# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบฝังตัว (Embedded systems) ก้าวเข้ามามีบทบาทสำคัญ เนื่องจากระบบฝังตัวถูกนำมาใช้ในอุปกรณ์เครื่องใช้ต่างๆ ที่ใช้ในชีวิตประจำวัน ตัวอย่างเช่น อุปกรณ์ไฟฟ้าภายในบ้าน อุปกรณ์ภายในรถยนต์ เครื่องเสียง โทรศัพท์เคลื่อนที่ และอุปกรณ์อื่นๆ อีกมากมาย ดังนั้นการลดต้นทุนของชิประบบฝังตัว (Chip) จึงเป็นประเด็นที่น่าสนใจสำหรับอุตสาหกรรมการผลิตชิป

ต้นทุนการผลิตชิประบบฝังตัวจะขึ้นอยู่กับปัจจัยหลายประการ เช่น เทคโนโลยีการผลิต จำนวนการผลิต ขนาดของแว่นผลึก (Wafer) ชนิดของแพ็คเก็จ (Package) เป็นต้น จากบทความของ Maly [1] ได้นำเสนอแบบจำลองในการคำนวณหาต้นทุนการผลิตชิปโดยการหาต้นทุนของทรานซิสเตอร์ (Cost of transistor, $C_{tr}$) ดังสมการที่ 1

$$C_{tr} = \frac{C_w}{N_{ch} N_{tr} Y} \qquad (1)$$

โดย $C_w$ คือ ต้นทุนการผลิตแว่นผลึก

$N_{ch}$ คือ จำนวนดายของวงจรรวมต่อหนึ่งแว่นผลึก

$N_{tr}$ คือ จำนวนทรานซิสเตอร์ต่อหนึ่งดาย

$Y$ คือ จุดคราก (Yield) การผลิต

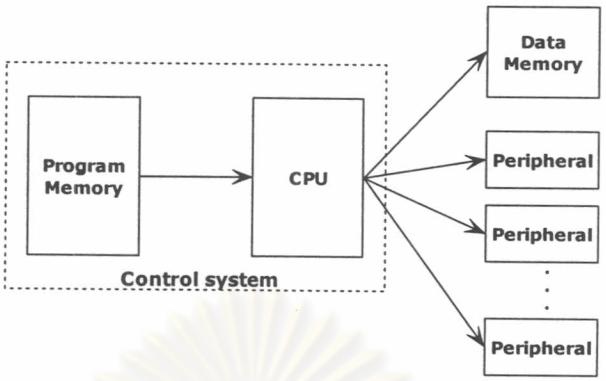
จากสมการที่ 1 พบว่าที่เทคโนโลยีการผลิตเดียวกัน ค่าที่มีผลต่อต้นทุนการผลิตคือจำนวนดายของวงจรรวมต่อหนึ่งแว่นผลึก ($N_{ch}$) เนื่องจากปัจจัยอื่นจะมีค่าคงที่ที่เทคโนโลยีการผลิตเดียวกัน ซึ่งค่า $C_{tr}$ จะแปรผกผันกับค่า $N_{ch}$ ดังนั้นถ้ายิ่งออกแบบให้ชิปมีขนาดดายยิ่งเล็กเท่าไหร่ก็จะสามารถลดต้นทุนได้มากขึ้นเท่านั้น

ส่วนประกอบที่สำคัญของระบบฝังตัวประกอบด้วยหน่วยประมวลผล (Processor) หน่วยความจำ (Memory) และอุปกรณ์บริวาร (Peripheral device)[1] ดังรูปที่ 1.1 การทำงานของระบบฝังตัวจะถูกควบคุมโดยหน่วยประมวลผล ซึ่งทำงานตามโปรแกรมฝังตัว (Embedded program) ที่จัดเก็บอยู่ในหน่วยความจำโปรแกรม (Program memory) ดังนั้นส่วนประกอบ

---

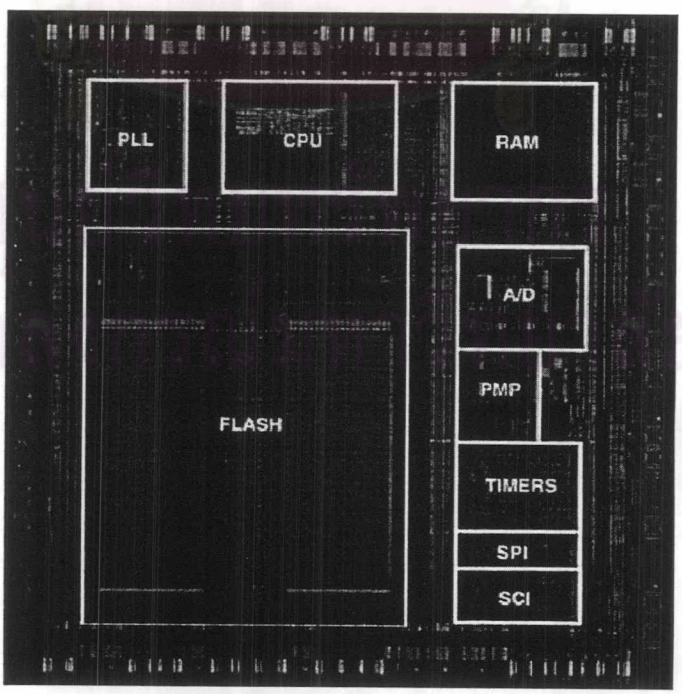
[1] ในวิทยานิพนธ์นี้ไม่สนใจเกี่ยวกับการติดต่อกับอุปกรณ์บริวาร ดังนั้นเนื้อหาที่จะกล่าวถึงต่อไปจึงละอุปกรณ์บริวารไว้

พื้นฐานที่สำคัญที่ทำหน้าที่ควบคุมระบบฝังตัวก็คือหน่วยประมวลผล หน่วยความจำโปรแกรม และหน่วยความจำข้อมูลนั่นเอง

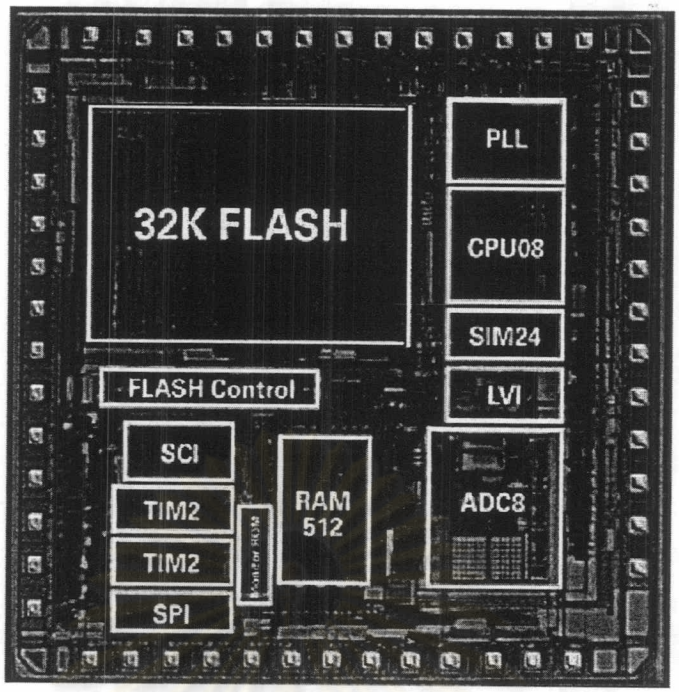



รูปที่ 1.1 ส่วนประกอบของระบบฝังตัว

ไมโครคอนโทรลเลอร์ MC68HC908GP20 [2] ของบริษัท Motorola ประกอบด้วยหน่วยประมวลผลขนาด 8 บิต และมีหน่วยความจำโปรแกรมชนิดแฟลชขนาด 20 กิโลไบต์ จากรูปที่ 1.2 แสดงถึงดายของไมโครคอนโทรลเลอร์นี้พบว่าหน่วยความจำแฟลชกินพื้นที่ดายถึง 36% ของพื้นที่ดายทั้งหมด และในไมโครคอนโทรลเลอร์ MC68HC908GP32 [3] เป็นไมโครคอนโทรลเลอร์ชนิดเดียวกับ MC68HC908GP20 แต่มีขนาดของหน่วยความจำแฟลชที่มากกว่า คือ 32 กิโลไบต์ และใช้เทคโนโลยีการผลิตที่เล็กกว่า พบว่าพื้นที่ดายของหน่วยความจำแฟลชกินพื้นที่ 18% ของพื้นที่ทั้งหมดดังรูปที่ 1.3 ซึ่งจากรูปที่ 1.2 และรูปที่ 1.3 จะเห็นว่าพื้นที่ขอหน่วยความจำแฟลชมีขนาดใหญ่กว่าส่วนประกอบอื่นๆ ภายในไมโครคอนโทรลเลอร์

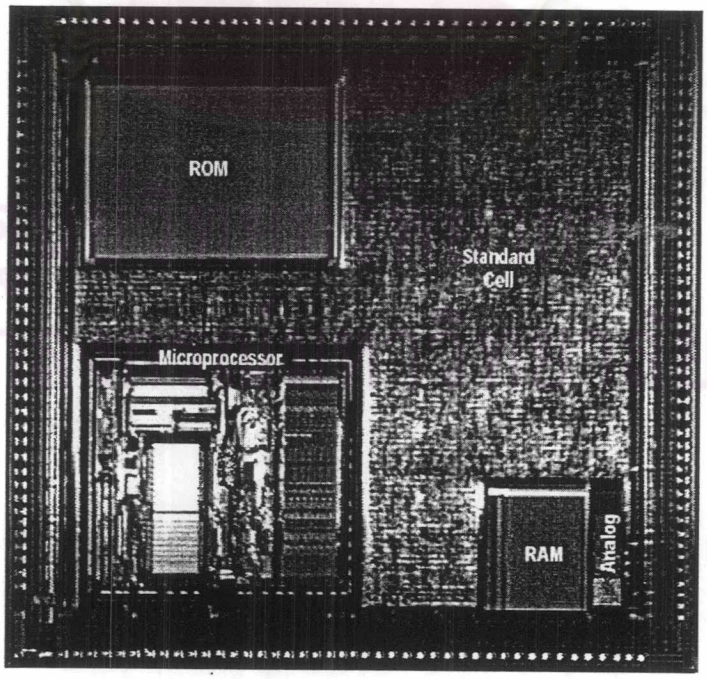



รูปที่ 1.2 ดายของไมโครคอนโทรลเลอร์ MC68HC908GP20




รูปที่ 1.3 ดายของไมโครคอนโทรลเลอร์ MC68HC908GP32

เช่นเดียวกันกับในไมโครคอนโทรลเลอร์ HP DeskJet 820C [4] ของบริษัท HP ที่ใช้หน่วยประมวลผล 68EC000 ซึ่งเป็นหน่วยประมวลผลขนาด 16 บิตของ Motorola โดยมีหน่วยความจำรอมขนาด 64 กิโลไบต์ โดยดายของไมโครคอนโทรลเลอร์ดังกล่าวแสดงดังรูปที่ 1.4 ซึ่งในบทความของ Sherwood และ Calder [5] ได้สรุปถึงพื้นที่ของหน่วยความจำรอมว่ากินพื้นที่ 14% ของพื้นที่ทั้งหมด และจะกินพื้นที่ถึง 25% เมื่อเปลี่ยนไปใช้หน่วยความจำแฟลช




รูปที่ 1.4 ดายของไมโครคอนโทรลเลอร์ HP DeskJet 820C

ในไมโครคอนโทรลเลอร์ Q-Chip [6] ที่ออกแบบขึ้นเพื่อใช้ในการควบคุมเครื่องรับโทรทัศน์ มีส่วนประกอบได้แก่หน่วยประมวลผลขนาด 16 บิต รอมสำหรับเก็บคำสั่ง รอมสำหรับเก็บตัวอักษร หน่วยความจำแรม และวงจรอุปกรณ์บริวาร ขนาดดายของส่วนประกอบต่างๆ แสดงได้ดังตารางที่ 1.1 พบว่าหน่วยความจำรอมสำหรับเก็บโปรแกรมนั้นมีขนาดพื้นที่ดาย 10% ของพื้นที่ทั้งหมดซึ่งเป็นส่วนประกอบที่ใหญ่รองมาจากหน่วยประมวลผลกลาง (13%) และหน่วยความจำรอมสำหรับเก็บตัวอักษร (12%)

ตารางที่ 1.1 พื้นที่ของวงจรในส่วนต่างๆ ของไมโครคอนโทรลเลอร์ Q-Chip

| วงจร | พื้นที่ ($mm^2$) |
|---|---|
| หน่วยประมวลผลกลาง | 8.1 |
| อุปกรณ์บริวาร | 3.2 |
| หน่วยความจำแบบ RAM | 4.4 |
| หน่วยความจำแบบ ROM สำหรับตัวอักษร | 7.8 |
| หน่วยความจำแบบ ROM สำหรับโปรแกรม | 6.3 |
| PAD | 10.8 |
| พื้นที่รวมทั้งหมด (รวมพื้นที่ที่ใช้ในการต่อเชื่อมแต่ละส่วนเข้าด้วยกัน) | 60.8 |

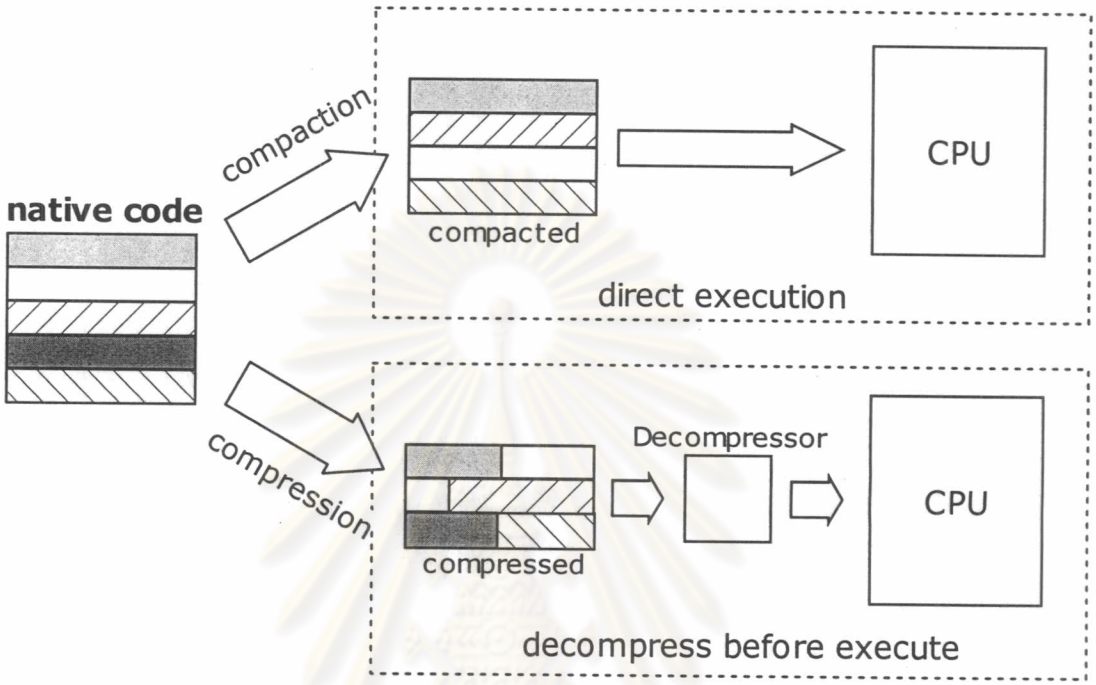
จะเห็นได้ว่าหน่วยความจำโปรแกรมในระบบฝังตัวนั้น (ทั้งชนิดแฟลชและรอม) เป็นส่วนที่กินพื้นที่ดายมากเมื่อเปรียบเทียบกับส่วนประกอบอื่นๆ ไม่ว่าจะเป็นหน่วยประมวลผล หน่วยความจำข้อมูล หรืออุปกรณ์บริวารอื่นๆ ดังนั้นการลดขนาดของโปรแกรมฝังตัวที่ใช้ในการควบคุมระบบลงจะส่งผลให้ขนาดดายของหน่วยความจำมีขนาดเล็กลง และทำให้ต้นทุนการผลิตลดลงด้วย การลดขนาดโปรแกรมจึงเป็นประเด็นที่น่าสนใจศึกษาเพื่อนำมาประยุกต์ใช้ในการผลิตหน่วยประมวลผลสำหรับระบบฝังตัว

### 1.1.1 การลดขนาดโปรแกรม

วิธีที่ใช้ในการลดขนาดโปรแกรมจะเรียกว่า "การลดขนาดโปรแกรม" (Code-size reduction) ในบทความของ Beszedes [7] ได้แบ่งประเภทของการลดขนาดของโปรแกรมไว้สองประเภทได้แก่ "การอัดแน่นโปรแกรม" (Code Compaction) และ"การบีบอัดโปรแกรม" (Code Compression)

การอัดแน่นโปรแกรมใช้ตัวแปลโปรแกรม (Compiler) ในการตัดคำสั่งบางคำสั่งในโปรแกรมที่ไม่จำเป็น และจัดเรียงลำดับของคำสั่งใหม่เพื่อลดขนาดของโปรแกรมโดยรวม ส่วนการ

บีบอัดโปรแกรมนั้นนำเอาหลักการของการบีบอัดข้อมูล (Data compression) มาประยุกต์ใช้กับการบีบอัดโปรแกรม ดังนั้นก่อนที่ระบบทำงานจำเป็นต้องคลายโปรแกรม (Decompression) ที่ถูกบีบอัดไว้เสียก่อนจึงสามารถทำงานได้ ซึ่งแตกต่างจากโปรแกรมที่ผ่านการอัดแน่นโปรแกรมสามารถทำงานกับระบบเดิมได้ทันที โดยไม่จำเป็นต้องมีการคลายโปรแกรม ดังรูปที่ 1.5




รูปที่ 1.5 การอัดแน่นโปรแกรมและการบีบอัดโปรแกรม

แต่เนื่องจากการบีบอัดโปรแกรมใช้หลักการของการบีบอัดทำให้สามารถบีบอัดโปรแกรมให้มีขนาดที่เล็กว่าการอัดแน่นโปรแกรม อีกทั้งวิธีการบีบอัดโปรแกรมยังสามารถประยุกต์ใช้กับสถาปัตยกรรมแบบต่างๆ ได้ ในขณะที่การอัดแน่นโปรแกรมจะต้องออกแบบมาเพื่อประยุกต์ใช้กับชุดคำสั่งแบบใดแบบหนึ่งเท่านั้น

