



## สรุปการวิจัย และข้อเสนอแนะ

การประเมินงานวิจัย

การดำเนินการวิจัย และ พัฒนาโปรแกรมจำลองซีพียู 8 บิต ให้มีคุณสมบัติเป็นโปรแกรมจำลองซีพียูแบบทั่วไป สามารถทำได้ผลตามเป้าหมายที่วางไว้ โปรแกรมที่พัฒนาขึ้นมีความสามารถจำลองการทำงานของซีพียู ที่มีข้อมูลระบุลักษณะเฉพาะตัว ซึ่งกำหนดโดยผู้ใช้โปรแกรม ทำให้โปรแกรมจำลองการทำงานของซีพียูแบบทั่วไปนี้ มีประโยชน์กว้างขึ้นกว่าโปรแกรมจำลองซีพียูแบบเฉพาะ เนื่องจากผู้ใช้โปรแกรมสามารถเปลี่ยนแปลงซีพียูจำลองได้โดยไม่ต้องอาศัยความรู้เกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์จำลองการทำงานของซีพียู เพราะโปรแกรมจำลองซีพียูแบบทั่วไปมีส่วนของ CPU Editor สำหรับให้ผู้ใช้เปลี่ยนแปลงแก้ไขข้อมูลเฉพาะของซีพียูได้ ผลที่ตามมาคือ การศึกษาการทำงานของซีพียูโดยการใช้โปรแกรมจำลองการทำงาน มีความสะดวกและประหยัดเพราะไม่ต้องหาระบบคอมพิวเตอร์ของซีพียูจริง และไม่ต้องหาโปรแกรมจำลองซีพียูแบบเฉพาะหลายๆ โปรแกรมเมื่อต้องการศึกษาซีพียูหลายๆ ตัวหรือเปลี่ยนแปลงไป หากผู้ใช้ไม่เห็นการจำลองอย่างสมบูรณ์แบบแล้ว โปรแกรมนี้จะให้ประโยชน์ได้พอสมควร และสามารถใช้งานโปรแกรมนี้ได้ด้วยไมโครคอมพิวเตอร์ PC ทั่วไป

อย่างไรก็ตามโปรแกรมนี้อาจยังมีข้อบกพร่องอยู่บ้าง ไม่ว่าจะเป็นในแง่ของการออกแบบหรือวิธีการเขียนโปรแกรม ซึ่งผู้วิจัยหวังว่างานวิจัยนี้จะให้ประโยชน์ในแง่ของการเรียนรู้ต่อผู้สนใจบ้างไม่มากก็น้อย

จุฬาลงกรณ์มหาวิทยาลัย

### สาระของงานวิจัย

ได้ดำเนินการวิจัยตามขั้นตอนที่ได้กล่าวไว้ในบทที่ 1 ครบถ้วนทุกขั้นตอน งานวิจัยในส่วนของการพัฒนาโปรแกรม เป็นงานพัฒนาโปรแกรม 2 ประเภทด้วยกันคือ งานพัฒนาโปรแกรมประเภท Editor ในส่วนของ CPU Editor และงานพัฒนาโปรแกรมประเภท Simulator ในส่วนของ CPU Simulator ซึ่งทำให้ผู้วิจัยมีความรู้ และทักษะเกี่ยวกับการพัฒนาโปรแกรมเพิ่มขึ้นมากกว่าที่ไม่เคยทำงานด้านนี้มาก่อนเลย แม้ว่าโปรแกรมในส่วน CPU Editor จะยังมีข้อบกพร่องอยู่ก็ตาม ซึ่งจะได้กล่าวถึงแนวทางในการปรับปรุงต่อไปในหัวข้อของข้อเสนอแนะ

ส่วนสำคัญของงานออกแบบและพัฒนาโปรแกรมจำลองการทำงานซีพียูพอจะสรุปได้ดังนี้

1. การออกแบบโครงสร้างข้อมูลรีจิสเตอร์ แฟล็ก และคำสั่ง เป็นกลไกสำคัญของ การหาคำตอบในการจำลองการทำงานของซีพียูแบบ generic
2. โครงสร้างข้อมูลของรีจิสเตอร์ แฟล็ก และคำสั่ง ที่ออกแบบสำหรับงานวิจัยนี้ มีข้อดีต่อความเร็วในการทำงานของซีพียูที่จำลอง แม้ว่าโดยหลักการแล้ว Generic CPU Simulator จะเพิ่มขั้นตอนการทำงานขึ้นมาอีก 2 ขั้นตอนจาก Specific CPU Simulator คือ การแปลงตัวแปรของคำสั่งจาก Specific register เป็น Generic register เพื่อจำลองการทำงาน แล้วแปลงกลับจาก Generic register เป็น Specific register เพื่อแสดงผลการทำงาน แต่ในความเป็นจริงแล้วไม่มีการแปลงระหว่าง Specific และ Generic register ในขั้นตอนของการเอ็กซีคิวต์คำสั่งแต่อย่างใด เนื่องจากโครงสร้างข้อมูลที่ออกแบบให้มีคุณสมบัติทั้ง generic และ specific ในตัวเอง หรือมีความสัมพันธ์ระหว่างส่วนจำลองการทำงานกับส่วนแสดงผลอยู่ในตัวข้อมูลนั่นเอง
3. การออกแบบ Generic flags ซึ่งมาจากการศึกษาและวิเคราะห์ข้อมูลได้ว่า แฟล็กของซีพียูแบ่งได้เป็น 2 จำพวกเท่านั้นคือ
  - พวกที่หนึ่ง แฟล็กซึ่งเป็นผลของการทำงานของ ALU เป็นแฟล็กส่วนใหญ่ของซีพียู

-พวกที่สอง แฟล็กเฉพาะงาน เป็นแฟล็กส่วนน้อยของซีพียู

Generic Flags ของโปรแกรม คือแฟล็กจำพวกที่หนึ่ง ที่มีการเปลี่ยนแปลงสถานะตามการทำงานของคำสั่ง generic ในลักษณะเดียวกับซีพียูทั่วไป รวมถึงโครงสร้างข้อมูลแฟล็กที่เป็นการจับคู่กันระหว่าง Generic flag กับแฟล็กของ Specific CPU ก็ทำให้สามารถแก้ไขปัญหาในเรื่องที่เกี่ยวกับแฟล็กได้ ส่วนแฟล็กจำพวกที่สอง ก็ให้ผู้ใช้งานโปรแกรมการทำงานของแฟล็กเองในขั้นตอนของการโปรแกรมคำสั่ง

4. Generic Assembler สามารถแปลแอสเซมบลีใดๆ ได้ เพราะออกแบบให้มีการทำงานที่ดัดแปลงจากแอสเซมเบลอร์ทั่วๆ ไปคือ

-ให้ผู้ใช้ระบุข้อมูลที่ใช้ในการแปลของแอสเซมเบลอร์

-ไม่ได้แปลแอสเซมบลีเป็น binary code ในหน่วยความจำ เนื่องจากถ้าทำงานแบบ Conventional assembler แล้ว จะยุ่งยากในการอิมพลีเมนต์มาก เพราะปัญหาของจำนวนไบต์ของ opcode และการวางตำแหน่งของ opcode ของซีพียูบางตัว

เมื่อเป็นแอสเซมเบลอร์แบบดัดแปลง คำสั่งที่แปลแล้ว จึงอยู่ในรูปของคำสั่ง generic อยู่ใน Execution Table การอิมพลีเมนต์ Execution Table ได้ใช้เทคนิค Pointer Array Approach ซึ่งจะสามารถแปลแอสเซมบลีต่างๆ ได้ สำหรับโปรแกรมนี้อาจกำหนดให้โปรแกรมแอสเซมบลีมีความยาวได้ไม่เกิน 500 คำสั่ง

ศูนย์วิทยพัทยาการ  
จุฬาลงกรณ์มหาวิทยาลัย



## ปัญหาและอุปสรรค

เนื่องจากความต้องการเริ่มต้นที่จะให้โปรแกรมนี้มีความยืดหยุ่นสูง ต่อผู้ใช้งาน และต่อลักษณะภายในของซีพียูเบอร์ต่างๆที่แตกต่างกันด้วย ขอบเขตของเนื้อหา งานวิจัยจึงค่อนข้างกว้างครอบคลุมตั้งแต่สถาปัตยกรรมของซีพียู 8 บิต เบอร์ต่างๆทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ วิธีการทำงานของแอสเซมเบลอร์ และวิธีการเขียนโปรแกรมจำลองการทำงานของซีพียูต่างๆไป ทำให้ต้องใช้เวลามากในขั้นตอนของการศึกษา งานและต้องใช้เวลามากขึ้นไปอีก กับขั้นตอนการออกแบบเพราะมีการปรับเปลี่ยนจนนับครั้งไม่ถ้วน กับการออกแบบการทำงานของแอสเซมเบลอร์และชุดคำสั่ง โดยเฉพาะอย่างยิ่งซีพียูเบอร์ Z80 ซึ่งเป็นซีพียูประเภท CISC (Complex Instruction Set Computer) ทำให้งานออกแบบยุ่งยากขึ้นมาก เพราะคำสั่งที่ซับซ้อนและชุดคำสั่งที่ใหญ่ อย่างไรก็ตามคำสั่ง Generic Instruction Set กับวิธีการโปรแกรมคำสั่งของโปรแกรมนี้อาจจะสามารถที่จะจำลองคำสั่งที่ซับซ้อนของซีพียู Z80 ได้ (ภาคผนวก ข) นอกจากนี้ความล่าช้าของงานออกแบบยังเกิดจากลักษณะความสัมพันธ์ของทุกส่วนของโปรแกรมที่มีความเกี่ยวข้องถึงกันเกือบทุกส่วน ความคิดในการออกแบบจึงต้องขยายออกให้ครอบคลุมให้หมด เมื่อยังคิดไม่หมดก็เกิดปัญหาทำให้ต้องรี้อความคิดเดิมไปตลอดเวลา

อย่างไรก็ตามโปรแกรมจำลองซีพียู 8 บิตทั่วไปก็สามารถทำงานได้ผลตามเป้าหมายที่วางไว้ คือ เป็นโปรแกรมที่สามารถจำลองการทำงานของซีพียูอย่างไม่ตายตัว

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

### ข้อเสนอแนะ

1. เกี่ยวกับแนวความคิดของโปรแกรมจำลองการทำงานของซีพียูแบบทั่วไป ผู้วิจัยเห็นว่าควรแบ่งโดเมนของโปรแกรมประเภทนี้ออกไปเป็นโปรแกรมจำลองการทำงานของซีพียูทั่วไปแบบซับซ้อน และแบบง่าย เพราะน่าจะทำให้เกิดความเหมาะสมได้ดีกว่า โปรแกรมจะเกิดความยืดหยุ่นสูงขึ้นไปอีกและสามารถอิมพลีเมนต์ได้อย่างมีประสิทธิภาพมากกว่า
2. โปรแกรมในส่วน Simulator Editor ที่โปรแกรมชุดคำสั่งยังไม่เหมาะสมนักกับการใช้งาน ควรปรับปรุงการโปรแกรมให้สามารถระบุแอดเดรสซึ่งโหมดทั้งหมดของแต่ละคำสั่งลงไปพร้อมกันได้ จะทุ่นเวลาในการป้อนข้อมูลลงมาก
3. อาจปรับปรุงให้สามารถโปรแกรมการทำงานของแต่ละแอดเดรสซึ่งโหมดได้ ด้วยวิธีการเดียวกันกับการโปรแกรมคำสั่ง

### สรุป

แนวความคิดหลักของการออกแบบ Generic machine คือ

1. ขยายขอบเขตของ Resource จาก Specific เป็น Generic
2. ให้ Generic machine มีการทำงานพื้นฐานเป็นหลัก
3. กำหนดโครงสร้างข้อมูลที่มีความสัมพันธ์ของข้อมูลตายตัว คือส่วน Generic และข้อมูลแปรเปลี่ยน คือส่วน Specific อยู่ในตัวเอง

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย