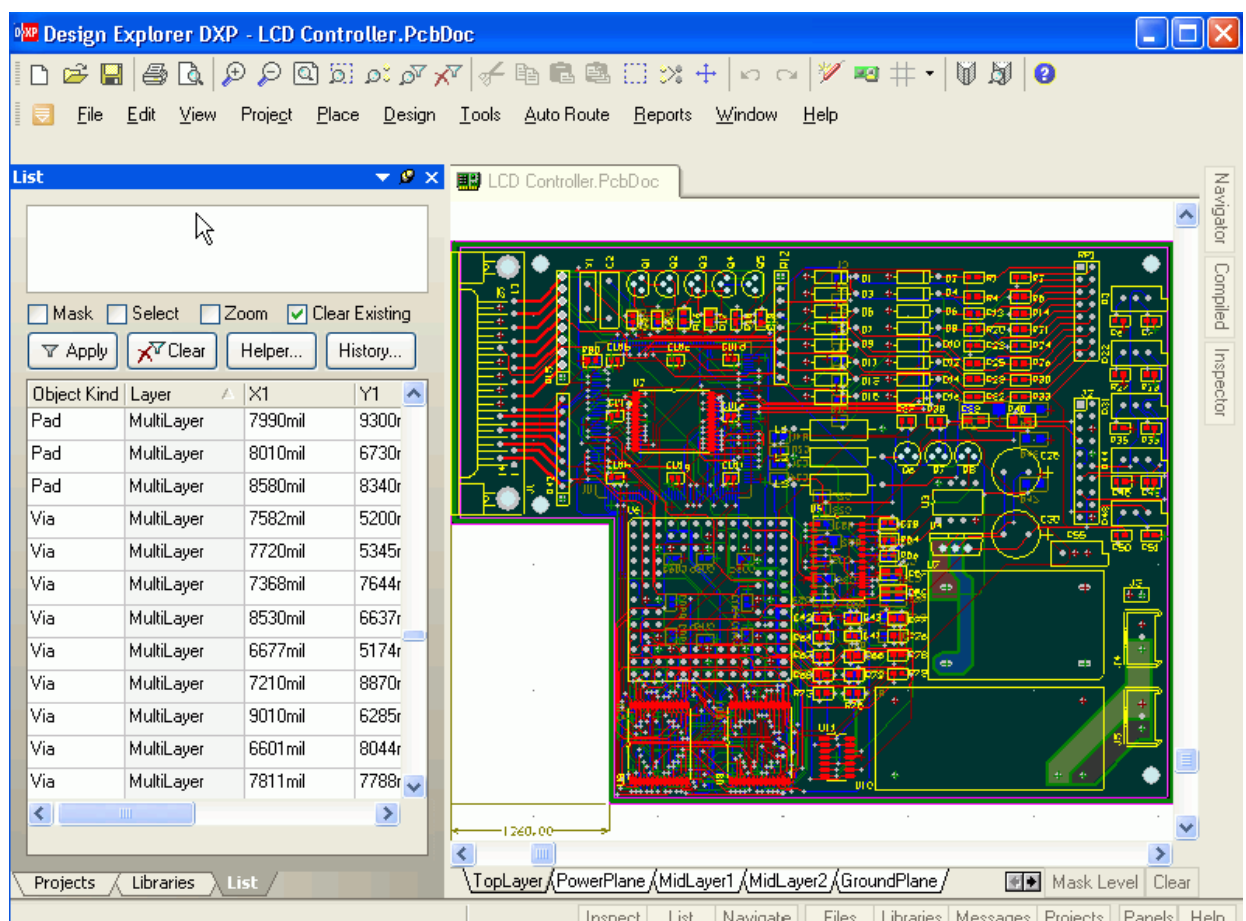
ภาพที่ 31 จำลองการทำงานของวงจรโดยใช้โปรแกรม MicroSim Design LAB 8.0

5. เมื่อผ่านการจำลองโดยใช้คอมพิวเตอร์ จึงดำเนินการสร้างวงจรต่างๆ ของชุดทดลอง โดยใช้ PCB แบบเอนกประสงค์ เพราะสามารถแก้ไขวงจรได้ง่าย



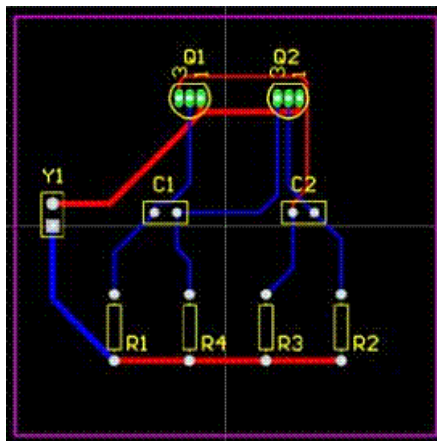
ภาพที่ 32 ตัวอย่าง PCB แบบเอนกประสงค์

6. ทดสอบวงจรต่างๆ ของชุดทดลอง โดยจ่ายแรงดันไฟฟ้าแล้ววัดค่าแรงดัน และกระแสที่จุดต่างๆ เพื่อตรวจสอบการทำงานของวงจร ถ้าวางจรใดทำงานไม่เต็มประสิทธิภาพ ต้องแก้ไขปรับปรุง หรือออกแบบวงจรนั้นใหม่ และดำเนินการตั้งแต่ขั้นตอนที่ 3 อีกครั้ง จนกว่าจะได้วงจรที่สมบูรณ์ ทำงานได้อย่างถูกต้องตามที่ออกแบบไว้
7. หลังจากทดสอบเรียบร้อยแล้วจะได้วงจรย่อย 14 วงจร ทำงานเป็นอิสระ ไม่ขึ้นต่อกัน
8. ทดสอบการรวมวงจรย่อยทั้ง 14 วงจร เข้าด้วยกัน โดยการเชื่อมต่อจุดแรงดันและกระแสไฟฟ้าของทุกวงจรเข้าด้วยกัน จากนั้นทำการตรวจสอบระบบทั้งหมดของวงจร ถ้ามีจุดบกพร่อง ต้องแก้ไข ปรับปรุง จนกว่าวงจรย่อยทั้ง 14 วงจร สามารถรวมเข้าด้วยกันได้
9. ออกแบบ PCB ของวงจรทั้งหมดโดยใช้คอมพิวเตอร์ ซึ่งผู้วิจัยใช้โปรแกรม Protel DXP



ภาพที่ 33 การออกแบบ PCB ของวงจรทั้งหมดโดยใช้โปรแกรม Protel DXP

10. ใช้คอมพิวเตอร์ตรวจสอบจุดเชื่อมต่อทั้งหมดของวงจร



ภาพที่ 34 การใช้โปรแกรม Protel DXP เติมนจุดเชื่อมต่อลายวงจร

11. ผลิต PCB ของวงจรทั้งหมด โดยใช้ข้อมูลที่ออกแบบจากคอมพิวเตอร์

12. ประกอบอุปกรณ์ลง PCB และทดสอบการทำงานของระบบของชุดทดลองต้นแบบ



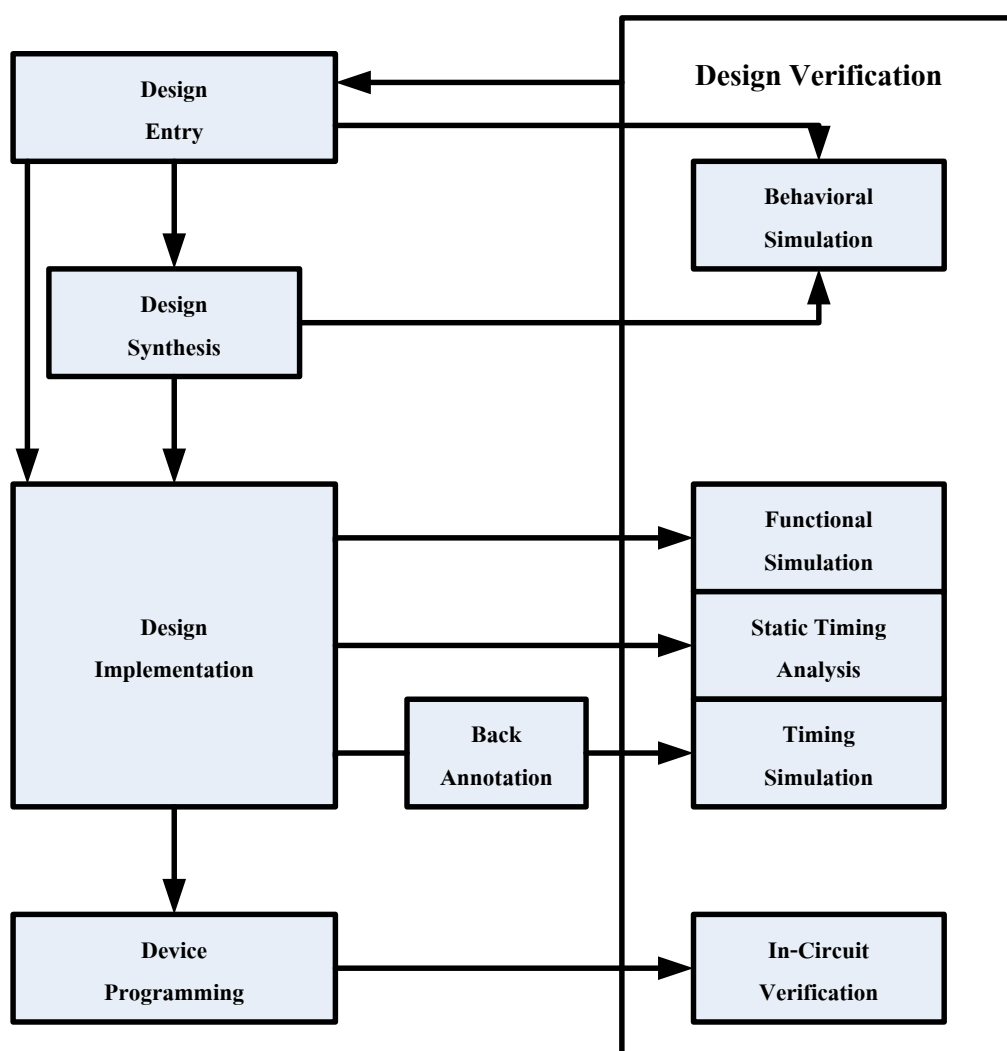
ภาพที่ 35 เครื่องมือที่ใช้ประกอบอุปกรณ์ลง PCB

13. ทดสอบการทำงานชุดทดลองต้นแบบโดยใช้เทคโนโลยีระบบสมองกลฝังตัว

14. ออกแบบและผลิต Package เพื่อใช้กับชุดทดลองออกแบบวงจรรวมดิจิทัล ซึ่งพร้อมที่จะนำไปใช้ในการออกแบบวงจรรวม และประยุกต์ใช้งานในขั้นตอนต่อไป

การออกแบบวงจรรวมดิจิทัลโดยใช้เทคโนโลยีระบบสมองกลฝังตัว

ขั้นตอนการออกแบบวงจรรวมดิจิทัลโดยใช้เทคโนโลยีระบบสมองกลฝังตัว มีการทำงานหลายขั้นตอน ผู้วิจัยยกตัวอย่างการใช้โปรแกรม Xilinx ISE WebPACK 8.1i ในการออกแบบ แสดงรายละเอียดดังภาพที่ 36

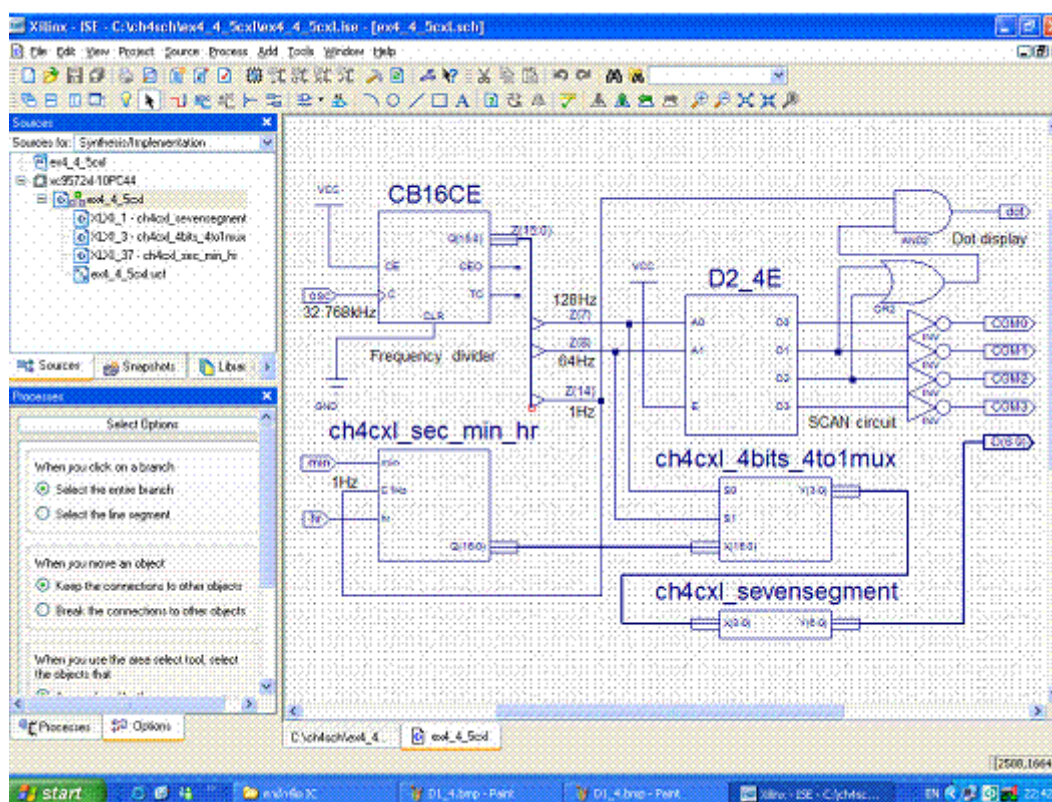


ภาพที่ 36 ขั้นตอนการออกแบบวงจรรวมดิจิทัล

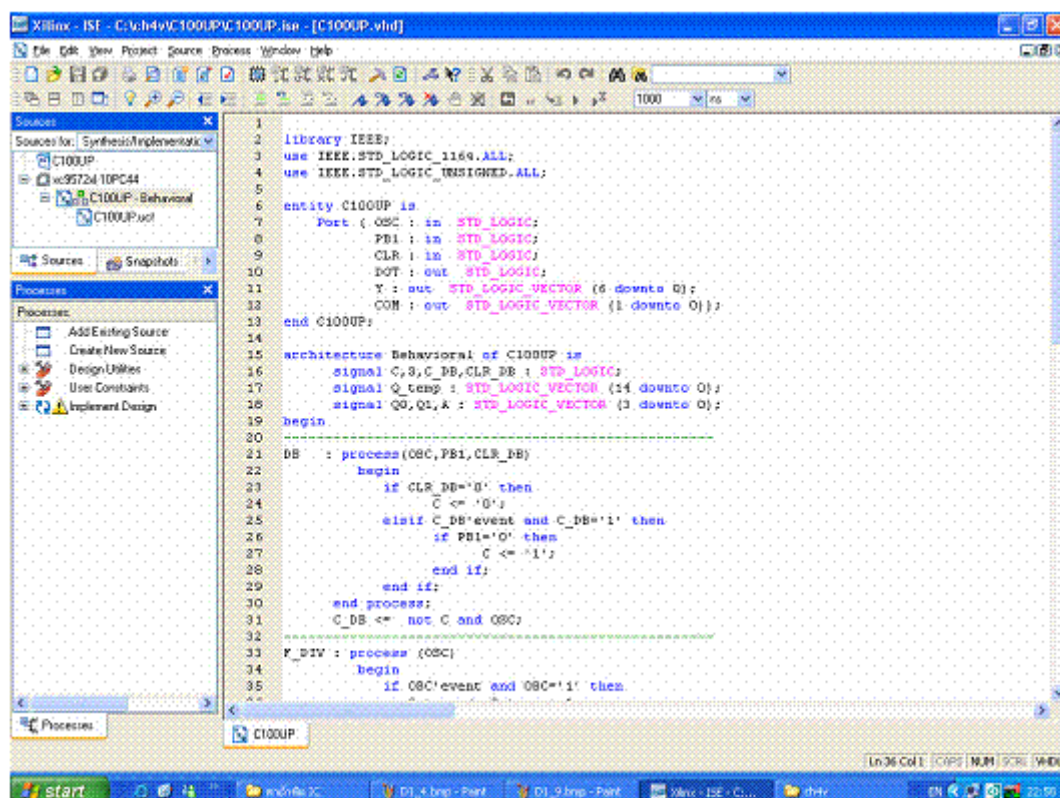
ที่มา : <http://toolbox.xilinx.com/docsan/xilinx82/books/manuals.htm>

Design Entry

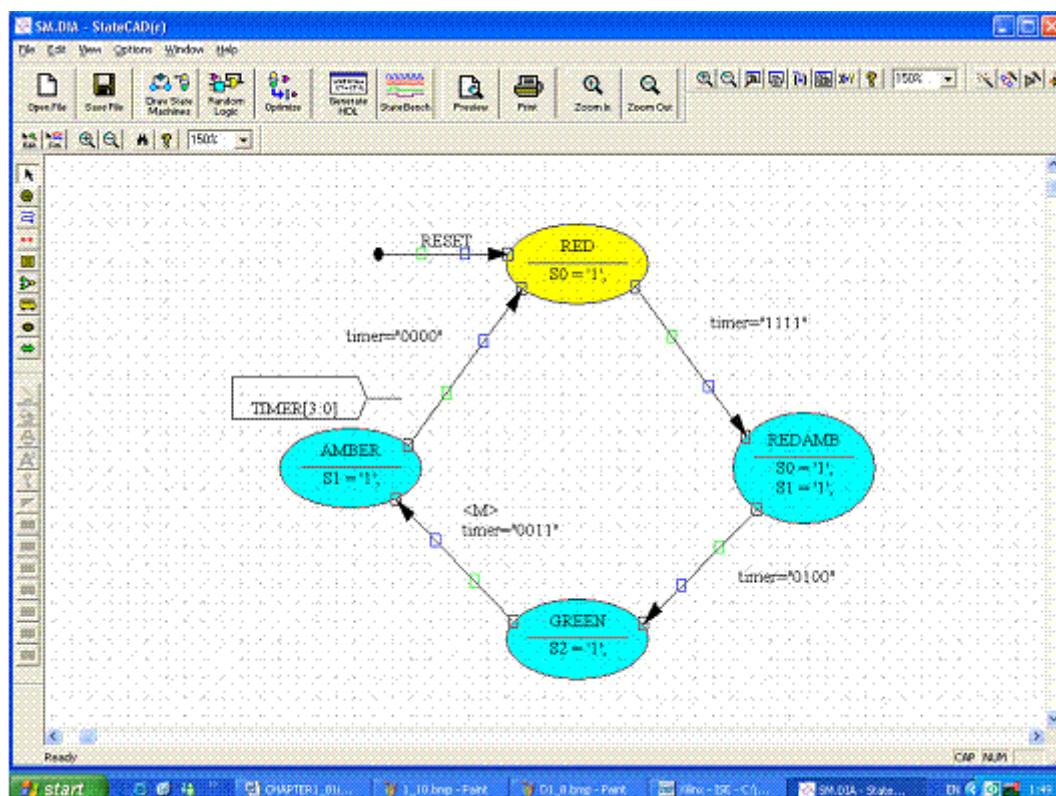
เป็นขั้นตอนเริ่มการออกแบบวงจรรวม สามารถเลือกออกแบบด้วยวิธีการต่างๆ ได้ 3 วิธี คือ วิธืวาดผังวงจร (Schematic) ดังภาพที่ 37 หรือการออกแบบด้วยภาษาระดับสูงที่เรียกว่า HDL (Hardware Description Language) ดังภาพที่ 38 และการออกแบบด้วยวิธีการเขียนแผนภูมิสถานะ (State Diagram) ดังภาพที่ 39 เพื่อให้ได้วงจรรวมตามที่ต้องการ



ภาพที่ 37 การออกแบบด้วยวิธืวาดผังวงจร (Schematic)



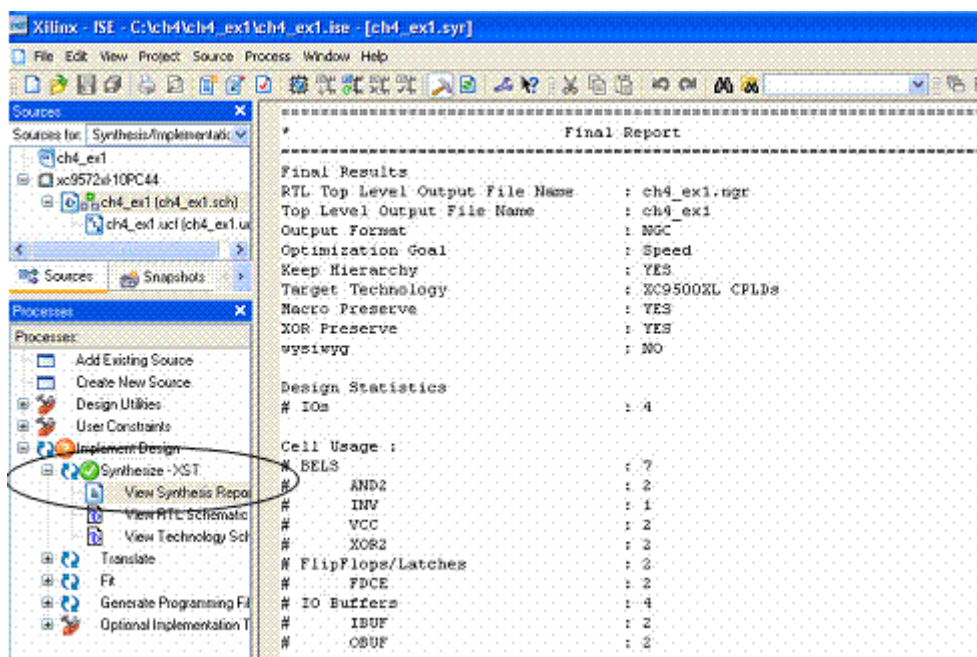
ภาพที่ 38 การออกแบบด้วยภาษาระดับสูง (HDL)



ภาพที่ 39 การออกแบบด้วยวิธีการเขียนแผนภูมิสถานะ (State Diagram)

Design Synthesis

ขั้นตอนการสังเคราะห์วงจร คือ ขั้นตอนการแปลงโค้ดที่ใช้ออกแบบด้วยภาษาระดับสูงที่เรียกว่า HDL ให้เป็นวงจรในระดับเกต (Gate Level) โดยใช้ Xilinx Synthesis Tool หรือ XST โดยวงจรจะถูกเขียนอยู่ในรูปไฟล์ Text File ที่เรียกว่า Net list ที่ระบุรายการอุปกรณ์ระดับเกต และรายการสายสัญญาณแต่ละเส้นที่เชื่อมต่อถึงกัน ส่วนการออกแบบด้วยวิธีวาดผังวงจร ซึ่งเป็นระดับเกตอยู่แล้วนั้น จะถูกเขียนอยู่ในรูปไฟล์ Net list เช่นกัน และอาจอยู่ในรูปไฟล์ Xilinx Net list Format (XNF) หรือ (EDIF) ที่เป็นมาตรฐานทางอุตสาหกรรมก็ได้ การสังเคราะห์วงจรนั้นจะต้องมีการระบุว่าเป็น CPLD รุ่นใด เพื่อป้อนค่าพารามิเตอร์ต่างๆ ให้กับเครื่องมือสังเคราะห์วงจร (Synthesis Tool) ซึ่งสามารถเลือก Optimize โดยเน้นความเร็วหรือเลือกประหยัดพื้นที่ (จำนวนเกต)



ภาพที่ 40 การสังเคราะห์วงจร

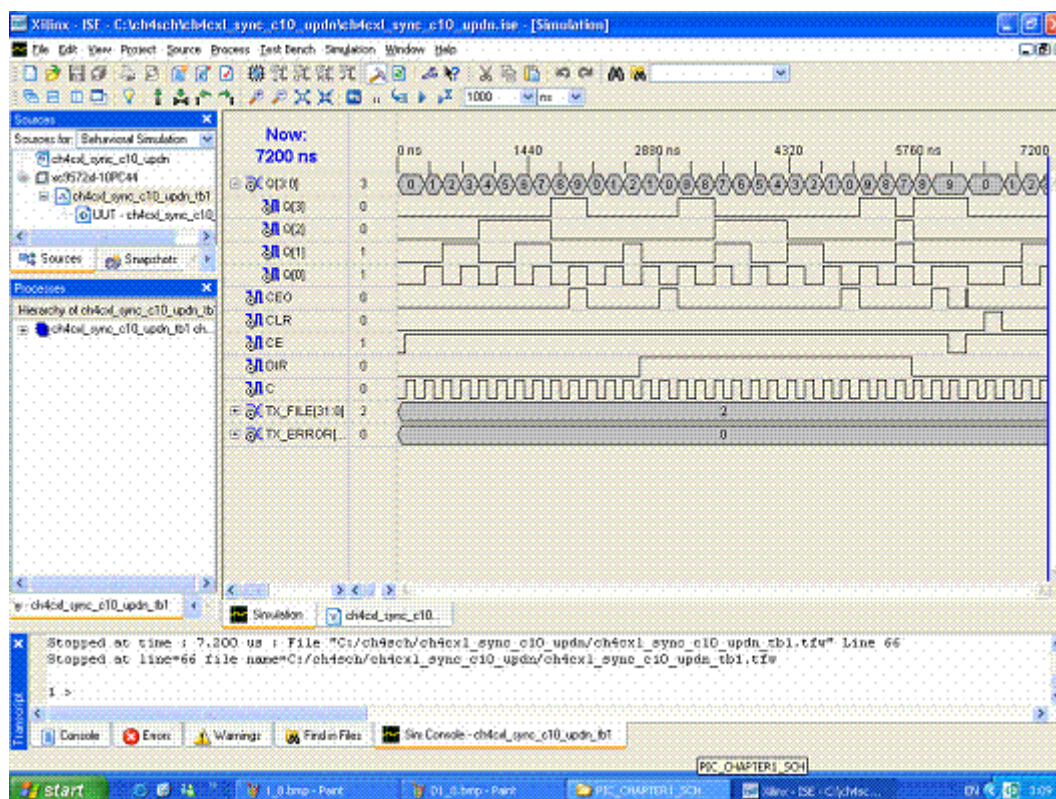
Design Verification

ขั้นตอนการตรวจสอบความถูกต้องของวงจรที่ออกแบบ เป็นการนำโค้ด HDL ของวงจรที่ออกแบบไปตรวจสอบความถูกต้อง ด้วยโปรแกรมจำลองการทำงาน (Simulation) โดยใช้ XST เช่น ISE Simulator หรือ ModelSim ตัวอย่างสัญญาณที่ได้จากการจำลองการทำงานแสดงดังภาพที่ 41 การจำลองการทำงานโดยทั่วไปมี 3 ระดับ คือ

- Behavioral Simulation เป็นการจำลองเฉพาะพฤติกรรมของวงจร โดยยังไม่คิดถึงโครงสร้างภายในของวงจร เพื่อให้ได้แบบจำลองการทำงานเบื้องต้น ซึ่งโค้ด HDL ที่เขียนนี้อาจจะนำไป Synthesis ไม่ได้ แต่เขียนโค้ดได้รวดเร็ว

- Functional Simulation เป็นการนำโค้ดในระดับ RTL (Register Transfer Level) ซึ่งเป็นระดับที่สามารถนำไปสังเคราะห์วงจรได้ มาทำการจำลองการทำงาน เพื่อตรวจสอบการทำงานของวงจรให้ถูกต้องก่อนนำไปสังเคราะห์วงจร

- Timing Simulation เป็นการจำลองการทำงานที่ใกล้เคียงกับ Hardware จริงมากที่สุด เนื่องจากการนำข้อมูล Timing ที่เกิดขึ้นมาใช้ ทำให้สามารถตัดสินใจเลือกเบอร์ชิปได้ถูกต้องก่อนทำการซื้อ



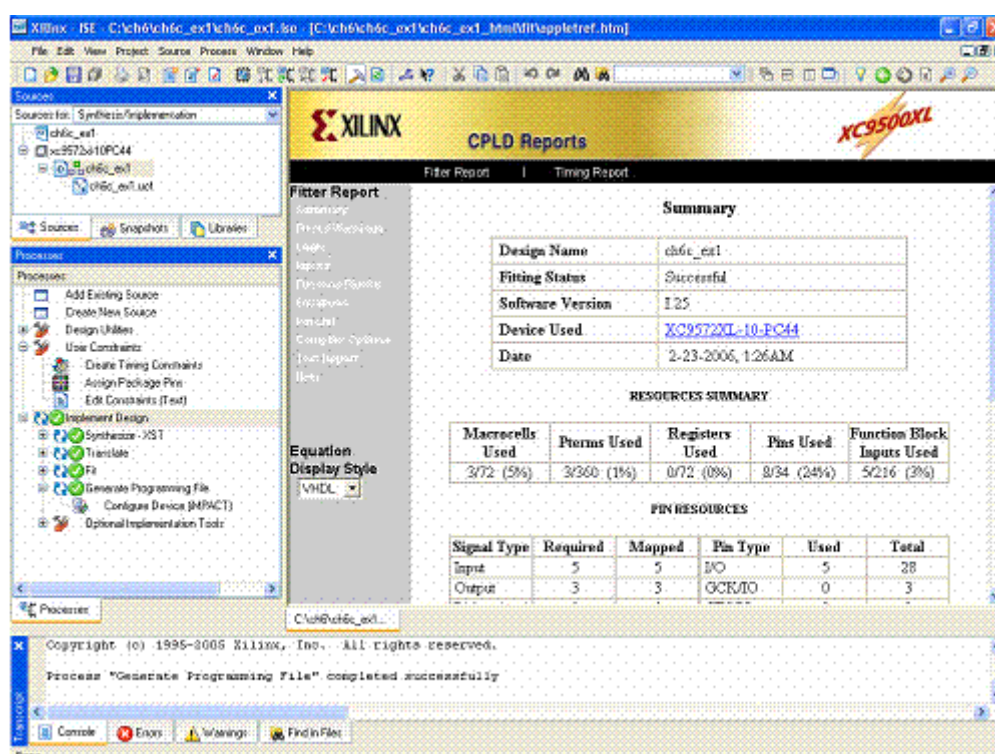
ภาพที่ 41 สัญญาณที่ได้จากการจำลองการทำงานของวงจรที่ออกแบบ

โปรแกรมที่ช่วยสร้างชุดข้อมูลที่ใช้ในการจำลองการทำงาน คือ HDL Bencher ที่ผู้ใช้สามารถสั่งป้อนสัญญาณขาของอินพุตต่างๆ ให้เป็นค่าตามที่ต้องการได้ในรูปแบบที่เรียกว่า Test Bench Waveform (ไฟล์ .tbw) ก่อนจะเรียกใช้โปรแกรม ISE Simulator หรือ ModelSim เพื่อแสดงผลลัพธ์ให้ผู้ทดสอบทราบในรูปของ Waveform ซึ่งในที่นี้ HDL Bencher ที่มีอยู่ใน ISE Simulator ติดตั้งมาพร้อมกับโปรแกรม ISE WebPACK 8.1i อยู่แล้ว จึงสามารถนำมาใช้จำลองการทำงานได้

Design Implementation

ในกรณีของ FPGA ขั้นตอนนี้เริ่มจากขั้นตอนการแปล (Translate) โดยนำไฟล์ Design Net list มาทำการ Optimize วงจร เพื่อตรวจสอบว่าสามารถวางหรือบรรจุวงจรลงในชิพเบอร์ที่เรากำหนดได้หรือไม่ ขั้นตอนต่อไปเป็นการ Map โดยการเลือกอุปกรณ์จากไฟล์ของวงจรเข้าไปวางในอุปกรณ์พื้นฐานต่างๆ ที่อยู่ภายใน FPGA เบอร์ที่เรากำหนด โดยจะต้องมีการคำนวณหาตำแหน่งที่เหมาะสมเพื่อที่จะนำเอาวงจรไปวาง (Place) และเมื่อวางเรียบร้อยแล้วจึงเชื่อมต่อสัญญาณต่างๆ เข้าด้วยกัน (Route) ตามลำดับ โดยใช้ XST ซึ่งข้อมูลวงจรที่ใช้ในขั้นตอนนี้จะได้มาจาก Design Net

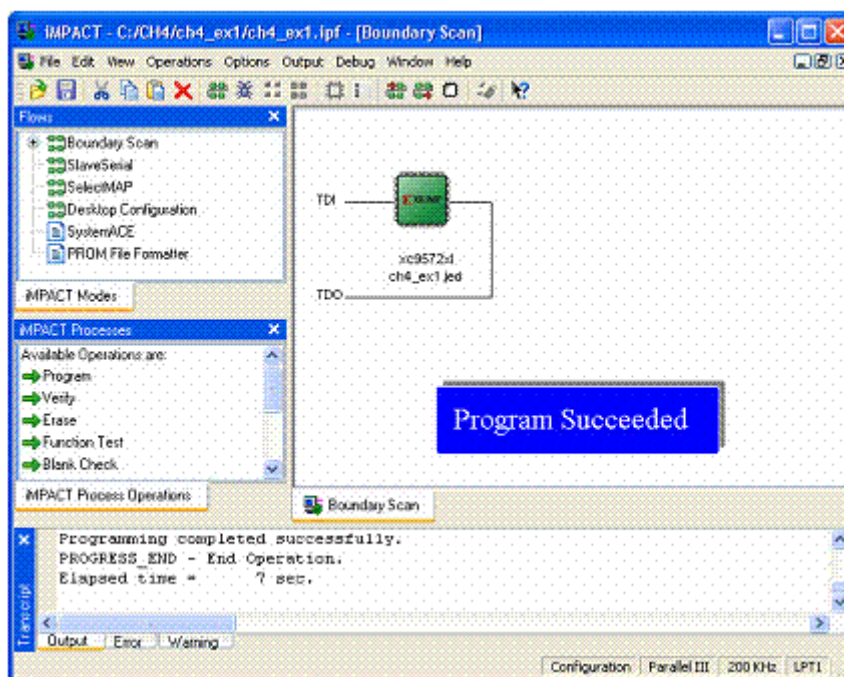
list ที่ได้จาก Design Entry ที่เป็น Schematic หรือจากการสังเคราะห์วงจรในกรณีที่เป็น HDL ส่วนกรณีของ CPLD จะมีความซับซ้อนน้อยกว่ามาก จึงเรียกรวมขั้นตอนหลังจากขั้นตอนการแปลว่าขั้นตอน Fitting จากนั้นจะเป็นขั้นตอน Bit Stream Generation เพื่อสร้างไฟล์ให้เหมาะสำหรับการ Download ลงชิพ โดยจะได้เป็นไฟล์นามสกุล .bit (ไฟล์ข้อมูลวงจรของ FPGA) หรือไฟล์นามสกุล .jed (ไฟล์ข้อมูลวงจรของ CPLD)



ภาพที่ 42 การเตรียมไฟล์ข้อมูลสำหรับ Download ลงชิพ CPLD

Device Programming

การโปรแกรมข้อมูลวงจรลงชิพนั้น สามารถทำได้โดยนำไฟล์ที่มีนามสกุล .bit (ไฟล์ข้อมูลวงจรของ FPGA) หรือไฟล์นามสกุล .jed (ไฟล์ข้อมูลวงจรของ CPLD) ที่ได้ในขั้นตอน Design Implementation หรือขั้นตอน Generate Programming มา Download ลงชิพ โดยใช้เครื่องโปรแกรม หรือใช้สาย Download ก็ได้ ซึ่งงานวิจัยครั้งนี้ผู้วิจัยใช้สาย Download มาตรฐาน JTAG ในการโปรแกรมข้อมูลลงชิพ CPLD



ภาพที่ 43 การ โปรแกรมข้อมูลลงชิพ CPLD

เมื่อ Download ข้อมูลลงบนชิพ CPLD เรียบร้อยแล้ว โปรแกรมจะแสดงรายละเอียดการ Download เป็นการบอกสถานะของการ Download โดยจะมีข้อความ “Program Succeeded” สำหรับการ Download ที่ถูกต้อง และข้อความ “Program Failed” เมื่อมีความผิดพลาดในการ Download หลังจากนั้นผู้ใช้สามารถนำ CPLD ไปใช้งานในการออกแบบวงจรรวมตามต้องการ