วิศวกรรมสาร มข. ปีที่ 32 ฉบับที่ 6 (741 -753) พฤศจิกายน - ธันวาคม 2548
KKU Engineering Journal Vol 32 No 6 (741 -753) November– December 2005

การออกแบบระบบตรรกะเชิงอะซิงโครนัสเบื้องต้น*

ดารณี หอมดี[1]

[1] อาจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น จังหวัดขอนแก่น 40002

Email: darhor@kku.ac.th

บทคัดย่อ

ระบบซิงโครนัส (synchronous system) แบบดั้งเดิมจะใช้สัญญาณนาฬิการะบบในการควบคุมจังหวะการทำงานของส่วนต่าง ๆ เมื่อสัญญาณนาฬิการะบบนี้มีความเร็วมากขึ้น รวมไปถึงระบบมีขนาดใหญ่ขึ้น และเส้นสัญญาณต่าง ๆ มีขนาดเล็กลง การที่จะทำให้ส่วนต่าง ๆ ของระบบยังทำงานได้อย่างเข้าจังหวะกันนั้นจะทำได้ยากขึ้นเรื่อย ๆ

ทางแก้ไขปัญหาข้างต้นทางหนึ่งคือ การออกแบบเชิงอะซิงโครนัส (Asynchronous design) ซึ่งวิธีนี้จะแก้ปัญหาที่เกี่ยวเนื่องกับคาบเวลาโดยการแทนที่ระบบสัญญาณนาฬิกาควบคุมรวมโดยการควบคุมที่อยู่ในรูปของข้อ ตกลงร่วมกันระหว่างระบบย่อยต่าง ๆ ในการสื่อสารโดยไม่ยึดเรื่องของเวลาเป็นหลัก

บทความนี้นำเสนอบทนำเกี่ยวกับการออกแบบเชิงอะซิงโครนัสนี้ โดยอธิบายถึงข้อแตกต่างพื้นฐานในการออกแบบระหว่างเชิงซิงโครนัสและเชิงอะซิงโครนัส และข้อดี-ข้อด้อยของการออกแบบเชิงอะซิงโครนัส

คำสำคัญ : การออกแบบเชิงอะซิงโครนัส ระบบอะซิงโครนัส การออกแบบเชิงตรรกะดิจิตอล

---

* รับต้นฉบับเมื่อวันที่ 31 ตุลาคม 2548 และได้รับบทความฉบับแก้ไขเมื่อวันที่ 3 พฤศจิกายน 2548

Introduction to Asynchronous Logic Design*

Daranee Hormdee[1)]

[1)] Lecturer, Department of Computer Engineering, Khon Kaen University, KhonKaen, 40002

Email:darhor@kku.ac.th

Conventional synchronous systems are based on global clocking whereby global synchronization signals control the rate at which different units operate. As the clocks get faster, the systems bigger and the wires finer, it is increasingly difficult to ensure that all parts in the system are operating in lockstep with one another.

One solution to this is the use of asynchronous design, which attacks clock-related timing problems by replacing global clock control with some form of agreement on a mutually acceptable protocol of data transmission and acknowledgement which is not regulated by time. This can happen locally within a unit or globally between subsystems.

The paper gives an introduction to the asynchronous design describing the fundamental difference between synchronous and asynchronous design styles and highlighting the pros and cons of asynchronous

Keywords : asynchronous design, asynchronous system, digital logic design

---

* Original manucript submitted: October 31, 2005 and Final manucript received: November 3, 2005

บทนำ

ความรุดหน้าของเทคโนโลยีด้านการพัฒนาระบบดิจิตอลในปัจจุบันได้สร้างศักยภาพและโอกาสให้นักออกแบบสามารถสร้างระบบที่มีประสิทธิภาพสูงและมีส่วนประกอบเชิงตรรกะในรูปของวงจร *VLSI (Very Large Scale Integrated)* ที่มีความหนาแน่นที่สูงด้วย ระบบ VLSI ในปัจจุบันนี้ส่วนใหญ่ถูกออกแบบโดยถือหลักปฏิบัติ 2 ประการคือ

1. ข้อมูลต่าง ๆที่ใช้อยู่ในรูปของเลขฐานสอง
2. เวลามีลักษณะไม่ต่อเนื่อง

ซึ่งนักออกแบบส่วนใหญ่จะยึดถือหลักปฏิบัติเหล่านี้สำหรับการออกแบบวงจรดิจิตอล เนื่องจากเป็นวิธีการที่เลี่ยงปัญหาต่าง ๆที่เกี่ยวเนื่องในการประมวลผลข้อมูลเชิงดิจิตอล โดยทั่วไปวงจร VLSI เหล่านี้ใช้นาฬิการ่วมที่ถูกส่งกระจายไปยังส่วนต่าง ๆของระบบเพื่อให้จังหวะและควบคุมลำดับการทำงานและการลำเลียงข้อมูล ระบบดังกล่าวเรียกว่าระบบซิงโครนัส *(Synchronous System)*

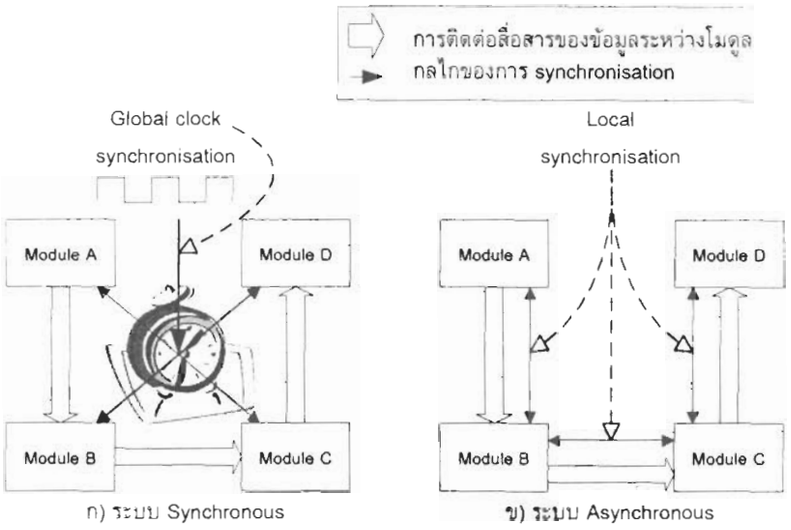
แต่ที่จริงแล้วโดยธรรมชาติวิธีการออกแบบวงจรเพื่อการประมวลผลข้อมูลกลับคือวิธีที่ยังไม่แพร่หลายนักที่เรียกว่า *การออกแบบเชิงอะซิงโครนัส (Asynchronous Design)* ซึ่งแต่ละส่วนประกอบย่อยภายในระบบประเภทนี้จะประมวลผลข้อมูลทันทีที่มีข้อมูลเข้ามาเป็น input ของโมดูลย่อยนั้น ๆ โดยไม่จำเป็นต้องรอความพร้อมของการทำงานของโมดูลอื่น ๆ

ตลอดระยะเวลากว่าครึ่งศตวรรษที่ผ่านมา นักพัฒนาระบบวงจรดิจิตอลส่วนใหญ่ได้เลือกการออกแบบเชิงซิงโครนัสที่ใช้นาฬิการะบบในการให้จังหวะแก่ทุกส่วนของระบบเพื่อความสะดวกในการสื่อสาร อย่างไรก็ตามแนวความคิดเกี่ยวกับระบบอะซิงโครนัสก็ไม่ได้ถูกละเลยไปทั้งหมดเสียทีเดียว แต่ยังคงถูกรักษาและพัฒนาภายในทั้งกลุ่มของงานวิจัยและภาคเอกชนต่าง ๆดังรายนามที่สามารถหาได้จาก Asynchronous Logic Homepage (Homepage) ซึ่งที่นี่และ Asynchronous Bibliography (Bibliography) ยังเป็นแหล่งค้นคว้างานทั้งเชิงวิจัยและวิชาการเกี่ยวกับการออกแบบเชิงอะซิงโครนัสตลอดระยะเวลากว่า 15 ปีที่ผ่านมา

บทความนี้จะกล่าวแนะนำการออกแบบระบบตรรกะประเภทที่เรียกว่า *Asynchronous Logic Design* โดยจะเริ่มด้วยความหมายและลักษณะเฉพาะของการออกแบบระบบตรรกะประเภทนี้ จากนั้นจะเป็นการสรุปถึงข้อเด่นต่าง ๆของระบบเมื่อเทียบกับวิธีการออกแบบระบบตรรกะทั่ว ๆไปที่ใช้นาฬิกาในการควบคุมการทำงาน รวมไปถึงสาเหตุที่เป็นฉนวนให้ระบบอะซิงโครนัสนี้ยังไม่เป็นที่แพร่หลายนัก

ระบบ(อะ)ซิงโครนัส

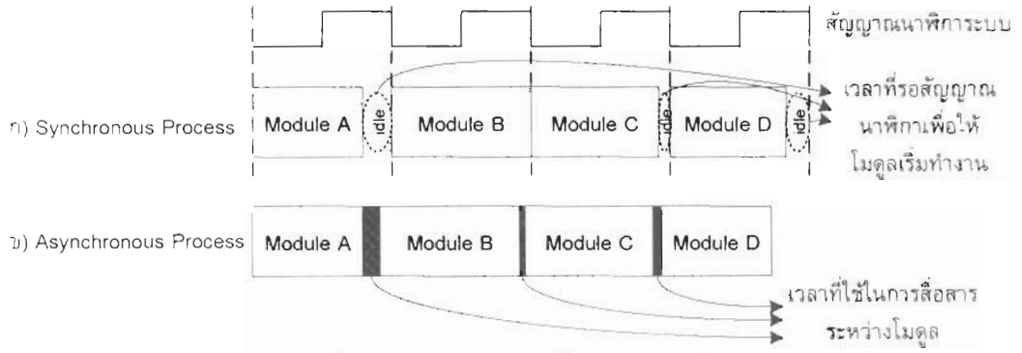
วงจรซิงโครนัส (Synchronous Circuit) ใช้*สัญญาณคาบเวลา (periodic signal)* หรือ *สัญญาณนาฬิกา (clock signal)* หรือที่เรียกได้ว่า *นาฬิการะบบ (system clock)* เพื่อทำให้ส่วน (หรือโมดูล) ต่าง ๆของระบบทำงานเข้ากันได้อย่างเป็นจังหวะ สัญญาณนาฬิกานี้จะถูกส่งกระจายไปยังทุกส่วนของระบบเพื่อใช้ในการให้จังหวะแก่กลไกต่าง ๆดังในรูปที่ 1(ก)



รูปที่ 1 *ระบบ(อะ)ซิงโครนัส (Janin,L. 2004)*

ในทางตรงกันข้าม *วงจรอะซิงโครนัส (Asynchronous Circuit)* จะไม่มีนาฬิการ่วมของระบบการให้จังหวะสามารถทำได้โดย สร้างสัญญาณสื่อสารโดดอบเฉพาะในส่วนที่ต้องการการเข้าจังหวะกันของการทำงานดังในรูปที่ 1(ข)

รูปที่ 2 อธิบายขั้นตอนการทำงานเชิงซิงโครนัสเมื่อเปรียบเทียบกับเชิงอะซิงโครนัส โดยที่สมมติว่าระบบนี้ประกอบด้วยโมดูลทั้งหมด 4 โมดูลได้แก่ A B C และ D ทำงานตามลำดับ ในระบบซิงโครนัสแต่ละโมดูลต้องรอสัญญาณนาฬิกาเพื่อที่จะเริ่มการทำงานในแต่ละครั้ง ซึ่งในระบบชนิดนี้อาจมีช่วงเวลาว่าง *(idle time)* หลังจากที่โมดูลหนึ่งได้เสร็จสิ้นภารกิจและรอสัญญาณนาฬิกาใหม่มาเพื่อเริ่มการทำงานของโมดูลอื่นต่อไป ทั้งนี้เนื่องจากโดยทั่วไประยะเวลาที่แต่ละโมดูลใช้ในการทำงานบ่อยครั้งที่จะไม่เท่ากัน ดังแสดงในรูปที่ 2(ก)



รูปที่ 2 *ขั้นตอนการทำงานเชิงซิงโครนัสและอะซิงโครนัส*

ในขณะที่ในระบบอะซิงโครนัสการสื่อสารระหว่างแต่ละโมดูลย่อยจะมีระบบการติดต่อเฉพาะของแต่ละคู่

