บทที่ 5

การวางโครงร่างของงาน

เนื่องจากโปรแกรมจำลองซีพียู 8 บิตทั่วไป เป็นโปรแกรมที่มีการทำงานใน 2 ระดับคือระดับแรกเป็นการรับข้อมูลจำเพาะต่างๆ ที่จำเป็นต่อการทำงานจากผู้ใช้ โดยผู้ใช้สามารถกำหนดลักษณะจำเพาะของซีพียูที่ตนต้องการได้เองภายใต้ขีดจำกัดของโปรแกรม ดังนั้นก่อนใช้งานโปรแกรมจำลองการทำงานของซีพียู ผู้ใช้งานจำเป็นต้องป้อนข้อมูลเกี่ยวกับการทำงานของซีพียูตามที่ต้องการเข้าไปในโปรแกรมก่อน เมื่อโปรแกรมรับข้อมูลต่างๆ ที่จำเป็นเข้ามาแล้วก็จะสามารถจำลองการทำงานของซีพียูได้ตามที่ผู้ใช้กำหนด การทำงานในระดับที่สองก็คือ การจำลองการทำงานของซีพียูให้ได้ตามกำหนดของผู้ใช้และสามารถแสดงข้อมูลภายในจากการทำงานให้ผู้ใช้เห็นได้เพื่อตรวจสอบการทำงานของซีพียู ฉะนั้นโครงร่างหลักของงานจึงแบ่งเป็น 2 ส่วนได้แก่

1. Simulator Editor สำหรับโต้ตอบกับผู้ใช้ในการรับข้อมูลจำเพาะเพื่อจำลองการทำงานของซีพียู และเก็บข้อมูลเป็นแฟ้มข้อมูลไว้ใช้ในการจำลองการทำงานของซีพียู

2. CPU Simulator ทำหน้าที่จำลองการทำงานของซีพียูตามความต้องการของผู้ใช้จากกลไกการทำงานของซีพียูแบบทั่วไปที่อยู่ภายใน ข้อมูลจากโปรแกรมส่วนที่หนึ่งสามารถเปลี่ยนแปลงแก้ไขได้ ซึ่งเมื่อข้อมูลส่วนนี้เปลี่ยนแปลงไปก็จะส่งผลถึงส่วนที่สองทุกครั้ง คือทำให้โปรแกรมจำลองการทำงานของซีพียู ทำงานในการจำลองอย่างไม่ตายตัวซึ่งส่วนแสดงผลที่จอภาพก็เช่นเดียวกัน

Simulator Editor

โปรแกรมในส่วนนี้จะทำหน้าที่รับ/เก็บข้อมูลทุกอย่าง ที่อนุญาตให้ผู้ใช้เป็นผู้กำหนดหรือออกแบบซีพียู ได้แก่ จำนวน ชื่อ ของรีจิสเตอร์ใช้งาน ชุดคำสั่งของซีพียู ลักษณะการอ่าน/เขียนข้อมูลกับหน่วยความจำ รูปแบบของภาษาแอสเซมบลีในการเขียนโปรแกรมสั่งงานซีพียูเป็นต้น

1. Register Editor

ให้ผู้ใช้สามารถกำหนดชื่อรีจิสเตอร์ 8 บิต 16 บิต และแฟล็กรวมทั้งจำนวนที่ต้องการ แบ่งเป็น 3 ส่วน

1.1 8-bit Register Editor

1.2 16-bit Register Editor

1.3 Flag Editor

2. Assembler Editor

ให้ผู้ใช้กำหนดรูปแบบและสัญลักษณ์ของภาษาแอสเซมบลีซึ่งเกี่ยวข้องกับการทำงานของแอสเซมเบลอร์ในการแปลแอสเซมบลีเป็นภาษาที่สั่งงานซีพียูได้ ซึ่งลักษณะการแปลของแอสเซมเบลอร์มีความสัมพันธ์โดยตรงกับคำสั่งของซีพียู