### 1.1.2 ชุดคำสั่งกลาง และการแปลคำสั่ง

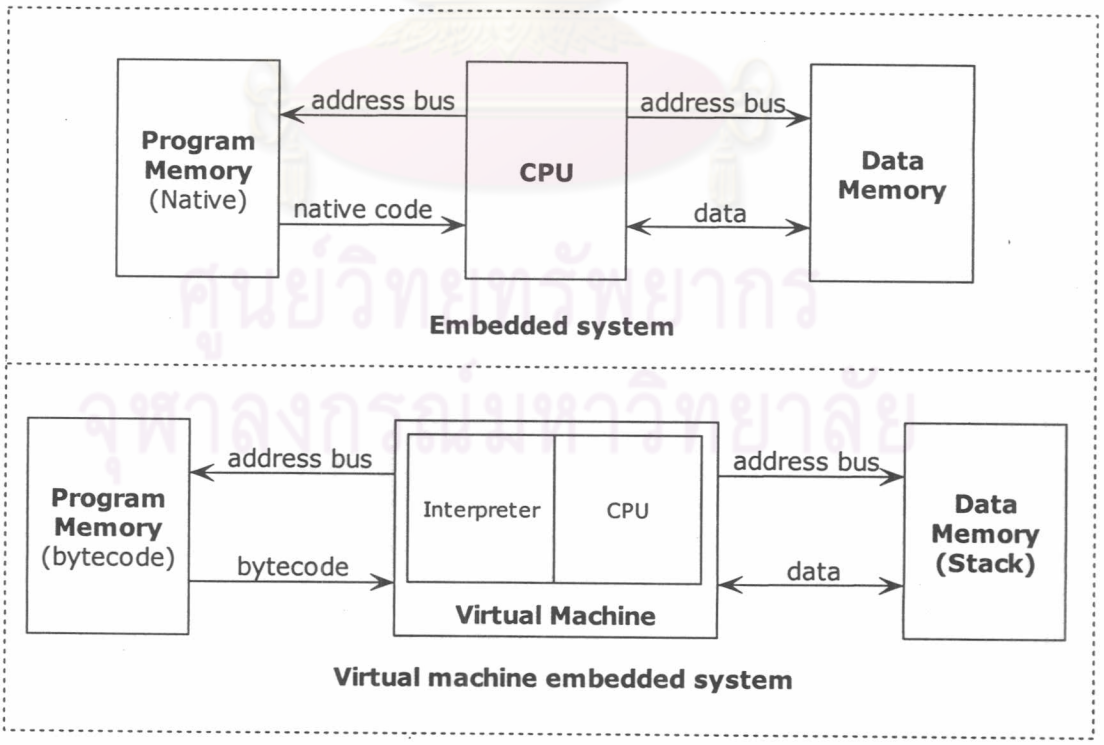
การแปลคำสั่ง (Interpretation) เป็นวิธีการหนึ่งที่ใช้ในการลดขนาดของโปรแกรม เนื่องจากเป็นวิธีการที่เปิดโอกาสให้ระบบสามารถทำงานกับโปรแกรมที่อยู่ในรูปแบบของชุดคำสั่งอื่นที่ไม่ใช่ชุดคำสั่งของระบบโดยตรง ซึ่งจะเรียกชุดคำสั่งรูปแบบอื่นนี้ว่าชุดคำสั่งกลาง (Intermediate code)

หน่วยประมวลผลที่นิยมนำมาใช้ระบบฝังตัวคือหน่วยประมวลผลแบบเรจิสเตอร์ (Register-based CPU) เนื่องจากเป็นหน่วยประมวลผลที่มีให้เลือกมากมายใช้ในท้องตลาด

ส่วนชุดคำสั่งกลางที่นิยมใช้จะเป็นชุดคำสั่งแบบแสตก (Stack-based instruction set) ที่มีขนาดของชุดคำสั่งเล็กกว่าชุดคำสั่งแบบเรจิสเตอร์ (Register-based instruction set) สาเหตุที่ชุดคำสั่งแบบแสตกนั้นมีขนาดที่เล็กว่าชุดคำสั่งแบบเรจิสเตอร์นั้นในบทความของประภาส [8] ได้กล่าวถึงเหตุผลไว้ 2 ประการดังนี้

1. คำสั่งแบบแสตกมีความหมาย (Semantic) สูงกว่าคำสั่งแบบเรจิสเตอร์ หรืออาจกล่าวได้ว่า 1 คำสั่งแบบแสตกเทียบเท่ากับหลายการทำงานของคำสั่งแบบเรจิสเตอร์นั่นเอง
2. ชุดคำสั่งแบบแสตกใช้แสตกในการทำงานจึงไม่ต้องกำหนดตัวถูกดำเนินการในคำสั่ง ซึ่งชุดคำสั่งแบบเรจิสเตอร์มีตัวถูกดำเนินการหลายตัวตามชนิดของแอดเดรสซิ่งโหมด (Addressing mode) ทำให้ขนาดของแต่ละคำสั่งของชุดคำสั่งแบบแสตกมีขนาดที่เล็กกว่าชุดคำสั่งแบบเรจิสเตอร์