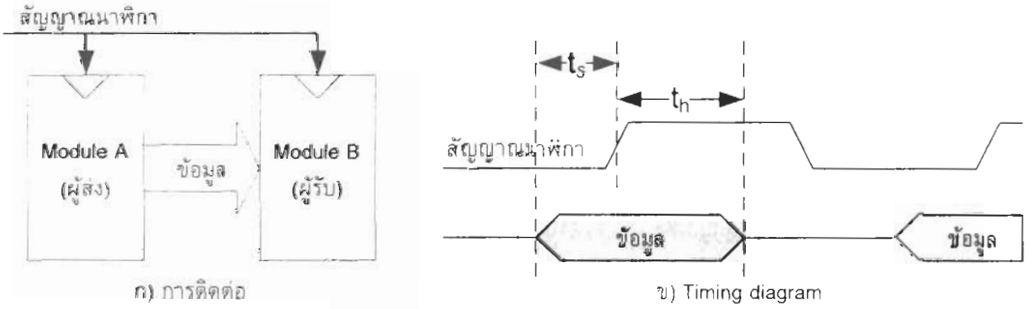
ไป ทำให้การทำงานของโมดูลใด ๆสามารถเริ่มได้ทันทีที่โมดูลคู่นั้นพร้อม โดยที่อาจมีการเสียเวลาอยู่บ้างเล็กน้อยเพื่อเริ่มต้นการติดต่อสื่อสารระหว่างโมดูลย่อยในแต่ละครั้ง ดังแสดงในรูปที่ 2(ข)

การติดต่อสื่อสารเชิง(อะ)ซิงโครนัส

ในการติดต่อสื่อสารนั้นข้อมูลจะถูกลำเลียงระหว่างโมดูล โดยผ่านกลุ่มของสายสัญญาณที่เรียกว่า*ช่องสัญญาณ (channel)* เมื่อคำนึงถึงทิศทางการไหลของข้อมูลระหว่างโมดูลคู่หนึ่ง ๆ จะได้ว่า

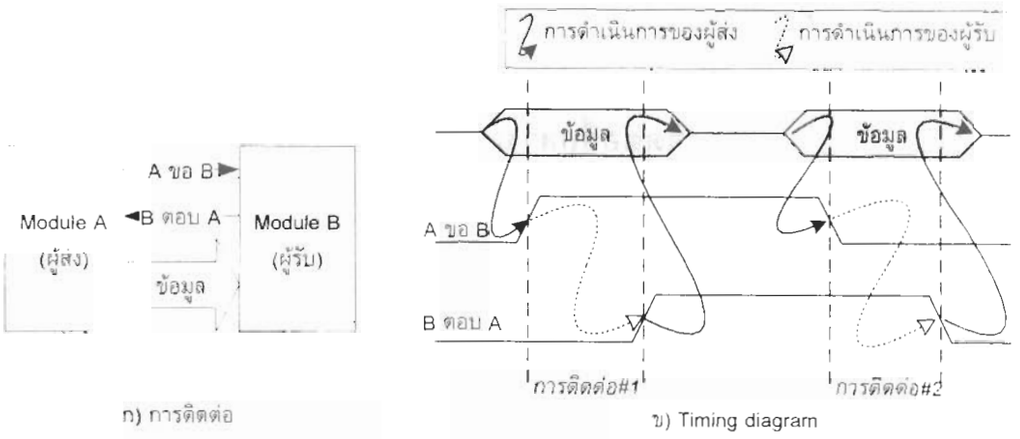
- *'ผู้ส่ง' (sender)* คือโมดูลที่วางข้อมูลลงบนช่องสัญญาณข้อมูล
- *'ผู้รับ' (receiver)* คือโมดูลที่มารับข้อมูลจากช่องสัญญาณข้อมูล

อย่างที่กล่าวไว้ข้างต้นว่าการออกแบบเชิงซิงโครนัสดั้งเดิมที่ใช้ในปัจจุบันอาศัยนาฬิการะบบ ในการควบคุมทุกการติดต่อสื่อสารของข้อมูลในระบบดังแสดงในรูปที่ 3(ก) โดยสัญญาณนาฬิการะบบนี้จะถูกส่งกระจายไปยังส่วนต่าง ๆของระบบ 'ผู้ส่ง'จะต้องวางข้อมูลบนช่องสัญญาณข้อมูลช่วงเวลาหนึ่งก่อนที่สัญญาณนาฬิกาจะมา ซึ่งช่วงเวลาดังกล่าวนั้นเรียกว่า *setup-time* (t_s) และข้อมูลบนช่องสัญญาณจะต้องยังคงค้างอยู่เป็นระยะเวลาหนึ่งที่เรียกว่า *hold-time* (t_h) เพื่อให้'ผู้รับ'ได้ทันเก็บค่าข้อมูลนั้นเพื่อนำไปประมวลผลต่อไป รูปที่ 3(ข) แสดงถึง timing diagram ของทั้งสองช่วงเวลานี้ ข้อได้เปรียบของวิธีนี้คือข้อจำกัดทางด้านเวลาดังที่ดังกล่าวมาทั้งในส่วนของ t_s และ t_h สำหรับการทำงานที่ถูกต้องของวงจรนั้น สามารถตรวจสอบได้โดยง่ายด้วยการวิเคราะห์เวลา static ทั่วๆไป



รูปที่ 3 *การติดต่อสื่อสารแบบซิงโครนัส (Hormdee, D. 2002)*

ในทางตรงกันข้ามระบบอะซิงโครนัส หรือที่เป็นที่รู้จักในนาม **ระบบไร้นาฬิกา** *(clockless)* หรือ *ระบบ self-timed* ตามลักษณะโครงสร้างของระบบ เป็นระบบที่ทำงานโดยปราศจากการให้จังหวะของนาฬิกาส่วนกลางเพื่อควบคุมการทำงานของส่วนต่าง ๆในระบบ โดยระบบชนิดนี้จะใช้การติดต่อสื่อสารที่เรียกว่า ***handshaking*** แทน ซึ่งจะเป็นการสื่อสารภายในระหว่างโมดูลย่อยที่เกี่ยวข้องเพื่อแจ้งให้ทราบว่าเมื่อใดที่ข้อมูลพร้อม 'ผู้ส่ง' A จะให้สัญญาณเพื่อขอ *(request)* ส่งข้อมูลไปยัง B และเมื่อใดที่'ผู้รับ'ได้รับข้อมูลแล้ว B จะให้สัญญาณตอบกลับ *(acknowledge)* มาที่ A ดังแสดงในรูปที่ 4 ในบางกรณีสัญญาณร้องขอนี้อาจถูกเข้ารหัสร่วมไปกับข้อมูลบนช่องสัญญาณข้อมูลเลยก็เป็นได้



รูปที่ 4 การติดต่อสื่อสารเชิงอะซิงโครนัส (Hormdee, D. 2002)

กลไก Handshaking ของระบบอะซิงโครนัส

กลไกที่สำคัญกลไกหนึ่งของการสื่อสารระหว่างโมดูลต่าง ๆในระบบอะซิงโครนัสหนึ่ง ๆคือ Handshaking ซึ่งใช้เพื่อให้จังหวะในการถ่ายโอนข้อมูลระหว่างโมดูลอะซิงโครนัส โดยขั้นตอนการสื่อสารระหว่างโมดูลทั้งสองมีดังนี้

1. 'ผู้ส่ง'วางข้อมูลที่ต้องการส่งบนช่องสัญญาณข้อมูล
2. 'ผู้ส่ง'ให้สัญญาณแจ้ง'ผู้รับ'เพื่อเป็นการบอกว่า "ข้อมูลอยู่บนช่องสัญญาณข้อมูลแล้ว พร้อมที่จะนำข้อมูลไปใช้ได้เลย"
3. 'ผู้รับ'เข้าไปอ่านข้อมูลจากช่องสัญญาณข้อมูล แล้วนำไปประมวลผล
4. 'ผู้รับ'ให้สัญญาณแจ้ง'ผู้ส่ง'ข้อมูลว่า "เสร็จสิ้นการใช้ข้อมูลแล้ว"
5. 'ผู้ส่ง'รับสัญญาณดังกล่าวนั้นแล้วจึงปิดการสื่อสารโดยถอนข้อมูลออกจากช่องสัญญาณข้อมูลเพื่อทำงานอื่นต่อไป

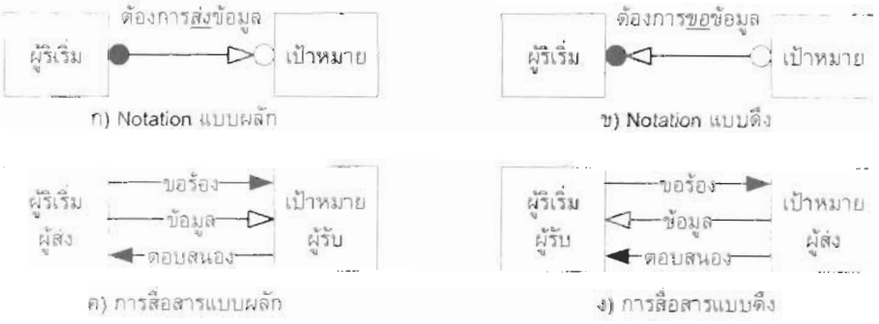
ก่อนหน้านี้ในการจำแนกประเภทของโมดูลนั้น จำแนกโดยอิงทิศทางการไหลของข้องมูล แต่ถ้าอาศัยทิศทางการควบคุมการทำงานมาจำแนก (ว่าโมดูลใดเป็นต้นกำเนิดของการสื่อสาร) จะได้ประเภทของโมดูลดังนี้

- *ผู้ริเริ่ม (initiator)* คือโมดูลที่ก่อให้เกิดการติดต่อสื่อสาร
- *เป้าหมาย (target)* คือโมดูลที่โต้ตอบกับ'ผู้ริเริ่ม'

การสื่อสารข้อมูลระหว่างโมดูลผ่านทางช่องสัญญาณนั้นทำได้โดยใช้สัญญาณ Handshake ซึ่งประกอบด้วย *สัญญาณร้องขอ (request signal)* และ *สัญญาณตอบสนอง (acknowledge signal)* โดย'ผู้ริเริ่ม'จะใช้'สัญญาณร้องขอ'นี้ในการเริ่มการสื่อสาร ส่วน'เป้าหมาย'จะใช้'สัญญาณตอบสนอง'ในการตอบโต้กับ 'ผู้ริเริ่ม'

ช่องสัญญาณนี้จะต่อเข้ากับโมดูลผ่านพอร์ต *(port)* โดยที่พอร์ตที่ต่อเข้ากับด้าน'ผู้ริเริ่ม'เรียกว่า *พอร์ตกระทำ (active port)* ซึ่งจะแทนด้วยรูปวงกลมทึบ ส่วนพอร์ตที่ต่อเข้ากับด้าน'เป้าหมาย'เรียกว่า *พอร์ตถูกกระทำ (passive port)* ซึ่งจะแทนด้วยรูปวงกลมกลวง ดังแสดงในรูปที่ 5(ก) และ 5(ข)

ความสัมพันธ์ของทิศทางการไหลของข้อมูล เมื่อเทียบกับทิศของการควบคุมเป็นสิ่งที่จำแนกประเภทของช่องสัญญาณของข้อมูลว่าเป็น*แบบผลัก (push)* เมื่อ'ผู้ส่ง'เป็นผู้ริเริ่มการสื่อสาร และในทางตรงข้ามว่าเป็น*แบบดึง (pull)* เมื่อ'ผู้รับ'เป็น'ผู้ริเริ่ม'การสื่อสาร **รูปที่ 5(ค)** และ 5(ง) แสดงถึงประเภทของช่องสัญญาณข้อมูลทั้งสองประเภท



รูปที่ 5 *ประเภทของการสื่อสารแบบผลักและแบบดึง (Janin, 2004)*

เกณฑ์วิธีการให้สัญญาณ handshake (handshake signaling protocol) อาจอยู่ในรูปแบบใดรูปแบบหนึ่ง ดังต่อไปนี้

การให้สัญญาณแบบ 2 ขั้นตอน (2-phase transition signaling)

การให้สัญญาณรูปแบบนี้ ความสำคัญทางสัญญาณจะไม่ขึ้นกับระดับของสัญญาณ นั่นคือ*ขอบขาขึ้น (rising edge)* และ*ขอบขาลง (falling edge)* ของสัญญาณมีผลเหมือนกัน รูปที่ 6 แสดงการให้สัญญาณแบบ 2 ขั้นตอนของการสื่อสารทั้งแบบผลักและแบบดึง โดยที่ลูกศรคำเส้นทึบ แสดงถึงกิจกรรมที่ก่อให้เกิดการทำงาน ในขณะที่ลูกศรโปร่งเส้นประ แสดงถึงช่วงเวลาการประมวลผล

โดยในการติดต่อสื่อสารแบบผลัก การทำงานของแต่ละการติดต่อมีขั้นตอนเหมือนกับกลไก handshake พื้นฐานที่ได้กล่าวมาข้างต้น ดังนี้

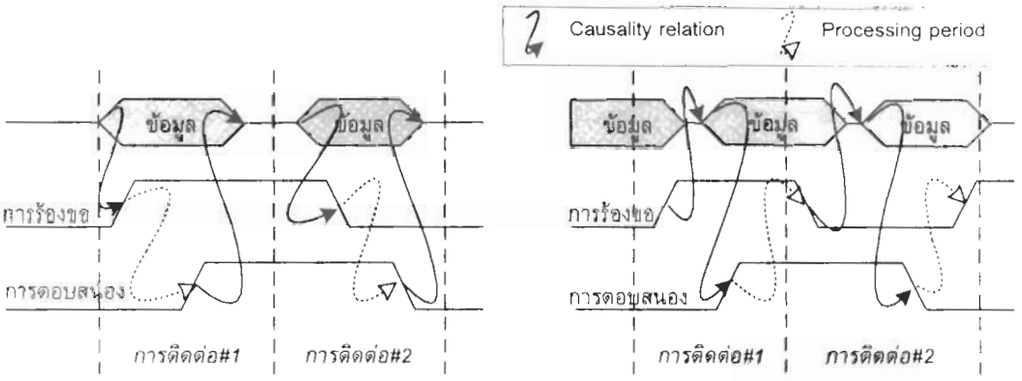
- 'ผู้ส่ง'วางข้อมูลบนช่องสัญญาณ
- 'ผู้ส่ง'ให้'สัญญาณร้องขอ'ไปยัง'ผู้รับ' ⇒ ขั้นที่ 1
- 'ผู้รับ'เข้าไปอ่านข้อมูลจากช่องสัญญาณ
- 'ผู้รับ'ให้'สัญญาณตอบสนอง'กลับไปยัง'ผู้ส่ง' ⇒ ขั้นที่ 2
- 'ผู้ส่ง'ปิดการสื่อสารโดยถอนข้อมูลออกจากช่องสัญญาณข้อมูล

ในขณะที่ในการติดต่อสื่อสารแบบดึง การร้องขอจะออกมาจาก'ผู้รับ' โดยมีขั้นตอนดังนี้

- 'ผู้รับ'ให้'สัญญาณร้องขอ'ข้อมูลจาก'ผู้ส่ง' ⇒ ขั้นที่ 1
- 'ผู้ส่ง'วางข้อมูลบนช่องสัญญาณ

ในขณะเดียวกัน

- 'ผู้ส่ง'ให้'สัญญาณตอบสนอง'การร้องขอข้อมูลจาก'ผู้รับ' ⇒ ขั้นที่ 2
- 'ผู้รับ'อ่านข้อมูลจากช่องสัญญาณ
- 'ผู้ส่ง'ปิดการสื่อสารโดยถอนข้อมูลออกจากช่องสัญญาณข้อมูล



ก) การให้สัญญาณแบบผลัก ข) การให้สัญญาณแบบดึง

รูปที่ 6 ***การให้สัญญาณ*** *handshake แบบ 2 ขั้นตอน (2-phase transition signaling)*

การให้สัญญาณแบบ 4 ขั้นตอน (4-phase level signaling)

การให้สัญญาณรูปแบบนี้ ความสำคัญทางสัญญาณจะขึ้นกับระดับของสัญญาณ นั่นคือขอบขาขึ้นและขอบขาลงของสัญญาณก่อให้เกิดผลไม่เหมือนกัน รูปที่ 7 แสดงการให้สัญญาณแบบ 4 ขั้นตอนของการสื่อสารทั้งแบบผลักและแบบดึง

โดยในการติดต่อสื่อสารแบบผลัก การทำงานของแต่ละการติดต่อ มีดังนี้

- 'ผู้ส่ง'วางข้อมูลบนช่องสัญญาณ
- 'ผู้ส่ง'ให้'สัญญาณร้องขอ'ไปยัง'ผู้รับ' ⇒ ขั้นที่ 1
- 'ผู้รับ'เข้าไปอ่านข้อมูลจากช่องสัญญาณ
- 'ผู้รับ'ให้'สัญญาณตอบสนอง'กลับไปยัง'ผู้ส่ง' ⇒ ขั้นที่ 2
- 'ผู้ส่ง'ถอน'สัญญาณร้องขอ'และถอนข้อมูลออกจากช่องสัญญาณข้อมูล ⇒ ขั้นที่ 3
- 'ผู้รับ'ปิดการสื่อสารโดยถอน'สัญญาณตอบสนอง' ⇒ ขั้นที่ 4

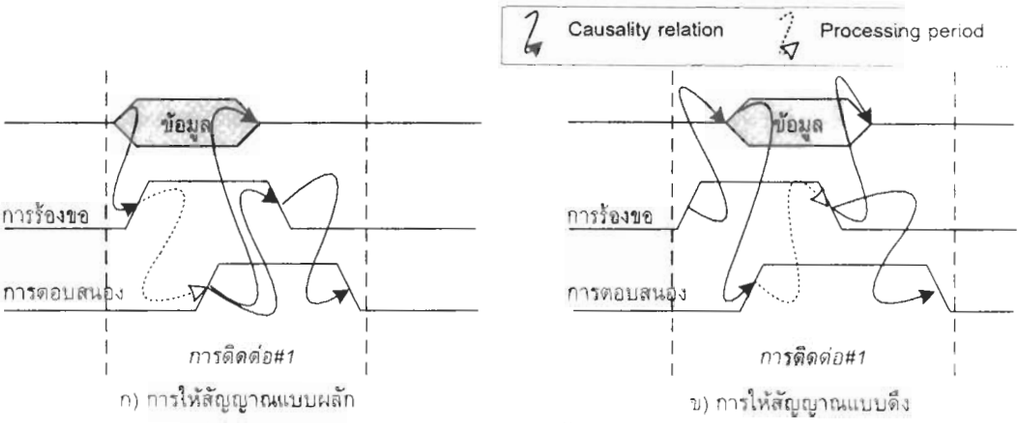
ในขณะที่ในการติดต่อสื่อสารแบบดึง การร้องขอจะออกมาจาก'ผู้รับ' โดยมีขั้นตอนดังนี้

- 'ผู้รับ'ให้'สัญญาณร้องขอ'ข้อมูลจาก'ผู้ส่ง' ⇒ ขั้นที่ 1
- 'ผู้ส่ง'วางข้อมูลบนช่องสัญญาณ

ในขณะเดียวกัน

- 'ผู้ส่ง'ให้'สัญญาณตอบสนอง'การร้องขอข้อมูลจาก'ผู้รับ' ⇒ ขั้นที่ 2

- 'ผู้รับ'อ่านข้อมูลจากช่องสัญญาณ
- 'ผู้รับ'ปิดการสื่อสารโดยถอน'สัญญาณร้องขอ' ⇒ ขั้นที่ 3
- 'ผู้ส่ง'ถอน'สัญญาณตอบสนอง'และถอนข้อมูลออกจากช่องสัญญาณข้อมูล ⇒ ขั้นที่ 4



รูปที่ 7 การให้สัญญาณ handshake แบบ 4 ขั้นตอน (4-phase level signaling)

โดยการให้สัญญาณ handshake ทั้ง 2 ประเภทนี้ต่างก็มีข้อได้เปรียบเสียเปรียบกันดังนี้ การให้สัญญาณแบบ 4 ขั้นตอน จะมีการเปลี่ยนแปลงของสัญญาณมากกว่าเป็น 2 เท่าของการให้สัญญาณแบบ 2 ขั้นตอนในการติดต่อสื่อสารจำนวนเท่ากัน เนื่องจากการให้สัญญาณแบบ 4 ขั้นตอนนี้ จะมีการกลับมาสู่สถานะศูนย์หรือสถานะเริ่มต้นของสัญญาณ *(Return-To-Zero: RTZ)* ซึ่งทำให้การให้สัญญาณประเภทนี้เป็นที่นิยมสืบเนื่องมาจากคุณสมบัติ RTZ จะทำให้วงจรควบคุมเพื่อออกแบบง่ายขึ้น ในทางตรงกันข้ามการให้สัญญาณแบบ 2 ขั้นตอนนั้น ถึงแม้ว่าวงจรควบคุมในส่วนการออกแบบวิธีนี้จะซับซ้อนกว่า แต่ข้อเด่นของวิธีนี้คือ ประสิทธิภาพในการทำงานที่ดีกว่า รวมไปถึงการใช้พลังงานในการควบคุมที่ต่ำกว่าในการติดต่อสื่อสารครั้งหนึ่งๆ เนื่องจากมีจำนวนการเปลี่ยนแปลงของสัญญาณที่น้อยกว่ากึ่งหนึ่ง

ข้อดีของระบบอะซิงโครนัส

เหตุผลที่ผลักดันให้นักออกแบบทั้งหลายยังคงมีความสนใจในการออกแบบเชิงอะซิงโครนัส นั่นก็คือ ศักยภาพของข้อได้เปรียบต่าง ๆ ที่การออกแบบนี้สามารถให้ได้เมื่อเปรียบเทียบกับการออกแบบเชิงซิงโครนัส ซึ่งสามารถสรุปได้ดังนี้

การออกแบบในลักษณะ plug-and-play

ในการออกแบบระบบอะซิงโครนัสนั้นสามารถทำได้โดยการออกแบบโมดูลหรือส่วนย่อย ซึ่งมีการจัดการต่างๆ สำเร็จรูปภายในทีละโมดูล เมื่อต้องการเชื่อมประสานส่วนต่างๆเข้าด้วยกันสามารถทำได้เพียงเพิ่มเติมการติดต่อสื่อสารเชิง handshaking ที่เกิดขึ้นระหว่างโมดูล ลักษณะโครงสร้างแบบนี้เมื่อโมดูลย่อยต่างๆ ไม่ขึ้นอยู่กับซึ่งกันและกันทำให้การออกแบบระบบที่มีขนาดใหญ่ การจำลองการทำงาน

การทดสอบการทำงาน รวมไปถึงการที่ต้องออกแบบโมดูลใหม่เพื่อใช้งานแทนโมดูลใดๆ ทำได้ง่ายขึ้น เนื่องจากการกระทำดังกล่าวสามารถเกิดขึ้นได้ ณ ส่วนใดส่วนหนึ่งซึ่งแยกจากระบบใหญ่ได้เลย หรือจะกระทำทั้งระบบพร้อมกันก็ได้

การหลีกเลี่ยงปัญหาการบิดเบือนของนาฬิกา

ถึงแม้ว่าการออกแบบเชิงซิงโครนัสจะมีข้อดีต่าง ๆ ที่ทำให้ระบบนี้ถูกใช้อย่างแพร่หลาย เช่น ความไม่ซับซ้อนในการออกแบบเนื่องจากการมีนาฬิการะบบ รวมไปถึงสิ่งต่างๆ ที่อำนวยความสะดวกในการออกแบบและเรียนรู้ระบบดังกล่าวที่มีอยู่มากมายในท้องตลาด อย่างไรก็ตามการพัฒนาระบบที่ประกอบด้วยทรานซิสเตอร์ (transistors) นับล้านๆ ตัว การทำให้ทุกส่วนในระบบเข้าจังหวะกันเป็นสิ่งที่มีความลำบากขึ้นเรื่อยๆ *clock skew* หรือ *การบิดเบือนของนาฬิกา* (Weste, N.H.E. and Eshraghian, K. 1993) คือนิยามที่ใช้อธิบายปรากฏการณ์ที่เกิดขึ้นเนื่องจาก *เวลาแพร่กระจาย (propagation time)* ซึ่งทำให้มีเกิดความต่างระหว่างขอบของสัญญาณนาฬิกาที่ไปถึงยังส่วนต่าง ๆ ของระบบ เมื่อระบบมีขนาดใหญ่ขึ้น การรับประกันการออกแบบเพื่อให้ได้มาซึ่งระบบที่ปลอดการบิดเบือนของนาฬิการะบบ ณ ส่วนต่างๆของระบบเป็นเรื่องที่ทำได้ยากขึ้น

ระบบอะซิงโครนัสจะมีความต้านทาน (tolerant) ต่อการบิดเบือนของสัญญาณที่ใช้ในการควบคุมการทำงานของระบบดีกว่า เนื่องจากสัญญาณเหล่านั้นมีเป็นเฉพาะแห่งๆไป

ประสิทธิภาพที่ดีขึ้นโดยเฉลี่ย

เพื่อทำให้การทำงานต่างๆ ในระบบซิงโครนัสเป็นไปในจังหวะเดียวกันความเร็วของระบบนี้จึงมักถูกกำหนดโดยความเร็วของการทำงานของโมดูลที่ช้าที่สุด ในขณะที่โมดูลต่างๆในระบบอะซิงโครนัสสามารถทำงานในความเร็วที่แต่ละโมดูลสามารถทำได้ โดยอาจมีการชะลอกันเกิดขึ้นเป็นครั้งคราวในกรณีที่ต้องรอการทำงานของโมดูลอื่น ทำให้โดยเฉลี่ยแล้วประสิทธิภาพการทำงานในระบบอะซิงโครนัสดีกว่าระบบซิงโครนัส

การใช้พลังงานที่ลดลง

เนื่องด้วยสัญญาณนาฬิการะบบในระบบซิงโครนัสต้องทำงานอยู่ตลอด ในการนี้จำเป็นต้องใช้พลังงานเป็นจำนวนมากในการกระจายสัญญาณนาฬิกานี้ไปยังทุกส่วนของระบบ ฉะนั้นการที่เกิดการสื่อสารเฉพาะส่วนที่มีความจำเป็นในเวลาหนึ่งๆนั้นทำให้ระบบอะซิงโครนัสมีการใช้พลังงานในส่วนการควบคุมและให้จังหวะที่มีประสิทธิภาพดีกว่า (van Gageldonk, H. 1998) ทางแก้ปัญหาที่นักออกแบบระบบซิงโครนัสที่ตระหนักถึงประสิทธิภาพของการใช้พลังงานวิธีหนึ่งที่เป็นที่นิยมคือการนำ *Clock Gating* มาประยุกต์ใช้เพื่อบริหารจัดการการใช้พลังงานอันเนื่องมาจากระบบนาฬิการวม

การรบกวนเชิงแม่เหล็กไฟฟ้า

ในระบบซิงโครนัสมีนาฬิการะบบที่คอยทำให้ทุกกิจกรรมทำงานพร้อมกันอย่างเป็นจังหวะ ทำให้เกิดการเปลี่ยนแปลงความเข้มของสนามแม่เหล็กและไฟฟ้าขึ้นพร้อมๆ กัน ในทางตรงกันข้ามกิจกรรมต่าง ๆ ในระบบอะซิงโครนัสจะกระจายออกไปตามช่วงเวลาต่าง ๆ ทำให้*การรบกวนเชิงแม่เหล็กไฟฟ้า*

(Electro Magnetic Interference: EMI) นั้นกระจายไปในทุก ๆ ส่วนของเวลาด้วยแอมพลิจูด (amplitude) ที่ต่ำลง

อุปสรรคของการออกแบบระบบอะซิงโครนัส

ถึงแม้ว่าระบบอะซิงโครนัสมีศักยภาพและข้อได้เปรียบในหลายด้าน อย่างไรก็ตามยังมีอุปสรรคอยู่จำนวนหนึ่งที่ทำให้การออกแบบวิธีนี้ไม่ง่ายอย่างที่นักออกแบบพัฒนาระบบหลายท่านหวังไว้ อุปสรรคเหล่านั้นได้แก่

ความซับซ้อนของตรรกะควบคุม

ระบบที่สามารถทำงานได้โดยปราศจากนาฬิการะบบนำมาซึ่งข้อดีหลายประการอย่างที่กล่าวมาข้างต้น แต่เนื่องจากหน้าที่ของนาฬิการะบบนี้คือ ควบคุมการทำงานของส่วนต่างๆ ของทั้งระบบ ซึ่งเป็นการอำนวยความสะดวกให้นักพัฒนาระบบสามารถออกแบบระบบได้โดยไม่ยุ่งยากนัก เมื่อไม่มีสัญญาณนาฬิกานี้แล้ว นักพัฒนาจำเป็นต้องออกแบบฮาร์ดแวร์ที่จะมาทำหน้าที่ควบคุมการให้จังหวะการทำงานแก่ส่วนต่างๆ เหล่านั้นเอง ซึ่งแน่นอนอยู่แล้วว่า *ตรรกะควบคุม (control logic)* เพื่อการนี้ในระบบอะซิงโครนัสจะมีความซับซ้อนกว่าการใช้สัญญาณนาฬิการะบบในการออกแบบเชิงซิงโครนัส

ความสัมพันธ์เชิงเวลาของส่วนต่าง ๆ ในระบบ

ในระบบซิงโครนัสนั้น ตำแหน่งหรือลำดับต่างๆ ของขั้นตอนการทำงานจะถูกควบคุมโดยนาฬิการะบบ ผลที่ได้คือ การได้มาซึ่งตำแหน่งของขั้นตอนการทำงานของโมดูลหนึ่งๆ นั้นสามารถทำได้โดยง่ายว่าอยู่ที่ช่วงคาบเวลาใด ในทำนองกลับกัน เนื่องจากการทำงานของแต่ละโมดูลย่อยในระบบอะซิงโครนัสควรที่จะดำเนินไปอย่างอิสระที่สุด โดยพยายามลดการเข้าจังหวะกันให้น้อยที่สุด เนื่องจากเป็นส่วนที่เสียเวลาด้วยเหตุที่โมดูลเหล่านั้นต้องรอกันเพื่อทำงานในขั้นตอนต่อไป ผลที่ได้จากการเป็นอิสระต่อกันของโมดูลต่างๆนั้นคือ การได้มาซึ่งตำแหน่งการทำงานของโมดูลหนึ่งๆในเชิงของคาบเวลาจะทำได้ไม่ง่ายนัก

ความลำบากในการตรวจสอบการทำงานของระบบ

การตรวจสอบการทำงานของการออกแบบระบบ (Design validation) เป็นขั้นตอนที่มีความสำคัญอย่างยิ่งยวดเพื่อตรวจสอบและแก้ไขข้อผิดพลาดต่างๆที่อาจเกิดขึ้นในการออกแบบระบบ แต่เนื่องด้วยคุณสมบัติที่ไม่แน่นอน *(non-deterministic)* ที่สืบทอดมาของระบบอะซิงโครนัสอันเนื่องมาจากขบวนการการตัดสิน *(arbitration)* (Seitz, C. 1980) ของระบบอะซิงโครนัส ซึ่งในหลายครั้งระบบจะอยู่ในสถานการณ์ที่ต้องเลือกว่าจะดำเนินการในส่วนใดต่อไปก่อน ทำให้ลำดับการทำงานของส่วนต่างๆในระบบอาจแตกต่างกันในการทำงานแต่ละครั้ง มีผลให้สำหรับลำดับของ input ชุดหนึ่งๆ อาจจะมีลำดับของ output ที่ได้มาอย่างถูกต้องจากระบบได้หลายชุด นอกเหนือจากนั้นคือระบบอะซิงโครนัสมีแนวโน้มค่อนข้างสูงที่จะมีจำนวนสถานการณ์ทำงาน (state) ในระบบมากกว่าในระบบซิงโครนัสซึ่งทำให้การตรวจสอบการทำงานต้องครอบคลุมจำนวนการทดสอบที่มากขึ้นนี้

ความขาดแคลนเครื่องมือที่ดี

ในการผลักดันให้ความนิยมในการออกแบบเชิงอะซิงโครนัสสามารถแข่งขันกับการออกแบบระบบนาฬิกาในได้นั้น *EDA (Electronic Design Automation) tools*[1] และหลักการและวิธีการออกแบบต่างๆ ที่อำนวยความสะดวกในการออกแบบระบบนี้ต้องได้รับการพัฒนาอย่างเร่งด่วน นับถึงปัจจุบันการออกแบบพัฒนาระบบอะซิงโครนัส โดยส่วนมากนักวิจัยและนักออกแบบต้องอาศัยประสบการณ์อย่างมากในการออกแบบสิ่งต่างๆ 'ด้วยมือ' (by hand) ซึ่งทำให้ต้องมีความพยายามและระยะเวลาที่ใช้ในการพัฒนาระบบหนึ่งๆ ค่อนข้างสูง เนื่องด้วยเครื่องมือในการออกแบบระบบอะซิงโครนัสนี้มีค่อนข้างจำกัดในเชิงของจำนวน ประสิทธิภาพ และกลุ่มผู้ใช้ อย่างไรก็ตามกลุ่มวิจัยและองค์กรทางธุรกิจหลายแห่งทั่วโลกได้เล็งเห็นถึงอุปสรรคนี้ และพยายามในการแก้ไขปัญหานี้อยู่ เครื่องมือต่างๆที่มีให้เห็นในปัจจุบันได้แก่ Petrify (Cortadella, J. et al. 1997) Minimalist (Fuhrer, R. M. et al. 1999) Balsa (Bardsley, A. 2000) Tangram (van Berkel, K. et al. 1991) LARD (LARD Home Page) ฯลฯ

ความไม่คุ้นเคย

ประเด็นนี้อาจเป็นประเด็นอุปสรรคที่ยากที่สุดในการฟันฝ่า เนื่องด้วยวิถีชีวิตที่เราคุ้นเคยแต่กับระบบซิงโครนัสรวมไปถึงการเรียนการสอนที่วางพื้นแนวความคิดเกี่ยวกับการออกแบบวิธีเดิมๆนี้ จึงเป็นการยากที่จะเปลี่ยนแปลงแนวปฏิบัติของคนทั่วไปให้มาสนใจวิธีการออกแบบระบบเชิงอะซิงโครนัสนี้

สรุป

บทความนี้เป็นการแนะนำถึงการออกแบบระบบตรรกะอีกประเภทหนึ่งที่เรียกว่า Asynchronous Logic Design โดยมีความแตกต่างจากระบบซิงโครนัสเดิมที่เป็นที่นิยมกันทั่วไป คือการมีเกณฑ์วิธีการให้สัญญาณ handshake แทนการใช้นาฬิการะบบในการให้จังหวะแก่ส่วนต่างๆของระบบ

จากนั้นเป็นการสรุปถึงข้อดีและอุปสรรคของการออกแบบเชิงอะซิงโครนัสนี้โดยเปรียบเทียบกับวิธีการออกแบบระบบตรรกะทั่วๆไปที่ใช้นาฬิกาในการควบคุมการทำงาน

ถึงแม้ว่าระบบที่ปราศจากสัญญาณนาฬิกาควบคุมนี้ดูเหมือนจะมีความลำบากและซับซ้อนในการออกแบบอยู่บ้าง อย่างไรก็ตามผู้เขียนต้องการที่จะให้ข้อคิดว่าอุปสรรคต่างๆเหล่านี้จะเป็นปัญหาให้กับนักออกแบบและพัฒนาระบบเท่านั้น ในทางกลับกับบุคคลอื่นๆที่จะนำระบบนี้ไปใช้งานจะเผชิญเฉพาะจุดเด่นต่างๆที่ระบบประเภทนี้จะพึงเสนอให้เท่านั้น

---

[1] เครื่องมือเชิงซอฟต์แวร์ในการพัฒนาระบบวงจรรวม (Integrated Circuit)

กิตติกรรมประกาศ

สาระของบทความนี้ สรุปจากประสบการณ์ที่ได้รับจากการทำงานที่ AMULET Group ณ the University of Manchester สหราชอาณาจักร ผู้เขียนจึงขอขอบคุณคณาจารย์และเพื่อนๆนักวิจัยทุกท่านที่ได้ช่วยเปิดโลกทัศน์แห่งงานวิจัยแขนงนี้

เอกสารอ้างอิง

Bardsley, A, 2000. **Implementing Balsa Handshaking Circuits**, *Ph.D. Thesis*, Department of Computer Science, The University of Manchester, UK.

Bibliography, **Bibliography on asynchronous circuits**, *http://liinwww.ira.uka.de/bibliography/Misc/async.circuits.html*

Cortadella, J. et al. 1997. "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers", in **Proceedings of IEICE Transactions on Information and Systems**, E80-D(3), pp. 315-325.

Fuhrer, R. M. et al. 1999. **Minimalist: An environment for the synthesis, verification and testability of burst-mode asynchronous machines**, *Technical Report TR CUCS-020-99*, Columbia University, New York.

Homepage, **The Asynchronous Logic Home Page**, *http://www.cs.man.ac.uk/async/*

Hormdee, D. 2002. **Copy-Back Cache Organization for an Asynchronous Microprocessor**. *Ph.D. Thesis*, Department of Computer Science, The University of Manchester, 2002.

Janin, L. 2004. **Simulation and Visualisation for Debugging Large Scale Asynchronous Handshake Circuits**. *Ph.D. Thesis*, Department of Computer Science, The University of Manchester, 2004.

LARD Home Page, **LARD Documentation Home Page**, *http://www.cs.manchester.ac.uk/apt/projects/tools/lard/*

Seitz, C. 1980. "System Timing", **Chapter 7 of Introduction to VLSI Systems**, Addison Wesley, Second Edition.

van Berkel, K. et al. 1991. "The VLSI-Programming Language Tangram and Its Translation into Handshake Circuits", **Proceedings of European Conference on Design Automation**, pp. 384-389.

van Gageldonk, H. 1998. **An Asynchronous Low-Power 80C51 Microcontroller**, *Thesis*, Eindhoven University of Technology, The Netherlands.

Weste, N.H.E. and Eshraghian, K. 1993. **Principles of CMOS VLSI Design: A Systems Perspective**, Addison Wesley, Second Edition.