3. Addressing Recognition Editor

ให้ผู้ใช้กำหนดรูปแบบของแอดเดรสชิงโหมด ที่ปรากฏแตกต่างกันในภาษาแอสเซมบลีทุกโหมดที่มี ซึ่งเป็นกลไกที่จะทำให้แอสเซมเบลอร์ สามารถทำการแปลแอสเซมบลีได้สำเร็จ โดยรูปแบบที่จะให้ผู้ใช้กำหนด จะต้องได้รับการออกแบบมาจากการออกแบบกลไกการทำงานภายในของแอสเซมเบลอร์

4. Instruction Set Editor

ให้ผู้ใช้กำหนดชุดคำสั่งทำงานของซีพียู จะต้องมีการออกแบบชุดคำสั่งภายในของ Generic CPU เสียก่อนซึ่งเรียกว่า Generic Instruction Set ในการกำหนดคำสั่งทำงานของซีพียูโดยผู้ใช้ ผู้ใช้จะเลือกคำสั่งทำงานจาก Generic Instruction Set นี้มาใช้ในคำสั่งทำงานของซีพียูของผู้ใช้ ซึ่งอาจประกอบด้วย Generic Instruction คำสั่งเดียว หรือเป็นชุดของคำสั่งก็ได้ และการทำงานของคำสั่งที่มีความสัมพันธ์กับแอดเดรส-ชิงโหมด ก็จะต้องมีการกำหนดด้วย ดังนั้นจำเป็นต้องออกแบบ Generic Addressing

Modes ก่อนด้วย

ข้อมูลส่วนนี้จะถูกนำไปใช้ในการแปลของแอสเซมเบลอร์ และการทำงานของตัวแปลคำสั่ง (Instruction Interpreter) ซึ่งมีความสัมพันธ์กันด้วยแอดเดรส-ชิงโหมด

โปรแกรม Editor ทั้ง 4 ส่วน จะต้องได้รับการออกแบบให้ผู้ใช้เลือกทำงานป้อนข้อมูล ลบ/แก้ไข/เพิ่มเติม/เรียก/เก็บ ข้อมูลได้ โครงสร้างของข้อมูลมีความสัมพันธ์กับการเรียกไปใช้งานโดยโปรแกรมส่วนที่สองคือ CPU Simulator

CPU Simulator

โปรแกรมส่วนนี้คือ โปรแกรมจำลองการทำงาน จะต้องมีการออกแบบเพื่อรองรับข้อมูลโครงสร้างและลักษณะการทำงานของซีพียูที่เปลี่ยนแปลงได้จากภายนอกคือผู้ใช้ และมีกลไกการทำงานภายในที่สามารถจำลองการทำงานได้ตามการโปรแกรมการทำงานโดยผู้ใช้ ซึ่งได้กล่าวไว้แล้วในส่วนแรก

จากแนวความคิดของการออกแบบที่ได้จากการศึกษาและวิเคราะห์ปัญหาทำให้แบ่งงานในการออกแบบโปรแกรมส่วนนี้เป็น

1. การออกแบบโครงสร้างจำลองแบบทั่วไป
   - Register
   - Flags
   - Memory and IO
   - Common Registers
2. การออกแบบ Generic Instruction Set และ Generic Addressing Modes
3. การออกแบบกลไกการทำงานของ Generic Assembler
4. การออกแบบกลไกการทำงานของ Simulator ซึ่งสามารถแปลภาษาแอสเซมบลีและทำงานตามโปรแกรมภาษาแอสเซมบลีของผู้ใช้ได้

งานส่วนที่ 1-3 จะต้องทำไปพร้อมๆกันหมายถึงการออกแบบแต่ละส่วนจำเป็นต้องคิดเผื่อถึงการทำงานของอีก 2 ส่วนด้วยเสมอและยังต้องคำนึงถึงข้อมูลในงานส่วน

แรกคือ Simulator Editor ด้วย

โครงสร้างแบบ Generic

1. Registers and Flags ในส่วนนี้กำหนดให้ผู้ใช้ระบุความต้องการผ่าน Editor ได้ซึ่งทั้งรีจิสเตอร์และแฟล็กเป็นตัวแปรของคำสั่งทำงานของซีพียู เพราะการทำงานของซีพียูส่วนใหญ่ ก็คือการถ่ายเทข้อมูลระหว่างรีจิสเตอร์ และการประมวลผลข้อมูลในรีจิสเตอร์ ซึ่งผลของการทำงาน สะท้อนให้เห็นโดยการเปลี่ยนแปลงของข้อมูลในรีจิสเตอร์ และการเปลี่ยนแปลงสถานะของแฟล็ก

ฉะนั้นโครงสร้างข้อมูลในส่วนนี้จำเป็นต้องแบ่งเป็น 2 ส่วน คือ ส่วนที่จำกัด ซึ่งทำให้ชุดคำสั่ง Generic Instruction ภายใน สามารถจัดการข้อมูลในรีจิสเตอร์และแฟล็กได้ โดยมีข้อมูลส่วนที่สองเป็นสื่อกลางในการติดต่อกับผู้ใช้ภายนอก และข้อมูลส่วนที่สองนี้เป็นส่วนที่จะต้องมีความยืดหยุ่นคือสามารถเปลี่ยนแปลงได้โดยผู้ใช้

ความสำคัญในการออกแบบโครงสร้างข้อมูลรีจิสเตอร์ ก็คือ ให้มีโครงสร้างที่จำกัดและโครงสร้างที่ยืดหยุ่น ซึ่งการติดต่อกับคำสั่งทำงานจะติดต่อกับโครงสร้างที่จำกัด และการติดต่อกับผู้ใช้จะติดต่อกับโครงสร้างที่ยืดหยุ่น

สำหรับแฟล็กนั้นค่อนข้างจะยุ่งยากกว่า เนื่องจากเราจะต้องคำนึงถึงการทำงานของซีพียูในการประมวลผลอย่างละเอียดว่า การทำงานประเภทใด ส่งผลต่อการเปลี่ยนแปลงของแฟล็กใด และมีกลไกในการเปลี่ยนแปลง จากการเปลี่ยนแปลงของข้อมูลใน รีจิสเตอร์อย่างไร ซึ่งที่กล่าวถึงนี้คือ ส่วนที่จำกัดหรือเป็นส่วนที่เหมือนกันตามหลักการทำงานของซีพียู และเช่นเดียวกันกับโครงสร้างข้อมูลรีจิสเตอร์คือ จะต้องมีโครงสร้างส่วนที่ยืดหยุ่นเพื่อใช้สำหรับติดต่อกับผู้ใช้ได้

2. Common Registers, Memory, I/O Devices ส่วนนี้เป็นการกำหนด Resources ที่โปรแกรมกำหนดไว้ตายตัว เพราะจากการวิเคราะห์ข้อมูลแล้วเป็นส่วนที่ถือได้ว่าเป็นคุณสมบัติทั่วไปของซีพียูขนาด 8 บิต คือมี Memory Address ขนาด 16 บิต และ I/O device address ขนาด 8 บิต ถ้าเป็น Separated I/O

Generic Instruction Set and Addressing Modes

งานในส่วนนี้จะต้องผ่านการวิเคราะห์ข้อมูลมาอย่างถี่ถ้วนและจะต้องพิจารณาถึงความเป็นไปได้ของซีพียูที่จะจำลองขึ้นในอนาคตโดยผู้ใช้คนใดก็ได้รวมทั้งซีพียูที่มีอยู่แล้วด้วย

จากการศึกษาแนวความคิดของ Risc (Reduce Instruction Set Computer) สามารถนำมาประยุกต์กับการออกแบบ Generic Instruction Set ได้ เนื่องจากถ้าจะพิจารณากันแล้ว คำสั่งในการทำงานของซีพียูใดๆ ของซีพียูแบบใดเบอร์ใดก็ตามเมื่อแตกเป็นขั้นตอนโดยละเอียดแล้วก็หนีไม่พ้นหลักการทำงานภายในของซีพียูซึ่งไม่สลับซับซ้อน แต่มีการผสมผสานอย่างเป็นขั้นตอนระหว่างการทำงานและการถ่ายเทข้อมูลด้วยเงื่อนไขต่างๆ จึงทำให้เกิดผลของการทำงานที่สลับซับซ้อนมากขึ้นๆ

อย่างไรก็ตามแม้ว่าเราจะวางแผนที่จะออกแบบให้มีเฉพาะคำสั่งง่ายๆ ใน Generic Instruction Set เพียงไม่กี่คำสั่ง แต่ก็ยังต้องคำนึงถึงตัวแปรของคำสั่งที่เป็นรีจิสเตอร์ แฟล็ก และ แอดเดรสชิงโหมดด้วย และควรทำการ Optimize ให้เป็นชุดคำสั่งที่เหมาะสมด้วย

Generic Assembler

Generic Assembler ก็เป็นงานอีกส่วนหนึ่งซึ่งมีความสลับซับซ้อน และจะต้องพิจารณาอย่างถี่ถ้วนในการออกแบบ เนื่องจากขอบเขตการทำงานของโปรแกรมที่ค่อนข้างกว้าง คือโปรแกรมจะต้องมีความสามารถในการแปลภาษาแอสเซมบลีซึ่งมีรูปแบบของการเขียนและสัญลักษณ์ที่ใช้ในคำสั่งแบบใดๆก็ได้ตามแต่การกำหนดของผู้ใช้ และที่กล่าวไว้ในขั้นตอนของการวิเคราะห์ปัญหาว่า เป็นปัญหาต่อการจำลองแบบ Generic มาก แต่การแปลภาษาแอสเซมบลีก็เป็นขั้นตอนที่สำคัญและจำเป็นต่อการทำงานของโปรแกรม ฉะนั้นจะต้องวิเคราะห์และพิจารณาให้ละเอียด ถึงรูปแบบภาษาแอสเซมบลีของซีพียู 8 บิตที่มีอยู่ ศึกษาและพิจารณาหลักการทำงานของแอสเซมเบลอร์ทั่วไป เพื่อให้เกิดความคิดในการที่จะออกแบบกลไกอันใดอันหนึ่งขึ้นมา ที่จะทำให้แอสเซมเบลอร์ สามารถทำงานแบบ Generic ให้ได้ และในกลไกการทำงานนั้น ก็มีความสัมพันธ์กับการให้แอส-

เซมเบลอร์ ได้รับรู้ถึงส่วนที่ยืดหยุ่นจากผู้ใช้ได้

Simulator

ส่วนนี้จะมีลักษณะการทำงานเหมือนกับ CPU Simulator ทั่วๆ ไปหรือเป็นโปรแกรมหลักที่จะเรียกโปรแกรมอื่นๆเพื่อทำงานในการจำลองการทำงานของซีพียูตามขั้นตอนที่เหมาะสมและตามการเลือกจากผู้ใช้ ที่ต้องการศึกษาการทำงานของซีพียูที่ได้ออกแบบหรือป้อนข้อมูลไว้แล้วด้วยโปรแกรมภาษาแอสเซมบลี

ฉะนั้นโปรแกรมในส่วนนี้ จะมีส่วนที่ทำหน้าที่ติดต่อกับจอภาพหรือผู้ใช้ ให้สามารถเห็นการทำงานของโปรแกรมจำลองการทำงานได้จากหน้าต่างของส่วนต่างๆของซีพียู เช่นรีจิสเตอร์ แฟล็ก หน่วยความจำ