# การออกแบบวงจรหน่วยประมวลผลขนาด 32 บิต ที่มีการอัดคำสั่ง

นายเฉลิมพงศ์ สัตยาวิบูล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# A DESIGN OF A 32-BIT PROCESSOR WITH INSTRUCTION PACKING

Mr. Chalermpong Satayavibul

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

492252

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การออกแบบวงจรหน่วยประมวลผลขนาด 32 บิต ที่มีการอัดคำสั่ง |
| โดย | นายเฉลิมพงศ์ สัตยาวิบูล |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| อาจารย์ที่ปรึกษา | รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา |

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(อาจารย์ ดร.ฐิต ศิริบูรณ์)

..... อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... กรรมการ

(อาจารย์ ดร.เศรษฐา ปานงาม)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เขมะฑัต วิภาตะวนิช)

เฉลิมพงศ์ สัตยาวิบูล : การออกแบบวงจรหน่วยประมวลผลขนาด 32 บิต ที่มีการอัดคำสั่ง. (A DESIGN OF A 32-BIT PROCESSOR WITH INSTRUCTION PACKING) อ.ที่ปรึกษา : รศ.ดร.ประภาส จงสถิตย์วัฒนา, 80 หน้า.

วิธีการอัดคำสั่งเป็นวิธีลดขนาดโปรแกรมวิธีหนึ่ง ทำโดยการนำคำสั่งหลายๆคำสั่งมารวมกันบรรจุไว้ด้วยกัน ส่งผลให้วิธีการนี้มีส่วนช่วยเพิ่มสมรรถนะของหน่วยประมวลผล โดยช่วยลดเวลาที่ใช้ในการอ่านคำสั่งจากหน่วยความจำ

วิทยานิพนธ์นี้นำเสนอขั้นตอนการออกแบบวงจรหน่วยประมวลผล พร้อมทั้งนำเสนอการนำวิธีการอัดคำสั่งมาประยุกต์ใช้ เพื่อเพิ่มสมรรถนะให้กับหน่วยประมวลผลนี้ เนื่องจากงานวิจัยนี้มีจุดมุ่งหมายที่จะสร้างหน่วยประมวลผลสำหรับระบบฝังตัวที่มีทรัพยากรจำกัด เป้าหมายการออกแบบวงจรหน่วยประมวลผลเพื่อให้มีขนาดเล็ก ใช้ทรัพยากรเท่าที่จำเป็น โดยที่ยังมีประสิทธิภาพในการทำงานที่ไม่ช้าเกินไป อีกทั้งวิธีการอัดคำสั่งที่ใช้ต้องมีคุณสมบัติในด้านใช้ทรัพยากรน้อย โดยพิจารณาจากขนาดวงจรที่ต้องเพิ่มเข้าไป เพื่อให้วงจรหน่วยประมวลผลรองรับการอัดคำสั่งได้

การอัดคำสั่งที่นำมาประยุกต์ใช้กับหน่วยประมวลผลนี้ ช่วยลดขนาดโปรแกรมได้ร้อยละ 37.9 ของขนาดโปรแกรมปกติ และทำงานได้เร็วขึ้น 1.22 เท่า จากการลดจำนวนรอบนาฬิกาที่ใช้ในการอ่านคำสั่งจากหน่วยความจำ ในด้านการใช้ทรัพยากร วงจรที่ต้องเพิ่มให้วงจรหน่วยประมวลผลเพื่อให้รองรับการอัดคำสั่งนั้น มีขนาดเพียงร้อยละ 3 ของขนาดวงจรหน่วยประมวลผลเดิม

หน่วยประมวลผลที่ได้ออกแบบในงานวิจัยนี้ ใช้จำนวนเกตสมมูล 13,060 เกต ในด้านประสิทธิภาพของหน่วยประมวลผลนี้ได้รับการเปรียบเทียบกับหน่วยประมวลผลไมโครแบลซ ซึ่งเป็นหน่วยประมวลผลขนาด 32 บิต ที่มีการใช้งานกันอย่างแพร่หลายในอุตสาหกรรมระบบฝังตัว พบว่าหน่วยประมวลผลที่ได้ออกแบบในงานวิจัยนี้ ใช้จำนวนรอบนาฬิกาในการทำงานใกล้เคียงกับหน่วยประมวลผลไมโครแบลซ ความถี่สัญญาณนาฬิกาสูงสุดของหน่วยประมวลผลที่ออกแบบในงานวิจัยนี้อยู่ที่ 63 เมกะเฮิร์ทซ์

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....

สาขาวิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....

ปีการศึกษา.....2549.....

## 4970268321 : MAJOR COMPUTER ENGINEERING

KEY WORD: PROCESSOR / INSTRUCTION PACKING / EMBEDDED SYSTEM

CHALERMPONG SATAYAVIBUL : A DESIGN OF A 32-BIT PROCESSOR WITH INSTRUCTION PACKING. THESIS ADVISOR : ASSOC. PROF. PRABHAS CHONGSTITVATANA, 80 pp.

Instruction packing is one of the code compression techniques. It is done by packing more than one instruction to form a new packed-instruction. The packed-instruction can be fetched from the program memory in one clock cycle. Hence the processor's performance is improved by reducing the number of memory access.

This thesis proposes a design of a 32-bit processor. Its performance has been improved by instruction packing method. The proposed processor is designed for a limitted resource embedded system. The goal of this work is to design a low-cost processor with adequate performance. The instruction packing method must requires low resource. It should not increase the size of the circuit too much.

The instruction packing proposed in this work reduces the code size by 37.9% compared to a normal code. In term of performance, the processor executes packed-code 1.22 time faster than normal code. The instruction packing method increases the proposed processor equivalent gate by 3%.

The proposed processor requires 13,060 equivalent gates. Its maximum frequency is 63 MHz. In term of cycles consumed, it compares very well to the commercial 32-bit Xilinx's microprocessor, Micro Blaze.

Department .....Computer Engineering... Student's signature Chalermpong Satayavibul

Field of study.....Computer Engineering... Advisor's signature .....

Academic year .....2006 .....

# กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จเรียบร้อยมิได้ หากปราศจากความช่วยเหลือจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ. ดร. ประภาส จงสถิตย์วัฒนา ที่คอยให้คำแนะนำและความคิดเห็นเกี่ยวกับงานวิจัยนี้ อีกทั้งความรู้ด้านอื่นๆ มากมายที่อาจารย์ได้ถ่ายทอดให้ ล้วนเป็นสมบัติอันล้ำค่าที่ข้าพเจ้าสามารถนำมาใช้ประโยชน์ได้โดยไม่มีวันหมด

นอกจากนี้ อยากขอขอบพระคุณเป็นพิเศษ สำหรับคณะกรรมการสอบวิทยานิพนธ์ ได้แก่ อ. ดร.ฐิต ศิริบูรณ์ อ. ดร.เศรษฐา ปานงาม และ ผศ. ดร.เขมะฑัต วิภาตะวนิช ที่สละเวลาอันมีค่ามาชี้ให้เห็นถึงข้อบกพร่อง รวมทั้งข้อแนะนำที่น่าสนใจ

ขอขอบคุณสมาชิกห้องปฏิบัติการ ISL (Inteligent System Laboratory) ทุกคน รวมทั้งพี่ๆ และเพื่อนๆ ที่สร้างบรรยากาศที่ดีในการทำงาน ให้คำปรึกษาและช่วยแก้ปัญหาต่างๆ ทั้งที่เกี่ยวข้องกับงานวิจัยนี้และปัญหาอื่นๆ

ขอขอบคุณจุฬาลงกรณ์มหาวิทยาลัย และโรงเรียนหอวัง สถานศึกษาที่ได้ประสิทธิประสาทวิชาความรู้ให้กับผู้เขียน โดยเฉพาะเหล่าคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้สอนวิชาความรู้ ซึ่งจะเป็นสิ่งที่เลี้ยงชีพข้าพเจ้าต่อไปในวันข้างหน้า

และสุดท้ายขอขอบพระคุณ คุณพ่อ คุณแม่ พี่สาวและญาติๆ ที่อุปการะเลี้ยงดู ให้การสนับสนุนข้าพเจ้าในด้านการศึกษาอย่างเต็มที่ อีกทั้งยังเปิดโอกาสให้ข้าพเจ้าได้ศึกษาในด้านที่ข้าพเจ้าสนใจ

ทั้งนี้งานวิจัยนี้ได้รับเงินทุนสนับสนุนจากโครงการจัดการศึกษาสาขาวิชาวิศวกรรมศาสตร์เพื่อเพิ่มศักยภาพทางด้านวิทยาศาสตร์ เทคโนโลยีและอุตสาหกรรม หมวดเงินอุดหนุนการศึกษาประจำปีการศึกษา 2549

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ญ
สารบัญภาพ ..... ฏ
บทที่ 1 บทนำ ..... 1
1.1 ความเป็นมาและความสำคัญของปัญหา ..... 1
1.2 วัตถุประสงค์ของการวิจัย ..... 2
1.3 ขอบเขตของการวิจัย ..... 2
1.4 ขั้นตอนการวิจัย ..... 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ ..... 3
1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ ..... 3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ..... 4
2.1 ทฤษฎีที่เกี่ยวข้อง ..... 4
2.1.1 ต้นทุนการผลิตชิปของระบบฝังตัว ..... 4
2.1.2 การลดขนาดโปรแกรมในระบบฝังตัว ..... 5
2.2 งานวิจัยที่เกี่ยวข้อง ..... 6
2.2.1 การดัดแปลงการออกแบบชุดคำสั่งของหน่วยประมวลผลแบบอาร์ไอเอสซี ..... 6
2.2.2 การออกแบบวงจรหน่วยประมวลผลแบบแสตก ..... 8
2.2.3 การอัดคำสั่งของหน่วยประมวลผลแบบแสตก ..... 9
2.2.4 การอัดคำสั่งลงในรีจิสเตอร์ ..... 10
บทที่ 3 การออกแบบวงจรหน่วยประมวลผล ..... 12
3.1 การออกแบบชุดคำสั่ง ..... 12
3.2 การออกแบบทางเดินข้อมูล ..... 14
3.3 การกำหนดความหมายของคำสั่ง ..... 15
3.4 การแทนรหัสดำเนินการแต่ละคำสั่ง ..... 16
3.5 การออกแบบสถานะและการออกแบบพฤติกรรมการทำงานของหน่วยประมวลผล ..... 17
3.5.1 กลุ่มสถานะที่อยู่ในขั้นตอนการดึงคำสั่งและถอดรหัสคำสั่ง ..... 19

3.5.2 กลุ่มสถานะที่อยู่ในขั้นตอนการกระทำที่ 1 ..... 20
3.5.3 กลุ่มสถานะที่อยู่ในขั้นตอนการกระทำที่ 2 ..... 20
บทที่ 4 การอัดคำสั่ง ..... 21
4.1 วิธีการอัดคำสั่ง ..... 21
4.2 การดัดแปลงทางเดินข้อมูลของหน่วยประมวลผลเพื่อให้รองรับการทำงานแบบอัดคำสั่ง.. 23
4.3 การทำงานโปรแกรมที่ได้รับการอัดคำสั่ง ..... 25
4.3.1 ขั้นตอนการทำงาน ..... 26
4.3.2 สถานะการทำงาน ..... 26
4.4 การดัดแปลงวงจรหน่วยควบคุมของหน่วยประมวลผลที่ให้รองรับการอัดคำสั่ง ..... 29
4.5 การถอดรหัสคำสั่งของหน่วยประมวลผลที่รองรับการอัดคำสั่ง ..... 30
4.5.1 การถอดรหัสคำสั่งในสถานะดึงคำสั่งและถอดรหัสคำสั่ง ..... 31
4.5.2 การถอดรหัสคำสั่งถัดไปในระหว่างที่กระทำคำสั่งปัจจุบัน ..... 31
4.5.3 การถอดรหัสคำสั่งในสถานะดึงคำสั่งและถอดรหัสคำสั่งแบบพิเศษ ..... 31
4.6 ข้อจำกัดของการอัดคำสั่ง ..... 31
4.6.1 คำสั่งถัดไปเป็นตำแหน่งปลายทางของการกระโดด ..... 31
4.6.2 คำสั่งถัดไปเป็นคำสั่งแรกของโปรแกรมย่อย ..... 32
4.6.3 คำสั่งถัดไปเป็นตำแหน่งปลายทางของการคืนค่าจากโปรแกรมย่อย ..... 32
4.7 วิเคราะห์ประสิทธิภาพการทำงานที่เพิ่มขึ้นจากคำสั่งอัด ..... 32
4.8 แนวทางการปรับปรุงเพื่อกำจัดข้อจำกัดของการอัดคำสั่ง ..... 32
บทที่ 5 การพัฒนาโปรแกรม ..... 33
5.1 อุปกรณ์ต่อเชื่อมกับหน่วยประมวลผล ..... 33
5.2 การจัดสรรพื้นที่หน่วยความจำ ..... 34
5.2.1 หน่วยความจำส่วนโปรแกรม ..... 34
5.2.2 หน่วยความจำส่วนแสตก ..... 35
5.2.3 หน่วยความจำส่วนข้อมูล ..... 35
5.2.4 หน่วยความจำส่วนอุปกรณ์ต่อเชื่อม ..... 35
5.3 การเข้าถึงตัวแปรส่วนกลาง ..... 36
5.4 การเข้าถึงข้อมูลแถวลำดับ ..... 36
5.5 การเข้าถึงตัวแปรเฉพาะที่ ..... 37
5.6 แอคทิเวชันเรคคอร์ด ..... 38
5.7 กลไกในการเรียกโปรแกรมย่อย ..... 40
5.7.1 การส่งผ่านพารามิเตอร์ ..... 40

5.7.2 การจองพื้นที่สำหรับตัวแปรเฉพาะที่ ..... 40
5.8 กลไกในการคืนค่าจากโปรแกรมย่อย ..... 41
5.9 การระบุระยะทางของการกระโดด ..... 41
5.10 ตัวอย่างโค้ดภาษาแอสเซมบลีของหน่วยประมวลผลนี้ ..... 42
บทที่ 6 การอัดคำสั่งแบบปรับปรุง ..... 44
6.1 การกำหนดเลขที่อยู่ของคำสั่งใหม่ ..... 44
6.2 การปรับเปลี่ยนวงจรหน่วยประมวลผลเพื่อรองรับการอัดคำสั่งแบบปรับปรุง ..... 45
6.2.1 วงจรส่วนที่ทำการบวกค่าของรีจิสเตอร์ PC ..... 45
6.2.2 วงจรส่วนนำค่าของรีจิสเตอร์ PC ไปใช้ ..... 46
6.2.3 วงจรหน่วยควบคุม ..... 47
6.3 วิเคราะห์ประสิทธิภาพของการอัดคำสั่งแบบปรับปรุง ..... 50
6.4 สรุปประสิทธิภาพของการอัดคำสั่งแบบปรับปรุง ..... 51
บทที่ 7 วิเคราะห์และสรุปผลการวิจัย ..... 54
7.1 โปรแกรมทดสอบที่ใช้ในงานวิจัยนี้ ..... 55
7.2 ประสิทธิภาพที่เพิ่มขึ้นจากการอัดคำสั่ง ..... 55
7.3 วิเคราะห์ขนาดของโปรแกรมของหน่วยประมวลผล ..... 57
7.4 วิเคราะห์ประสิทธิภาพของหน่วยประมวลผล ..... 59
7.5 แนวทางการวิจัยในอนาคต ..... 61
7.6 สรุปผลการวิจัย ..... 62
รายการอ้างอิง ..... 65
ภาคผนวก ..... 67
ภาคผนวก ก ความหมายของคำสั่งและการให้รหัสคำสั่ง ..... 68
ก.1 ความหมายของคำสั่ง ..... 68
ก.2 การให้รหัสดำเนินการ ..... 75
ภาคผนวก ข สถานะของหน่วยประมวลผล ..... 77
ประวัติผู้เขียนวิทยานิพนธ์ ..... 80

# สารบัญตาราง

หน้า

ตารางที่ 3.1 คำสั่งในชุดคำสั่งของหน่วยประมวลผลของงานวิจัยนี้ ..... 12
ตารางที่ 3.2 รูปแบบของชุดคำสั่งแต่ละประเภทของหน่วยประมวลผล ..... 13
ตารางที่ 4.1 สัญลักษณ์ที่ใช้ในการอธิบายพฤติกรรมการทำงานของรีจิสเตอร์ IR ..... 29
ตารางที่ 4.2 การรับค่าเข้ารีจิสเตอร์ IR ในแต่ละสถานการณ์ ..... 30
ตารางที่ 4.3 การเลื่อนข้อมูลในรีจิสเตอร์ IR ในแต่ละสถานการณ์ ..... 30
ตารางที่ 4.4 ทรัพยากรที่ใช้ในการปรับปรุงวงจรหน่วยประมวลผลเพื่อให้รองรับการอัดคำสั่ง ..... 30
ตารางที่ 5.1 สัญลักษณ์ที่ใช้ในแอสเซมเบลอร์ ..... 42
ตารางที่ 6.1 ค่าที่เพิ่มให้กับรีจิสเตอร์ PC ในแต่ละสถานการณ์ ..... 48
ตารางที่ 6.2 การรับค่าเข้ารีจิสเตอร์ IR ในแต่ละสถานการณ์ ..... 48
ตารางที่ 6.3 สัญลักษณ์ที่ใช้ในการอธิบายการทำงานของรีจิสเตอร์ IR ..... 49
ตารางที่ 6.4 โปรแกรมที่ใช้ในการทดสอบ ..... 50
ตารางที่ 6.5 การเปรียบเทียบขนาดของโปรแกรมที่ได้รับการอัดคำสั่งแบบปรับปรุงกับโปรแกรมที่ได้รับการอัดคำสั่งแบบธรรมดา ..... 51
ตารางที่ 6.6 ความถี่สูงสุดของสัญญาณนาฬิกาที่หน่วยประมวลผลสามารถทำงานได้ ..... 51
ตารางที่ 6.7 ทรัพยากรที่ใช้ในการสร้างหน่วยประมวลผล ..... 51
ตารางที่ 7.1 โปรแกรมที่ใช้ในการทดสอบ ..... 55
ตารางที่ 7.2 การเปรียบเทียบขนาดของโปรแกรมที่ได้รับการอัดคำสั่งกับโปรแกรมปกติ ..... 56
ตารางที่ 7.3 การเปรียบเทียบประสิทธิภาพการทำงานของโปรแกรมที่ได้รับการอัดคำสั่งกับโปรแกรมปกติ ..... 56
ตารางที่ 7.4 การเปรียบเทียบรอบนาฬิกาที่ใช้ในการทำงานหนึ่งคำสั่งของโปรแกรมที่ได้รับการอัดคำสั่งกับโปรแกรมปกติ ..... 57
ตารางที่ 7.5 การเปรียบเทียบขนาดโปรแกรมระหว่างโปรแกรมที่ได้รับการอัดคำสั่งกับโปรแกรมของหน่วยประมวลผลแบบแสตก ..... 58
ตารางที่ 7.6 การเปรียบเทียบขนาดโปรแกรมระหว่างโปรแกรมที่ได้รับการอัดคำสั่งกับโปรแกรมของหน่วยประมวลผลไมโครแบลซ ..... 58
ตารางที่ 7.7 การเปรียบเทียบประสิทธิภาพการทำงานระหว่างหน่วยประมวลผลไมโครแบลซกับหน่วยประมวลผลที่มีการอัดคำสั่ง ..... 59
ตารางที่ 7.8 ความถี่สูงสุดของสัญญาณนาฬิกาที่หน่วยประมวลผลสามารถทำงานได้ ..... 60
ตารางที่ 7.9 ทรัพยากรที่ใช้ในการสร้างหน่วยประมวลผล ..... 61

ตารางที่ ก.1 แสดงความหมายของสัญลักษณ์ที่ใช้ในการอธิบายการทำงานของแต่ละคำสั่ง..... 68
ตารางที่ ก.2 อธิบายความหมายและการทำงานของคำสั่งประเภทเลขคณิตและตรรกะ ..... 70
ตารางที่ ก.3 อธิบายความหมายและการทำงานของคำสั่งประเภทพิเศษ..... 72
ตารางที่ ก.4 อธิบายความหมายและการทำงานของคำสั่งประเภทจัดกการข้อมูล ..... 73
ตารางที่ ก.5 อธิบายความหมายและการทำงานของคำสั่งประเภทควบคุม ..... 74
ตารางที่ ก.6 รหัสดำเนินการของคำสั่งทั้งหมดของชุดคำสั่งนี้..... 76
ตารางที่ ข.1 สัญลักษณ์ที่ใช้แทนชื่อสถานะของหน่วยประมวลผล..... 77
ตารางที่ ข.2 แสดงสถานะถัดไปของสถานะดึงคำสั่งและถอดรหัสคำสั่ง..... 78
ตารางที่ ข.3 พฤติกรรมการขนถ่ายข้อมูลระหว่างรีจิสเตอร์ในสถานะต่างๆ ..... 79

# สารบัญภาพ

หน้า

รูปที่ 2.1 แสดงพื้นที่ดายของไมโครคอนโทรลเลอร์ MC68HC908GP20 ..... 5
รูปที่ 2.2 ตัวอย่างการลดจำนวนบิตในชุดคำสั่งเอ็มไอพีเอส16ของหน่วยประมวลผลเอ็มไอพีเอส ... 7
รูปที่ 2.3 การทำงานของหน่วยประมวลผลเอ็มไอพีเอสกับชุดคำสั่งแบบเอ็มไอพีเอส16 ..... 7
รูปที่ 2.4 ตัวอย่างการดำเนินการบนแสตก ..... 8
รูปที่ 2.5 รูปแบบคำสั่งของหน่วยประมวลผลไว ..... 10
รูปที่ 2.6 รูปแบบการอัดคำสั่งของหน่วยประมวลผลไว ..... 10
รูปที่ 2.7 โครงสร้างการอัดคำสั่งด้วยวิธีการใช้รีจิสเตอร์คำสั่ง ..... 11
รูปที่ 2.8 รูปแบบของคำสั่งที่ทำการอัดคำสั่งด้วยรีจิสเตอร์ ..... 11
รูปที่ 3.1 แสดงส่วนประกอบของชุดคำสั่งรูปแบบต่างๆของหน่วยประมวลผล ..... 13
รูปที่ 3.2 ทางเดินข้อมูลของหน่วยประมวลผลที่พัฒนาในงานวิจัยนี้ ..... 15
รูปที่ 3.3 ส่วนของรหัสดำเนินการที่ใช้ในการแบ่งรูปแบบของคำสั่ง ..... 17
รูปที่ 3.4 แผนภูมิแสดงขั้นตอนการทำงานของหน่วยประมวลผล ..... 18
รูปที่ 3.5 แผนภูมิสถานะของหน่วยประมวลผลนี้ ..... 19
รูปที่ 4.1 ตัวอย่างการอัดคำสั่งที่ใช้ในงานวิจัยนี้ ..... 22
รูปที่ 4.2 ตัวอย่างคำสั่งที่ได้รับการอัดคำสั่ง ..... 22
รูปที่ 4.3 ส่วนประกอบของคำสั่งอัด ..... 23
รูปที่ 4.4 รีจิสเตอร์ IR ที่ได้รับการปรับปรุง ..... 24
รูปที่ 4.5 ทางเดินข้อมูลของวงจรหน่วยประมวลผลที่รองรับการทำงานแบบอัดคำสั่ง ..... 25
รูปที่ 4.6 แผนภูมิแสดงขั้นตอนการทำงานโค้ดที่ได้รับการอัดคำสั่ง ..... 27
รูปที่ 4.7 แผนภูมิสถานะของหน่วยประมวลผลที่มีการอัดคำสั่ง ..... 28
รูปที่ 5.1 การเชื่อมต่ออุปกรณ์ต่อเชื่อมของหน่วยประมวลผล ..... 33
รูปที่ 5.2 ตัวอย่างการจัดสรรพื้นที่หน่วยความจำที่ใช้ในงานวิจัยนี้ ..... 34
รูปที่ 5.3 ส่วนประกอบของแอคทิเวชันเรคคอร์ด ..... 38
รูปที่ 5.4 การสร้างแอคทิเวชันเรคคอร์ดในคำสั่ง CALL ..... 39
รูปที่ 6.1 ตัวอย่างการอัดคำสั่งแบบปรับปรุง ..... 45
รูปที่ 6.2 การปรับเปลี่ยนวงจรส่วนที่ทำการบวกค่าของรีจิสเตอร์ PC ..... 46
รูปที่ 6.3 การปรับเปลี่ยนวงจรส่วนนำค่าของรีจิสเตอร์ PC ไปใช้ ..... 46
รูปที่ 6.4 ทางเดินข้อมูลของหน่วยประมวลผลที่รองรับการอัดคำสั่งแบบปรับปรุง ..... 49
รูปที่ 6.5 ตัวอย่างการจัดเรียงคำสั่งที่ไม่มีประสิทธิภาพ ..... 53