บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของการวิจัย

อุปกรณ์คอมพิวเตอร์ล้วนแต่ใช้พลังงานเป็นตัวขับเคลื่อนให้สามารถทำงานได้ ซึ่งในอดีตจะใช้พลังงานจากไฟบ้านโดยตรงจึงสามารถใช้พลังงานไฟฟ้าได้อย่างเต็มที่โดยไม่ต้องกังวลเกี่ยวกับพลังงานที่หมดไป แต่ในปัจจุบันอุปกรณ์คอมพิวเตอร์ได้ถูกออกแบบมาให้สามารถพกพาอุปกรณ์เหล่านั้นไปยังที่ต่าง ๆ ได้โดยใช้พลังงานที่สะสมไว้ในแบตเตอรี่ แต่พลังงานที่เก็บไว้ในแบตเตอรี่สามารถเก็บพลังงานได้อย่างจำกัดจึงส่งผลให้ระยะเวลาในการทำงานโดยใช้แบตเตอรี่ช่วยนั้นเป็นไปได้ช่วงระยะเวลาหนึ่งเท่านั้น ก่อให้เกิดความไม่ต่อเนื่องในการทำงานถ้าแบตเตอรี่หมดในระหว่างการทำงาน นอกจากนี้ระบบหลายระบบหรืออุปกรณ์หลายชนิดจำเป็นต้องเปิดให้ทำงานตลอดเวลาทำให้ความต้องการในการใช้พลังงานไฟฟ้าของระบบเหล่านี้สูง หากไม่มีการควบคุมการใช้กำลังงานอย่างเหมาะสมแล้วสิ่งที่ตามมาก็คือ พลังงานที่สูญเสียไปจำนวนมหาศาล และผลข้างเคียงเมื่อมีการใช้กำลังงานไฟฟ้าในระดับที่สูงมาก ๆ เป็นระยะเวลานาน ๆ คือ ความร้อนที่เกิดขึ้นกับอุปกรณ์ ซึ่งจะทำให้อุปกรณ์เหล่านี้มีอายุการใช้งานสั้นลงอีกทั้งระบบหรืออุปกรณ์อาจจะมีปัญหาเรื่องเสถียรภาพในการทำงานได้ถ้าระบบทำงานโดยปล่อยความร้อนออกมาสูงมากโดยไม่มีการระบายความร้อนที่ดี ดังนั้นการหาวิธีการลดพลังงานที่ใช้ในการทำงานของอุปกรณ์ต่าง ๆ เป็นสิ่งที่สำคัญและจำเป็นมากสำหรับเทคโนโลยีในปัจจุบันและอนาคต

การลดการใช้พลังงานไฟฟ้าอย่างเหมาะสมสามารถทำได้ 2 วิธีหลัก ๆ คือ วิธีทางด้านฮาร์ดแวร์และวิธีทางด้านซอฟต์แวร์

เทคนิคที่ช่วยให้การใช้พลังงานใช้ได้อย่างคุ้มค่ามากที่สุด ประกอบด้วย [1]

1) เทคนิคระดับอุปกรณ์ (device) หรือสถาปัตยกรรม (architecture) ต้องออกแบบอุปกรณ์ให้ใช้แรงดันต่ำและมีขนาดเล็ก วิธีนี้มีข้อจำกัด คือ สามารถลดแรงดันได้ถึงจุด ๆ หนึ่งเท่านั้น เพราะจะมีจุดวิกฤต (critical) ที่เป็นตัวกำหนดไม่ให้ลดแรงดันมากเกินกว่าจุดนี้ สำหรับการออกแบบสถาปัตยกรรมต้องมีการเคลื่อนย้ายข้อมูลออกไปนอกหน่วยประมวลผลน้อยที่สุด เช่น การใช้แคช (cache) ที่อยู่ในหน่วยประมวลผลจะช่วยประหยัดพลังงานได้ โดยเฉพาะอย่างยิ่งถ้าแคชมีขนาดใหญ่แล้วจะทำให้สามารถเก็บ

ข้อมูลไว้ในแคชได้มากด้วยซึ่งช่วยลดความจำเป็นที่หน่วยประมวลผลต้องเคลื่อนย้ายข้อมูลกับหน่วยความจำภายนอกช่วยให้ลดพลังงานลงได้มาก

2) เทคนิคระดับตัวแปลภาษา (complier) สามารถแปลงภาษาระดับสูงเป็นภาษาเครื่องโดยตระหนักถึงพลังงานที่ใช้ได้โดยจัดลำดับภาษาเครื่องที่สร้างให้ใช้พลังงานในการประมวลผลคำสั่งเหล่านั้นน้อยลง

3) เทคนิคระดับระบบปฏิบัติการ (OS) ระบบปฏิบัติการต้องฉลาดพอที่จะรู้ความจำเป็นในการใช้หน่วยประมวลผลของงานในแต่ละงาน และสามารถควบคุมความถี่และปรับแรงดันให้เหมาะสมกับความต้องการใช้งาน

4) เทคนิคระดับเครือข่าย (network) โปรแกรมต้องสามารถคาดคะเนความต้องการใช้งานเครือข่ายได้ โดยสามารถระงับการใช้งาน (disable) ระบบเครือข่ายได้เมื่อไม่มีความจำเป็นต้องใช้งานเครือข่าย และเมื่อมีความจำเป็นใช้งานอีกก็สามารถใช้งาน (enable) ระบบเครือข่ายเพื่อเป็นการประหยัดพลังงาน

การทำงานโดยตระหนักถึงกำลังงานที่ใช้ในส่วนของระบบปฏิบัติการ สามารถทำได้โดยการอนุญาตให้แอพลิเคชันใช้กำลังงานเท่าที่จำเป็นเท่านั้น พิจารณาได้จากงานบางอย่างที่ทำงานพร้อมกันแต่ความจำเป็นในการทำงานต่างกัน ระยะเวลาที่ใช้ในการทำงานแต่ละรอบไม่เหมือนกัน ซึ่งถ้าระบบปฏิบัติการไม่ฉลาดพอที่จะจัดสรรให้มีการใช้หน่วยประมวลผลได้ตามความต้องการของแอพลิเคชันแล้วจะทำให้สูญเสียกำลังงานโดยเปล่าประโยชน์ เช่น ถ้าให้งาน 2 งาน ทำงานเชิงเวลาจริงโดยให้งานแรกมีความจำเป็นในการใช้เวลาในการประมวลผลรอบละ 2 วินาที และงานที่สองมีความจำเป็นในการใช้เวลาในการประมวลผลรอบละ 1.5 วินาที ซึ่งถ้าระบบปฏิบัติการมีการจัดสรรทรัพยากรเท่ากันทั้ง 2 งาน คือ 2 วินาที จะทำให้การประมวลผลงานที่สองมีการใช้กำลังงานอย่างไม่เหมาะสม หรือถ้าจัดสรรให้ 1.5 วินาทีเท่ากัน ส่งผลให้งานแรกไม่สามารถทำงานได้ทันจะทำให้เกิดความเสียหายตามมาได้ เนื่องจากงานแรกทำงานเชิงเวลาจริงต้องทำงานให้เสร็จทันตามเวลาที่ได้กำหนดไว้ ดังนั้นการจัดสรรทรัพยากรให้เหมาะสมจึงเป็นสิ่งที่จำเป็นมากในระบบปฏิบัติการเชิงเวลาจริง

ระบบปฏิบัติการโดยทั่วไปไม่สามารถที่จะจัดสรรทรัพยากรโดยพิจารณาถึงเวลาในการทำงานเข้ามาเกี่ยวข้องได้แต่ระบบปฏิบัติการเชิงเวลาจริง (Real-Time Operating System : RTOS) เป็นระบบปฏิบัติการที่นำเวลาในการทำงานเข้ามาพิจารณาในการจัดสรรทรัพยากรได้ ซึ่งหมายความว่างานที่ทำเป็นงานแบบวิกฤตต้องทำให้เสร็จตามเวลาที่กำหนด สามารถจัดลำดับความสำคัญและแบ่งเวลาในการเข้าใช้หน่วยประมวลผลของงานแต่ละงานให้ได้อย่างเหมาะสม

