บทที่ 3

ข้อพิจารณาในการออกแบบ

หลังจากได้ทำการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับเครื่องจัดการจราจรในระบบเครือข่าย ทำให้ได้พบเห็นการออกแบบระบบในรูปแบบต่าง ๆ จึงได้นำข้อมูลดังกล่าวมาเป็นแนวทางในการออกแบบเครื่องจัดการจราจรในระบบเครือข่าย โดยมีวัตถุประสงค์สำคัญในการออกแบบคือการสร้างฮาร์ดแวร์ที่มีความสามารถในการจัดคิวแพ็คเกตและสามารถลดการเคลื่อนย้ายข้อมูลที่เกิดจากการเขียนและลบแพ็คเกตออกจากคิว ซึ่งกระบวนการสำคัญในการออกแบบที่จะกล่าวถึงในบทนี้คือ การจำแนกประเภทของงาน รูปแบบของสถาปัตยกรรมในการจัดคิว การเลือกวิธีการอินเตอร์เฟสกับระบบเครือข่าย การตรวจสอบความถูกต้องของข้อมูล (Error Checking) และการเลือกใช้อุปกรณ์ ส่วนรายละเอียดในการออกแบบวงจรจะกล่าวถึงในบทต่อไป

3.1 การจำแนกประเภทของงาน

ฟังก์ชันการทำงานต่าง ๆ ที่พบในเครื่องจัดการจราจรในระบบเครือข่ายประกอบด้วย

1) การจำแนกแพ็คเกต เป็นการระบุชนิดของแพ็คเกต โดยพิจารณาจากส่วนเฮดเดอร์ (Header) ของแพ็คเกตก่อนจะนำไปประมวลผลตามชนิดของแพ็คเกตต่อไป
2) การกรองแพ็คเกต (Filtering) เป็นการเลือกที่จะรับหรือทิ้งแพ็คเกตที่ได้รับเข้ามาตามที่ผู้ใช้กำหนด
3) การจัดคิวแพ็คเกต
4) การส่งต่อแพ็คเกต เป็นการกำหนดพอร์ตปลายทางสำหรับแพ็คเกตที่ได้รับเข้ามา จากนั้นจึงส่งแพ็คเกตไปที่พอร์ตดังกล่าว
5) การทำสัญลักษณ์ (Marking) เป็นการเขียนค่าลงไปในฟิลด์ Type of Service : ToS ของไอพีแพ็คเกตหลังจากผ่านการจำแนกแพ็คเกตแล้วเพื่อให้ง่ายต่อการประมวลผลต่อไป
6) การจัดการกับแพ็คเกตตามข้อกำหนดของผู้ใช้ (Policing)
7) การให้บริการที่เน้นคุณภาพการให้บริการ
8) การเก็บข้อมูลทางสถิติ เป็นการบันทึกข้อมูลทางสถิติของการรับส่งแพ็คเกต

9) การปรับแต่งการจราจรในเครือข่าย (Traffic Shaping) เป็นการควบคุมการไหลของข้อมูลให้เป็นไปตามที่ผู้ใช้ระบุ

10) การหาเส้นทางให้แพ็คเกตโดยพิจารณาจากชนิดของการให้บริการ (ToS – Based Routing) ซึ่งเป็นฟิลด์ขนาด 8 บิตในส่วนเฮดเดอร์ของไอพีแพ็คเกต

ฟังก์ชันการทำงานที่กล่าวมาข้างต้นมีความซับซ้อนแตกต่างกันขึ้นอยู่กับอัลกอริทึมที่ใช้ สำหรับวิทยานิพนธ์นี้เครื่องจัดการจราจรในระบบเครือข่ายถูกออกแบบให้รองรับฟังก์ชันต่อไปนี้คือ การจำแนกแพ็คเกต การกรองแพ็คเกต การส่งต่อแพ็คเกตและการจัดคิวแพ็คเกต ซึ่งฟังก์ชันดังกล่าวจะเกี่ยวข้องกับการประมวลผลและการเคลื่อนย้ายข้อมูล โดยเฉพาะการจัดคิวแพ็คเกตซึ่งจะเกี่ยวข้องกับการเคลื่อนย้ายข้อมูลเป็นอย่างมาก เมื่อพิจารณาฟังก์ชันการทำงานเหล่านี้พบว่าสามารถจำแนกตามความซับซ้อนของงานได้เป็น 2 ประเภท ได้แก่

3.1.1 **งานที่ทำงานที่ความเร็วสูง** งานประเภทนี้มักจะมีความซับซ้อนต่ำ และใช้ฮาร์ดแวร์ที่ออกแบบมาโดยเฉพาะมาดำเนินการ เพื่อช่วยลดภาระงานของไมโครโปรเซสเซอร์ทำให้ไมโครโปรเซสเซอร์ไม่ต้องรับภาระในการประมวลผลมากเกินไป

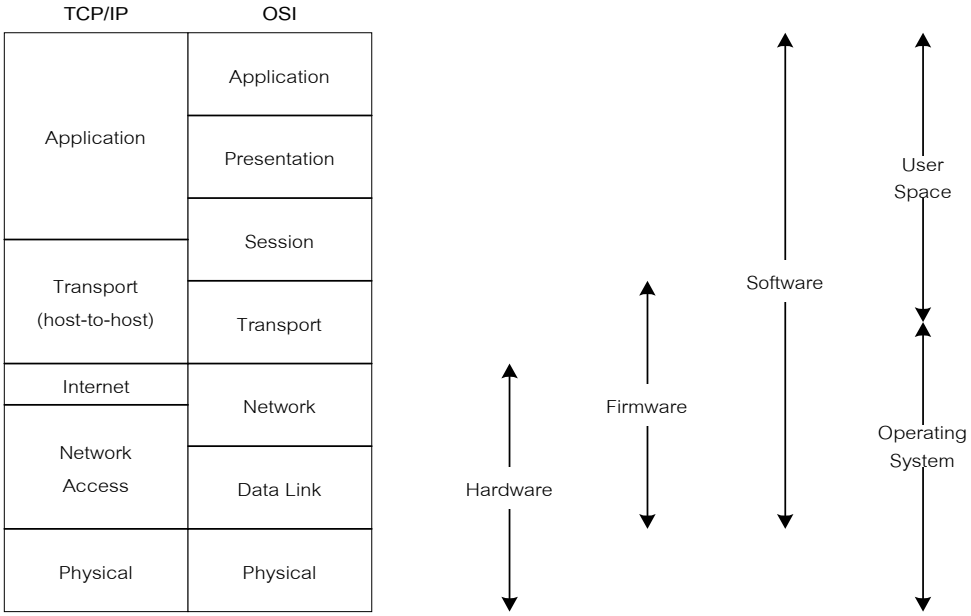
3.1.2 **งานที่ทำงานที่ความเร็วต่ำ** งานประเภทนี้มักจะมีความซับซ้อนสูง ต้องอาศัยไมโครโปรเซสเซอร์ในการประมวลผล

3.2 การเลือกสถาปัตยกรรมในการจัดคิว

การจัดคิวเป็นวิธีการหนึ่งในการควบคุมการไหลของการจราจรในระบบเครือข่าย โดยเฉพาะจากระบบเครือข่ายความเร็วสูงไปยังระบบเครือข่ายที่มีความเร็วต่ำกว่าหรือในระบบเครือข่ายที่มีแบนด์วิดท์จำกัด การจัดคิวในเครื่องจัดการจราจรในระบบเครือข่ายสามารถทำได้หลายวิธี ได้แก่ ซอฟท์แวร์ ฮาร์ดแวร์ หรือ เฟิร์มแวร์ (Firmware) ขึ้นอยู่กับการเลือกใช้ ดังภาพประกอบ 3-1 ซึ่งแสดงให้เห็นว่าฟังก์ชันการทำงานในเลเยอร์ต่าง ๆ นั้นสามารถทำได้โดยใช้ซอฟท์แวร์ ฮาร์ดแวร์ หรือ เฟิร์มแวร์ ในทางปฏิบัตินั้นการจัดคิวส่วนใหญ่จะใช้ซอฟท์แวร์

กระบวนการสำคัญในการจัดคิวโดยใช้ซอฟท์แวร์นั้นประกอบด้วยการประมวลผลตามอัลกอริทึมที่กำหนด การร้องขอการเข้าถึงหน่วยความจำ การเคลื่อนย้ายแพ็คเกตจากหน่วยความจำมาไว้ในคิว การลบแพ็คเกตออกจากคิวและส่งต่อ ซึ่งกระบวนการเหล่านี้จะมีไมโครโปรเซสเซอร์ทำหน้าที่ควบคุมการทำงานและประมวลผล

ข้อเสียของการจัดคิวโดยใช้ซอฟท์แวร์อยู่ที่ไมโครโปรเซสเซอร์ต้องเสียเวลาอย่างมากในการเคลื่อนย้ายข้อมูลซึ่งเกิดจากการเขียนและลบแพ็คเกตออกจากคิว



ภาพประกอบ 3-1 สถาปัตยกรรมโปรโตคอล

(ที่มา : William Stallings. 1997. “Data and Computer Communications”.)

