บทที่ 4

การออกแบบส่วนควบคุม

ไมโครโพรเซสเซอร์นอกจากจะมีส่วนเส้นทางข้อมูลเพื่อทำงานตามคำสั่งต่างๆ แล้ว ยังจำเป็นต้องมีส่วนควบคุมเพื่อควบคุมการทำงานด้วย ส่วนควบคุมถือว่าเป็นส่วนที่มีบทบาทสำคัญต่อการทำงานและสมรรถนะที่แท้จริงของตัวไมโครโพรเซสเซอร์มาก ถ้าหากส่วนควบคุมสามารถควบคุมการวิ่งของสัญญาณที่วิ่งอยู่ในไมโครโพรเซสเซอร์ทั้งส่วนเส้นทางและส่วนควบคุมเองได้ดีก็จะทำให้ไมโครโพรเซสเซอร์ที่ออกแบบมีสมรรถนะสูงขึ้นได้อีกพอสมควร วิธีการที่ใช้ในการออกแบบส่วนควบคุมก็ส่งผลต่อความเร็วของไมโครโพรเซสเซอร์มากเช่นเดียวกัน

4.1 การออกแบบส่วนควบคุม

การออกแบบส่วนควบคุมนั้นโดยทั่วๆ ไปมีอยู่ 2 วิธี คือ Hardwired Control และ Microprogrammed Control การออกแบบด้วย Hardwired Control นั้นเป็นวิธีการที่ตรงไปตรงมาสามารถทำได้โดยการออกแบบวงจรตามลักษณะการทำงานของแต่ละคำสั่ง ดูแต่ละคำสั่งว่าต้องทำอะไรบ้างจึงจะทำงานนั้นได้อย่างสมบูรณ์แล้วออกแบบการทำงานไปทีละขั้นตอน การออกแบบส่วนควบคุมด้วยวิธีนี้จะส่งผลทำให้ไมโครโพรเซสเซอร์ที่ได้มีความเร็วกว่าการออกแบบด้วยวิธี Microprogrammed Control มาก แต่ก็มีข้อเสียเปรียบอยู่หลายประการเช่นกัน คือมีความยุ่งยากในการออกแบบมาก อีกทั้งยังมีความไม่สะดวกในการปรับเปลี่ยนการทำงานในอนาคต เช่น การเพิ่มเติมหรือลดทอนคำสั่งอีกด้วย โดยเฉพาะอย่างยิ่งถ้าเป็นไมโครโพรเซสเซอร์ที่มีชุดคำสั่งที่มีความสลับซับซ้อนมาก การออกแบบด้วยวิธีนี้จึงเป็นทางเลือกที่ไม่เหมาะสมเท่าใดนัก ดังนั้นโดยมากแล้วไมโครโพรเซสเซอร์รุ่นใหม่ๆ นิยมออกแบบด้วยวิธี Microprogrammed Control มากกว่า การออกแบบด้วยวิธี Microprogrammed Control มีความง่ายและสะดวกกว่าวิธี Hardwired Control มาก แต่ความเร็วก็ต่ำกว่ามากเช่นกัน อย่างไรก็ตามความเร็วที่ต่ำกว่านั้นเป็นสิ่งที่ยอมรับได้เมื่อเทียบกับความสะดวกในการปรับเปลี่ยนชุดคำสั่งหรือลักษณะการทำงานในอนาคตของวิธีการออกแบบด้วย Microprogrammed Control ซึ่งเป็นจุดที่ต้องคำนึงมากในการใช้งานจริง ดังนั้นไมโครโพรเซสเซอร์ที่ออกแบบในวิทยานิพนธ์นี้จึงเลือกใช้วิธีการออกแบบด้วย Microprogrammed Control

Microprogrammed Control นั้นมีแนวคิดเช่นเดียวกับการเขียนโปรแกรม การออกแบบเริ่มจากการดูจำนวนสายสัญญาณควบคุมว่ามีจำนวนเท่าใดเพื่อนำมากำหนดรูปแบบของ