ที่สุด สำหรับงานวิจัยนี้จะทำการศึกษาระบบปฏิบัติการเชิงเวลาจริง uC/OS-II [2] และปรับปรุง uC/OS-II ให้ใช้กำลังงานได้อย่างเหมาะสมที่สุดโดยนำไปใช้กับระบบควบคุมเชิงเวลาจริงแบบฝังตัว (Embedded real-time system) ที่มีติดต่อสื่อสารระหว่างอุปกรณ์หลาย ๆ ตัวโดยผ่านโพรโตคอล Modbus ซึ่งเป็นโพรโตคอลที่อำนวยความสะดวกให้การติดต่อสื่อสารระหว่างอุปกรณ์ในระบบควบคุมเชิงเวลาจริง ทั้งชนิดเดียวกันและต่างชนิดกันสามารถส่งข้อมูลและทำงานร่วมกันได้บนเครือข่ายที่หลากหลาย [3]

ในงานวิจัยชิ้นนี้ผู้วิจัยได้ทำการศึกษาค้นคว้าพบองค์ความรู้ที่จะเป็นประโยชน์ต่อวิชาการด้านวิทยาการคอมพิวเตอร์ดังนี้

- เทคนิคในการลดพลังงานไฟฟ้าของระบบปฏิบัติการเชิงเวลาจริง uC/OS-II สำหรับหน่วยประมวลผล ARM-7
- เทคนิคการตรวจสอบความถูกต้องและความน่าเชื่อถือของระบบควบคุมแบบฝังตัวเชิงเวลาจริงที่ทำงานโดยใช้โพรโตคอล Modbus

1.2 การตรวจเอกสาร บทความ และงานวิจัยที่เกี่ยวข้องกับเทคนิคในการออกแบบระบบปฏิบัติการเชิงเวลาจริงที่ตระหนักถึงกำลังงานที่ใช้

กำลังงานและพลังงานไฟฟ้าเป็นสิ่งจำเป็นมากในระบบที่มีการทำงานโดยใช้ไฟฟ้าเป็นตัวขับเคลื่อนโดยเฉพาะอย่างยิ่งระบบที่มีข้อจำกัดทางด้านกำลังงานที่สามารถใช้ได้ ดังนั้นจึงได้มีการคิดค้นหาเทคนิคใหม่ ๆ ที่สามารถจัดการกำลังงานที่สูญเสียไปอย่างเหมาะสมเพื่อเป็นการยืดอายุการทำงานของอุปกรณ์ให้สามารถทำงานได้นานยิ่งขึ้น

ระบบที่ตระหนักถึงกำลังงานที่ใช้มีความแตกต่างจากระบบที่ใช้กำลังงานต่ำ นั่นคือระบบที่ใช้กำลังงานต่ำมีจุดประสงค์เพื่อให้มีการใช้กำลังงานน้อยที่สุดในขณะที่ระบบที่ตระหนักถึงกำลังงานที่ใช้พิจารณาถึงการเปลี่ยนแปลงพฤติกรรมในการใช้พลังงานและกำลังไฟฟ้า ความแตกต่างของ 2 ระบบดังกล่าวอธิบายได้ดังนี้

ระบบที่ตระหนักถึงกำลังงานที่ใช้ไม่ได้หมายความว่าจะทำให้ระบบมีการใช้พลังงานหรือกำลังงานน้อยลงเพราะในบางครั้งการออกแบบระบบที่ตระหนักถึงกำลังงานที่ใช้อาจเป็นการเพิ่มปริมาณการใช้พลังงานได้ เช่น การลดค่าสูงสุดของกำลังงานไฟฟ้าที่ใช้ในการประมวลผลโดยเลื่อนให้บางคำสั่งทำงานช้าลงส่งผลให้ระยะเวลาที่ใช้ในการประมวลผลนานขึ้นทำให้พลังงานที่ใช้เพิ่มสูงขึ้นเนื่องจากค่าพลังงานขึ้นอยู่กับเวลาที่ใช้ในการทำงานด้วย ดังนั้นการออกแบบระบบที่ตระหนักถึงกำลังงานที่ใช้ไม่ได้เป็นการลดพลังงานที่ใช้โดยตรง

การลดค่าเฉลี่ยของกำลังงานไม่ได้เป็นการลดค่าสูงสุดของกำลังงานที่ใช้ เนื่องจากค่าเฉลี่ยของกำลังงานที่สูญเสียไปคิดจากเวลาในการทำงานตั้งแต่ต้นจนจบ ส่วนค่าสูงสุดของกำลังงานคิด ณ เวลาหนึ่งเท่านั้น การลดค่าเฉลี่ยของกำลังงานที่ใช้อาจนำไปสู่การใช้กำลังงานที่สูงขึ้นก็เป็นได้

เป้าหมายในการออกแบบระบบให้ใช้กำลังงานและพลังงานอย่างมีประสิทธิภาพมีความแตกต่างกันเนื่องจากพลังงาน คือ ตัวเลขการใช้กำลังงานในช่วงเวลาหนึ่ง ๆ สำหรับการออกแบบการใช้กำลังงานให้มีประสิทธิภาพวิธีหนึ่ง คือ การลดจำนวนรอบของสัญญาณนาฬิกา วิธีการนี้ไม่ได้เพิ่มประสิทธิภาพในการใช้พลังงานเพราะส่งผลให้ระยะเวลาในการทำงานนานขึ้น พลังงานที่ใช้สูงขึ้น

ระบบเชิงเวลาจริงแบ่งออกเป็น 2 ประเภทคือ hard real-time system และ soft real-time system สำหรับ hard real-time system เป็นระบบที่จะต้องทำงานให้เสร็จทันตามเวลาที่กำหนดไว้หากไม่สามารถทำงานได้เสร็จทันตามกำหนดเวลาจะก่อให้เกิดความเสียหายตามมา ส่วน soft real-time system เป็นระบบที่พยายามทำงานให้เสร็จทันตามกำหนดเวลาหากไม่สามารถทำงานได้เสร็จทันตามเวลาที่กำหนดไว้ก็จะไม่ก่อให้เกิดความเสียหาย

เทคนิคการออกแบบระบบเชิงเวลาจริงที่ตระหนักถึงกำลังงานที่ใช้แบ่งออกเป็น 4 ระดับใหญ่ ๆ ได้แก่ ระดับอุปกรณ์หรือสถาปัตยกรรม ตัวแปลภาษา ระบบปฏิบัติการ และเครือข่าย

1.2.1 ระดับอุปกรณ์หรือสถาปัตยกรรม

CMOS (Complementary Metal Oxide Semiconductor) เป็นวงจรที่ใช้กำลังงานต่ำเมื่อไม่มีการเปลี่ยนแปลงข้อมูลหรือสัญญาณใด ๆ และใช้กำลังงานสูงมากเมื่อมีการเปลี่ยนแปลงข้อมูลหรือสัญญาณเกิดขึ้น เนื่องจาก CMOS เป็นวงจรที่ใช้กำลังงานต่ำจึงได้นำมาใช้พัฒนากับอุปกรณ์ที่ใช้กำลังงานจากแบตเตอรี่เพื่อเป็นการยืดระยะเวลาการทำงานของอุปกรณ์ให้นานขึ้นหรือระบบที่ต้องการลดปริมาณการใช้กำลังงาน ดังนั้นการออกแบบระบบสิ่งที่ควรคำนึงถึงนอกจากประสิทธิภาพในการทำงานแล้วต้องคำนึงถึงกำลังงานที่ใช้ด้วยเช่นกัน

เมื่อมีการเปลี่ยนสภาวะของข้อมูลหรือสัญญาณจะเกิดการสูญเสียกำลังงานในวงจร CMOS เพิ่มขึ้น ค่าเฉลี่ยของกำลังงานที่ใช้แสดงดังสมการ (1)

$$P = C_L \cdot N_{SW} \cdot V^2{}_{DD} \cdot f \qquad (1)$$

C_L คือปริมาณความจุไฟฟ้าทั้งหมด N_{SW} คือค่าเฉลี่ยของ circuit switch ต่อรอบของสัญญาณนาฬิกา V_{DD} คือแรงดัน และ f คือความถี่ของสัญญาณนาฬิกา

จากสมการ (1) การลดกำลังงานที่ใช้ด้วยค่าความจุไฟฟ้าทำได้โดยใช้เทคนิคการออกแบบให้วงจรมีขนาดเล็กลง ส่วนค่าเฉลี่ยของ circuit switch ใช้เทคนิคให้มีการทำงานต่อหนึ่งคำสั่งลดลงด้วยวิธี Hamming distance หรือเป็นการลดจำนวนคำสั่งให้น้อยลงนั่นเอง วิธีการนี้ใช้เทคนิคในการออกแบบทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ร่วมกัน ในส่วนของการลดแรงดันใช้เทคนิคในการผลิตชิพให้มีขนาดเล็กลงแต่ก็ส่งผลให้การทำงานของวงจรช้าลงด้วย เมื่อการทำงานของวงจรช้าลงจึงไม่สามารถทำงานที่ความถี่เดิมได้อีกดังนั้นต้องปรับความถี่ให้ต่ำลง การตระหนักถึงกำลังงานที่ใช้ในระบบเชิงเวลาจริงส่วนมากจะใช้เทคนิคในการลดแรงดันและความถี่ของสัญญาณนาฬิกาวิธีการนี้เรียกว่า Dynamic Voltage Scaling (DVS) ซึ่งเป็นวิธีที่ใช้กันมากในระดับระบบโดยไม่เน้นถึงประสิทธิภาพในการทำงานมากนักเพราะทำให้การทำงานของระบบช้าลง

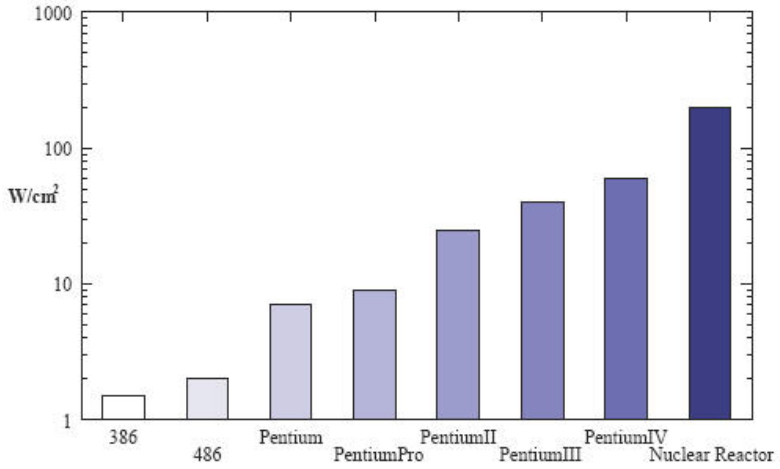
การลดขนาดของวงจรส่งผลให้ความจุไฟฟ้าและแรงดันลดลงด้วยซึ่งสามารถลดลงได้ถึง X^3 (X คืออัตรากำลังงานที่สูญเสีย) โดยที่ความถี่ยังคงเท่าเดิม (พิจารณาจากสมการที่ 1) แต่การลดขนาดอุปกรณ์มีข้อจำกัดนั่นคือสามารถลดการใช้กำลังงานได้ถึงจุด ๆ หนึ่งเท่านั้น เนื่องจากการลดขนาดอุปกรณ์ทำให้มีการเพิ่มความหนาแน่นของวงจรและความถี่ของสัญญาณนาฬิกาส่งผลให้ใช้กำลังงานสูงขึ้น ผลที่ได้จากการลดขนาดของวงจรที่ทำงานบนหน่วยประมวลผล Alpha พบว่ามีการสูญเสียกำลังงานเพิ่มมากขึ้นถึง 5 เท่าแสดงดังตาราง 1-1

ตาราง 1-1 Case Study For Scaling : The Compaq Alpha

Processor	Power (Watts)	Freq.(MHz)	Die Size (mm^2)	V_{dd}
21064	30	200	234	3.3
21164	50	300	299	3.3
21264	90	575	313	2.2
21364	100	1000	340	1.5
21464	150	2000	396	1.2

การออกแบบอุปกรณ์และสถาปัตยกรรมมีข้อจำกัด นั่นคือสามารถทำได้ถึงจุด ๆ หนึ่งเท่านั้นซึ่งนำไปสู่การคิดค้นและหาแนวทางอื่น ๆ เพื่อช่วยสนับสนุนให้ระบบมีการตระหนักถึงกำลังงานที่ใช้ได้ดียิ่งขึ้น คือ การออกแบบทางด้านตัวแปลภาษา ระบบปฏิบัติการ และเครือข่าย

ปัจจัยหนึ่งที่ทำให้อุปกรณ์มีความร้อนสูงขึ้น คือ ความหนาแน่นของกำลังงาน ซึ่งความหนาแน่นของกำลังงานมีหน่วยเป็นวัตต์ต่อตารางเซนติเมตร (watt/cm^2) ถ้ามีค่าสูงส่งผลให้เกิดความร้อนเพิ่มขึ้น ผลที่ตามมาคือต้องเสียค่าใช้จ่ายที่สูงมากในการระบายความร้อนออกจากอุปกรณ์ ดังนั้นการออกแบบระบบที่ตระหนักถึงกำลังงานที่ใช้นอกจากการคำนึงถึงกำลังงานที่ใช้แล้วต้องคำนึงถึงความร้อนที่เกิดขึ้นด้วย ความหนาแน่นของกำลังงานในหน่วยประมวลผลแต่ละชนิดเมื่อเปรียบเทียบกับความหนาแน่นที่เกิดขึ้นในระบบ Nuclear reactor พบว่าอุปกรณ์ที่ผลิตบนเทคโนโลยีที่สูงขึ้นมีความหนาแน่นของกำลังงานสูงขึ้นมากจนเกือบเทียบเท่ากับระบบ Nuclear reactor แสดงดังภาพประกอบ 1-1



ภาพประกอบ 1-1 ความหนาแน่นของกำลังงานในชิพ Intel

1.2.2 ระดับตัวแปลภาษา

วัตถุประสงค์หลักในการออกแบบระบบคอมพิวเตอร์ฝังตัวที่ทำงานบนระบบปฏิบัติการเชิงเวลาจริงที่ตระหนักถึงกำลังงานที่ใช้ในระดับตัวแปลภาษา คือ ออกแบบให้มีการใช้งานหน่วยความจำได้อย่างเหมาะสมที่สุดภายใต้ข้อจำกัดของหน่วยความจำที่สามารถใช้ได้ ในที่นี้จะกล่าวถึง 2 เทคนิคหลัก ๆ ที่เกี่ยวข้องกับกำลังงานที่ใช้ในระดับนี้ คือ การปรับแรงดันควบคู่กับความถี่ และการจัดโครงสร้างหน่วยความจำ

1.2.2.1 การปรับแรงดันและความถี่ (Voltage and frequency scaling)

Azavedo และคณะ [4] ได้พัฒนาตัวแปลภาษาด้วยการตรวจสอบจุดที่ใช้ตรวจสอบพลังงานที่ใช้ (checkpoint) และใช้เทคนิค DVS ซึ่งการ checkpoint จะเกิดขึ้นเมื่อมีการแปลคำสั่งจากโปรแกรมต้นฉบับ (source code) จากนั้นจะระบุส่วนที่ต้องมีการคำนวณซ้ำและตรวจสอบการเปลี่ยนแปลงแรงดันและความถี่ของหน่วยประมวลผล ข้อมูลที่ได้จากการทำ checkpoint ในแต่ละครั้งเป็นข้อมูลระดับภายในงาน การทำ checkpoint ผลที่ได้เก็บไว้ในโพรไฟล์ (profile) ที่รายงานข้อมูลกำลังงาน พลังงาน และประสิทธิภาพต่ำสุดและสูงสุดต่อการตรวจสอบในแต่ละ checkpoint เพื่อที่จะสามารถปรับแรงดันและความถี่ได้อย่างเหมาะสมในรอบต่อ ๆ ไป

1.2.2.2 การจัดโครงสร้างหน่วยความจำ (Memory organization)

Levy และคณะ [5] ได้กล่าวว่าระบบคอมพิวเตอร์ฝังตัวมีการสูญเสียพลังงานอันเนื่องมาจากการใช้และการทำงานบนระบบหน่วยความจำมากกว่า 50% เป้าหมายหลักของเอกสารนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องที่ตระหนักถึงกำลังงานที่ใช้ด้วยตัวแปลภาษาโดยพิจารณาถึงการใช้หน่วยความจำและลักษณะเฉพาะสำหรับระบบคอมพิวเตอร์ฝังตัวที่ตระหนักถึงกำลังงานที่ใช้ เช่น การแบ่งหน่วยความจำเป็นบล็อกเล็ก ๆ (block granularity) และขั้นตอนวิธีที่ใช้ในการแทนที่แคช (cache replacement algorithm)

สำหรับ soft real-time system การวิเคราะห์/สำรวจหน่วยความจำเพื่อตระหนักถึงกำลังงานที่ใช้ด้วยเทคนิคทางด้านตัวแปลภาษาเป็นสิ่งที่สำคัญมาก จากการทดสอบประสิทธิภาพในการใช้พลังงาน [6] ด้วยแปลภาษาที่เหมาะสมในการถอดรหัส MPEG บนระบบคอมพิวเตอร์ฝังตัวที่ใช้พลังงานจากแบตเตอรี่ พบว่านอกจากการใช้ตัวแปลที่เหมาะสมแล้วการเขียนโปรแกรมต้นฉบับที่ดีสามารถประหยัดพลังงานได้มากกว่าการใช้ตัวแปลภาษาที่เหมาะสม ซึ่งเป็นประเด็นหนึ่งที่ช่วยเพิ่มประสิทธิภาพในการทำงานให้มีการใช้กำลังงานที่ต่ำลง นั่นคือ โปรแกรมต้นฉบับที่ไม่เหมาะสมเมื่อนำไปแปลบนตัวแปลภาษาที่ดีสามารถประหยัดพลังงานได้น้อยกว่า 1% แต่ถ้านำ source code ที่ดีไปแปลบนตัวแปลภาษาที่เหมาะสมสามารถประหยัดพลังงานได้มากถึง 90%

Grun และคณะ [7] ได้สร้างระบบคอมพิวเตอร์ฝังตัวที่ทำงานร่วมกับโพรไฟล์และจัดหมวดหมู่การเข้าถึงหน่วยความจำ นำผลที่ได้จากโพรไฟล์มาใช้ในการสร้างสถาปัตยกรรมของหน่วยความจำที่มีรูปแบบและตำแหน่งการเข้าถึงที่แตกต่างกันด้วยการแบ่งแคชที่มีขนาดใหญ่

ออกเป็นส่วน ๆ และจัดกลุ่มการเข้าถึงหน่วยความจำให้เหมาะสม นอกจากนี้การใช้แคชชั่วคราว (temporal cache) สามารถลดกำลังงานที่หน่วยความจำใช้ได้ถึง 30% โดยที่ประสิทธิภาพในการทำงานไม่ลดลง

1.2.3 ระดับระบบปฏิบัติการ

ระบบปฏิบัติการเชิงเวลาจริงมีลักษณะหรือโครงสร้างในการออกแบบแบ่งเป็นโมดูล ตามหน้าที่การทำงาน เช่น โมดูลที่เกี่ยวกับเครือข่ายหรือโมดูลที่เกี่ยวกับการติดต่อสื่อสารผ่านทาง serial port เป็นต้น นอกจากนี้ยังสามารถเพิ่มโมดูลได้ตามความต้องการ ระบบปฏิบัติการทั่วไปทำงานอยู่บน micro-kernel จากลักษณะดังกล่าวสามารถออกแบบการประมวลผลที่ตระหนักถึงกำลังงานที่ใช้ได้โดยการตัดโมดูลที่ไม่จำเป็นหรือไม่มีการใช้งานออกไป เทคนิคที่ใช้สำหรับการตระหนักถึงกำลังงานในระดับนี้ ประกอบด้วย DVS การจัดการอุปกรณ์ I/O การจัดการใน soft real-time system และการวัดกำลังงานที่สูญหายไปในระหว่างที่ระบบปฏิบัติการกำลังดำเนินการอยู่ เป็นต้น

1.2.3.1 การปรับแรงดันและความถี่ (Voltage and Frequency Scaling)

Pillai และ Shin [8] ได้ทดลองและพัฒนาการจัดการกำลังงานในระบบปฏิบัติการเชิงเวลาจริงด้วยวิธีการหน่วงเวลาการทำงานในแต่ละช่วงเวลา และทำการปรับแรงดันและความถี่ให้เหมาะสมด้วยเงื่อนไขของการทำงานที่ไม่ขึ้นต่อกันในแต่ละงานที่งาน นำผลที่ได้รวมเข้าเป็นส่วนหนึ่งของ Linux kernel 2.2.16 จากผลการทดลองเมื่อทำการวัดกำลังงานที่ใช้พบว่าวิธีการนี้สามารถประหยัดกำลังงานที่ใช้ได้จริง

1.2.3.2 อุปกรณ์รับเข้า/ส่งออกข้อมูล (I/O devices)

Swaminathan และคณะ [9] ได้ศึกษาเกี่ยวกับการจัดการตารางเวลาการเข้าใช้ I/O บน hard real-time system ที่ตระหนักถึงกำลังงานที่ใช้โดยเสนอขั้นตอนวิธีการกำหนดตารางการเข้าใช้อุปกรณ์ต่าง ๆ ไว้ล่วงหน้าซึ่งประกอบด้วยรายการของข้อมูลเข้า (input) และผลลัพธ์ (output) ของอุปกรณ์ในสถานะหลับ (sleeping state) และสถานะที่มีการดำเนินงาน (working state) โดยมีข้อกำหนดที่ว่างานแต่ละงานจะเป็นอิสระต่อกันและต้องทำงานให้เสร็จทันตามเวลาที่

กำหนดไว้ วิธีการนี้ไม่สามารถทำได้ถ้าไม่มีฐานความรู้เกี่ยวกับการร้องขอหรือเรียกใช้อุปกรณ์ล่วงหน้า

1.2.3.3 การวิเคราะห์กำลังงานและพลังงานในระบบปฏิบัติการเชิงเวลาจริง (Power and Energy Analysis of RTOS)

Dick และคณะ [10] ได้วิเคราะห์การใช้กำลังงานบนระบบปฏิบัติการ uC/OS ซึ่งเป็นระบบปฏิบัติการเชิงเวลาจริงชนิดหนึ่ง โดยรันแอพลิเคชัน 2 ตัวบนระบบคอมพิวเตอร์ฝังตัวที่ใช้หน่วยประมวลผล Fujitsu SPARClite-processor-based และหาค่ากำลังงานที่ใช้เมื่อเรียกใช้ระบบปฏิบัติการ (operating system call) และแสดงให้เห็นว่าการใช้กำลังงานของระบบปฏิบัติการเชิงเวลาจริงสัมพันธ์กับการใช้กำลังงานของระบบ Baynes และคณะ [11] ได้วิเคราะห์ระบบปฏิบัติการ uC/OS ควบคู่กับระบบปฏิบัติการเชิงเวลาจริงอื่น ๆ อีก 2 ระบบปฏิบัติการ คือ Ecnidna และ NOS โดยให้ NOS เป็นตัวเปรียบเทียบกับตัวอื่น ๆ ผลการทดสอบพบว่าระบบปฏิบัติการเชิงเวลาจริง Ecnidna และ uC/OS มีปัจจัยแฝงที่เป็น overhead ในการทำงานเกิดขึ้นถึง 2-4 เท่าเมื่อเปรียบเทียบกับ NOS สาเหตุหนึ่งพบว่าการออกแบบการวนรอบแบบไม่เกิดงาน (idle loop) ที่ไม่ดีทำให้ระบบทำงานเพิ่มขึ้นเป็น 2 เท่า ส่งผลให้มีการใช้พลังงานสูงขึ้น Acquaviva และคณะ [12] ทดลองวัดกำลังงานของระบบปฏิบัติการเชิงเวลาจริงที่มีการรันแอพลิเคชันที่ทำงานไม่ขึ้นต่อกันและทดสอบกับระบบปฏิบัติการหลาย ๆ แบบพบว่าการเพิ่มความถี่ในการทำ context switch จาก 0 KHz ไปจนถึง 10 KHz ไม่ส่งผลกระทบต่อการใช้กำลังงาน สรุปได้ว่ากระบวนการในการทำ context switching ของระบบปฏิบัติการเชิงเวลาจริงใช้พลังงานอย่างมีประสิทธิภาพ แต่เมื่อรวมกระบวนการ flushing ของแคชในระหว่างการทำ context switching เข้ามาพิจารณาด้วยจะทำให้เกิด overhead ซึ่งส่งผลกระทบให้กำลังงานที่ใช้สูงขึ้น นอกจากนี้ Acquaviva และคณะได้ทดลองเกี่ยวกับ I/O driver พบว่าเมื่อหน่วยประมวลผลสั่งให้มีการส่งข้อมูลขนาดใหญ่มากเมื่อเทียบกับ output buffer และเกิดกรณีบัฟเฟอร์เต็มหรือมีการเข้าจังหวะกันของระบบปฏิบัติการส่งผลให้มีการใช้พลังงานเพิ่มมากขึ้น ดังนั้นสรุปได้ว่าการส่งข้อมูลขนาดเล็ก ๆ เป็นอีกวิธีหนึ่งที่ตระหนักถึงพลังงานที่ใช้ในระบบปฏิบัติการเชิงเวลาจริง

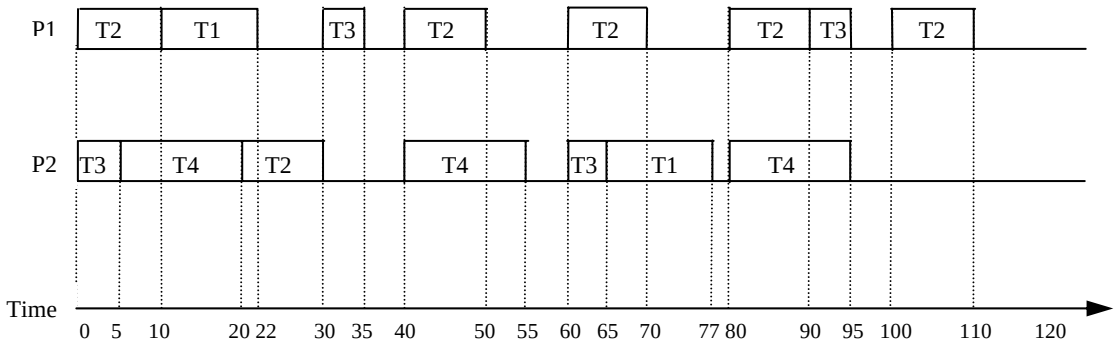
1.2.3.4 ระบบเชิงเวลาจริงที่มีการทำงานแบบกระจาย (Distributed real-time systems)

Unsal และคณะ [13] ได้ทดลองเกี่ยวกับการจองและการจัดการตารางเวลาในระดับสูงบน hard real-time system โดยพิจารณาทั้งระบบที่ผูกเข้าด้วยกันแบบหลวม ๆ และแบบแน่น สำหรับระบบที่ผูกเข้าด้วยกันแบบหลวม ๆ การตระหนักถึงกำลังงานที่ใช้ทำได้โดยการแบ่งกลุ่มงานที่มีโครงสร้างข้อมูลเหมือนกันหรือมีโครงสร้างการประมวลผลใกล้เคียงกันให้ใช้โครงสร้างข้อมูลร่วมกันได้ เนื่องจากปัญหานี้เป็นปัญหาที่ยังหาวิธีที่ดีที่สุดไม่ได้ (NP-complete) จึงทำได้เพียงจำลองสถานการณ์ที่เหมาะสมเพื่อหาวิธีการแก้ปัญหาที่ดีที่สุด สำหรับคุณสมบัติของระบบที่ผูกกันอย่างแน่น ๆ คือ ระบบต้องมีความสามารถในการทำงานหลาย ๆ งานที่ใช้เวลาในการประมวลผลพร้อมกันบนส่วนของหน่วยความจำ (memory bank) ที่ต่างกัน ดังนั้น Unsal และคณะจึงพิจารณาการจัดงานเข้าสู่ memory bank พบว่าเมื่อปิด memory bank ขณะที่ไม่มีความจำเป็นในการใช้งานสามารถประหยัดพลังงานได้ส่วนหนึ่ง จากประเด็นดังกล่าวการทำงานของระบบที่ผูกกันอย่างแน่น ๆ ขัดแย้งกับคุณลักษณะการจัดการ memory bank ที่ตระหนักถึงกำลังงานที่ใช้ Unsal และคณะจึงได้ทดสอบและวิเคราะห์การใช้พลังงานของระบบนี้ ซึ่งประกอบด้วยหน่วยประมวลผล 2 หน่วย สามารถปรับเปลี่ยนกำลังงานที่ใช้ได้โดยเปลี่ยนสถานะของ memory bank ให้เข้าสู่สถานะหลับ โดยที่ memory bank มีหน้าที่เก็บข้อมูลในส่วน data section และ text section มีคิวสำหรับการจัดลำดับการทำงาน 1 คิว สำหรับระบบที่ตระหนักถึงกำลังงานที่ใช้ถ้าในเวลาหนึ่ง ๆ มีการใช้งาน memory bank เพียง X หน่วย memory bank ที่เหลือ คือ 2-X หน่วยต้องเข้าสู่สถานะหลับ (sleeping mode) การที่ memory bank อยู่ในสถานะหลับจะไม่มีการสูญเสียกำลังงานแต่อย่างใด โดยไม่นำความล่าช้าที่เกิดจากการเข้าถึง memory bank พร้อม ๆ กันของงานมากกว่าหนึ่งงานมาใช้ในการพิจารณาระบบที่ทดสอบข้อมูลของแต่ละงานแสดงดังตาราง 1-2 สมมติให้การใช้ประโยชน์จากหน่วยความจำของงานแต่ละงานสัมพันธ์กับการทำงาน (execution utilization) แบบเชิงเส้น การจัดตารางการเข้าใช้หน่วยประมวลผล (multiprocessor) จะจัดให้งานที่มีกำหนดเวลาน้อย ๆ เข้าไปอยู่ในคิวก่อน ผลการทำงานแสดงดังภาพประกอบ 1-2

การใช้กำลังงานของ memory bank ที่ตระหนักถึงกำลังงานที่ใช้ขึ้นอยู่กับการจอง memory bank ให้กับงาน และต้องพยายามให้มีการใช้งาน memory bank อย่างสมดุลและเกิดประโยชน์มากที่สุดด้วยวิธีการ load balancing (LB) ซึ่งเป็นวิธีที่จัดให้งานที่เกี่ยวข้องกันอยู่ใน memory bank เดียวกัน

ตาราง 1-2 ตัวอย่างกลุ่มงาน

Task id	Execution Time	Period	Utilization (%)
T1	12	60	20.0
T2	10	20	50.0
T3	5	30	16.6
T4	15	40	37.5



ภาพประกอบ 1-2 Execution timeline ของงาน

1.2.3.5 Soft real-time system

ระบบ soft real-time ส่วนใหญ่เป็นระบบที่ทำงานอยู่บน pocket PC แอพลิเคชันที่ทำงานบนระบบนี้ ได้แก่ การจำลายมือ การเล่นเสียงและวีดีโอ และ ระบบตรวจสอบตำแหน่งของโลก : Global Positioning System (GPS) Farks และคณะ [14] พิจารณาคุณสมบัติของกำลังงานที่ใช้ใน pocket PC และแสดงให้เห็นว่าช่วงของกำลังงานที่ใช้มีค่ากว้างและไม่คงที่เมื่อเปรียบเทียบกับกำลังงานที่ใช้บนเครื่องคอมพิวเตอร์แบบ notebook จากนั้นพิจารณาระบบที่ออกแบบมาเพื่อสนับสนุน Java เพิ่มเติมเนื่องจาก Java ใช้ Java Virtual Machine (JVM) ในการทำงานซึ่งมีหน้าที่เป็นตัวกลางการติดต่อระหว่างแอพลิเคชันและระบบปฏิบัติการ จากการวิเคราะห์พบว่าการใช้ JVM เพียงตัวเดียวที่ผู้ใช้สามารถใช้งานร่วมกับระบบได้จะใช้พลังงานได้อย่างมี

ประสิทธิภาพมากกว่าการใช้หนึ่ง JVM ต่อหนึ่งแอพลิเคชัน และอีกเทคนิคหนึ่งคือการใช้ just-in time จะช่วยให้ระยะเวลาในการทำงานทั้งหมดลดลงส่งผลให้ประสิทธิภาพในการใช้พลังงานดีขึ้น Yuan และ Nahrstedt [15] ได้พิจารณาถึงวิธีการที่ตัวกลางใช้ในการติดต่อระหว่างแอพลิเคชันกับระบบปฏิบัติการ (middleware) เช่นเดียวกันแต่ทดสอบกับแอพลิเคชันที่ทำงานอยู่บน laptop แทน pocket PC โดยใช้ระบบปฏิบัติการ Windows NT และพัฒนาตัวกลางสำหรับติดต่อระบบที่สามารถปรับค่ากำลังงานที่ใช้ได้ (Advanced Configuration and Power Interface : ACPI) ทดสอบกับการถอดรหัสของ MPEG ที่ทำงานเป็นช่วงเวลาแบบไม่คงที่ และทดสอบกับแอพลิชันที่ทำงานทางด้านคณิตศาสตร์ที่มีการประมวลผลเป็นช่วงเวลาที่แน่นอน พบว่าการจำลองในลักษณะนี้สามารถประหยัดพลังงานได้ถึง 39.5%

1.2.3.6 แบตเตอรี่ (Batteries)

Ma และ Shin [16] ทดสอบคุณภาพการทำงานที่ได้รับกับการพัฒนาขั้นตอนวิธีในการจัดตารางเวลาเพื่อตระหนักถึงพลังงานที่ใช้บนระบบงานแบบวิกฤติ ผลที่ได้สามารถยืดอายุการใช้งานของแบตเตอรี่ได้แต่จะส่งผลต่อประสิทธิภาพในการทำงาน ผู้ใช้สามารถปรับระดับประสิทธิภาพและอายุการทำงานของแบตเตอรี่ได้ตามความต้องการ นั่นคือถ้าต้องการขยายอายุการใช้งานแบตเตอรี่มากขึ้นเท่าไรประสิทธิภาพการทำงานก็จะลดลงตามลำดับ ผลการทดสอบที่ได้จากการจำลองพบว่าสามารถยืดอายุการใช้แบตเตอรี่ได้สูงถึง 100% โดยที่ประสิทธิภาพลดลงถึง 40% Benini และคณะ [17] ทดสอบการเปิดและปิดกล้องบันทึกวีดีโอหลาย ๆ รอบเพื่อพิสูจน์ข้อเท็จจริงเกี่ยวกับค่าเฉลี่ยของกำลังงาน พวกเขาพบว่าการเปิดและปิดกล้องบันทึกวีดีโอหลาย ๆ รอบสามารถลดค่าเฉลี่ยของกำลังงานที่ใช้ได้แต่ไม่ได้เป็นการยืดอายุการทำงานของแบตเตอรี่ การเพิ่มช่วงเวลาในการทำงานของแบตเตอรี่ให้สามารถจ่ายพลังงานให้ได้นานที่สุดเป็นวัตถุประสงค์หลักของการทดสอบนี้ Flinn และ Satyanarayanan [18] ได้นำเสนอวิธีการที่ช่วยขยายระยะเวลาในการจ่ายพลังงานของแบตเตอรี่ให้กับแอพลิเคชันและระบบปฏิบัติการเช่นเดียวกับ Benini และคณะโดยศึกษาแอพลิเคชันที่ใช้สำหรับเล่นวีดีโอ การจำเสียง แผนที่อิเล็กทรอนิกส์ (map viewer) และ เว็บเบราเซอร์ (web browser) เป็นต้น ผลที่ได้สามารถยืดระยะเวลาในการทำงานของแบตเตอรี่ได้นานกว่าเดิมถึง 30%

1.2.3.7 Web server

อินเทอร์เน็ต (Internet) เป็นแอพลิเคชันที่ได้รับความนิยมเป็นอย่างมาก เครื่องที่ทำหน้าที่ให้บริการเครื่องอื่น (Server) ต้องทำงานตลอดเวลาและใช้พลังงานสูงมาก Bohrer และคณะ [19] จึงทดสอบประสิทธิภาพการใช้พลังงานของเครื่องให้บริการเว็บและวัดประสิทธิภาพการทำงานของเครื่องเหล่านี้ในช่วงที่ทำงานหนัก พบว่าการทำงานในช่วงนี้เหมือนการทำงานบนระบบ hard real-time ดังนั้นเครื่องให้บริการที่ตระหนักถึงกำลังงานที่ใช้ต้องมีความฉลาดและสามารถคาดคะเนถึงการใช้งานได้ ช่วงใดที่มีการใช้งานต่ำก็ปรับให้ใช้กำลังงานเท่าที่จำเป็น ข้อมูลจริงจากเว็บล็อค (web log) ในปี 1998 จากงานโอลิมปิกที่ Nagano พบว่ามีการเข้าใช้งานสูงถึง 1,840 ครั้งต่อวินาทีในขณะที่ค่าเฉลี่ยมีค่าเป็น 459 ครั้งต่อวินาทีเท่านั้น ข้อมูลนี้แสดงให้เห็นว่าค่าเฉลี่ยของกำลังงานที่ใช้เป็นเพียง 25% ของจุดสูงสุด การทำนายช่วงที่มีกิจกรรมการใช้ทรัพยากรต่ำและนำค่าที่ได้กำหนดแรงดันและความถี่ให้เหมาะสมสามารถประหยัดกำลังงานได้ถึง 23-36% โดยยังรักษาประสิทธิภาพในการตอบสนองของเครื่องให้บริการได้

1.2.3.8 Jitter

Jitter เป็นคุณสมบัติอย่างหนึ่งที่ไม่พึงปรารถนาในการทำงานของแอพลิเคชัน แต่คุณสมบัตินี้สามารถนำมาใช้ในการวิเคราะห์เพื่อปรับให้ระบบ soft real-time สามารถใช้กำลังงานได้อย่างเหมาะสมในการรับส่งข้อมูลโดยพิจารณาจากความเร็วของเฟรมข้อมูลที่มาถึง ถ้ามาถึงเร็วให้ปรับลดความเร็วของเฟรมข้อมูลที่มาถึงด้วยวิธี DVS แต่ถ้าเฟรมข้อมูลมาถึงช้าเนื่องจากมีบางเฟรมที่ไม่สำคัญถูกส่งมาด้วยให้ทิ้งเฟรมข้อมูลนั้น ๆ เพื่อเป็นการประหยัดกำลังงานและเพิ่มประสิทธิภาพการทำงานให้ดียิ่งขึ้น

1.2.4 ระดับเครือข่าย

งานวิจัยที่เกี่ยวกับระบบเชิงเวลาจริงในระดับนี้ส่วนใหญ่เป็นการออกแบบโพรโตคอล สำหรับติดต่อสื่อสารโดยต้องอยู่ในขอบเขตของเวลาการตอบสนองที่ได้กำหนดไว้และใช้กำลังงานได้อย่างเหมาะสม Qu และคณะ [20] ทดสอบการลดพลังงานที่ใช้ในระบบการสื่อสารและมีข้อจำกัดของเวลา โดยสร้างแบบจำลองของระบบสื่อสารให้ทำงานในลักษณะของการเก็บและส่งต่อไปตามลำดับ และพัฒนาขั้นตอนวิธีสำหรับการคำนวณหาค่า k โดยที่ k คือค่า

กำลังงานที่เหมาะสมในการแบ่งแพ็คเก็ต (packet) ออกเป็นส่วนย่อย ๆ วิธีการนี้ต้องผ่านกระบวนการ Myrinet pipeline 4 ขั้นตอนด้วยกัน ผลที่ได้สามารถลดกำลังงานที่ใช้ได้ถึง 27 – 93% ขึ้นอยู่กับขั้นตอนของการทำ pipeline Gruenwald และ Banik [21] ศึกษาฐานข้อมูลเชิงเวลาจริงบน mobile host และนำเสนอแบบจำลองที่ใช้ในการจัดการงานเพื่อลดงานที่ไม่สามารถทำได้ทันตามเวลาที่กำหนดไว้โดยที่การใช้พลังงานยังอยู่ในสภาวะที่สมดุล Unsal และคณะ [22] ทดสอบการสำรองข้อมูลบนระบบเชิงเวลาจริงแบบกระจาย (distributed real-time system) ที่ตระหนักถึงกำลังงานที่ใช้ ศึกษาโครงสร้างของเครือข่ายและการ broadcast กับกำลังงานที่ใช้ในระบบเชิงเวลาจริง โดยทดสอบการ multicast ของระบบที่มีโครงสร้างแบบ Steiner tree heuristic Lee และคณะ [23] สร้างแบบจำลองที่มีพื้นฐานการทำงานแบบพีชคณิตบนระบบเชิงเวลาจริงที่ตระหนักถึงกำลังงานที่ใช้ โดยแบบจำลองดังกล่าวตั้งอยู่บนสมมติฐานที่ระบบยังสามารถทำงานได้ดีแม้ว่ามีข้อผิดพลาดเกิดขึ้น โดยทดสอบกับโพรโตคอลที่ทำงานบนเครือข่ายแบบไร้รูปแบบ (ad-hoc)

Qiao และคณะ [24] ได้ดัดแปลงการทำงานของ IEEE 802.11a wireless LAN ซึ่งเป็นมาตรฐานหนึ่งที่ใช้ในระบบเครือข่าย โดยนำเทคนิคการควบคุมการใช้กำลังงานมาใช้ในการออกแบบร่วมกับการออกแบบทางด้านกายภาพเพื่อให้ระบบตระหนักถึงกำลังงานที่ใช้ได้ดียิ่งขึ้น Lahiri และคณะ [25] กล่าวถึงการพัฒนาประสิทธิภาพของแบตเตอรี่ใน Media Access Control (MAC) ของ 802.11 เช่น การรวมชั้นเครือข่ายกับชั้นสถาปัตยกรรมเข้าด้วยกันโดยนำสถาปัตยกรรมเดิมมาประยุกต์ใช้กับโครงสร้างของ MAC และวิเคราะห์โพรไฟล์เพื่อตรวจสอบส่วนที่มีการใช้แบตเตอรี่อย่างไม่มีประสิทธิภาพ ระบุจุดที่ทำให้เกิดคอขวด (bottleneck) ในการประมวลผลและปรับปรุงโพรโตคอลที่ทำงานบน MAC processor bus เพื่อลดจุดที่ทำให้เกิดคอขวด Lahiri และคณะได้คิดค้นหาแนวทางเพิ่มเติมที่จะทำให้ประสิทธิภาพการใช้แบตเตอรี่ตรงตามระดับที่ตั้งใจโดยนำเทคนิคการตระหนักถึงกำลังงานที่ใช้ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์มาใช้ในการออกแบบร่วมกัน ทดสอบกับแอพลิเคชันทั่ว ๆ ไปที่ทำงานโดยคำนึงถึง QoS และแอพลิเคชันที่ไม่ได้ทำงานเชิงเวลาจริง จากการทดสอบดังกล่าวสามารถลดปริมาณกำลังงานที่ใช้ลงได้

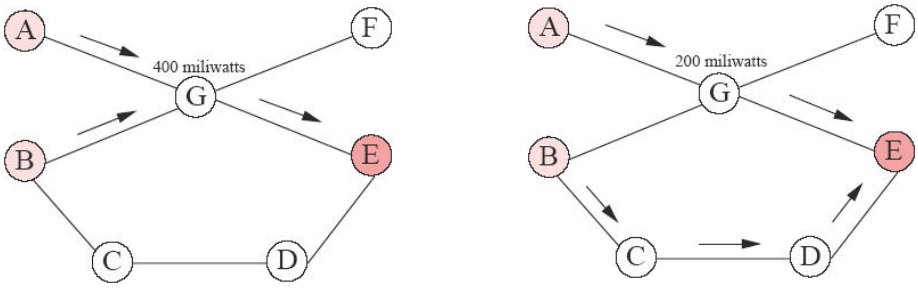
Havinga และ Smit [26] พิจารณากำลังงานที่ใช้ในระดับ MAC เช่นกันแต่ทดสอบบนคอมพิวเตอร์มือถือและโพรโตคอลของเครือข่ายที่ตระหนักถึงกำลังงานที่ใช้ โดยออกแบบให้ network interface เข้าสู่สถานะ standby นานขึ้นจะช่วยลดปริมาณพลังงานที่ใช้ได้ จากทีมวิจัยเดียวกันกับข้างต้น [27] ได้ศึกษาองค์ประกอบของสถาปัตยกรรม ATM-like switching และหากลวิธีการจัดการบัฟเฟอร์ที่เหมาะสมบนแอพลิเคชันที่ทำงานแบบมัลติมีเดีย องค์ประกอบของสถาปัตยกรรมที่กล่าวถึง เช่น clock gate นั่นคือออกแบบให้การสวิชชิ่ง (switching) ใช้กำลัง

งานอย่างเหมาะสมที่สุด Gomez และคณะ [28] ทดสอบการหาเส้นทางกับกำลังงานที่ใช้ที่เหมาะสมในเครือข่ายไร้สายด้วยการปรับเปลี่ยนแผนการส่งแพ็คเก็ต โดยให้โหนดที่อยู่ในระหว่างเส้นทางการส่งข้อมูล (intermediate node) เป็นตัวแทนในการเปลี่ยนเส้นทาง (redirecting) และหาเส้นทางใหม่ที่เหมาะสมที่สุดเพื่อลดปริมาณการใช้พลังงานของเครือข่าย

แอพลิเคชันที่ใช้ในทางทหารส่วนมากทำงานอยู่บนเครือข่าย wireless sensor ซึ่งมีความสามารถในการทำงานต่ำ bandwidth ต่ำ แบตเตอรี่ที่ใช้ไม่สามารถเปลี่ยนใหม่แทนที่ได้ และสามารถส่งแพ็คเก็ตได้ในระยะทางสั้น ๆ เท่านั้น สามารถทนต่อความผิดพลาดได้สูงและปรับเปลี่ยนค่าได้ (dynamic configurability) จากลักษณะดังกล่าวระบบนี้เป็นระบบที่ต้องการทำงานบนระบบที่ตระหนักถึงกำลังงานที่ใช้ ด้วยเหตุนี้จึงนำไปสู่การวิจัยของระบบดังกล่าวที่เกี่ยวกับการสื่อสารบนชั้นกายภาพ MAC และชั้นย่อย ๆ ของเครือข่าย (network sublayer) การหาเส้นทาง และการส่งต่อข้อมูล ในที่นี้กล่าวสรุปถึงการตระหนักถึงกำลังงานที่ใช้ใน network sensor Rabaey และคณะ [29] ได้พัฒนาแบบจำลองที่ใช้สำหรับการส่งต่อแพ็คเก็ตโดยพิจารณากรณีที่มีการสูญหายของเส้นทาง (path-loss term) ร่วมด้วย จากการทดสอบพบว่าการส่งข้อมูลด้วย hop สั้น ๆ หลาย ๆ hop ทำให้การใช้พลังงานมีประสิทธิภาพมากกว่าการส่งด้วย hop ยาว ๆ เพียง hop เดียว สำหรับนักวิจัยอื่น ๆ ได้มุ่งประเด็นที่การทำ clustering ในระหว่างทางของการส่งข้อมูล ซึ่งการส่งข้อมูลไปยังแต่ละ sensor node อาจส่งเป็น cluster หลาย ๆ cluster และสามารถปรับเปลี่ยนค่าคอนฟิก (configure) ของ sensor network ได้แบบไดนามิค สรุปได้ว่าการทำ dynamic cluster จึงเป็นวิธีการที่เหมาะสมสำหรับ sensor network ที่ตระหนักถึงกำลังงานที่ใช้ Hienzelman และคณะ [30] ได้นำเสนอวิธีการทำ dynamic clustering ด้วยวิธี Low-Energy Adaptive Clustering Hierarchy (LEACH) เป็นวิธีการสุ่มสถานีที่มีหน้าที่ในการทำ cluster (สามารถใช้ได้ผลกับการติดต่อสื่อสารที่มีการทำ cluster ร่วมกัน) เพื่อให้มีการกระจายโหลดของพลังงานที่เกิดขึ้นในแต่ละโหนดให้เท่า ๆ กัน สำหรับโพรโตคอลที่ใช้ในการหาเส้นทางมีหน้าที่ในการรวมข้อมูลเข้าด้วยกันเพื่อลดปริมาณข้อมูลที่ส่งไปยัง sensor node ที่ base station เพราะสามารถลดปริมาณการใช้พลังงานได้ การออกแบบ sensor network ในชั้น MAC มีเป้าหมายที่แตกต่างจาก wireless MAC ทั่ว ๆ ไป (IEEE 802.11) คือ การให้ความสำคัญกับการการกำหนดค่าต่าง ๆ การอนุรักษ์พลังงาน และความยุติธรรมในการทำงานของโหนดแต่ละโหนดมากกว่าปัจจัยแฝงอื่น ๆ Ye และคณะ [31] ได้พิจารณาการออกแบบ MAC protocol โดยพัฒนาประสิทธิภาพการใช้พลังงานในส่วนของ S-MAC พบว่าการกำหนดให้โหนดที่ไม่มีการทำงานเข้าสู่สถานะหลับช่วยลดการใช้พลังงานได้ ส่วนโหนดใกล้เคียงที่มีการส่งข้อมูลจะมีตารางเวลาในการเข้าสู่สถานะหลับด้วยเช่นกัน จากผลที่ได้พบว่า S-MAC ใช้พลังงานน้อยกว่า MAC ใน IEEE

802.11 2 ถึง 6 เท่า Tsiatsis และคณะ [32] ศึกษาประสิทธิภาพการใช้พลังงานสำหรับการส่งแพ็คเก็ตใน wireless sensor network ของแต่ละ sensor node ซึ่งประกอบด้วยบล็อกจำนวน 2 บล็อกด้วยกัน คือ sensor node CPU และ radio board และเสนอว่าการให้ radio board ทำหน้าที่ในการตอบสนองแทน sensor node CPU ที่อยู่ในระดับชั้นเครือข่ายสามารถประหยัดพลังงานที่ใช้ได้ เนื่องจากเกิดการประมวลผลที่ radio board โดยไม่ต้องย้ายแพ็คเก็ตไปประมวลผลที่ sensor node CPU

การลดกำลังงานที่ใช้ให้มากที่สุดเป็นสิ่งสำคัญมากในระบบที่มีข้อจำกัดด้านอุณหภูมิ ตัวอย่างเช่น กรณีการตั้งค่าเราเตอร์ที่ต้องการลดปริมาณกำลังงานที่ใช้ให้สูงที่สุดสำหรับการหาเส้นทางเพื่อส่งข้อมูลบนระบบเชิงเวลาจริงกับการตั้งค่าเราเตอร์ทั่วไปที่พิจารณาเพียงเส้นทางที่สั้นที่สุดเท่านั้นโดยไม่คำนึงถึงกำลังงานที่ใช้ โดยทั่วไปจะได้ประสิทธิภาพการทำงานสูงสุดเนื่องจากได้เส้นทางที่สั้นที่สุด และสามารถลดค่าเฉลี่ยของเวลาที่ใช้ในการส่งข้อมูลจากต้นทางไปยังปลายทางได้ แต่วิธีการนี้อาจทำให้จุด ๆ หนึ่งมีการทำงานบ่อยมากส่งผลต่อสภาพความแออัดและเกิดการใช้กำลังงานในจุด ๆ นั้นสูงขึ้น การกำหนดค่าเราเตอร์ที่ตระหนักถึงกำลังงานที่ใช้เป็นการลดขีดสูงสุดของกำลังงานที่ใช้ในจุดใดจุดหนึ่งด้วยการกำหนดจุดที่มีการใช้งานบ่อย ๆ (hotspot) และหลีกเลี่ยงความแออัดที่เกิดขึ้นกับจุดนั้น ๆ วิธีการนี้อาจส่งผลให้พลังงานที่สูญเสียโดยรวมมีค่าเพิ่มมากขึ้น สมมติให้ระบบหนึ่งประกอบด้วยหน่วยประมวลผล 7 หน่วย ดังแสดงในภาพประกอบ 1-3 โหนด A และโหนด B ส่งข้อมูลไปยังโหนด E พร้อม ๆ กันและให้ข้อความที่ส่งไปยังโหนด E ต้องใช้เวลาไม่เกิน 20 ไมโครวินาที (μsecond) ใช้เวลาในการส่ง 5 μsecond ในแต่ละ hop การทำงานของเราเตอร์ในแต่ละโหนดใช้กำลังงาน 200 มิลลิวัตต์ (milliwatt) จากรูป 1.3a พบว่ากำลังงานสูงสุดที่สูญเสียไปในระบบที่ไม่คำนึงถึงกำลังงานที่ใช้ จะเลือกเส้นทางที่สั้นที่สุดแต่สูญเสียกำลังงานถึง 400 มิลลิวัตต์ที่โหนด G โดยที่ข้อความถึงโหนด E ในเวลา 10 μsecond สำหรับระบบที่ตระหนักถึงกำลังงานที่ใช้จะใช้เส้นทางต่างจากแบบแรกในการส่งข้อความจากโหนด B ไปยังโหนด E ดังแสดงในรูปที่ 1.3b โหนด E ได้รับข้อความสุดท้ายในเวลา 15 μsecond ซึ่งใช้เวลานานกว่าแบบแรกแต่ยังส่งข้อมูลได้ทันตามระยะเวลาที่กำหนดโดยสามารถลดขีดสูงสุดของการสูญเสียกำลังงานที่ใช้ได้ถึง 200 มิลลิวัตต์ นั่นคือประสิทธิภาพในการลดการสูญเสียกำลังงานดีขึ้นถึง 100% วิธีการนี้เป็นวิธีการสุ่มหาเส้นทางที่สั้นที่สุดจากจุดที่อยู่รอบ ๆ จุดที่มีการใช้งานสูงสุด จากตัวอย่างที่กล่าวมาข้างต้นสรุปได้ว่าการตระหนักถึงโครงสร้างเครือข่ายเป็นสิ่งหนึ่งที่สำคัญมากในการกำหนดเส้นทางการส่งข้อมูลของเราเตอร์ขึ้นอยู่กับความต้องการของระบบว่าต้องการลดปริมาณพลังงานที่ใช้หรือประสิทธิภาพในการส่งข้อมูล



(a) Non-power-aware routing (b) Power-aware routing

ภาพประกอบ 1-3 การหาเส้นทางโดยตระหนักถึงกำลังงานที่ใช้กับการหาเส้นทางที่สั้นที่สุด

1.3 วัตถุประสงค์

1.3.1 ศึกษาระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

1.3.2 ศึกษาและวิเคราะห์ปัจจัยที่มีผลต่อการใช้กำลังงานเกินความจำเป็นของระบบควบคุมเชิงเวลาจริงแบบฝังตัวภายใต้ระบบปฏิบัติการ uC/OS-II

1.3.3 ปรับปรุงระบบปฏิบัติการ uC/OS-II ให้จัดสรรทรัพยากรเพื่อให้สามารถใช้กำลังไฟฟ้าเท่าที่จำเป็น

1.4 ขอบเขตการวิจัย

1.4.1 ศึกษาและวิเคราะห์การทำงานของระบบปฏิบัติการ uC/OS-II

1.4.2 ศึกษาแนวทางในการตรวจวัดกำลังไฟฟ้าของงานภายใต้ระบบปฏิบัติการ uC/OS-II

1.4.3 ปรับปรุงระบบปฏิบัติการ uC/OS-II ให้สามารถควบคุมการใช้กำลังงานได้อย่างเหมาะสมโดยปรับตารางการทำงานให้มีการใช้กำลังงานอย่างมีประสิทธิภาพมากที่สุด

1.5 ขั้นตอนและวิธีการวิจัย

1.5.1 ศึกษาและค้นคว้ารายละเอียดที่เกี่ยวข้องกับโครงการวิจัย

1.5.1.1 ศึกษาการทำงานของระบบปฏิบัติการเชิงเวลาจริง

1.5.1.2 ศึกษาและวัดการใช้กำลังไฟฟ้าของงานภายใต้ระบบปฏิบัติการเชิงเวลาจริง

1.5.2 ทำการทดลองวัดกำลังไฟฟ้าของงานจากแอพลิเคชันจริง ๆ

1.5.3 แก้ไข/ปรับปรุงระบบปฏิบัติการเชิงเวลาจริง

1.5.4 ทดสอบกับแอพลิเคชันจริง

1.5.5 ปรับปรุงระบบปฏิบัติการเชิงเวลาจริง uC/OS-II ให้มีประสิทธิภาพมากยิ่งขึ้น

1.5.6 จัดทำเอกสารประกอบการปรับปรุงและพัฒนาระบบคอมพิวเตอร์ฝังตัวที่ตระหนักถึงกำลังงานที่ใช้บนระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 เรียนรู้ระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

1.6.2 ทำให้เกิดความเข้าใจถึงการใช้กำลังไฟฟ้าของงานในแอพลิเคชันจริง ๆ

1.6.3 ได้ระบบปฏิบัติการเชิงเวลาจริง uC/OS-II ที่ตระหนักถึงกำลังไฟฟ้าและจัดสรรกำลังงานให้เหมาะสมกับงานในแต่ละงานโดยทำงานบนระบบคอมพิวเตอร์ฝังตัวที่เป็นระบบเชิงเวลาจริง

1.6.4 สามารถถ่ายทอดผลที่ได้รับจากการวิจัยสู่การประยุกต์ใช้งานในระบบคอมพิวเตอร์ฝังตัว ซึ่งใช้พลังงานไฟฟ้าเท่าที่จำเป็น

1.7 ระยะเวลาที่ใช้ในการวิจัย

พฤษภาคม 2548 – พฤษภาคม 2549

1.8 แผนการดำเนินการตลอดโครงการ

กิจกรรมขั้นตอนการดำเนินงาน	2548								2549				
	5	6	7	8	9	10	11	12	1	2	3	4	5
1. ศึกษาการทำงานของระบบปฏิบัติการเชิงเวลาจริง	←	→											
2. ศึกษาและวัดกำลังไฟฟ้าของงานภายใต้ระบบปฏิบัติการเชิงเวลาจริง		←	→										
3. ทำการทดลองวัดกำลังไฟฟ้าของงานจากแอพลิเคชันจริง ๆ				↔									
4. แก้ไข/ปรับปรุงระบบปฏิบัติการเชิงเวลาจริง					←		→						
5. ทดสอบกับแอพลิเคชันจริง								←		→			
6. ปรับปรุง uC/OS-II ให้มีประสิทธิภาพในการใช้กำลังงานมากยิ่งขึ้น										←		→	
7. จัดทำเอกสารประกอบการปรับปรุงและพัฒนาระบบ						←							→