การทดสอบเพื่อเปรียบเทียบเวลาในการเคลื่อนย้ายข้อมูลกับเวลาในการทำงานโดยรวมของไมโครโปรเซสเซอร์ทำโดยการใช้เครื่องคอมพิวเตอร์ที่ใช้ซีพียูเพนเทียมทูความเร็ว 300 เมกกะเฮิร์ท อ่านแพ็คเกตขนาด 64 ไบต์มาเก็บไว้ในหน่วยความจำ ประมวลผลเพื่อจัดคิวแพ็คเกต เขียนและลบแพ็คเกตออกจากคิว ซึ่งการจัดคิวที่ใช้เป็นการจัดคิวอย่างง่าย กล่าวคือ มีคิวทั้งหมด 8 คิว ซึ่งคิวที่ 0 มีลำดับความสำคัญสูงที่สุด โดยกำหนดให้แพ็คเกตที่ได้รับเข้าถูกจัดเข้าในคิวที่ 0 ทั้งหมด ดังนั้นแพ็คเกตที่ได้เข้ามาจะถูกส่งออกทันทีซึ่งหมายความว่าเวลาที่วัดได้เป็นค่าที่ดีที่สุดสำหรับการจัดคิวด้วยวิธีนี้ จากนั้นทำการทดสอบในทำนองเดียวกันแต่เปลี่ยนขนาดแพ็คเกตเป็น 1500 ไบต์ แล้วบันทึกเวลาที่ใช้ ผลการทดสอบแสดงดังตาราง 3-1

ตาราง 3-1 เวลาในการเคลื่อนย้ายข้อมูลขนาด 64 และ 1,500 ไบต์เทียบกับเวลาโดยรวม

ขนาดแพ็คเกต (ไบต์)	เวลาในการเคลื่อนย้ายข้อมูล (ไมโครวินาที)	เวลาในการทำงานโดยรวม (ไมโครวินาที)
64	187.5	190
1,500	4,300	4,305

จากค่าที่วัดได้พบว่าเวลาในการเคลื่อนย้ายข้อมูลคิดเป็น 98.68% (มาจาก (187.5/190) x 100) ของเวลาโดยรวมทั้งหมดสำหรับแพ็คเกตขนาด 64 ไบต์และ 99.88% (มาจาก (4,300/4,305) x 100) สำหรับแพ็คเกตขนาด 1,500 ไบต์ ซึ่งถ้าใช้การจัดคิวที่ซับซ้อนมากขึ้นจะทำให้เวลาโดยรวมของการเคลื่อนย้ายข้อมูลและประมวลผลมีค่าเพิ่มขึ้น แม้ว่าระยะเวลาที่วัดได้จะขึ้นอยู่กับความเร็วในการประมวลผลของไมโครโปรเซสเซอร์ที่เลือกใช้ แต่อย่างไรก็ตามไม่ว่าจะเลือกใช้ไมโครโปรเซสเซอร์ใด ๆ เวลาที่เสียไปกับการเคลื่อนย้ายข้อมูลก็ยังคงมีค่าสูงมากเมื่อเทียบกับเวลาทั้งหมดที่ใช้ในการจัดคิวนอกเหนือจากการประมวลผลเพื่อตัดสินใจว่าจะเลือกแพ็คเกตใดที่จะจัดส่งต่อไป ซึ่งนับว่าเป็นภาระที่หนักสำหรับไมโครโปรเซสเซอร์ ทำให้ไมโครโปรเซสเซอร์ที่ใช้ต้องมีประสิทธิภาพสูงมากเพื่อที่จะมีขีดความสามารถเพียงพอที่จะจัดคิวแพ็คเกตสำหรับเครือข่ายอินเตอร์เน็ตความเร็วสูงได้ทัน ด้วยเหตุนี้จึงได้เลือกใช้ฮาร์ดแวร์เข้ามาช่วยในการจัดคิวแพ็คเกตซึ่งจะช่วยลดการเคลื่อนย้ายข้อมูลที่เกิดขึ้นโดยอาศัยหลักการเก็บพอยน์เตอร์ที่ชี้ไปยังตำแหน่งของแพ็คเกตไว้ภายในคิวแทนการเก็บแพ็คเกตไว้ในคิว การใช้สถาปัตยกรรมในลักษณะนี้จะช่วยลดภาระของไมโครโปรเซสเซอร์ลง ทำให้หน้าที่หลัก ๆ ของไมโครโปรเซสเซอร์คือการประมวลผลเพื่อจัดคิวแพ็คเกตและส่งสัญญาณเพื่อไปควบคุมฮาร์ดแวร์ที่ได้ออกแบบไว้ ซึ่งจะส่งผลให้ไมโครโปรเซสเซอร์ที่ใช้ไม่จำเป็นต้องมีประสิทธิภาพสูงมากนัก

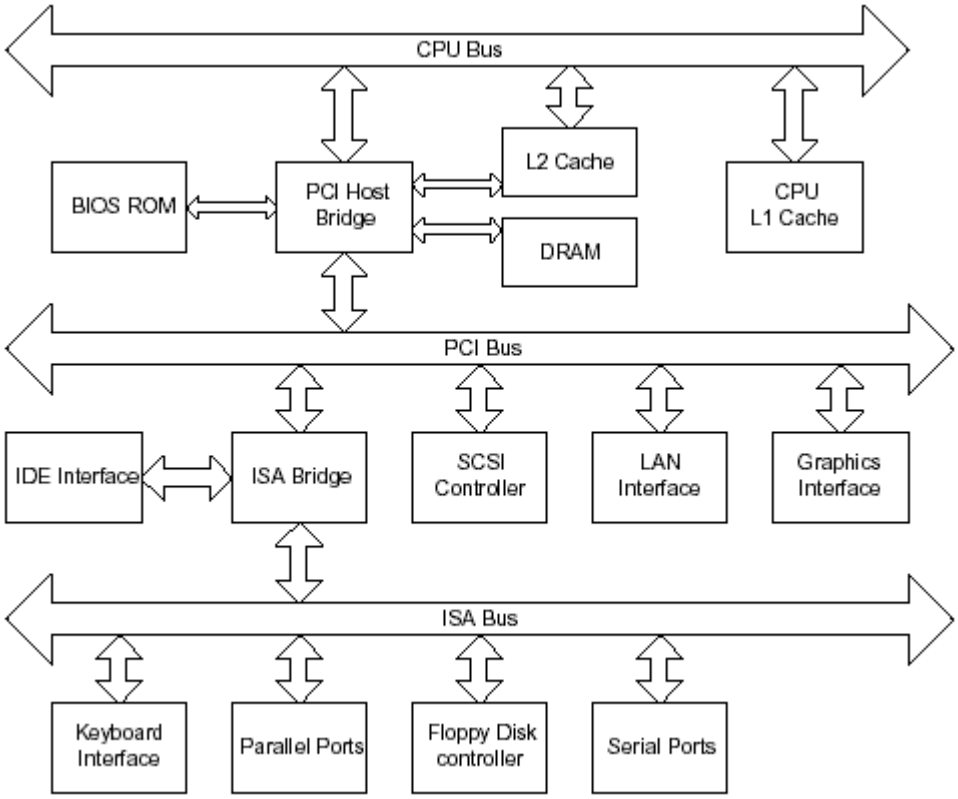
3.3 การอินเตอร์เฟสกับระบบเครือข่าย

ปัจจุบันแนวโน้มของความเร็วในการรับส่งข้อมูลของเครือข่ายอีเทอร์เน็ตเพิ่มสูงขึ้นเรื่อย ๆ จากเดิมซึ่งใช้งานอยู่ที่ระดับความเร็ว 10 เมกกะบิตต่อวินาที พัฒนามาสู่ความเร็ว 100 เมกกะบิตต่อวินาทีและระดับกิกะบิตในขณะนี้ ในวิทยานิพนธ์นี้จะเน้นความสนใจที่เครือข่ายอีเทอร์เน็ตความเร็ว 100 เมกกะบิตต่อวินาทีซึ่งนิยมใช้งานกันอย่างแพร่หลายในปัจจุบัน รูปแบบการอินเตอร์เฟสกับระบบเครือข่ายในความเร็วระดับนี้ที่นิยมใช้ในปัจจุบันมีด้วยกัน 2 วิธีได้แก่

3.3.1 การอินเตอร์เฟสผ่านทาง PCI Bus

การอินเตอร์เฟสผ่านทาง PCI Bus นั้นแพ็คเกตจะถูกรับส่งผ่านทางการ์ดแลน โดยข้อมูลจะถูกส่งผ่าน PCI Bus และมี North Bridge หรือ Host Bridge ทำหน้าที่ในการติดต่อระหว่าง Local Bus ของซีพียู และ PCI Bus ดังแสดงในภาพประกอบ 3-2 นอกจากนี้บางระบบอาจมี South Bridge ทำหน้าที่ในการติดต่อระหว่าง PCI Bus กับระบบบัสแบบอื่น ๆ เช่น ISA Bus อุปกรณ์ที่มีความสำคัญในการรับส่งข้อมูลผ่านทาง PCI Bus คือชิพ Host Bridge ซึ่งโดยส่วนใหญ่แล้วมักจะพบเห็น Host Bridge ในลักษณะของชิพเซ็ต ตัวอย่างของชิพ Host Bridge เช่น ชิพ

82443BX Host Bridge/Controller [Intel 440BX AGPset: 82443BX HostBridge/Controller Datasheet. April 1998.] ซึ่งอยู่ภายในชิพเซ็ต 440BX AGPset ของบริษัทอินเทล



ภาพประกอบ 3-2 การเชื่อมต่อกับระบบเครือข่ายผ่านทาง PCI Bus

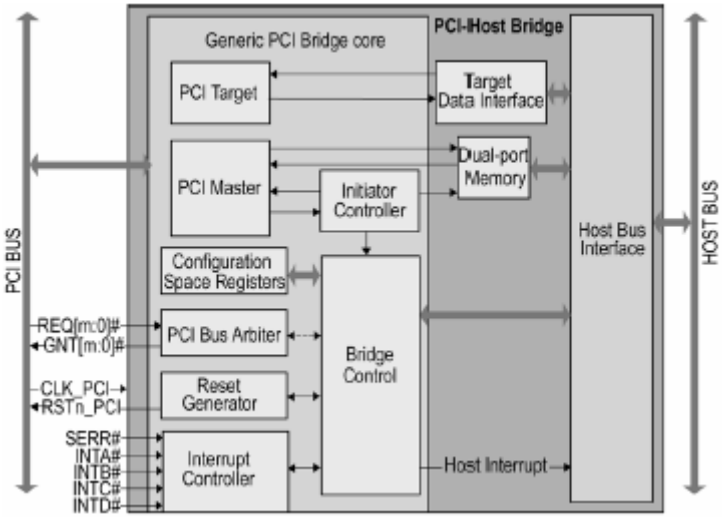
(ที่มา : Ehud Finkelstein. 1997. “Design and Implementation of PCI Bus Based Systems” PhD thesis,Tel Aviv University.)