Microinstruction ซึ่ง Microinstruction แต่ละชุดก็คือสายสัญญาณที่ต้องส่งไปควบคุมที่ส่วนเส้นทางข้อมูลนำมาเรียงเข้าเป็นกลุ่มกันนั่นเอง ถ้าแทนค่า 1 เป็นการทำงานและ 0 เป็นการไม่ทำงาน Microinstruction แต่ละชุดก็คือกลุ่มของค่า 1 และ 0 ของสายสัญญาณที่ถูกเก็บไว้ใน ROM เรียกว่า Control Storage เมื่อมีการเรียกให้ทำงานก็จะส่งค่าที่ตัวเองเก็บไว้ทั้งหมดไปที่สายสัญญาณที่ส่งไปควบคุมที่ส่วนเส้นทางข้อมูล Microinstruction แต่ละชุดจะส่งผลให้สายสัญญาณแต่ละเส้นทำงานไม่เหมือนกัน เช่น ถ้าเป็นการ Fetch ก็จะต้องเริ่มจากการส่งสัญญาณให้มีการส่งข้อมูลจากรีจิสเตอร์ PC ไปยังรีจิสเตอร์ MAR ก่อน เสร็จแล้วก็เรียกให้ Microinstruction ต่อไปทำงาน โดยที่ตัวต่อไปก็สั่งให้มีการส่งข้อมูลจากรีจิสเตอร์ MAR ไปที่ ROM เพื่อ Fetch ค่ามาเก็บไว้ที่รีจิสเตอร์ MI แล้ว Microinstruction ต่อไปก็สั่งให้รีจิสเตอร์ MI ส่งค่าไปที่รีจิสเตอร์ IR เพื่อส่งมาที่ส่วนควบคุม Decode และ Execute ต่อไป อย่างนี้เป็นต้น จากวิธีการดังที่กล่าวมาจะเห็นว่า Microinstruction แต่ละตัวก็จำเป็นต้องมีตัวควบคุมให้ทำงานเป็นลำดับก่อนหลังเช่นกัน ตัวควบคุมที่ว่านี้เรียกว่า Microsequencer ซึ่งทำหน้าที่ควบคุมลำดับก่อนหลังของการทำงาน กำหนดให้มีตัวชี้ไปยังตำแหน่งต่างๆ ใน Control Storage ที่เก็บ Microinstruction ทั้งหมดไว้ เมื่อสั่งให้ตัวชี้ไปที่ใดก็เป็นการเรียกใช้งาน Microinstruction นั้น

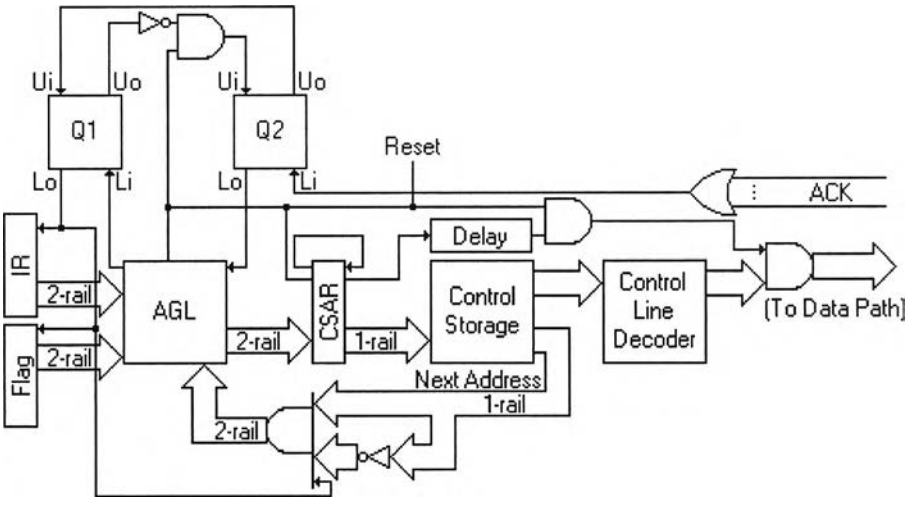
จากลักษณะการทำงานดังกล่าวจะเห็นว่าเหมือนกับการทำงานของโปรแกรม โดยมี Microsequencer คอยจัดการให้ตัวชี้ที่เปรียบไปก็เหมือนกับ Pointer ที่ชี้ไปยังตำแหน่งต่างๆ ใน Control Storage แต่ละตำแหน่ง มีผลให้การควบคุมส่วนเส้นทางข้อมูลทำงานแตกต่างกัน การออกแบบด้วยวิธีการนี้มีประสิทธิภาพมาก เพราะว่ามีความง่ายต่อการออกแบบและสะดวกในการปรับเปลี่ยนคำสั่งในภายหลังกว่าวิธีการแบบ Hardwired Control มาก การออกแบบส่วนควบคุมด้วยวิธี Microprogrammed Control นี้จึงได้รับความนิยมมากกว่าแม้ว่าจะทำงานได้ช้ากว่าก็ตาม

4.2 โครงสร้างของส่วนควบคุม

ไมโครโพรเซสเซอร์ที่ออกแบบนี้มีสายสัญญาณควบคุมที่ส่งจากส่วนควบคุมไปยังส่วนเส้นทางข้อมูลอยู่ 42 เส้นซึ่งได้แสดงรายละเอียดไว้ในภาคผนวก ง แล้ว และมีสายสัญญาณตอบรับที่ส่งจากส่วนเส้นทางข้อมูลกลับมายังส่วนควบคุมเพื่อระบุการสิ้นสุดรอบการทำงานอีก 27 เส้นซึ่งก็ได้แสดงรายละเอียดไว้ในภาคผนวก ง แล้วเช่นกัน

การออกแบบส่วนควบคุมนั้นถือว่ามีความยุ่งยากและซับซ้อนกว่าการออกแบบในส่วนเส้นทางข้อมูลมาก ดังนั้นในการนี้จึงได้กำหนดให้การทำงานในแต่ละรอบเป็นไปเพียงทีละขั้นเพื่อความง่ายและสะดวกในการควบคุมสัญญาณตอบรับที่ส่งกลับมา โดยในรอบการทำงานแต่ละรอบ

จะส่งผลให้การทำงานส่งสัญญาณตอบรับกลับมาเพียงเส้นเดียวเท่านั้น โครงสร้างของส่วนควบคุมเป็นดังรูปที่ 4.1



รูปที่ 4.1 ส่วนควบคุม

จากรูปที่ 4.1 ส่วนควบคุมประกอบไปด้วยส่วนหลักๆ ดังต่อไปนี้

1. อุปกรณ์ชนิดคิว ต้องใช้ 2 ตัวเพื่อควบคุมการทำงานให้ทำเป็นรอบๆ และเป็นการทำงานแบบกลับสู่ศูนย์ มีทั้งขั้นทำงานและขั้นว่าง
2. AGL (Address Generation Logic) จะทำหน้าที่เป็น Microsequencer และจะมีตัวถอดรหัสคำสั่ง (Instruction Decoder) อยู่ภายในด้วย จะรับอินพุตมาจากรีจิสเตอร์ IR และ Flag จากทางส่วนเส้นทางข้อมูล และยังมีอินพุตจากในส่วนควบคุมเอง ก็คือจาก Control Storage เป็นค่าที่ระบุตำแหน่งถัดไปใน Control Storage ที่ต้องทำงาน และค่าที่ใช้ในการควบคุมลำดับการทำงานของส่วนควบคุม ทางเอาท์พุตของตัว AGL จะมีรีจิสเตอร์ชั่วคราวอยู่ภายในเพื่อเก็บค่าก่อนส่งค่าออกเป็นเอาท์พุต
3. CSAR (Control Storage Address Register) เป็นตัวส่งค่าตำแหน่งของ Control Storage ที่จะต้องให้ทำงาน โดยที่จะส่งเป็นรหัสรางเดี่ยวไม่ใช่รางคู่เนื่องจาก Control Storage นั้นใช้เป็นรหัสรางเดี่ยวเหมือนกับวงจรสมวาร
4. Control Storage เป็น ROM ที่เก็บ Microinstruction ทั้งหมดที่ต้องใช้ควบคุมการทำงานของส่วนเส้นทางข้อมูลเอาไว้ โดยจะเป็นการทำงานตามชุดคำสั่งทั้งหมดตามที่ได้แสดงไว้ในภาคผนวก ก Control Storage เป็น ROM 43 บิตที่มีขนาด 256 ไบต์ ซึ่ง Microinstruction ที่ออกแบบได้แสดงในภาคผนวก จ ไว้แล้ว และรายละเอียดของ

Microprogrammed Control Storage ซึ่งเป็น Source Code ภาษา Verilog ก็ได้แสดงไว้ในภาคผนวก ฉ แล้ว

5. Control Line Decoder เป็นส่วนที่ใช้ถอดรหัสสัญญาณที่มาจาก Control Storage เนื่องจากอุปกรณ์บางตัวนั้นมีอินพุตมาจากอุปกรณ์อื่นหลายตัว แต่สัญญาณอินพุตจะเข้ามาครั้งละ 1 ตัวเท่านั้น ถ้ามีสัญญาณอินพุตเข้ามามากกว่า 1 ตัวสัญญาณจะชนกันและทำให้ทำงานผิดพลาด ดังนั้นจึงสามารถนำสายสัญญาณควบคุมที่ส่งเข้ามาทางด้านอินพุตมาจัดเป็นกลุ่มเดียวกันได้ จะสามารถลดขนาดของ Control Storage ได้พอสมควร ทำให้ไม่เป็นการสิ้นเปลือง เช่น ถ้าสัญญาณที่เข้ายังรีจิสเตอร์ Acc มีอยู่ 3 เส้น สามารถใช้เพียง 2 บิตในการเก็บค่าได้โดยนำมาเข้ารหัสไว้แล้วเก็บใน Control Storage เมื่อจะใช้งานก็ถอดรหัสออกมาว่าเป็นสัญญาณเส้นใดที่เข้ามา ยิ่งถ้ามีสัญญาณมากเส้นก็จะสามารถประหยัดไปได้มาก
6. ส่วนหน่วงการทำงาน ประกอบด้วย D Flipflop 2 ตัวเพื่อถ่วงการทำงานรอไว้ 2 สัญญาณนาฬิกา เพื่อให้แน่ใจว่าข้อมูลได้ส่งเข้าไปให้ตัว Control Storage เรียบร้อยแล้ว เพราะ Control Storage เป็น ROM จึงต้องมีความหน่วงในการเข้าถึงอยู่ด้วย

นอกจากส่วนประกอบหลักที่กล่าวไปแล้วก็ยังมีส่วนย่อยอื่นๆ อีก เพื่อให้การควบคุมทำได้อย่างสมบูรณ์และมีความถูกต้องเชื่อถือได้ เช่น เกต AND ที่เห็นในรูป 4.1 นั้นส่วนใหญ่ใช้งานเปรียบเสมือนเป็นสวิตช์ที่คอย Enable สัญญาณให้ออกหรือไม่ให้ออกไป และค่าตำแหน่งของ Control Storage ที่ส่งกลับมาที่ AGL ในแต่ละรอบการทำงานนั้นเป็นรหัสรางเดี่ยว ก็ต้องทำให้เป็นรหัสรางคู่ เป็นต้น

รีจิสเตอร์ที่ใช้ในส่วนควบคุมนั้นใช้ตามโมเดลที่ออกแบบ กล่าวคือถ้าเป็นโมเดลที่ไม่ไวต่อความหน่วงชนิดเสมือนก็ใช้รีจิสเตอร์แบบรูปที่ 3.4 ถ้าเป็นโมเดลที่ไม่ไวต่อความหน่วงชนิดปรับมาตราส่วนได้ก็เป็นรีจิสเตอร์อีกแบบรูปที่ 3.5 ตามที่ได้เสนอไปแล้วในบทที่ 3

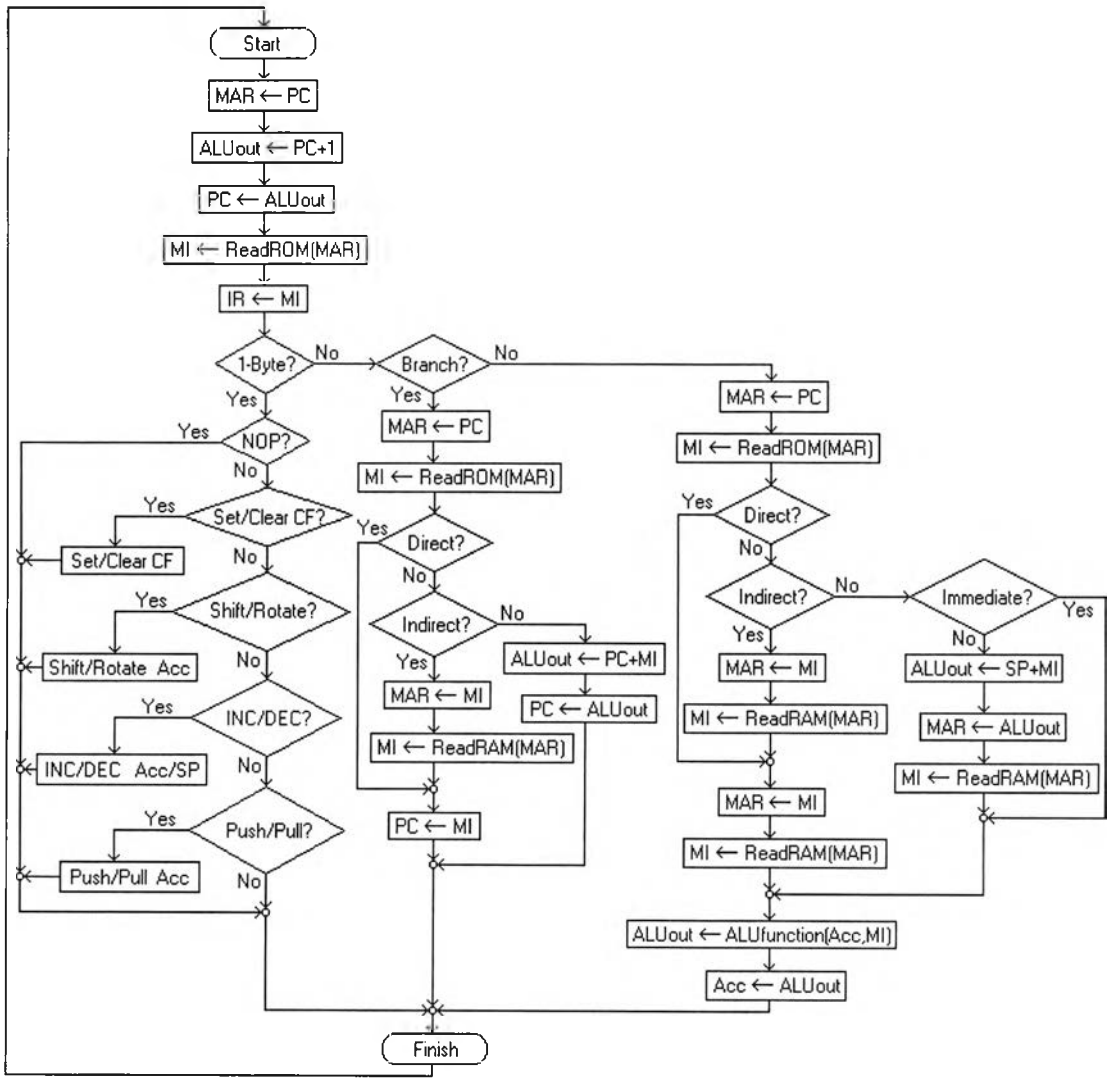
4.3 การทำงานของส่วนควบคุม

ไมโครโพรเซสเซอร์นั้นจะเริ่มทำงานที่ส่วนควบคุมก่อน จะเริ่มทำงานหลังจากกดปุ่มรีเซ็ตแล้ว ซึ่งมีขั้นตอนต่างๆ ดังต่อไปนี้

I20341039

1. เมื่อกดปุ่มรีเซ็ต รีจิสเตอร์ทุกตัวจะถูกรีเซ็ตค่าเป็น 0 ทั้งหมด
2. เมื่อปล่อยปุ่มรีเซ็ต ตัวแรกที่เริ่มทำงานคือ Q2 โดยจะส่งสัญญาณไปที่ Lo ให้ค่า 0 ออกมาจากรีจิสเตอร์ชั่วคราวใน AGL
3. ค่าจะส่งผ่านจาก AGL ไป CSAR เพื่อเปลี่ยนข้อมูลเป็น 1-rail เข้า Control Storage
4. Control Storage ตำแหน่งที่ 0 จะเริ่มทำงาน ซึ่งก็คือการ Fetch
5. Control Storage ส่งสัญญาณไปควบคุม พร้อมกับส่ง Address ถัดไปกลับไปให้ AGL
6. เมื่อมีสัญญาณตอบรับเป็น 1 กลับมาจะถูกส่งต่อไปที่ Li ของ Q2
7. เมื่อมีสัญญาณ 1 เข้ามาที่ Li จะส่งผลให้ Lo กลับเป็น 0
8. เมื่อ Lo กลับเป็น 0 จะทำให้สัญญาณต่างๆ ที่ส่งไปควบคุมเป็น 0 ทั้งหมด ซึ่งนี่ก็คือ Idle Phase
9. เมื่อสัญญาณ 0 ส่งมาถึง Li จะส่งผลให้ Uo ของ Q2 รวมทั้ง Ui และ Lo ของ Q1 ขึ้นเป็น 1 ด้วยตามลำดับ
10. Lo ของ Q1 จะเป็นสัญญาณ Request ให้รีจิสเตอร์ IR และ Flag ส่งค่าเข้ามาที่ AGL
11. เมื่อ AGL ได้รับค่าจะสัญญาณตอบรับไปที่ Li ที่ Q1
12. เมื่อ Q1 ได้สัญญาณจาก Li แล้วจะทำให้ Lo กลับเป็น 0
13. เมื่อ Lo กลับเป็น 0 จะส่งผลให้ Li กลับเป็น 0 เช่นกัน
14. เมื่อทั้ง Li และ Lo กลับเป็น 0 แล้ว Uo จะกลายเป็น 1
15. เมื่อ Uo เป็น 1 จะทำให้ Ui ของ Q2 กลายเป็น 0
16. เมื่อ Ui เป็น 0 จะทำให้ Uo เป็น 0 ส่งกลับไปที่ Q1
17. ที่ Q1 เมื่อ Ui เป็น 0 จะทำให้ Uo กลายเป็น 0
18. เมื่อ Uo เป็น 0 จะส่งผลให้ Ui ของ Q2 กลายเป็น 1 เริ่มการทำงานรอบต่อไป

แผนผังการทำงานหลักของไมโครโพรเซสเซอร์ได้แสดงไว้ดังรูปที่ 4.2



รูปที่ 4.2 แผนผังการทำงานหลักของไมโครโพรเซสเซอร์