การทำให้ระบบที่ทำงานด้วยหน่วยประมวลผลแบบเรจิสเตอร์จะสามารถทำงานกับชุดคำสั่งแบบแสตกได้นั้นต้องทำให้ระบบทำงานเป็นเครื่องเสมือนแบบแสตก (Virtual stack machine) หรือเรียกสั้นๆ ว่าเครื่องเสมือน (Virtual machine) ซึ่งมีส่วนประกอบที่สำคัญคือตัวแปลคำสั่ง (Interpreter) ซึ่งจะทำหน้าที่ในการแปลคำสั่งแบบแสตกเพื่อให้หน่วยประมวลผลทำงานตามคำสั่งได้ ดังรูปที่ 1.6 ซึ่งแสดงการเปรียบเทียบระหว่างระบบฝังตัวธรรมดากับระบบที่ทำงานเป็นเครื่องเสมือนแบบแสตก




รูปที่ 1.6 การเปรียบเทียบโครงสร้างระหว่างระบบฝังตัวปกติกับระบบเครื่องเสมือน

แต่การที่ระบบจำเป็นต้องมีการแปลคำสั่งก่อนทำให้การทำงานของระบบช้าลงมาก เนื่องจากตัวแปลคำสั่งเองนั้นก็เป็นโปรแกรมหนึ่งที่ทำงานอยู่บนหน่วยประมวลผลซึ่งในที่นี้จะขอเรียกว่าซอฟต์แวร์แปลคำสั่ง (Software interpreter) ในงานวิจัยของ Ernst และคณะ [9] และ Hoogerbrugge และคณะ [10] ได้รายงานถึงสมรรถนะการทำงานของซอฟต์แวร์แปลคำสั่งที่ส่งผลทำให้การทำงานโดยรวมของระบบช้าลงประมาณ 12 เท่าในงานวิจัยของ Ernst และคณะ และประมาณ 9 เท่าในงานวิจัยของ Hoogerbrugge และคณะของการทำงานปกติ ในขณะที่สามารถลดขนาดของโปรแกรมลงได้ประมาณ 40% ของโปรแกรมขนาดปกติ

จะเห็นได้ว่าด้วยวิธีการแปลโปรแกรมสามารถทำให้ขนาดของโปรแกรมลดลงได้เกือบครึ่งหนึ่ง แต่ในขณะเดียวกันกลับทำให้การทำงานช้าลงถึง 10 เท่า ดังนั้นในวิทยานิพนธ์นี้จึงนำเสนอวิธีการลดขนาดโปรแกรมฝังตัวในระบบฝังตัวขนาดเล็ก โดยการใช้วิธีการแปลโปรแกรมฝังตัวให้อยู่ในรูปของชุดคำสั่งกลาง เพื่อให้ได้โปรแกรมขนาดเล็ก แต่เนื่องจากการทำงานโดยซอฟต์แวร์แปลคำสั่งนั้นทำให้การทำงานของระบบช้าลงมาก ในวิทยานิพนธ์นี้จึงได้เสนอการสร้าง "วงจรแปลคำสั่ง" (Interpreter circuit) ขึ้นมาเพื่อใช้แทนซอฟต์แวร์แปลคำสั่ง โดยมีสมมติฐานว่าวงจรแปลคำสั่งจะลดเวลาที่ใช้ในการแปลคำสั่งลงได้มากเมื่อเทียบกับการใช้ซอฟต์แวร์แปลคำสั่ง โดยที่ขนาดของวงจรแปลคำสั่งที่เพิ่มขึ้นมาจะมีขนาดเล็กเมื่อเทียบกับขนาดของหน่วยความจำที่ลดลงได้

## 1.2 วัตถุประสงค์

1. เพื่อออกแบบระบบฝังตัวที่ใช้โปรแกรมขนาดเล็ก โดยใช้ชุดคำสั่งกลางและวงจรแปลคำสั่งในระบบฝังตัว
2. วัดประสิทธิภาพของระบบฝังตัวดังกล่าวใน 2 ประเด็นได้แก่ประสิทธิภาพในการลดขนาดของโปรแกรม และสมรรถนะการทำงานโดยรวมของระบบ

## 1.3 ขอบเขตงานวิจัย

สร้างระบบฝังตัวต้นแบบที่ใช้วิธีการลดขนาดของโปรแกรมโดยใช้หลักการการแปลคำสั่งของชุดคำสั่งกลาง ระบบดังกล่าวจะถูกออกแบบในระดับการโอนย้ายเรจิสเตอร์ (Register Transfer Level: RTL) และพัฒนาด้วยภาษาอธิบายลักษณะฮาร์ดแวร์เวอร์ริลอก (Verilog HDL)

การทดลองทำบนการจำลองของระบบเพื่อวัดค่าในการลดขนาดของโปรแกรม และสมรรถนะการทำงานโดยรวม หลังจากนั้นนำวงจรมาสังเคราะห์(Synthesis) เพื่อระบุขนาดของวงจรแปลคำสั่ง

## 1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาแนวความคิดต่างๆ ที่ใช้ในการลดขนาดโปรแกรม
2. ออกแบบการทำงานโดยรวมของระบบฝังตัวที่ใช้หลักการเครื่องเสมือน เลือกชุดคำสั่ง ชุดคำสั่งกลางที่เป็นชุดคำสั่งแบบแสตก และหน่วยประมวลผลที่จะนำมาใช้ในการสร้างระบบต้นแบบ
3. ปรับปรุงชุดคำสั่งแบบแสตกและหน่วยประมวลผลที่เลือกมาเพื่อให้รองรับการทำงานของวงจรแปลคำสั่ง
4. ออกแบบและสร้างวงจรแปลคำสั่ง
5. ทำการทดลองโดยจำลองการทำงานเพื่อทดสอบว่าแนวความคิดที่ใช้ในการบีบอัดและคลายโปรแกรมสามารถทำงานได้ถูกต้อง และขนาดของโปรแกรมเพื่อหาอัตราการบีบอัด รวมไปถึงสมรรถนะของระบบเครื่องเสมือนเปรียบเทียบกับระบบปกติ
6. สังเคราะห์ระบบเป็นวงจรจริงและวัดขนาดของวงจรแปลคำสั่ง
7. สรุปผลการทดลองและเขียนวิทยานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. วิธีในการลดขนาดของโปรแกรม ซึ่งทำให้ต้นทุนในการผลิตอุปกรณ์แบบฝังตัวลดลง
2. กระตุ้นให้เกิดการใช้การลดขนาดของหน่วยความจำในวงการอุตสาหกรรมผลิตระบบฝังตัวภายในประเทศ

## 1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บทดังนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงที่มาและความสำคัญของปัญหา รวมทั้งวัตถุประสงค์ของงานวิจัย บทที่ 2 สรุปงานวิจัยที่เกี่ยวข้องในด้านการลดขนาดโปรแกรม บทที่ 3 กล่าวถึงการทำงานพื้นฐานของแสตก (Stack) และนำเสนอรายละเอียดของชุดคำสั่งและหน่วยประมวลผลที่เลือกใช้ บทที่ 4 กล่าวถึงรายละเอียดการออกแบบเครื่องเสมือนแบบแสตก การทดลองและผลการทดลองแสดงไว้ในบทที่ 5 และสุดท้ายข้อสรุปจากการวิจัยถูกกล่าวถึงในบทที่ 6

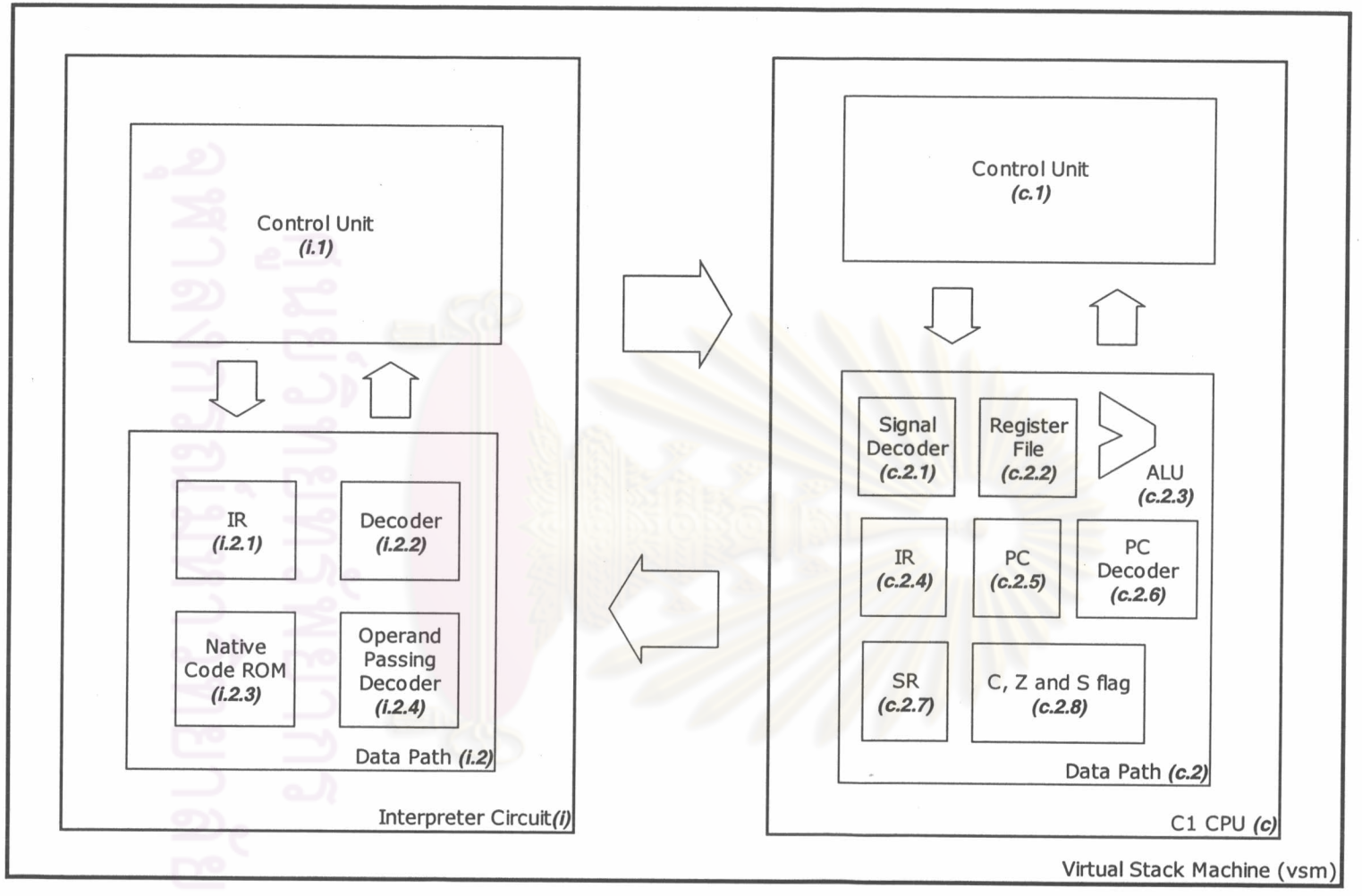
วิทยานิพนธ์นี้อธิบายถึงรายละเอียดการออกแบบของเครื่องเสมือนแบบแสตก ซึ่งอาจเกิดความสับสนในองค์ประกอบต่างๆ ทั้งหมด ดังนั้นในที่นี้ขอแสดงองค์ประกอบทั้งหมดในเครื่องเสมือนออกมาเป็นแผนภาพบล็อกดังรูปที่ 1.7 เวลาที่กล่าวถึงรายละเอียดของส่วนต่างๆ จะใช้แบบ

อักษร (Font) ***Helvetica ขนาด 10 แบบตัวหนาเอียง***อ้างอิงถึงหมายเลขที่กำกับของส่วนประกอบแต่ละส่วนตามรูปที่ 1.7

อนึ่งภายในวิทยานิพนธ์ใช้ศัพท์บางคำที่ไม่สอดคล้องกับชื่อวิทยานิพนธ์ได้แก่คำว่า Embedded Systems และ Bytecode Translator Unit โดยในกรณีของคำแปลของคำสั่ง Embedded Systems นั้นในชื่อวิทยานิพนธ์จะใช้คำว่า *ระบบฝังใน* ซึ่งไม่สอดคล้องกับคำที่บัญญัติในราชบัณฑิตดังนั้นในส่วนเนื้อหาจะขอใช้คำว่า *ระบบฝังตัว* แทน ส่วนคำว่า *Bytecode Translator Unit (วงจรแปลงรหัสไบต์)* ที่ใช้เรียกส่วนประกอบหนึ่งภายในวงจรที่ปรากฏในชื่อวิทยานิพนธ์นั้นยังสื่อความหมายไม่ดีนัก ดังนั้นจึงขอเปลี่ยนไปใช้คำว่า *Interpreter Circuit (วงจรแปลคำสั่ง)* แทน

## 1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง "Code-Size Reduction for Embedded Systems using Bytecode Translation Unit" โดย ภานุพันธ์ นันทนาวุฒิ และประภาส จงสถิตย์วัฒนา ในงานประชุมวิชาการ "ECTI Annual Conference (ECTI-CON 2004)" ซึ่งจัดโดยสมาคมวิศวกรรมไฟฟ้าแห่งประเทศไทย ณ โรงแรมอมารีออคิดรีสอร์ท พัทยา ชลบุรี ในระหว่างวันที่ 13-14 พฤษภาคม 2547




รูปที่ 1.7 แผนภาพบล็อกแสดงองค์ประกอบทั้งหมดของเครื่องเสมือนแบบแสตก