ฟังก์ชันการทำงานของชิพเซ็ต Host Bridge ประกอบด้วย

- การอินเตอร์เฟสเข้ากับ Local Bus ของซีพียู
- การทำฟังก์ชัน PCI Initiator ให้กับซีพียู
- การทำฟังก์ชัน PCI Target ให้กับหน่วยความจำของซีพียู
- การทำหน้าที่ PCI Bus Arbiter
- การควบคุมการอินเตอร์รัพท์ PCI
- การควบคุมการทำงานของหน่วยความจำ

- การทำหน้าที่รีจิสเตอร์ควบคุม (Control Registers) เช่น การแม็ปแอดเดรสบนสำหรับซีพียูที่มีบัสแอดเดรสต่ำกว่า 32 บิต

ตัวอย่างบล็อกไดอะแกรมของ PCI Host Bridge แสดงดังภาพประกอบ 3-3



ภาพประกอบ 3-3 ตัวอย่างบล็อกไดอะแกรมของ PCI Host Bridge

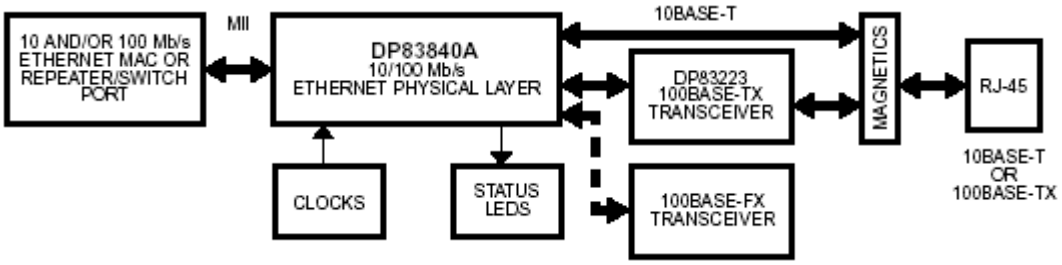
(ที่มา : CAST, Inc. “PCI-HB 32-bit/33,66 MHz PCI Host Bridge Core”. 2003.)

เนื่องจากเครื่องจัดการจราจรในระบบเครือข่ายที่ออกแบบนั้นต้องการใช้งานชิพเซ็ต Host Bridge เพียงบางฟังก์ชันเท่านั้นคือส่วนของการรับส่งข้อมูลระหว่าง Local Bus กับ PCI Bus ด้วยเหตุนี้จึงคิดที่จะตัดส่วนของชิพเซ็ต Host Bridge ออกและหาอุปกรณ์อื่นเพื่อมาทำหน้าที่นี้แทน ซึ่งการจะทำเช่นนั้นได้จะต้องทราบโครงสร้างและฟังก์ชันการทำงานโดยละเอียดของวงจรภายใน Host Bridge แต่ทางบริษัทผู้ผลิตชิพเซ็ต Host Bridge ไม่เปิดเผยข้อมูลในส่วนนี้เนื่องจากเหตุผลในเรื่องของลิขสิทธิ์ จึงทำให้ค่อนข้างยุ่งยากมากในการหาอุปกรณ์อื่นเพื่อมาใช้ทดแทนชิพเซ็ต Host Bridge รวมทั้งการ์ดแลนซึ่งติดต่อผ่าน PCI Bus ที่มีอยู่นั้นใช้ชิพ RTL8139 (Realtek Semiconductor Corp. Realtek 3.3 Single Chip Fast Ethernet Controller with Power Management RTL8139C(L).) ของบริษัท Realtek การอินเตอร์เฟสและเขียนโปรแกรมควบคุมชิพนี้ทำได้ค่อนข้างลำบากเพราะบริษัทผู้ผลิตให้ข้อมูลในการเขียนโปรแกรมน้อยมาก

ด้วยเหตุผลที่กล่าวมาข้างต้น ในวิทยานิพนธ์นี้จึงเลือกการติดต่อกับระบบเครือข่ายผ่านทาง MII แทนการติดต่อผ่านทาง PCI Bus

3.3.2 การอินเตอร์เฟสผ่านทาง MII

ในปัจจุบันชิพที่รองรับการติดต่อผ่าน MII ถูกผลิตขึ้นจากหลากหลายผู้ผลิต สำหรับในวิทยานิพนธ์นี้ชิพที่เลือกใช้คือชิพ DP83223 TWISTER High Speed Networking Transceiver Device และชิพ DP83840A 10/100 Mb/s Ethernet Physical Layer ของบริษัท National Semiconductor สาเหตุที่เลือกใช้ชิพทั้งสองเนื่องจากเป็นชิพที่รองรับการรับส่งข้อมูลผ่านทาง MII ตามมาตรฐาน IEEE802.3u และติดต่อรับส่งข้อมูลกับระบบเครือข่ายที่ความเร็ว 100 เมกกะบิตต่อวินาทีได้ ภาพรวมการรับส่งข้อมูลผ่าน MII โดยใช้ชิพทั้งสองแสดงดังภาพประกอบ 3-4 โดยสามารถอธิบายการทำงานได้ดังนี้คือ แพ็คเกตจะถูกส่งผ่านคอนเนกเตอร์ RJ45 จากนั้นชิพ DP83223 ทำหน้าที่ปรับแต่งสัญญาณและแปลงสัญญาณจากรูปแบบของ Multi–Level-Transmit / 3 levels (MLT-3) ไปเป็นแบบไบนารีแล้วส่งต่อให้ชิพ DP83840A โดยชิพ DP83840A จะทำการแปลงสัญญาณที่ได้รับเข้ามาให้กลายเป็นสัญญาณข้อมูลขนาด 4 บิตส่งผ่าน MII เพื่อนำไปจัดเก็บและประมวลผลต่อไป ในทำนองเดียวกันสำหรับการส่งแพ็คเกตนั้นแพ็คเกตที่ต้องการส่งจะถูกแปลงให้อยู่ในรูปสัญญาณ MII และส่งไปยังชิพ DP83840A และชิพ DP83223 ต่อไป โดยกระบวนการต่าง ๆ ภายในชิพทั้งสองจะเป็นไปในลักษณะที่ตรงข้ามการรับแพ็คเกต



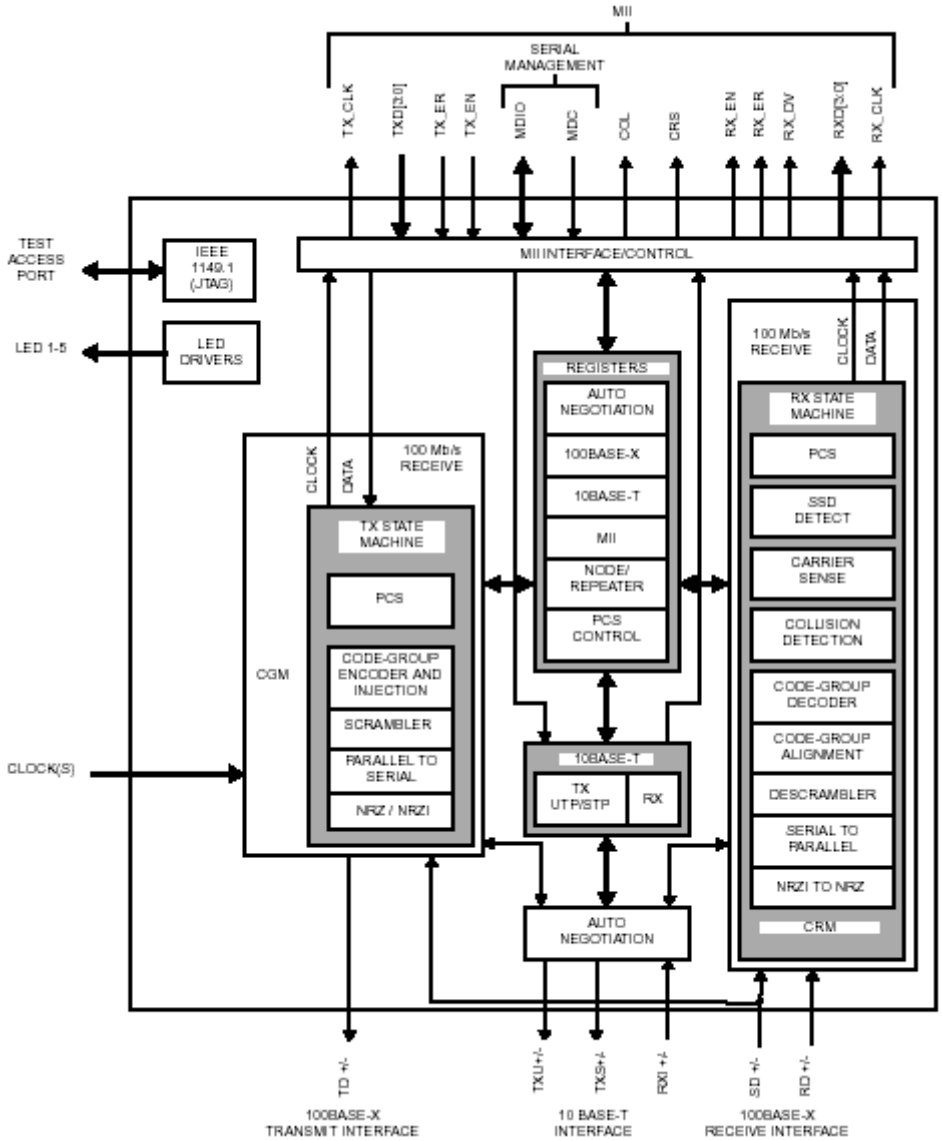
ภาพประกอบ 3-4 บล็อกไดอะแกรมของระบบรับส่งข้อมูลผ่าน MII

(ที่มา : National Semiconductor. “DP83840A 10/100 Mb/s Ethernet Physical Layer”.)

สำหรับบล็อกไดอะแกรมของชิพ DP83840A และชิพ DP83223 แสดงในภาพประกอบ 3-5 และ 3-6 ตามลำดับ โดยมีรายละเอียดการทำงานของชิพ DP83840A ดังนี้คือ สำหรับการทำงานของวงจรภาคส่งนั้น ชิพ DP83840A จะทำการแปลงสัญญาณ MII ขนาด 4 บิตไปเป็นสัญญาณอนุกรมที่ถูก Scrambled แล้วที่อัตราเร็ว 125 เมกกะบิตต่อวินาที ส่วนประกอบของวงจรภาคส่งได้แก่

- ตัวเข้ารหัส Code-Group และ Injection ทำหน้าที่แปลงสัญญาณ MII 4 บิตเป็นสัญญาณ Code-Group ขนาด 5 บิต

- Scrambler ใช้เพื่อควบคุมการแพร่คลื่นแม่เหล็กไฟฟ้าที่คอนเน็กเตอร์และบนสายตีเกลียวคู่
- ตัวเข้ารหัส NRZ ไปเป็น NRZI หลังจากที่ข้อมูลถูก Scrambled และแปลงเป็นสัญญาณอนุกรมแล้วนั้น ข้อมูลดังกล่าวจะต้องถูกเข้ารหัสแบบ NRZI เพื่อให้สอดคล้องกับมาตรฐาน TP-PMD สำหรับการส่งข้อมูล 100Base-TX บนสายตีเกลียวคู่ชนิด Category 5



ภาพประกอบ 3-5 บล็อกไดอะแกรมของชิพ DP83840A

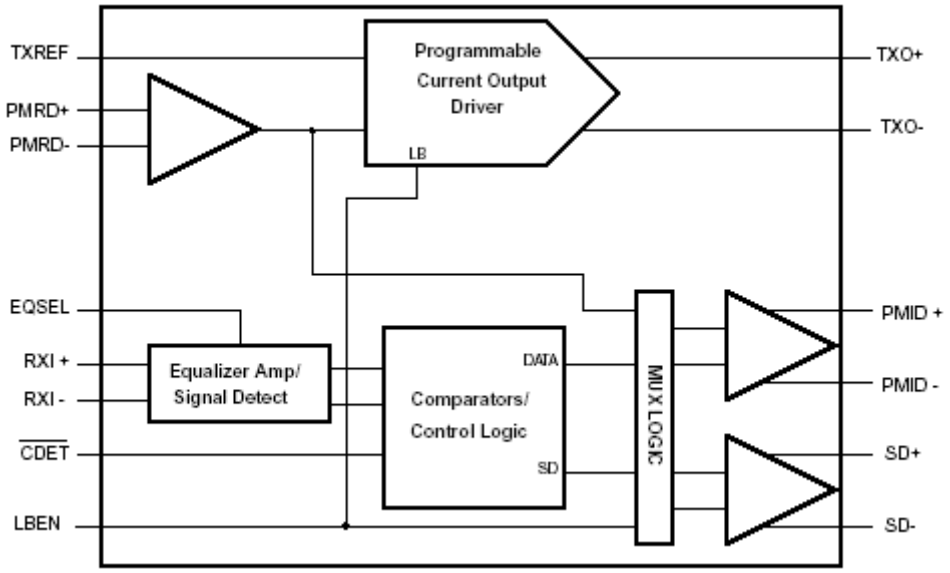
(ที่มา : National Semiconductor. “DP83840A 10/100 Mb/s Ethernet Physical Layer”.)

ส่วนการทำงานของวงจรภาครับของชิพ DP83840A นั้นจะเป็นการแปลงสัญญาณข้อมูลที่ถูก Scrambled ที่อัตราเร็ว 125 เมกกะบิตต่อวินาทีเป็นสัญญาณข้อมูล MII ขนาด 4 บิต โดยประกอบด้วยวงจรหลัก ๆ ดังนี้คือ

- Clock Recovery Module ทำการสร้างสัญญาณนาฬิกา 125 เมกกะเฮิร์ทจากสัญญาณข้อมูลที่ได้รับเข้ามา
- ตัวถอดรหัส NRZI เป็น NRZ เนื่องจากสัญญาณที่ได้รับเข้ามาถูกเข้ารหัสแบบ NRZI ดังนั้นต้องแปลงเป็น NRZ ก่อนจะนำไปประมวลผลต่อไป
- Descrambler ทำหน้าที่ De-Scrambling ข้อมูล NRZ ที่ได้รับเข้ามาให้กลับเป็นข้อมูลเดิมก่อนถูก Scrambled
- Code-Group Alignment
- ตัวถอดรหัส 5B/4B Code-Group ทำการแปลงข้อมูล Code-Group ขนาด 5 บิตไปเป็นข้อมูล MII ขนาด 4 บิต
- วงจรตรวจหาการชนกันของข้อมูล (Collision Detect)
- Carrier Sense
- 100 Mb/s Receive State Machine
- Far End Fault Indication
- Link Integrity Monitor
- Carrier Integrity Monitor

ส่วนการทำงานของชิพ DP83223 สามารถอธิบายได้ดังนี้คือ วงจรภาคส่งจะประกอบด้วยวงจรบัฟเฟอร์ที่มีอินพุตแบบ ECL คือขา PMRD+/- และวงจรไดรเวอร์ขับกระแสที่สามารถโปรแกรมกระแสเอาท์พุตได้ ซึ่งวงจรไดรเวอร์ขับกระแสจะทำหน้าที่แปลงสัญญาณไบนารีที่ได้รับเข้ามาให้กลายเป็นสัญญาณข้อมูลที่เข้ารหัสแบบ MLT-3 วงจรภาคส่งประกอบด้วยวงจรอิควอไลเซอร์แอมพลิไฟเออร์ชนิดที่มีอินพุตแบบดิฟเฟอเรนเชียลซึ่งมีวงจรตรวจจับสัญญาณอยู่ภายใน (Differential Input Equalization Amplifier with Signal Detect Circuitry) วงจรคอมพาราเตอร์ (Comparator) วงจรลูปแบ็กมัลติเพล็กเซอร์ (Loopback Multiplexer Logic) และวงจรไดรเวอร์ที่มีเอาท์พุตแบบดิฟเฟอเรนเชียลของสัญญาณ PMID และสัญญาณ Signal Detect โดยข้อมูลที่ได้รับเข้ามาจะผ่านการอิควอไลซ์หรือไม่ขึ้นอยู่กับการควบคุมสัญญาณ EQSEL จากนั้นวงจรคอมพาราเตอร์จะทำการควอนไตซ์และแปลงสัญญาณที่เข้ารหัสแบบ MLT-3 เป็นสัญญาณไบนารี สัญญาณควบคุมจากขา $\overline{\text{CDET}}$ และ ENCSEL ทำให้ทราบว่ามีข้อมูลอยู่ในสายสัญญาณ

หรือไม่และยังใช้ควบคุมการแปลงข้อมูลตามลำดับ ส่วนวงจรมัลติเพล็กเซอร์ทำหน้าที่เลือกเส้นทางให้สัญญาณข้อมูลที่ได้รับที่ขา PMRD+/- ลูปกลับไปออกที่ขา PMID+/- หรือส่งตรงไปออกที่ขา TXO+/- และวงจรไดรเวอร์ถูกใช้เพื่อส่งสัญญาณ PMID+/- และสัญญาณ SD+/- ไปยังอุปกรณ์ PHY



ภาพประกอบ 3-6 บล็อกไดอะแกรมของชิพ DP83223

(ที่มา : National Semiconductor. “DP83223 TWISTER High Speed Networking Transceiver Device”.)

3.4 การตรวจสอบความถูกต้องของข้อมูล

สำหรับการอินเตอร์เฟสกับระบบเครือข่ายผ่านทาง MII นั้นทั้งภาครับและส่งจะใช้สัญญาณ RX_ER (Receive Error) และสัญญาณ TX_ER (Transmit Error) เป็นสัญญาณที่บอกให้ทราบถึงความผิดพลาดของข้อมูลหากมีความผิดพลาดของข้อมูลเกิดขึ้นในการรับและส่งข้อมูลตามลำดับ ในกรณีที่มีความผิดพลาดของข้อมูลเกิดขึ้นสัญญาณดังกล่าวจะเปลี่ยนสถานะจากลอจิกต่ำเป็นลอจิกสูง

ในวิทยานิพนธ์นี้ได้ตัดการตรวจสอบความถูกต้องของข้อมูลออกไปเพื่อลดความซับซ้อนของวงจร โดยมีสมมติฐานว่าระยะทางในการรับส่งข้อมูลนั้นสั้นและความผิดพลาดของข้อมูลเกิดขึ้นต่ำ ซึ่งหากมีความผิดพลาดของข้อมูลเกิดขึ้นเป็นหน้าที่ของปลายทางที่จะต้องตรวจสอบและจัดการกับข้อมูลที่ผิดพลาดต่อไป เช่น อาจส่งข้อมูลไปบอกต้นทางให้ส่งแพ็คเกตมาใหม่ เป็นต้น

3.5 การเลือกอุปกรณ์

3.5.1 หน่วยความจำ

หน่วยความจำถือเป็นส่วนประกอบที่สำคัญสำหรับเครื่องจัดการจราจรในระบบเครือข่าย เนื่องจากเป็นส่วนที่ใช้สำหรับจัดเก็บแพ็คเกตและมีผลกระทบกับประสิทธิภาพโดยรวมของระบบอย่างมาก จึงมักพบว่าหน่วยความจำกลายเป็นปัญหาคอขวดของระบบในแพ็คเกตโปรเซสเซอร์ทั่วไป การเลือกหน่วยความจำนั้นสิ่งที่ต้องพิจารณาได้แก่ ขนาด ชนิดของหน่วยความจำและเวลาในการเข้าถึงหน่วยความจำ

ชนิดของหน่วยความจำที่เลือกใช้ พิจารณาจากหน่วยความจำที่นิยมใช้งานในปัจจุบันได้แก่

1) DRAM หรือ Dynamic Random Access Memory มีราคาถูก ความจุสูง กำลังสูญเสียต่ำ ใช้เวลาในการเข้าถึงมากกว่า จำเป็นต้องรีเฟรชอยู่ตลอดเวลา
2) SRAM หรือ Static Random Access Memory มีราคาสูงกว่าเมื่อเทียบกับ DRAM ที่ความจุเท่ากัน มีความจุต่ำกว่าและมีกำลังสูญเสียมากกว่า แต่ใช้เวลาในการเข้าถึงน้อยกว่า

ในวิทยานิพนธ์นี้ต้องการหน่วยความจำที่มีเวลาในการเข้าถึงหน่วยความจำ (Access Time) ต่ำจึงเลือกใช้หน่วยความจำแบบ SRAM เนื่องจากหน่วยความจำชนิดนี้ใช้เวลาในการเข้าถึงหน่วยความจำน้อยกว่า DRAM แม้ว่า SRAM จะมีราคาที่สูงกว่า แต่เนื่องจากราคาของหน่วยความจำในท้องตลาดมีราคาลดลงมาก จึงทำให้ราคาของ SRAM มีราคาไม่สูงจนเกินไป

ขนาดของหน่วยความจำพิจารณาจากจำนวนแพ็คเกตที่ต้องการจัดเก็บและอัลกอริทึมที่เลือกใช้ ในวิทยานิพนธ์นี้เลือกใช้หน่วยความจำขนาด 512 กิโลไบต์ โดยการประมาณขนาดสูงสุดของแพ็คเกตเท่ากับ 2 กิโลไบต์ เพราะฉะนั้นจำนวนแพ็คเกตที่สามารถจัดเก็บได้เท่ากับ 256 แพ็คเกต ในกรณีที่ใช้อัลกอริทึมในการจัดคิวแพ็คเกตตามลำดับความสำคัญ โดยแบ่งกลุ่มออกเป็น 8 กลุ่มคือ 0-7 จำนวนคิวในแต่ละกลุ่มจะมีค่าเท่ากับ 32 คิว จะเห็นว่าขนาดของหน่วยความจำที่เลือกใช้จะมีผลกับขนาดของคิวที่สามารถมีได้ รวมทั้งดีเลย์ที่จะเกิดขึ้นเมื่อมีจำนวนคิวมาก ๆ เพราะฉะนั้นการเลือกขนาดหน่วยความจำต้องพิจารณาให้เหมาะสมกับอัลกอริทึมที่เลือกใช้ด้วย

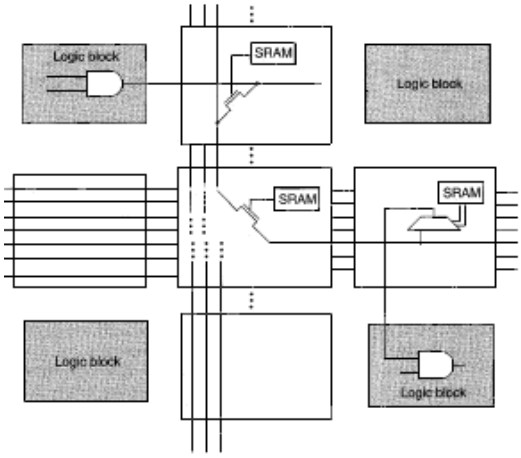
เวลาในการเข้าถึงหน่วยความจำเป็นอีกปัจจัยหนึ่งที่มีผลต่อประสิทธิภาพการทำงานของเครื่องจัดการจราจรในระบบเครือข่าย การเลือกค่าเวลาในการเข้าถึงหน่วยความจำพิจารณาจากอัตราการรับส่งข้อมูลที่สามารถทำได้ เช่น การเลือกเวลาในการเข้าถึงหน่วยความจำเท่ากับ 100 นาโนวินาทีโดยที่การเข้าถึงหน่วยความจำแต่ละครั้งสามารถกระทำได้ทีละ 16 บิตนั้น อัตราเร็วใน

การรับและส่งข้อมูลมีค่าเท่ากับ 160 เมกกะบิตต่อวินาที (มาจาก 16 บิต/100 นาโนวินาที) แต่เนื่องจากการใช้งานหน่วยความจำเป็นแบบ Dual Port หมายถึงทำการรับและส่งข้อมูลพร้อม ๆ กัน เพราะฉะนั้นอัตราเร็วในการส่งข้อมูลหรืออัตราเร็วในการรับข้อมูลจึงมีค่าเท่ากับ 80 เมกกะบิตต่อวินาที นั่นคือ อาจเกิดการสูญหายของแพ็คเกต (Packet Loss) ขึ้นได้ ถ้าหากเปลี่ยนค่าเวลาในการเข้าถึงหน่วยความจำเท่ากับ 70 นาโนวินาที อัตราเร็วในการส่งข้อมูลหรืออัตราเร็วในการรับข้อมูลจะมีค่าเท่ากับ 114 เมกกะบิตต่อวินาที ค่าที่ได้เป็นการประมาณโดยคร่าว ๆ ซึ่งอัตราเร็วในการรับส่งข้อมูลที่ทำได้จริงจะมีค่าต่ำกว่าที่คำนวณได้เนื่องจากเวลาที่สูญเสียไปกับการทำงานของวงจร *Arbiter* ซึ่งควบคุมการเข้าถึงหน่วยความจำ ดังนั้นในวิทยานิพนธ์นี้จึงเลือกใช้หน่วยความจำที่มีเวลาในการเข้าถึงหน่วยความจำเท่ากับ 70 นาโนวินาที

3.5.2 ชิพเอฟพีจีเอ

ชิพเอฟพีจีเอเป็นฮาร์ดแวร์ที่จะนำมาช่วยในการจัดคิวแพ็คเกต สาเหตุที่เลือกใช้ชิพเอฟพีจีเอเนื่องจากเป็นอุปกรณ์ดิจิตอลชนิดโปรแกรมซ้ำได้ (Reconfiguration Logic Device) ทำให้ง่ายต่อการปรับเปลี่ยนวงจรในการออกแบบ และช่วยตัดปัญหาความยุ่งยากในการเดินสาย จึงเหมาะสำหรับการนำมาเป็นต้นแบบในการออกแบบ ชิพเอฟพีจีเอสามารถแบ่งได้เป็น 2 ประเภทได้แก่

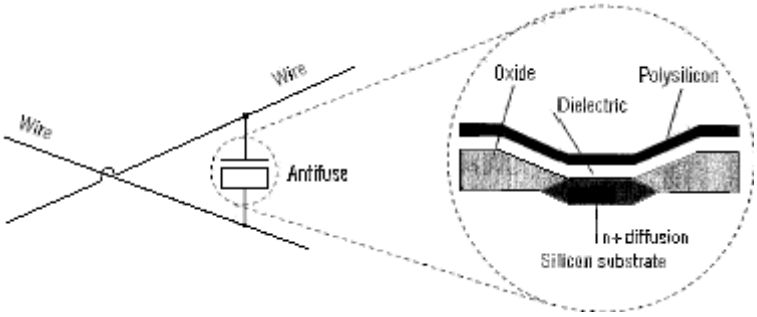
1) SRAM-based การเชื่อมต่อภายในจะใช้ทรานซิสเตอร์ เกต หรือมัลติเพล็กเซอร์ซึ่งควบคุมโดยใช้เซลล์ SRAM ทำให้การกำหนดค่าคอนฟิกกูเรชันใหม่ทำได้สะดวก รวดเร็ว ตัวอย่างสถาปัตยกรรมของชิพเอฟพีจีเอประเภทนี้ แสดงดังภาพประกอบ 3-7



ภาพประกอบ 3-7 ตัวอย่างสถาปัตยกรรมของชิพเอฟพีจีเอประเภท SRAM-based

(ที่มา : Stephen Brown and Jonathan Rose. 1996. “FPGA and CPLD Architectures : A Tutorial”, IEEE Design & Test of Computers (Summer 1996), 42-57.)

2) Antifuse-based โดยปกติสถาปัตยกรรมแบบนี้มักจะมีอยู่ในสถานะซึ่งมีค่าอิมพีแดนซ์สูงหรือเปิดวงจร (Open Circuit) และมีค่าอิมพีแดนซ์ต่ำในขณะที่ถูกโปรแกรม เทคโนโลยีที่ใช้ผลิตชิพประเภทนี้เป็นเทคโนโลยี CMOS ที่ได้รับการดัดแปลง ตัวอย่างของสถาปัตยกรรมของชิพเอฟพีจีเอประเภท Antifuse-based แสดงดังภาพประกอบ 3-8



ภาพประกอบ 3-8 ตัวอย่างสถาปัตยกรรมของชิพเอฟพีจีเอประเภท Antifuse-based
(ที่มา : Stephen Brown and Jonathan Rose. 1996. “FPGA and CPLD Architectures : A Tutorial”, IEEE Design & Test of Computers (Summer 1996), 42-57.)

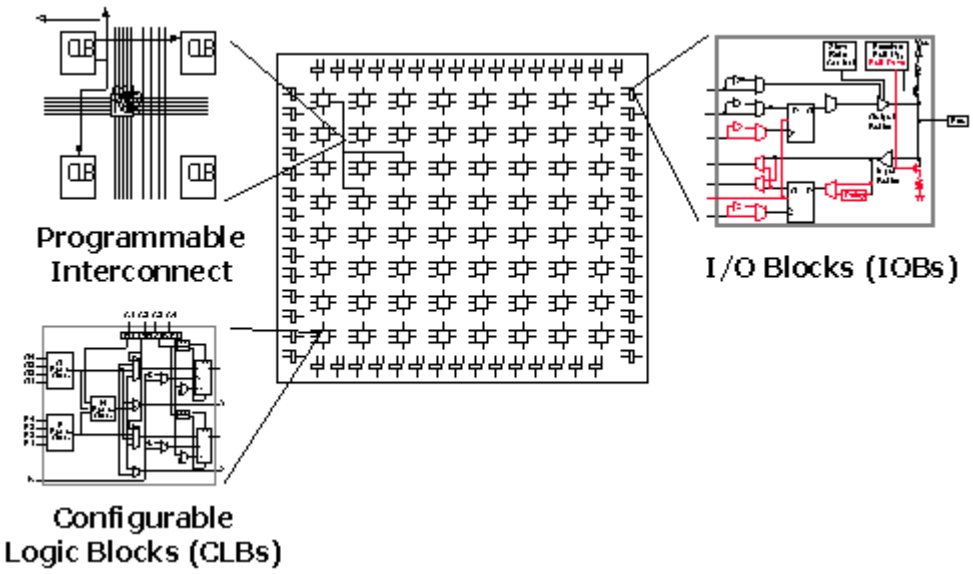
ตาราง 3-2 ความแตกต่างของชิพเอฟพีจีเอประเภท SRAM และ Antifuse-based
(ที่มา : Stephen Brown and Jonathan Rose. 1996. “FPGA and CPLD Architectures : A Tutorial”, IEEE Design & Test of Computers (Summer 1996), 42-57.)

Switch Type	Reprogrammable?	Volatile?	Technology
SRAM	(in circuit) Yes	Yes	CMOS
Antifuse	(in circuit) No	No	CMOS+

ความแตกต่างของชิพเอฟพีจีเอทั้งสองประเภทนี้แสดงดังตาราง 3-2 เนื่องจากชิพเอฟพีจีเอประเภท SRAM-based มีลักษณะเป็นแบบ Programming in Circuit ทำให้สามารถโปรแกรมเอฟพีจีเอโดยไม่จำเป็นต้องถอดชิพเอฟพีจีเอออกจากวงจร ซึ่งจะสะดวกในการพัฒนา

และปรับปรุงแก้ไขระบบเพราะจะมีความยืดหยุ่นมากกว่า ดังนั้นจึงเลือกที่จะใช้ชิพเอฟพีจีเอประเภท SRAM-based มาใช้งาน

ชิพเอฟพีจีเอประเภท SRAM-based ของบริษัท Xilinx ที่เป็นที่นิยมคือชิพเอฟพีจีเอในตระกูล XC4000 โดยสถาปัตยกรรมของชิพเอฟพีจีเอในตระกูลนี้จะประกอบไปด้วยตรรกะบล็อกหรือที่เรียกว่า Configuration Logic Block (CLB) ซึ่งเชื่อมต่อกันด้วยช่องทางเชื่อมต่อ (Routing Channels) ทั้งในแนวตั้งและแนวนอน ดังแสดงในภาพประกอบ 3-9 ประเด็นสำคัญของการเชื่อมต่อระหว่างซีแอลบี (Interconnect) ภายในชิพเอฟพีจีเอคือ การเคลื่อนที่ของสัญญาณจากซีแอลบีหนึ่งไปยังอีกซีแอลบีจะต้องเคลื่อนที่ผ่านสวิตช์ซึ่งจำนวนของสวิตช์ที่ผ่านนั้นขึ้นอยู่กับเส้นทางที่เลือกใช้ ดังนั้นประสิทธิภาพในเรื่องความเร็วของวงจรที่ออกแบบจะขึ้นอยู่กับการกำหนดเส้นทางให้กับสัญญาณ ซีแอลบีหนึ่ง ๆ จะมีส่วนประกอบสำคัญคือ Lookup Table ซึ่งเป็นอะเรย์ของหน่วยความจำขนาด 1 บิตอยู่ภายในบล็อกไดอะแกรมของซีแอลบีแสดงดังภาพประกอบ 3-10

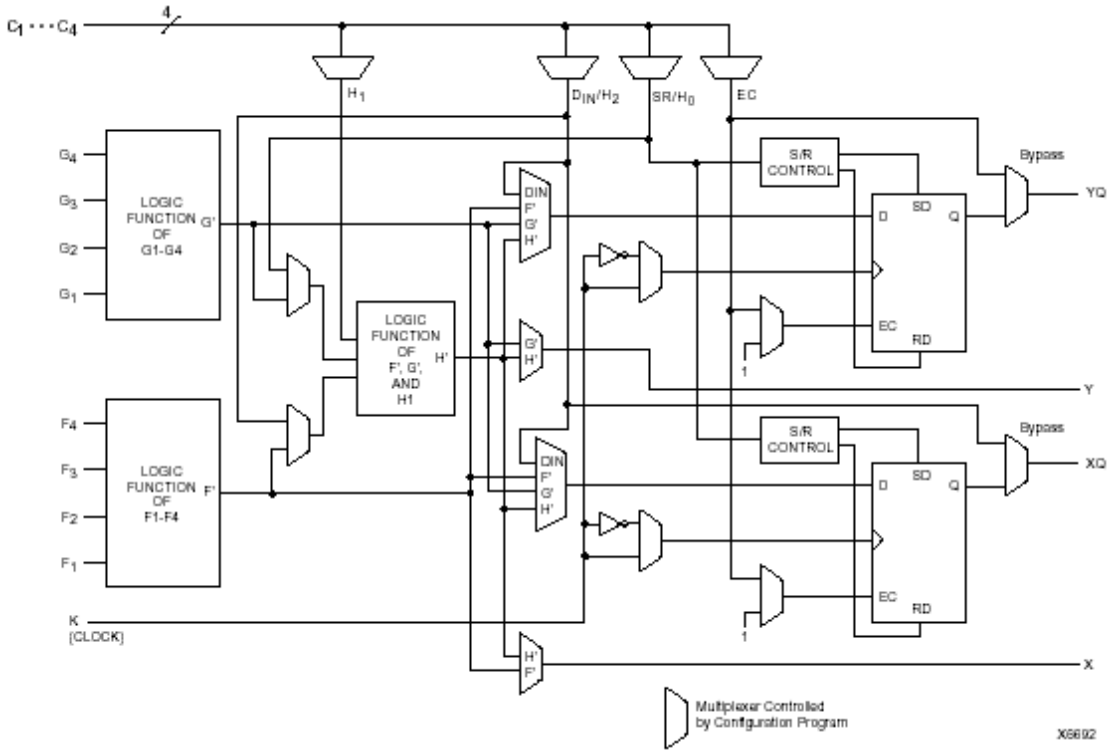


ภาพประกอบ 3-9 สถาปัตยกรรมของชิพตระกูล XC4000

(ที่มา: XC4000 Architecture. www.cedcc.psu.edu/ee497i/xilinx/day2/sld006.htm)

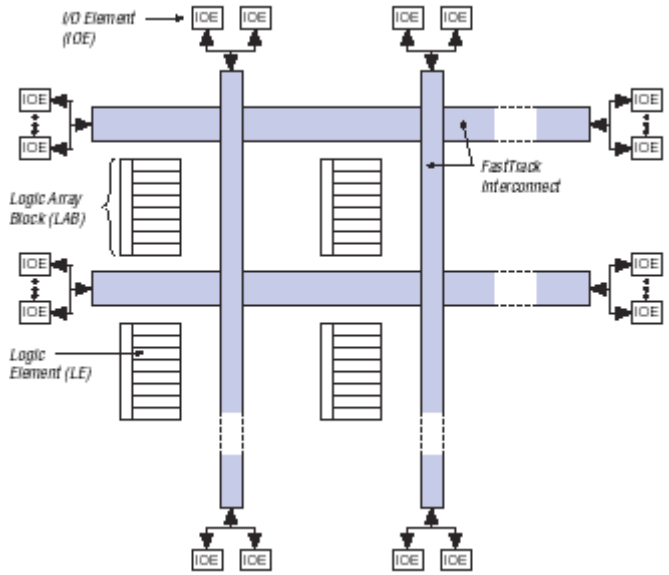
ชิพเอฟพีจีเอประเภท SRAM-based ของบริษัท Altera ที่เป็นที่นิยมคือชิพเอฟพีจีเอในตระกูล Altera Flex 8000 ชิพตระกูล Altera Flex 8000 จะรวมเทคโนโลยีของเอฟพีจีเอและซีพีแอลดี (Complex Programmable Logic Device : CPLD) เข้าไว้ด้วยกัน โดยอุปกรณ์จะประกอบด้วย 3 ลำดับชั้น ซึ่งชั้นที่ต่ำที่สุดคือกลุ่มของ Lookup Table สถาปัตยกรรมของชิพตระกูล Altera Flex 8000 แสดงดังภาพประกอบ 3-11 โดยตรรกะเซลล์ในที่นี้จะถูกเรียกว่า Logic

Element ดังภาพประกอบ 3-12 และ Logic Element จะถูกจัดกลุ่มรวมกันกลุ่มละ 8 Logic Element เรียกว่า Logic Array Block ดังภาพประกอบ 3-13



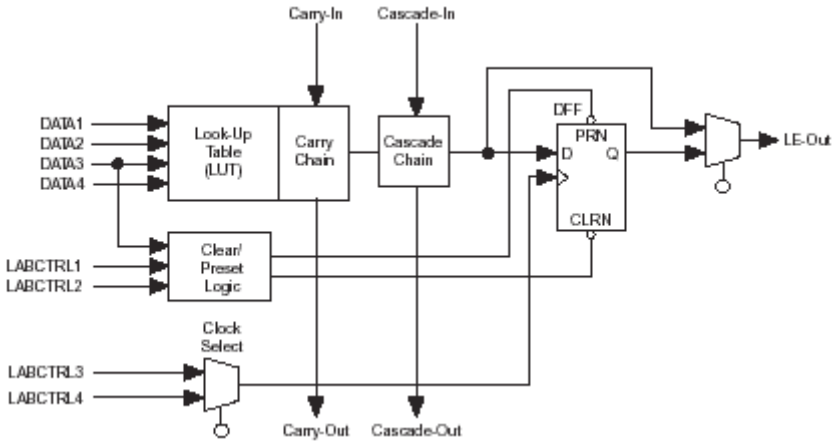
ภาพประกอบ 3-10 บล็อกไดอะแกรมของซีแอลบีของชิพตระกูล XC4000

(ที่มา: Xilinx,Inc. 1999. “XC4000E and XC4000X Series Field Programmable Gate Arrays”)



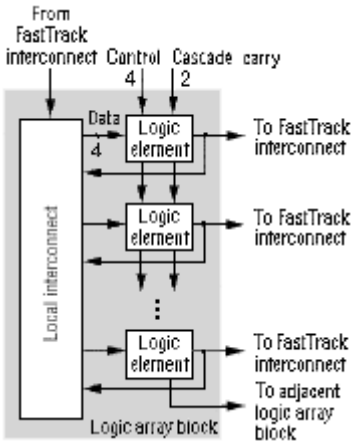
ภาพประกอบ 3-11 สถาปัตยกรรมของชิพตระกูล Altera Flex 8000

(ที่มา : Altera Corporation. 2003. “Flex 8000 Programmable Logic Device Family”.)



ภาพประกอบ 3-12 Logic Element ของชิพตระกูล Altera Flex 8000

(ที่มา : Altera Corporation. 2003. “Flex 8000 Programmable Logic Device Family”.)



ภาพประกอบ 3-13 Logic Array Block ของชิพตระกูล Altera Flex 8000

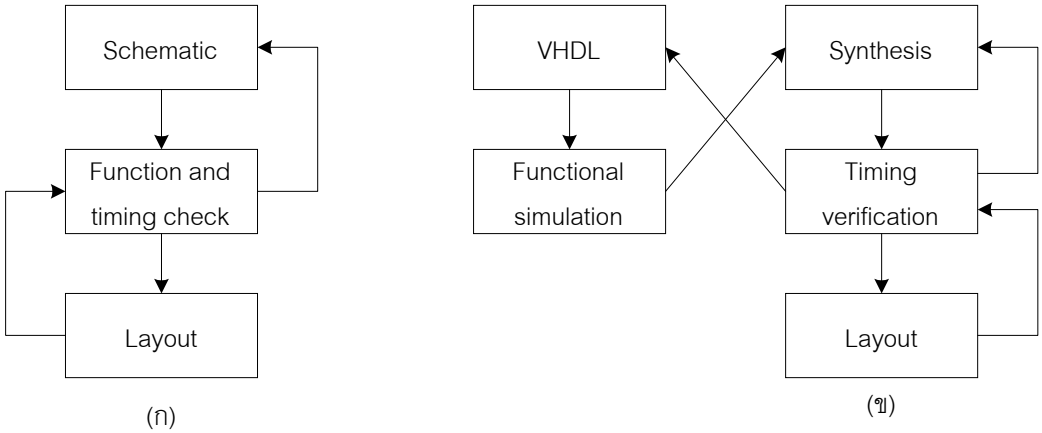
(ที่มา : Stephen Brown and Jonathan Rose. 1996. “FPGA and CPLD Architectures : A Tutorial”, IEEE Design & Test of Computers (Summer 1996), 42-57.)

ความแตกต่างระหว่างชิพเอฟพีจีเอตระกูล XC4000 และตระกูล Altera Flex 8000 คือ การเชื่อมต่อของชิพตระกูล Altera Flex 8000 ที่เรียกว่า FastTrack จะมีเฉพาะการเชื่อมต่อประเภท Long Lines ด้วยเหตุนี้ทำให้สามารถคาดการณ์ค่าหน่วงเวลาของชิพตระกูล Altera Flex 8000 ได้ดีกว่าชิพเอฟพีจีเอซึ่งใช้การเชื่อมต่อที่สั้นกว่าจำนวนมากเพราะว่าการเชื่อมต่อที่ยาวกว่าจะมีจำนวนสวิตช์น้อยกว่า

แม้ว่าชิพเอฟพีจีเอตระกูล Altera Flex 8000 จะเหนือกว่าชิพเอฟพีจีเอตระกูล XC4000 ในเรื่องของการคาดการณ์ค่าหน่วงเวลา แต่การที่ชิพเอฟพีจีเอตระกูล XC4000 มีรูปแบบการเชื่อมต่อที่หลากหลายทั้งแบบ Single Lines, Double Lines และ Quad Lines เป็นต้น ทำให้ขั้นตอนการ Place and Route จะทำได้ง่ายกว่า ดังนั้นในวิทยานิพนธ์นี้จึงได้เลือกใช้ชิพเอฟพีจีเอตระกูล XC4000ของบริษัท Xilinx รุ่น XC4020XLA-PQ208 เป็นวงจรรวมต้นแบบประกอบด้วย 1,862 ตรรกะเซลล์หรือ 784 ซีแอลบี เทียบเท่าวงจรดิจิตอลที่ประกอบด้วยเกต 20,000 ตัว มีจำนวนขาสัญญาณและขาควบคุมรวม 208 ขา

หลังจากได้ชิพเอฟพีจีเอที่ต้องการแล้ว ขั้นตอนต่อไปคือการเลือกวิธีที่จะใช้ในการออกแบบชิพเอฟพีจีเอ วิธีที่ใช้ในการออกแบบที่นิยมใช้มีด้วยกัน 3 วิธีได้แก่

1) Schematic Design วิธีการนี้เริ่มต้นจากการวาด Schematic จากนั้นทำการจำลองการทำงานของวงจรที่ออกแบบโดยการทำ Functional and Timing Simulation ถ้าหากพบข้อผิดพลาดในการออกแบบก็จะกลับไปแก้ไขใน Schematic ขั้นตอนของ Schematic Design แสดงดังภาพประกอบ 3-14 (ก)



ภาพประกอบ 3-14 ขั้นตอนการออกแบบชิพเอฟพีจีเอด้วย Schematic และ VHDL Design
(ที่มา : “VHDL Hardware Description Language”.
http://members.tripod.com/o_alshibami/project/ch3.html)

2) HDL Design (Hardware Description Language Design) เป็นการออกแบบโดยใช้ภาษาในการอธิบายพฤติกรรมหรือฟังก์ชันการทำงานของระบบ ซึ่งในปัจจุบันการออกแบบที่นิยมกันคือการใช้ VHDL วิธีการนี้เริ่มต้นจากเขียนฟังก์ชันการทำงาน

ของวงจรหรือระบบที่ต้องการโดยใช้ VHDL จากนั้นตรวจสอบความถูกต้องโดยการใช้ VHDL Simulation การแก้ไขวงจรที่เขียนด้วย VHDL ผู้ออกแบบสามารถทำได้รวดเร็วและมีประสิทธิภาพมากกว่า ขั้นตอนของ VHDL Design แสดงดังภาพประกอบ 3-14 (ข)

3) Finite State Machine Design วิธีการนี้เหมาะสำหรับวงจรควบคุม เนื่องจากวิธีการนี้จะนำเสนอในลักษณะการไหลของการควบคุม (Control Flow) ซึ่งจะคล้ายคลึงกับวิธีคิดของมนุษย์ จึงทำให้ง่ายต่อการทำความเข้าใจ นอกจากนี้ยังสามารถแปลงเป็น HDL ได้

ในทางปฏิบัติการออกแบบระบบที่มีขนาดใหญ่มาก ๆ อาจจำเป็นต้องใช้วิธีการออกแบบทั้งสามวิธีร่วมกัน สำหรับในวิทยานิพนธ์นี้เลือกใช้การออกแบบเอฟพีจีเอโดยใช้ Schematic Design เนื่องจากวงจรที่ต้องการจะมีลักษณะเป็น Sequential Circuits ซึ่งการออกแบบด้วย Schematic Design น่าจะออกแบบได้ง่ายกว่า

3.5.3 **ไมโครโปรเซสเซอร์** เนื่องจากเครื่องจัดการจราจรในระบบเครือข่ายที่ต้องการออกแบบนั้นเพื่อนำไปใช้งานกับเครือข่ายความเร็วสูงในระดับ 100 เมกกะบิตต่อวินาที นั่นคือเครื่องจัดการจราจรในระบบเครือข่ายดังกล่าวต้องมีขีดความสามารถในการทำงานที่ความเร็วสูง ดังนั้นไมโครโปรเซสเซอร์ที่เลือกใช้จึงต้องสอดคล้องกับวัตถุประสงค์การใช้งานของเครื่องจัดการจราจรในระบบเครือข่าย ประเด็นที่นำมาพิจารณาในการเลือกไมโครโปรเซสเซอร์ได้แก่

1) สถาปัตยกรรมของไมโครโปรเซสเซอร์ที่เลือกใช้
2) ภาระงานของไมโครโปรเซสเซอร์
3) จำนวนบัสข้อมูลและบัสแอดเดรสของไมโครโปรเซสเซอร์
4) ความเร็วของไมโครโปรเซสเซอร์
5) กำลังสูญเสียของไมโครโปรเซสเซอร์

สำหรับสถาปัตยกรรมของไมโครโปรเซสเซอร์นั้น เริ่มต้นจากการพิจารณารูปแบบของเครื่องจัดการจราจรในระบบเครือข่าย จากที่ได้นำเสนอในหัวข้อที่ 2.1.5 นั้น รูปแบบของเราเตอร์ซึ่งถือว่าเป็นแพ็คเกตโปรเซสเซอร์ชนิดหนึ่งมีด้วยกัน 3 รูปแบบ ในวิทยานิพนธ์นี้ได้เลือกใช้รูปแบบ Software-Based เนื่องจากง่ายต่อการปรับเปลี่ยนค่าต่าง ๆ รวมทั้งอัลกอริทึมที่เลือกใช้ แต่รูปแบบ Software-Based นั้นจะประสบปัญหาในเรื่องของการเสียเวลาอย่างมากในการเคลื่อนย้ายข้อมูล จึงได้นำฮาร์ดแวร์เข้ามาช่วยลดการเคลื่อนย้ายข้อมูล ดังนั้นรูปแบบของเครื่องจัดการจราจรในระบบเครือข่ายนี้จึงเป็นการผสมผสานกันระหว่างรูปแบบ Software-Based และ ASIC-Based

ดังนั้นขั้นต่อไปคือการเลือกสถาปัตยกรรมที่เหมาะสมกับการใช้ Software-Based โดยพิจารณาจากสถาปัตยกรรมของไมโครโปรเซสเซอร์ที่มีอยู่หลัก ๆ 2 ประเภทคือ

1) RISC (Reduced Instruction Set Computer) คุณลักษณะที่สำคัญคือ
- มีรีจิสเตอร์จำนวนมากเพื่อลดการติดต่อกับหน่วยความจำภายนอก
- การทำงานของชุดคำสั่งเป็นแบบไปป์ไลน์
- มีชุดคำสั่งที่สั้นและง่าย
- โอเปอร์เรชันส่วนใหญ่จะเป็นการติดต่อระหว่างรีจิสเตอร์กับรีจิสเตอร์เช่น คำสั่ง LOAD และ STORE
- การประมวลผล 1 ชุดคำสั่งภายใน 1 แมชชีนไซเคิล

2) CISC (Complex Instruction Set Computer) คุณลักษณะที่สำคัญคือ
- มีชุดคำสั่งที่ซับซ้อนกว่าแบบ RISC
- โอเปอร์เรชันส่วนใหญ่จะเป็นการติดต่อระหว่างหน่วยความจำกับหน่วยความจำ

เมื่อพิจารณาภาระงานของไมโครโปรเซสเซอร์ ซึ่งก็คือการจำแนกแพ็คเกต การกรองแพ็คเกต การส่งต่อแพ็คเกต การจัดคิว โดยโอเปอร์เรชันที่เกิดขึ้นส่วนใหญ่นั้นคือการกำหนดค่า การเปรียบเทียบค่า การเคลื่อนย้ายข้อมูล ซึ่งเป็นโอเปอร์เรชันที่ง่าย ไม่จำเป็นต้องมีการคำนวณทางคณิตศาสตร์ที่สลับซับซ้อน ด้วยเหตุนี้สถาปัตยกรรมที่เหมาะสมกับภาระงานของไมโครโปรเซสเซอร์และรูปแบบของเครื่องจัดการจราจรในระบบเครือข่ายคือ RISC

สำหรับจำนวนบัสแอดเดรสของไมโครโปรเซสเซอร์พิจารณาจากการอ้างตำแหน่งหน่วยความจำสูงสุดที่มีได้ จากขนาดหน่วยความจำที่เลือกใช้นั้น บัสแอดเดรสต้องอ้างอิงหน่วยความจำได้ไม่ต่ำกว่า 512 กิโลไบต์ หรือ มีบัสแอดเดรสมากกว่า 19 บิต แต่ในทางปฏิบัติบัสแอดเดรสที่ส่งมาจากไมโครโปรเซสเซอร์คือค่าแอดเดรส 8 บิตบน ส่วนบัสแอดเดรสอีก 11 บิตล่างกำหนดโดยฮาร์ดแวร์ที่ทำการออกแบบ ดังนั้นไมโครโปรเซสเซอร์ที่เลือกใช้ควรมีบัสแอดเดรสมากกว่า 8 บิต ส่วนการกำหนดจำนวนบัสข้อมูลนั้นพิจารณาจากความเร็วในการรับส่งข้อมูลและความซับซ้อนในการออกแบบลายวงจรพิมพ์ เนื่องจากระบบเครือข่ายมีอัตราเร็วในการรับส่งข้อมูลเท่ากับ 100 เมกกะบิตต่อวินาที เมื่อพิจารณาบัสข้อมูลขนาดต่าง ๆ พบว่า

หากใช้บัสข้อมูลขนาด 8 บิต ข้อมูลจะถูกส่งมาทุก ๆ 12.5 เมกกะเฮิร์ทหรือเท่ากับ 80 นาโนวินาที

หากใช้บัสข้อมูลขนาด 16 บิต ข้อมูลจะถูกส่งมาทุก ๆ 6.25 เมกกะเฮิร์ทหรือเท่ากับ 160 นาโนวินาที

หากใช้บัสข้อมูลขนาด 32 บิต ข้อมูลจะถูกส่งมาทุก ๆ 3.125 เมกกะเฮิร์ทหรือเท่ากับ 320 นาโนวินาที

ในที่นี้เลือกใช้บัสข้อมูลขนาด 16 บิต เพื่อไม่ให้การออกแบบลายวงจรพิมพ์ซับซ้อนจนเกินไป ซึ่งไมโครโปรเซสเซอร์จะสามารถประมวลผลได้ทันหรือไม่ขึ้นอยู่กับความซับซ้อนของซอฟท์แวร์ที่ใช้ประมวลผล หากไมโครโปรเซสเซอร์ประมวลผลไม่ทันผู้เขียนโปรแกรมจะต้องปรับแก้ซอฟท์แวร์เพื่อลดความซับซ้อนของซอฟท์แวร์

นอกเหนือจากสถาปัตยกรรมที่เลือกใช้แล้วสิ่งที่ต้องพิจารณาคือกำลังสูญเสียของไมโครโปรเซสเซอร์ที่เลือกใช้ เนื่องจากเครื่องจัดการจราจรในระบบเครือข่ายมีลักษณะเป็นระบบแบบฝังตัว ซึ่งระบบดังกล่าวมักจะมีกำลังสูญเสียต่ำ ดังนั้นไมโครโปรเซสเซอร์ที่เลือกใช้ควรจะมีกำลังสูญเสียต่ำด้วยเช่นกัน

ไมโครโปรเซสเซอร์ที่เลือกใช้คือ ชิพ DS89C420 ของบริษัท Dallas Semiconductor ซึ่งสามารถทำงานกับสัญญาณนาฬิกาได้เร็วสูงสุด 50 เมกกะเฮิร์ท และประมวลผลคำสั่ง 1 คำสั่งภายในสัญญาณนาฬิกา 1 ลูกหรือเท่ากับ 20 นาโนวินาที มีบัสแอดเดรส 16 เส้น

เมื่อพิจารณาฟังก์ชันการทำงานพบว่าได้ตัดภาระงานในเรื่องของการเคลื่อนย้ายข้อมูลออกไป ฟังก์ชันการทำงานที่เหลือคือการส่งสัญญาณควบคุม การกำหนดค่า *Base Address* ไมโครโปรเซสเซอร์ใช้เวลาในการทำคำสั่งเหล่านี้น้อยมาก สำหรับการประมวลผลเพื่อจัดคิวแม้จะไม่สามารถระบุเวลาในการประมวลผลอย่างชัดเจนเนื่องจากขึ้นอยู่กับอัลกอริทึมที่เลือกใช้ แต่ก็คาดว่าไมโครโปรเซสเซอร์น่าจะทำงานได้ทัน ในกรณีที่ไมโครโปรเซสเซอร์ไม่สามารถประมวลผลได้ทัน โปรโตคอลในระดับชั้นสูง ๆ (High Level Protocol) ของเครื่องต้นทางจะทำการส่งแพ็คเกตมาใหม่

ข้อพิจารณาในการออกแบบที่นำเสนอในบทนี้ก็เพื่อที่จะบอกให้ทราบถึงประเด็นสำคัญที่ใช้ในการออกแบบและเหตุผลในการตัดสินใจเลือกรูปแบบ วิธีการ และอุปกรณ์ที่ใช้ โดยรายละเอียดการออกแบบเครื่องจัดการจราจรในระบบเครือข่ายจะกล่าวถึงในบทถัดไป