หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การวิเคราะห์การทำงานของวงจรตรรกะ

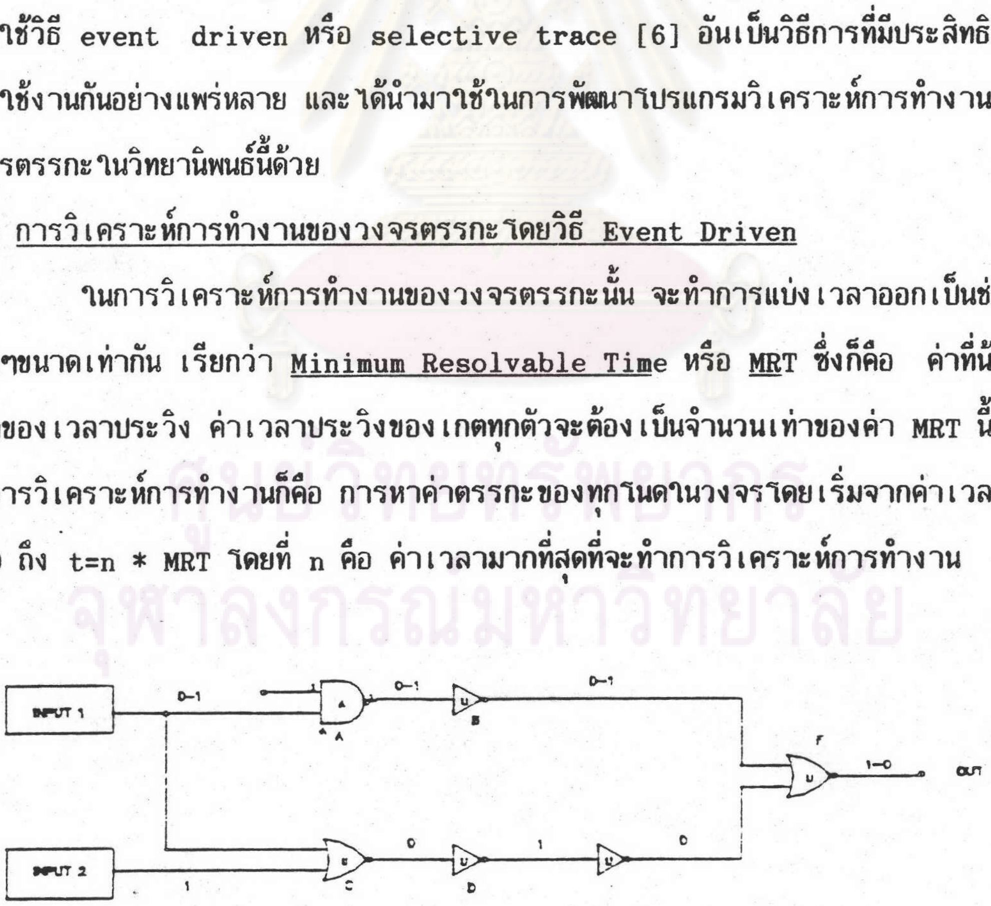
3.1 บทนำ

เราจะนิยามการวิเคราะห์การทำงานของวงจรตรรกะว่า เป็นการสร้างแบบจำลองของเกตต่างๆในวงจรขึ้น และใช้แบบจำลองหาค่าของสัญญาณ ณ จุดต่างๆในวงจร โดยขึ้นกับเวลา และสัญญาณที่เข้ามาสู่วงจร เทคนิคที่ใช้ในการวิเคราะห์วงจรตรรกะนั้น จะต้องมีความสามารถเลียนแบบการทำงานของวงจรตรรกะ ดังที่ได้กล่าวไปในบทที่ 2 ได้อย่างใกล้เคียงและมีประสิทธิภาพมากที่สุด ในบทนี้เราจะกล่าวถึงการวิเคราะห์การทำงานของวงจรตรรกะโดยใช้วิธี event driven หรือ selective trace [6] อันเป็นวิธีการที่มีประสิทธิภาพสูงและใช้งานกันอย่างแพร่หลาย และได้นำมาใช้ในการพัฒนาโปรแกรมวิเคราะห์การทำงานของวงจรตรรกะในวิทยานิพนธ์นี้ด้วย

3.2 การวิเคราะห์การทำงานของวงจรตรรกะโดยวิธี Event Driven

ในการวิเคราะห์การทำงานของวงจรตรรกะนั้น จะทำการแบ่งเวลาออกเป็นช่วงเล็กๆขนาดเท่ากัน เรียกว่า Minimum Resolvable Time หรือ MRT ซึ่งก็คือ ค่าที่น้อยที่สุดของเวลาประวิง ค่าเวลาประวิงของเกตทุกตัวจะต้องเป็นจำนวนเท่าของค่า MRT นี้ ในการวิเคราะห์การทำงานก็คือ การหาค่าตรรกะของทุกโนดในวงจรโดยเริ่มจากค่าเวลา t=0 ถึง t=n * MRT โดยที่ n คือ ค่าเวลามากที่สุดที่จะทำการวิเคราะห์การทำงาน

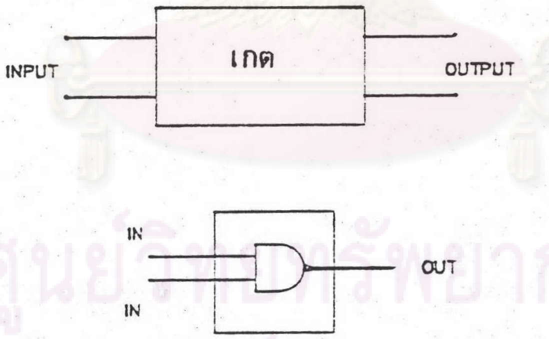
ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1 แสดงการกระจายของการเปลี่ยนค่าตรรกะในวงจรตรรกะ

จากรูปที่ 3.1 จะสามารถสังเกตลักษณะบางประการของวงจรตรรกะได้ กล่าวคือเมื่อไม่มีการเปลี่ยนแปลงที่แหล่งกำเนิดสัญญาณ INPUT1 และ INPUT2 วงจรตรรกะจะอยู่ในสภาวะสงบ ไม่มีการเปลี่ยนแปลงใดๆ หาก INPUT1 มีการเปลี่ยนค่าจาก 0 เป็น 1 พบว่ามีเพียง เกต A, B และ F เท่านั้นที่มีการเปลี่ยนแปลงค่าสัญญาณทางด้านออก ส่วนเกต C, D และ E จะยังคงสถานะเดิมไว้ ดังนั้น ในการหาค่าตรรกะของวงจร จึงไม่จำเป็นต้องทำการหาค่าตรรกะทางด้านออกของเกตทุกตัวในวงจร หากทำการหาค่าตรรกะเฉพาะเกตที่มีการเปลี่ยนแปลงทางด้านเข้าเท่านั้น วิธีการที่อาศัยหลักการนี้เรียกว่า Event Driven หรือ Selective Trace นั่นเอง

ในวงจรที่จะทำการวิเคราะห์นี้จะประกอบไปด้วยเกตและโนด โดยที่เกตประกอบไปด้วยขั้วเข้า (INPUT) และขั้วออก (OUTPUT) ซึ่งค่าตรรกะทางด้านออกจะแปรตามค่าตรรกะเข้าในเกตแบบจัดหมู่ และ ขึ้นกับสถานะ (State) ของเกตในกรณีเกตแบบลำดับ เช่น ฟลิปฟลอป เป็นต้น (รูปที่ 3.2)

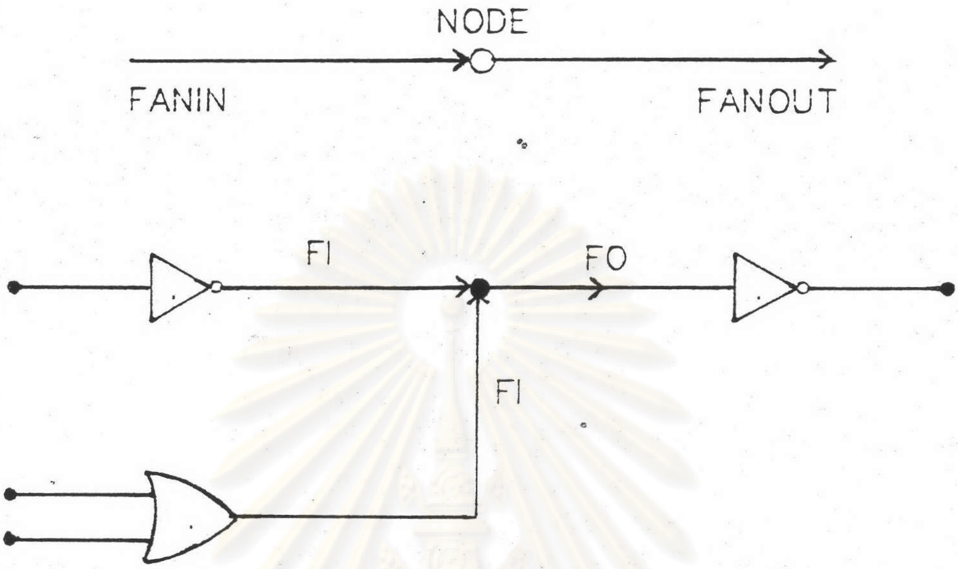


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.2 เกตในวงจรตรรกะ

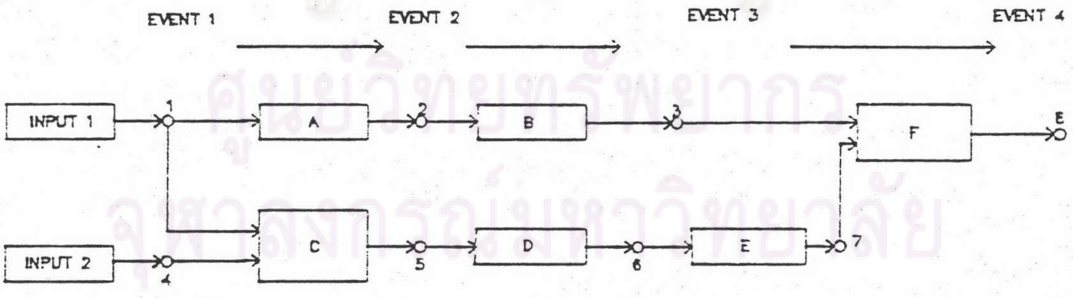
ขั้วเข้าของเกตจะมีหน้าที่สุ่มค่าสัญญาณที่โนดที่ต่ออยู่ ส่วนขั้วออกจะเป็นตัวกำหนดสถานะของโนดที่เกตนั้นกำลังขับอยู่

โนดนั้นจะมี fan in และ fan out ซึ่ง fan in หมายถึง จุดที่โนดมีการต่อกับขั้วออกของเกต ส่วน fan out หมายถึง จุดที่โนดมีการต่อไปยังขั้วเข้าของเกต ดังแสดงไว้ในรูปที่ 3.3



รูปที่ 3.3 แสดง fan in และ fan out

ในการวิเคราะห์การทำงานจะแบ่งเวลาออกเป็นช่วงเล็กๆ เรียกว่า MRT ดังที่กล่าวไปแล้ว ในแต่ละ MRT ของการวิเคราะห์การทำงาน จะมีการตรวจสอบว่ามี event เกิดขึ้นหรือไม่ event หมายถึง การเปลี่ยนแปลงของค่าตรรกะที่โนดใดโนดหนึ่ง



รูปที่ 3.4 แสดงการกระจายของ event

จากรูปที่ 3.4 เมื่อมี event เกิดที่โนดใดก็ตาม จะมีการตรวจสอบเกตทุกตัวที่ fan out ของโนดนั้นขับอยู่ ถ้าพบว่าสถานะทางด้านออกของเกตมีการเปลี่ยนแปลง แสดงว่ามี event ใหม่เกิดขึ้นที่โนดนั้น จะมีการบันทึกโนดที่เปลี่ยนแปลงไว้ เพื่อจะทำการวิเคราะห์การทำงานต่อไป ส่วนโนดที่ขับอยู่จะถูกเปลี่ยนค่าตรรกะเป็นค่าใหม่ การที่โนดที่มี event เกิดขึ้นใหม่ยังไม่เปลี่ยนสถานะทันทีเป็นเพราะในเกตทุกชนิดจะมีเวลาประวิงอยู่ ขั้นตอนในการจัดการกับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การบันทึก event และเวลาประวิงจะใช้เทคนิคซึ่งจะกล่าวถึงในตอนต่อไป

โดยสรุปแล้ว การวิเคราะห์การทำงานของวงจรตรรกะโดยใช้วิธี event driven จะเป็นไปตามขั้นตอนดังนี้

<u>Event driven simulation algorithm:</u>

```
set time to 0;
while (time <= maxtime)
begin
    for (all events at that time)
    begin
        modify logic stat at event node;
        for (all fan-out of node)
        begin
            evaluate gate at fan-out;
            if (new event at output)
                schedule event at time + timedelay;
        end; (* all fan out *)
    end; (* all event at that time *)
    goto next time step;
end;
```

เกตใดๆก็ตามจะเกิดการเปลี่ยนแปลงค่าตรรกะที่ขั้วออกก็ต่อเมื่อมีเหตุการณ์หรือ event เกิดขึ้นที่ขั้วเข้า เกตทุกตัวซึ่งเป็น fan out ของโนดที่มีการเปลี่ยนแปลงนั้นจะเรียกว่า potentially active [9] หรือมีแนวโน้มที่จะเปลี่ยนค่าตรรกะ แต่ค่าตรรกะออกนั้นจะมีการเปลี่ยนแปลงจริงหรือไม่ จะต้องรอผลจากการหาค่าตรรกะของเกตนั้นก่อน ในวงจรขนาดใหญ่ในแต่ละ MRT ที่ทำการหาค่าตรรกะ จะมีเกตที่จะมีการเปลี่ยนแปลงประมาณ 2-10% เท่านั้น [7]

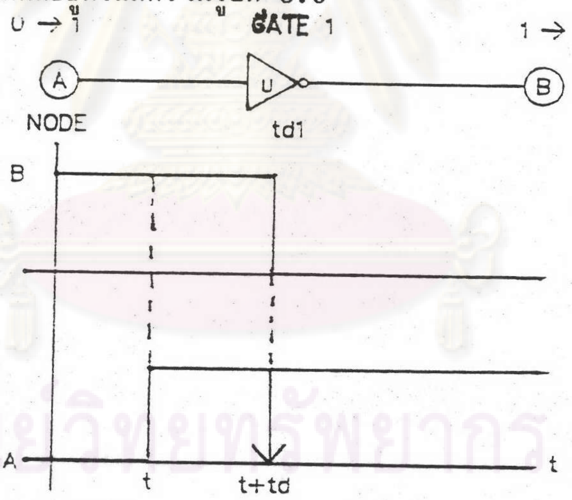
ดังนั้นวิธีการนี้จะทำให้การวิเคราะห์การทำงานเป็นไปได้อย่างรวดเร็วเนื่องจากต้องทำการหาค่าตรรกะของเกตในวงจรจำนวนน้อยลง ซึ่งจะแสดงไว้ในบทที่ 7 ซึ่งวิเคราะห์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การทำงานของโปรแกรม นอกจากนี้แล้ววิธีการนี้ยังเอื้ออำนวยให้จัดการกับเวลาประวิงของวงจรตรรกะได้อย่างมีประสิทธิภาพ โดยอาศัยเทคนิคที่เรียกว่า Time queue [6] หรือ Time wheel [7]

3.3 <u>การทำงานของ time queue</u>

ในการวิเคราะห์การทำงานของวงจรตรรกะโดยใช้วิธี event driven นี้ทุกครั้งที่เกิด event คือมีการเปลี่ยนแปลงค่าตรรกะที่โนดใดๆ จะต้องมีการตรวจสอบ fan out ของโนดนั้นว่าขับเกตใดบ้าง แล้วหาค่าตรรกะทางด้านออกของเกตนั้น กรณีที่มีการเปลี่ยนค่าตรรกะด้านออกเกิดขึ้นพบว่า ค่าตรรกะที่โนดทางด้านออกของเกตในขณะนั้นจะยังไม่เปลี่ยนค่าทันที แต่จะเปลี่ยนค่าเมื่อเวลาผ่านไป td โดยที่ td คือ ค่าของเวลาประวิงแผ่กระจายของเกตที่ขับโนดนั้นอยู่ดังแสดงในรูปที่ 3.5

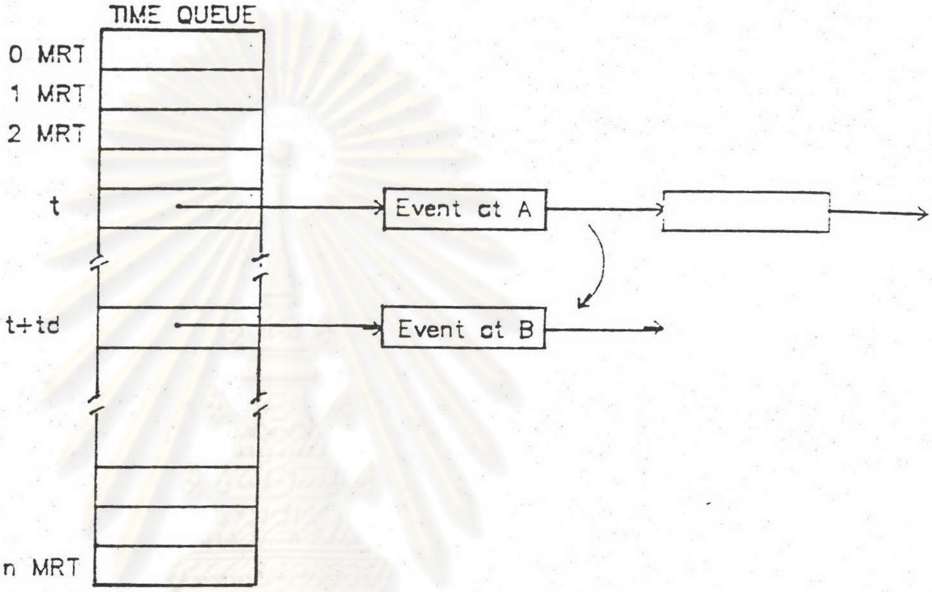


รูปที่ 3.5 แสดง propagation delay time

ค่าเวลาประวิงนี้เป็นค่าขึ้นกับคุณสมบัติของเกตแต่ละชนิด และมีค่าต่างกันได้มาก การจัดการจะใช้เทคนิค time queue เข้ามาช่วย โดยจะอธิบายได้ดังนี้

กำหนดให้ time queue เป็น array ขนาด n โดย n คือค่าเวลามากที่สุดที่จะทำการวิเคราะห์แต่ละจุดของ time queue จะหมายถึงเวลาเริ่มจาก 0 ถึง n MRT เมื่อมีการเปลี่ยนแปลงหรือ event ที่เกิดขึ้นที่โนดใดก็ตาม จะถูกบันทึกลงไปใน time queue ที่เวลาที่ถูกต้อง โปรแกรมจะตรวจสอบและจัดการ time queue โดยเริ่มจากเวลาที่ 0 ไปเรื่อยๆ เช่น ในรูป 3.5 และ 3.6 เมื่อทำการวิเคราะห์มาถึงเวลาที่ t พบว่ามีการเปลี่ยนค่าตรรกะที่โนด A จาก 0 เป็น 1 เนื่องจากเกต 1 มีค่าเวลาประวิงเท่ากับ td ดังนั้นค่าตรรกะออกของ

เกต 1 ที่โนด B จะมีการเปลี่ยนค่าใหม่เป็น 0 ที่เวลา t + td ซึ่งโนด B และค่าที่เปลี่ยนไปจะถูกบันทึกไว้ใน time queue ที่เวลา t + td ค่าตรรกะที่โนด A ที่เวลา t จะมีค่าใหม่เป็น 1 ส่วนค่าตรรกะที่โนด B ที่เวลาเดียวกันยังคงเป็น 1 อยู่ จนกว่าการวิเคราะห์จะไปทำที่เวลา t + td ซึ่งมีการจัดการ event ที่เกิดขึ้นใหม่ที่โนด B จึงจะมีการเปลี่ยนแปลงค่าตรรกะที่โนด B เป็นค่าใหม่ที่เวลา t+td



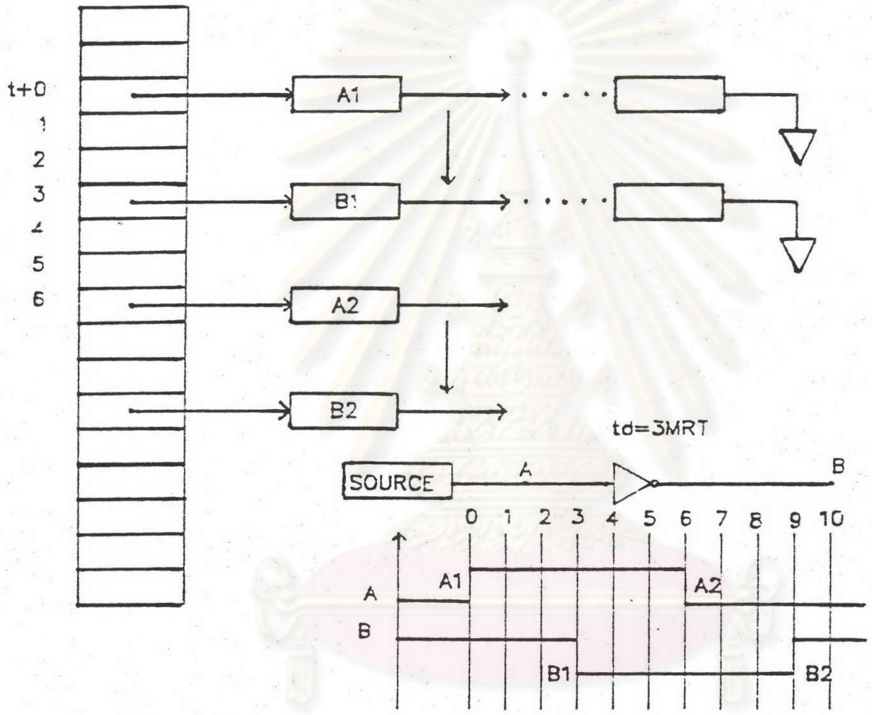
รูปที่ 3.6 แสดง time queue

วิธีการนี้สามารถวิเคราะห์การทำงานของวงจรตรรกะที่เกตแต่ละชนิดมีค่าเวลาประวิงต่างกันได้ ส่วนที่ของโปรแกรมที่จะควบคุมการจัดการ time queue นี้โดยรวมถึงโครงสร้างข้อมูลจะกล่าวถึงในบทที่ 5

ในบทที่ 2 ได้มีการกล่าวถึงการเกิดสัญญาณยอดแหลมในวงจรตรรกะ ว่า เป็นการเปลี่ยนแปลงทางด้านเข้าของเกตใดๆที่เกิดขึ้นเร็วกว่าเวลาประวิงของเกตนั้น นอกจากนั้นยังมี สัญญาณยอดแหลม อีกชนิดหนึ่งซึ่งเกิดในการวิเคราะห์การทำงานเรียกว่า delta spike เนื่องจากการเปลี่ยนแปลงค่าตรรกะที่ขั้วเข้าของเกตมากกว่า 1 ครั้งพร้อมกัน อันจะก่อให้เกิดการเปลี่ยนแปลง 2 ครั้งที่ด้านออกในเวลาเดียวกัน ทำให้โปรแกรมคิดว่าเป็นสัญญาณพัลซ์คาบเวลาเป็นศูนย์ การเปลี่ยนแปลงนี้อาจมีผลกระทบต่อวงจร เช่น ถ้าเป็นจุดต่อไปยังพลิบพลอบอาจก่อให้เกิดการเปลี่ยนสถานะของเกตได้ ซึ่งจะมีตัวอย่างในบทที่ 6 ในงานวิจัยนี้ยังไม่สามารถทำการหยุดยั้งการแพร่ของสัญญาณยอดแหลม ได้ แต่อย่างไรก็ดีเราจะแสดงถึงวิธีการตรวจสอบการเกิดของสัญญาณยอดแหลมในวงจรและสามารถทำการรายงานให้ผู้ใช้ทราบ

3.4 การตรวจสอบสัญญาณยอดแหลมในการวิเคราะห์การทำงานของวงจรตรรกะ

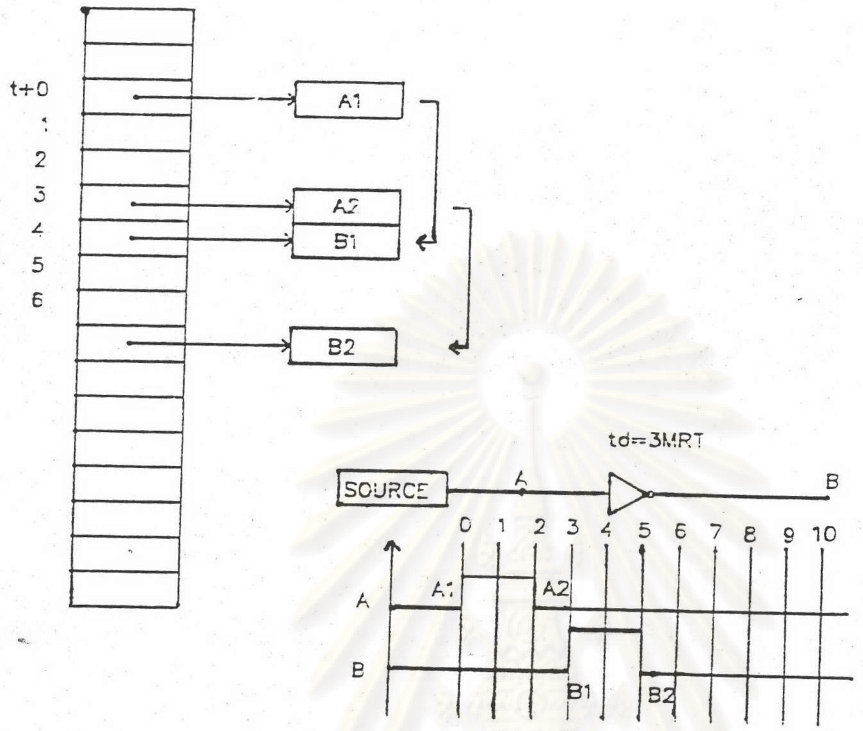
เราจะพิจารณาถึงกรณีต่างๆ ในการวิเคราะห์การทำงานของวงจรตรรกะโดยใช้ time queue ทั้งในกรณีที่ไม่มีการเกิดสัญญาณยอดแหลม การเกิดสัญญาณยอดแหลมแบบธรรมดา และการเกิด delta spike จากนั้นจึงจะสรุปถึงเทคนิคที่สามารถนำไปใช้ในการตรวจสอบการเกิดสัญญาณยอดแหลม ในการวิเคราะห์ซึ่งจะนำมาใช้ในโปรแกรมที่พัฒนาขึ้น



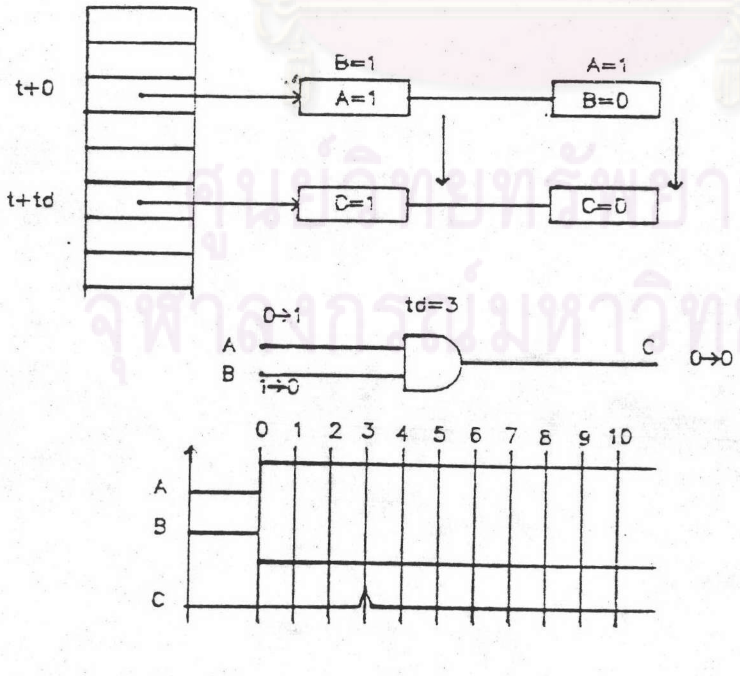
รูปที่ 3.7 แสดง time queue ปกติ

รูปที่ 3.7 แสดงถึงสัญญาณเข้ารูปพัลซ์ที่มีคาบเวลามากกว่าค่าเวลาประวิงของเกต ในกรณีนี้จะไม่เกิดสัญญาณยอดแหลมในวงจร หากลดคาบเวลาของสัญญาณพัลซ์ลงเหลือ 2 MRT จะได้ผลดังรูปที่ 3.8 ซึ่งจะมี สัญญาณยอดแหลม เกิดขึ้น (เวลาประวิงของเกตเท่ากับ 3 MRT)

ในรูปที่ 3.9 สัญญาณเข้าที่ A และ B เปลี่ยนแปลงพร้อมกัน ถ้าหากโปรแกรมตรวจสอบพบการเปลี่ยนแปลงที่ A ก่อน การทำงานจะเป็นดังภาพ สังเกตได้ว่าโนด C จะถูกบันทึกไว้ใน time queue หลายครั้งในเวลาเดียวกัน ซึ่งโปรแกรมจะทำการแสดงผลว่าเกิดสัญญาณยอดแหลม ที่โนดนั้น



รูปที่ 3.8 แสดง time queue เมื่อเกิดสัญญาณยอดแหลม



รูปที่ 3.9 แสดงการเกิด delta spike

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากตัวอย่างที่ยกมา ทำให้เรามองเห็นลักษณะบางอย่างในการวิเคราะห์การทำงาน เราสามารถนิยามสัญญาณยอดแหลมโดยดูจาก time queue ว่าเป็นเหตุการณ์ที่มี event เกิดขึ้นที่เกตใดๆทางด้านขาเข้าขณะที่ยังมี event ทางด้านขาออกโดยมี event ที่ยังมิได้ประมวลผลค้างอยู่ใน time queue

ดังนั้นการตรวจสอบการเกิดสัญญาณยอดแหลม ในวงจรนั้นจะสามารถทำได้โดยตรวจสอบว่า ในขณะที่เกิดการเปลี่ยนแปลงทางด้านขาเข้าของเกตยังมี event ทางด้านขาออกค้างอยู่ใน time queue หรือไม่ ซึ่งในการนี้จะใช้วิธีการซึ่งมีขั้นตอนดังต่อไปนี้

<u>การตรวจสอบการเกิด สัญญาณยอดแหลม</u>

กำหนดว่าสำหรับทุกโนดจะมีตัวแปร flag 1 ตัว

ขั้นที่ 1 : ก่อนเริ่มการวิเคราะห์การทำงานให้ Flag เป็น 0 สำหรับทุกโนด

ขั้นที่ 2 : ทุกครั้งที่ทำการเพิ่ม event ที่โนดใดใน time queue เพิ่มค่า Flag ขึ้น 1

ขั้นที่ 3 : ทุกครั้งที่ดึง event จาก time queue มาประมวลผลจะลดค่า Flag ลง 1

ขั้นที่ 4 : สภาวะเกิด สัญญาณยอดแหลม คือ กรณีที่มีการบันทึก event ของโนดใดแล้วพบว่า Flag > 0 ที่โนดนั้น

การนำเอาหลักการนี้ไปเขียนลงในโปรแกรมจะได้กล่าวถึงต่อไปในบทที่ 5 ส่วนผลของ delta spike ซึ่งจะเกิดกับการวิเคราะห์จะแสดงไว้ในบทที่ 6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย