บทที่ 8
สรุปผลและข้อเสนอแนะ

8.1 สรุปผล

วิทยานิพนธ์นี้แสดงการสร้างและพัฒนา PLC ขนาดเล็กมีความเร็วสูงสามารถใช้งานในอุตสาหกรรมจริงได้ สร้างและพัฒนา CPU ของ PLC ขึ้นเองใช้ FPGA ตระกูล FLEX 10K เบอร์ EPF10K70RC240-4 ที่มีความความจุเกต 70,000 เกต ผลการจำลองการทำงาน (Simulate) ได้ผลดี และการทำ PLC จริงใช้บอร์ด UP1X ประกอบกับอินพุต/เอาต์พุตบอร์ด โดย PLC ต้นแบบที่ได้มีจำนวน 16 อินพุต 8 เอาต์พุต คำสั่งทั้งหมด 23 คำสั่งแบ่งเป็นคำสั่งพื้นฐาน 19 คำสั่งความเร็วเฉลี่ย 0.31 ไมโครวินาที และมีจำนวนคำสั่งจัดการข้อมูลอีก 4 คำสั่งความเร็วเฉลี่ย 0.66 ไมโครวินาที จำนวนขั้นของโปรแกรมขั้นบันไดมีได้สูงสุด 384 ขั้น ซึ่งก็มีความสามารถเพียงพอในการใช้งานอุตสาหกรรมจริงสำหรับ PLC ขนาดเล็ก

จากการทดสอบการทำงานทุกคำสั่งของ PLC ทำงานได้อย่างถูกต้อง และจากการทดสอบกับโปรแกรมขั้นบันไดตัวอย่าง PLC ก็สามารถทำงานได้อย่างถูกต้องทั้งในด้านความถูกต้องในการดำเนินการตามคำสั่งและในด้านความเร็ว โดยช่วงเวลา Scan time จากการคำนวณและการวัดมีค่าตรงกัน

จุดเด่นของ PLC นี้ คือการรวมส่วนต่างๆ ที่สำคัญของ PLC ไว้ใน PLC คอนโทรลเลอร์บนชิป FPGA ตัวเดียวให้มากที่สุด ทำให้ PLC มีขนาดเล็กและมีความเร็วในการทำงานสูง นอกจากนี้ยังอาศัยจุดเด่นของเทคโนโลยี FPGA ในการลดเวลาการสร้างและพัฒนา PLC ต้นแบบ และง่ายในการเพิ่มฟังก์ชันการทำงานของ PLC ต่อไปในอนาคตอีกด้วย ซึ่งทำให้ PLC ขนาดเล็กที่สร้างมาจาก FPGA มีความสามารถมากและขนาดเล็กกว่า PLC ที่สร้างมาจากไมโครโพรเซสเซอร์ อีกทั้งง่ายในการเพิ่มฟังก์ชันการทำงาน เพิ่มจำนวนคำสั่งอีกด้วย แต่เมื่อพัฒนาเป็น PLC ที่มีขนาดใหญ่ขึ้น มีจำนวนอินพุต/เอาต์พุตและจำนวนคำสั่งมากขึ้น จะทำให้จำนวน state machine มีมากขึ้น จะมีปัญหาเกี่ยวกับการหน่วงเวลา (Timing) ทำให้ไม่สามารถจะทำงานที่สัญญาณนาฬิกา (clock) ความถี่สูงได้

8.2 ปัญหาและข้อเสนอแนะ

ปัญหาที่พบคือ PLC คอนโทรลเลอร์ที่พัฒนาขึ้นนี้คือไม่สามารถจะทำงานที่ความถี่สูงสุด 25.17 MHz ของบอร์ดทดลอง UP1X ได้ เนื่องผลของการจำลองการทำงานในโปรแกรม MAX PLUS II แสดงว่าค่าการหน่วงเวลามีค่ามากเกินกว่าที่จะทำงานที่ 25.17 MHz ได้ จึงได้ทำการแก้ปัญหาเริ่มต้นด้วยการแก้ไขที่โปรแกรม VHDL โดยเขียนโปรแกรมให้มีประสิทธิภาพมากที่สุด พยายามลดจำนวน state machine ที่ไม่จำเป็นออก ผลที่ได้ดีขึ้นแต่ก็ยังไม่พอ สุดท้ายจึงต้องลดความถี่สัญญาณนาฬิกาลงครึ่งหนึ่งเหลือเป็น 12.58 MHz PLC คอนโทรลเลอร์นี้จึงทำงานได้

PLC คอนโทรลเลอร์ยังมีจุดที่ควรปรับปรุงและพัฒนาต่อไป เพื่อเพิ่มเติมความสามารถให้ดียิ่งขึ้น โดยมีข้อเสนอแนะดังต่อไปนี้

1. ทดสอบคำสั่งอย่างละเอียดเพื่อทดสอบว่าคำสั่งทำงานถูกต้องไม่มีข้อผิดพลาดเมื่อใช้ในการควบคุม ปรับปรุงเพิ่มความสามารถของคำสั่งเดิมพร้อมเพิ่มคำสั่งใหม่ เช่น Off Delay Time, Reversible Counter และคำสั่งด้านการจัดการข้อมูล (Data Handling) เช่น Data Exchange, Move ระหว่างรีจิสเตอร์ เป็นต้น ซึ่งตามโครงสร้าง Instruction Machine ที่ได้ออกแบบไว้ตามหัวข้อ 4.11 สามารถรองรับคำสั่งได้สูงสุดถึง 46 คำสั่ง
2. PLC คอนโทรลเลอร์ที่พัฒนาขึ้นประกอบด้วยฟังก์ชันการทำงานพื้นฐานเพื่อให้ PLC ทำงานได้แล้ว แต่ยังสามารถเพิ่มฟังก์ชันตรวจสอบการโปรแกรมในหน่วยความจำภายในว่าถูกต้องหรือไม่ เพิ่มฟังก์ชันในการอ่านโปรแกรมขั้นบันไดจากหน่วยความจำโปรแกรมกลับไปที่คอมพิวเตอร์ส่วนบุคคลเพื่ออ่านหรือแก้ไข เป็นการเพิ่มประสิทธิภาพการทำงานของ PLC ให้มากขึ้น

   แต่การเพิ่มจำนวนคำสั่งตามข้อที่ 1 และเพิ่มฟังก์ชันการทำงานตามข้อที่ 2 ผลที่ตามมาคือจำนวน State Machine ของ CPU จะมากขึ้น จะมีผลโดยตรง 2 ส่วนคือ

   a. ต้องการ FPGA ที่มีจำนวนเกตมากขึ้น
   b. จำนวนเกตที่ต่ออนุกรมกันในวงจรภายใน FPGA จะมีมากขึ้น ซึ่งแต่ละเกตจะมีการหน่วงเวลา เมื่อเกตต่ออนุกรมกันมากผลรวมการหน่วงเวลา

ก็จะมีมากด้วย ถ้ามีค่ามากเกินกว่าสัญญาณนาฬิกาที่ใช้จะทำให้การทำงานทั้งหมดจะเกิดผิดพลาด [11] ซึ่งแก้ไขได้ด้วยการเขียนโปรแกรมให้สั้นและมีประสิทธิภาพ มีจำนวน State Machine ให้น้อยที่สุด เลือกชิปที่มีความเร็วสูงขึ้น (Delay time ของเกตภายในแต่ละตัวมีค่าน้อย) หรือทางแก้สุดท้ายคือลดสัญญาณนาฬิกาที่ใช้ลง

3. เพิ่มความเร็วของคำสั่งโดยลดสถานะการทำงานของแต่ละคำสั่งให้น้อยที่สุด ในส่วนนี้ต้องมีการศึกษาภาษา VHDL ให้มากขึ้น เพื่อเรียนรู้เทคนิคเขียนโปรแกรมให้มีประสิทธิภาพสูงขึ้น โดยใช้ทรัพยากรให้น้อยที่สุด
4. ควรเพิ่มความสามารถในการรับจำนวนขั้น (Step) ของโปรแกรมขั้นบันไดให้มากขึ้นอย่างน้อยได้ 1000 ขั้น, เพิ่มจำนวน อินพุต/เอาต์พุต ให้มากเป็น 32 อินพุต 16 เอาต์พุต ในส่วนนี้ต้องพิจารณาเลือกใช้ FPGA ที่มีขนาดหน่วยความจำภายในใหญ่ขึ้น
5. ในส่วนโปรแกรมซอฟต์แวร์พัฒนาโปรแกรมขั้นบันได ก็ยังสามารถเพิ่มความสามารถให้สูงขึ้นเพื่อที่จะให้ผู้ใช้ทำงานได้มีประสิทธิภาพมากขึ้น เช่น สามารถเปลี่ยนรูปแบบการโปรแกรมจากภาษาแบบขั้นบันไดเป็นโปรแกรมภาษานิวโมนิค หรือเปลี่ยนเป็นภาษา Logic Diagram หรือให้ผู้ใช้สามารถเลือกระบบภาษาในการโปรแกรมได้ และเพิ่มความสามารถให้การจำลองการทำงานของโปรแกรมขั้นบันไดที่เขียนถึง ก่อนที่จะโหลดโปรแกรมลง PLC จริง ทำให้ผู้ใช้ง่ายในการตรวจสอบโปรแกรม หรือลดความเสียหายที่อาจจะเกิดขึ้นเมื่อโปรแกรมที่เขียนทำงานไม่ถูกต้อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย