ร่าง

รายงานวิจัยฉบับสมบูรณ์

โครงการ "การให้จังหวะสัญญาณนาฬิกาสำหรับการออกแบบที่ใช้ FPGA ที่ใช้กำลังงานต่ำ"

"Clock-gating for low power FPGA based designs"

โดย

ผู้ช่วยศาสตราจารย์ ดร. ดารณี หอมดี

สัญญาเลขที่ MRG4680195

ร่าง

รายงานวิจัยฉบับสมบูรณ์

โครงการ "การให้จังหวะสัญญาณนาฬิกาสำหรับการออกแบบที่ใช้ FPGA ที่ใช้กำลังงานต่ำ"

"Clock-gating for low power FPGA based designs"

ชื่อหัวหน้าโครงการ: ผู้ช่วยศาสตราจารย์ ดร. ดารณี หอมดี

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

(งานวิจัยยังไม่สมบูรณ์ โปรดอย่านำไปใช้อ้างอิง)

บทคัดย่อ

สัญญาเลขที่ MRG4680195

โครงการ "การให้จังหวะสัญญาณนาฬิกาสำหรับการออกแบบที่ใช้ FPGA ที่ใช้กำลังงานต่ำ"
"Clock-gating for low power FPGA based designs"

ชื่อหัวหน้าโครงการ: ผู้ช่วยศาสตราจารย์ ดร. ดารณี หอมดี คณะวิศวกรรมศาสตร์
มหาวิทยาลัยขอนแก่น

ในระบบดิจิตอลแบบซิงโครนัส (Synchronous System) โครงข่ายสัญญาณนาฬิกา (Clock Network) ถือว่าเป็นโครงข่ายสัญญาณที่มีขนาดใหญ่และมีความถี่ในการเปลี่ยนแปลงของระดับสัญญาณมากที่สุด ซึ่งการเปลี่ยนแปลงระดับสัญญาณนี้จะไปขับดันตัวเก็บประจุ (Capacitor) ขนาดใหญ่ของระบบโดยรวม จึงกล่าวได้ว่าโครงข่ายสัญญาณนาฬิกาเป็นแหล่งที่มีการใช้พลังงานแบบพลวัต (Dynamic Power) แหล่งใหญ่ที่สุดในระบบ ดังนั้นจุดนี้จึงเป็นอีกจุดหนึ่งที่มีผู้ให้ความสนใจหาวิธีลดอัตราการใช้พลังงานในส่วนนี้เป็นจำนวนมาก เทคนิคประตูสัญญาณนาฬิกาก็เป็นวิธีหนึ่งที่มีผู้นิยมใช้ในการลดอัตราการใช้พลังงาน โดยมีหลักการว่าในระบบดิจิตอลแบบซิงโครนัสมีการใช้สัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการทำงานให้กับทุกส่วนการทำงานภายในวงจร และมีบางช่วงเวลาที่บางส่วนไม่จำเป็นต้องทำงาน การตัดสัญญาณนาฬิกาที่เข้าไปกำหนดจังหวะการทำงานในส่วนนี้จึงเป็นการลดอัตราการใช้พลังงานแบบพลวัตของระบบในบางช่วงเวลาได้โดยไม่มีผลกระทบต่อประสิทธิภาพการทำงาน

งานวิจัยนี้เป็นการรวบรวมรูปแบบการใช้งานเทคนิคประตูสัญญาณนาฬิกาในรูปแบบต่าง ๆ จากงานวิจัยที่ผ่านมาและนำมาทดลองประยุกต์ใช้กับระบบดิจิตอลที่มีอยู่แล้วเพื่อลดอัตราการใช้พลังงานลงรวมทั้งศึกษาผลการทำงานที่เกิดขึ้นในแต่ละรูปแบบเพื่อสรุปหาข้อดีข้อด้อยและลักษณะที่ควรนำเทคนิคประตูสัญญาณนาฬิกาแต่ละรูปแบบไปประยุกต์ใช้ โดยใช้โปรแกรม ISE 6.3i และ ModelSim SE 6.1 จำลองการใช้พลังงานและการทำงาน ผลการวิจัยพบว่าระบบดิจิตอลที่นำเทคนิคประตูสัญญาณนาฬิกาไปประยุกต์ใช้สามารถลดอัตราการใช้พลังงานได้จริงโดยไม่มีผลกระทบต่อประสิทธิภาพการทำงานหากใช้รูปแบบวงจรประตูสัญญาณนาฬิกาที่เหมาะสม

คำหลัก: Clock Gating, FPGA design, low power

สารบัญ

สารบัญตาราง

สารบัญภาพ

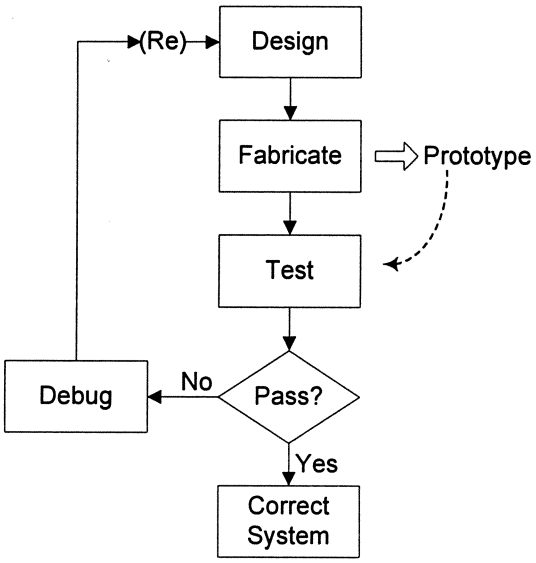
บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงความสำคัญและที่มาของการทำวิจัย หลังจากนั้นจะได้กล่าวถึงทฤษฎีพื้นฐานที่เกี่ยวข้อง

1. ความสำคัญและที่มาของการทำวิจัย

การออกแบบและพัฒนาระบบดิจิตอลโดยทั่วไปนั้นมีจุดมุ่งหมายหลักเน้นไปที่ว่าทำอย่างไรจึงจะได้ระบบที่มีประสิทธิภาพการทำงานสูงสุด ภายใต้งบประมาณ และระยะเวลาในการพัฒนาที่กำหนดไว้ ถึงแม้ว่าในปัจจุบันจุดมุ่งหมายหลักในการออกแบบระบบดิจิตอลจะยังคงเดิมคือเน้นประสิทธิภาพการทำงาน แต่รูปแบบการนำระบบดิจิตอลไปใช้งานก็มีหลากหลายมากขึ้นด้วย โดยเฉพาะในงานระบบสมองกลฝังตัว (Embedded System) ซึ่งวงจรนั้นจะฝังอยู่ในอุปกรณ์อิเล็กทรอนิกส์ เช่น รีโมทสำหรับควบคุมอุปกรณ์ต่างๆ ระบบควบคุมอิเล็กทรอนิกส์ (Electronics Control Unit: ECU) ในรถยนต์ เครื่องจักรควบคุมต่างๆ เป็นต้น โดยจุดประสงค์ในการออกแบบระบบดิจิตอลเหล่านี้มีแนวโน้มเพื่อให้สามารถพกพาหรือเคลื่อนย้ายได้สะดวก และสามารถทำงานได้ในทุกที่ทุกเวลา (Anywhere Anytime) มากขึ้น ซึ่งแนวโน้มนี้ก็ยังรวมไปถึงคอมพิวเตอร์ส่วนบุคคลชนิดพกพารูปแบบต่างๆ ด้วย อุปกรณ์เหล่านี้จึงจำเป็นต้องมีแบตเตอรี่เป็นแหล่งพลังงานในตัว แต่เนื่องจากแบตเตอรี่สามารถจ่ายพลังงานให้ได้ในระยะเวลาที่จำกัด ดังนั้นความสามารถของแบตเตอรี่ และอัตราการใช้พลังงานของอุปกรณ์ดิจิตอลจะเป็นตัวกำหนดระยะเวลาในการใช้งานในแต่ละครั้ง ในขณะที่ความก้าวหน้าทางเทคโนโลยีในการผลิตแบตเตอรี่ไม่สามารถก้าวตามความต้องการนี้ได้ทัน ดังนั้นการลดความต้องการการใช้พลังงานของระบบดิจิตอลจึงมีความจำเป็นควบคู่ไปกับการพัฒนาประสิทธิภาพที่ควรจะเพิ่มขึ้นอีกอย่างด้วย

วิธีการออกแบบพัฒนาระบบดิจิตอล (ภาพที่ 1) ในอดีตนั้น จะต้องใช้ผู้ที่มีทักษะสูงในการออกแบบวงจรเป็นผู้ออกแบบ โดยใช้โปรแกรมช่วยในลักษณะที่เป็นการช่วยวาดหรือช่วยวางผังวงจร (Schematic) ซึ่งการออกแบบวิธีนี้จำเป็นต้องใช้ระยะเวลาในการออกแบบมาก นอกจากนี้ในขบวนการจำลองการทำงานและการแก้ไขส่วนผิดพลาดยังยุ่งยากและใช้เวลานานอีกด้วย โดยเมื่อเสร็จสิ้นกระบวนการในการออกแบบแล้วนั้นจะต้องส่งสิ่งที่ได้ออกแบบนี้ไปสร้าง (Fabrication) เป็นชิพต้นแบบ (Prototype) ที่โรงงาน เพื่อใช้ทดสอบและประเมินผลการทำงานจริงของระบบนั้นๆ ซึ่งถ้าหากพบว่ามีข้อผิดพลาดภายในระบบ จะต้องมีการวนกลับไปแก้ไขวงจรตั้งแต่ในขั้นตอนของการออกแบบ ตามด้วยการสร้างเป็นชิพต้นแบบรุ่นที่ 2 รุ่นที่ 3 และรุ่นต่อๆ ไป โดยที่กระบวนการเหล่านี้จะทำวนซ้ำไปเรื่อยๆ จนกว่าจะได้ระบบดิจิตอลที่มีประสิทธิภาพตามต้องการ ซึ่งวงรอบขบวนการในการพัฒนาเหล่านี้จะต้องใช้เวลาและค่าใช้จ่ายในการพัฒนาเป็นจำนวนมาก



ภาพที่ 1 วิธีการออกแบบพัฒนาระบบดิจิตอล

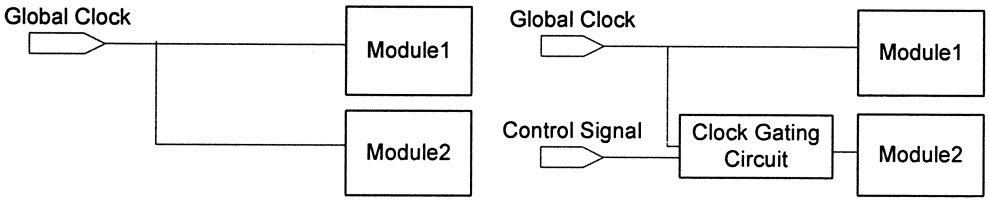
ปัจจุบันเครื่องมือที่ใช้ในการออกแบบวงจรดิจิตอลได้พัฒนาขึ้นอย่างมาก ทำให้มีกระบวนการออกแบบรูปแบบใหม่ ๆ เกิดขึ้น รูปแบบหนึ่งซึ่งมีประสิทธิภาพสูงและน่าสนใจ คือการใช้ภาษาบรรยายฮาร์ดแวร์ (HDL : Hardware Description Language) ในการออกแบบ (Design) จำลองการทำงาน (Simulate) และสังเคราะห์วงจร (Synthesize) การออกแบบจะใช้ภาษาบรรยายฮาร์ดแวร์เช่น ภาษา VHDL และ ภาษา Verilog เขียนบรรยายพฤติกรรมการทำงานของฮาร์ดแวร์นั้น (คล้ายการเขียนโปรแกรม) จากนั้นก็นำไปจำลองการทำงานและแก้ไขจุดบกพร่อง ซึ่งในกระบวนการรูปแบบนี้ไม่จำเป็นต้องใช้ผู้มีทักษะสูงมากในเชิงการออกแบบวงจร และยังใช้เวลาน้อยกว่าวิธีการในอดีตอีกด้วย เมื่อได้ผลเป็นที่พอใจแล้วก็สามารถนำภาษาบรรยายฮาร์ดแวร์นั้นไปทำการสังเคราะห์วงจร เพื่อสร้างเป็นชิพต้นแบบเพื่อทดสอบจริงได้ และในปัจจุบันขั้นตอนนี้เรายังสามารถใช้ชิพชนิดหนึ่งที่สามารถโปรแกรมวงจรได้เรียกว่า FPGA หรือ Field Programmable Gate Arrays ใช้แทนวงจรต้นแบบเพื่อใช้ทดสอบการทำงานจริง ซึ่งในปัจจุบัน FPGA ก็ได้รับความสนใจเป็นอย่างมาก เนื่องจากสามารถใช้งานได้ในหลายระดับตั้งแต่การออกแบบวงจรดิจิตอลทั้งที่มีขนาดเล็กเช่น วงจรนาฬิกาอย่างง่าย จนถึงวงจรขนาดใหญ่อย่างการออกแบบไมโครโพรเซสเซอร์ก็สามารถที่จะทำได้ เหมาะสำหรับขั้นตอนในการออกแบบวงจรก่อนที่จะผลิตจริง หรือใช้เพื่อการศึกษา เนื่องจากสามารถทำการโปรแกรมวงจรที่ออกแบบลงในชิพได้เองทันทีโดยไม่ต้องส่งไปทำชิพต้นแบบที่โรงงาน และยังสามารถโปรแกรมวงจรใหม่แทนที่วงจรเดิมซ้ำ ๆ บ่อยครั้งตามที่ต้องการได้ ทำให้ใช้ค่าใช้จ่ายและระยะเวลาในการออกแบบน้อยกว่าในอดีตมาก

งานวิจัยนี้จะคำนึงถึงพลังงาน และการนำเทคนิคที่ใช้ในการลดพลังงานมาใช้กับระบบดิจิตอล โดยการทำงานจะใช้ VHDL ร่วมกับ FPGA

2. หลักการของเทคนิคประตูสัญญาณนาฬิกา

การทำงานของวงจรแบบซิงโครนัสจะเป็นการทำงานแบบลำดับขั้นเมื่อจบการทำงานในแต่ละขั้นก็จะมีการส่งต่อการทำงานไปยังอีกขั้นตอนหนึ่งต่อ ๆ ไปจนสิ้นสุดกระบวนการ โดยใช้สัญญาณ

นาฬิกาเป็นตัวกำหนดจังหวะการทำงานและการส่งต่อการทำงานในแต่ละขั้น เราสามารถมองการทำงานในแต่ละขั้นให้เป็นกลุ่มของวงจรโดยมีสัญญาณนาฬิกาเป็นตัวกระตุ้นการทำงาน ดังแสดงในภาพที่ 4ก)

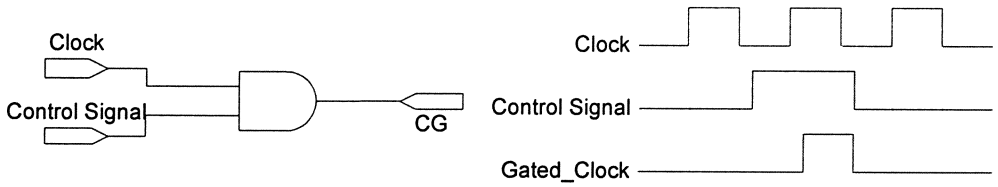


ก) ก่อนทำ Clock Gating **ข) หลังทำ Clock Gating**

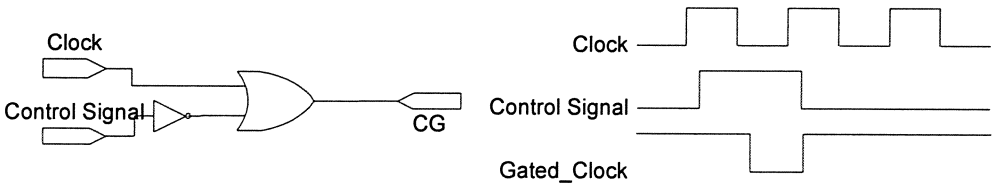
ภาพที่ 2 หลักการพื้นฐานการทำ Clock Gating

การทำประตูสัญญาณนาฬิกาคือการที่เรากระทำการบางอย่างเพื่อหยุดสัญญาณนาฬิกาไว้ในบางช่วงเวลา เพื่อไม่ให้ไปกระตุ้นการทำงานของกลุ่มวงจรที่ไม่จำเป็นจะต้องทำงานในช่วงเวลานั้น ดังแสดงในภาพที่ 4ข) โดยจะมีวงจรประตูสัญญาณนาฬิกา (Clock Gating Circuit) เป็นตัวกรองสัญญาณนาฬิกาจาก Global Clock และมีสัญญาณควบคุม (Control Signal) เป็นตัวควบคุมก่อนจะต่อเข้าสู่กลุ่มวงจรนั้น

วงจรประตูสัญญาณนาฬิกาสามารถแบ่งออกได้เป็น 2 แนวทางดังนี้ [8]



ก) วงจรแบบ Logic Low Disable



ข) วงจรแบบ Logic High Disable

ภาพที่ 3 วงจรประตูสัญญาณนาฬิกาและรูปแบบสัญญาณวงจร

2.1 Logic Low Disable

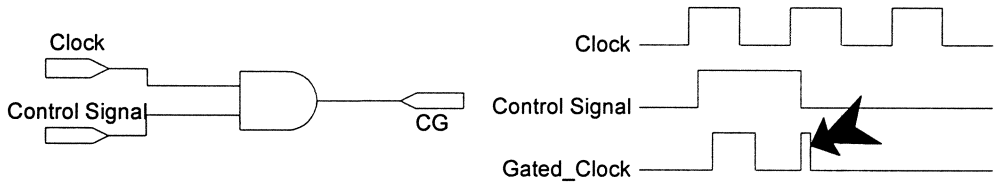
เป็นการสร้างวงจรควบคุมสัญญาณนาฬิกาให้ไปขับดันวงจรดิจิตอล (ภาพที่ 5ก)) ที่เป็น Edge Triggered แบบทำงานในสัญญาณขาขึ้น (Rising Edge) โดยวงจรจะทำการคงสัญญาณนาฬิกาขาออก (CG) ไว้ที่ 0 (Low) ตลอดเมื่อสัญญาณควบคุม (Control Signal) มีค่าเป็น 0 (Low) และจะมีค่าเปลี่ยนแปลงขึ้นลงตามสัญญาณนาฬิกาเมื่อสัญญาณควบคุมมีค่าเป็น 1 (High)

2.2 Logic High Disable

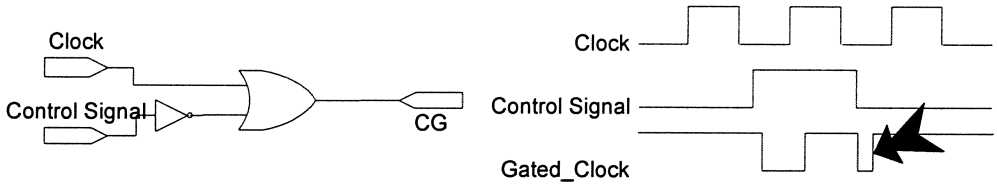
เป็นการสร้างวงจรควบคุมสัญญาณนาฬิกาให้ไปขับดันวงจรดิจิตอล (ภาพที่ 5ข)) ที่เป็น Edge Triggered แบบทำงานในสัญญาณขาลง (Falling Edge) โดยวงจรจะทำการคงสัญญาณนาฬิกาขาออก (CG) ไว้ที่ 1 (High) ตลอดเมื่อสัญญาณควบคุม (Control Signal) มีค่าเป็น 0 (Low) และจะมีค่าเปลี่ยนแปลงขึ้นลงตามสัญญาณนาฬิกาเมื่อสัญญาณควบคุมมีค่าเป็น 1 (High)

3. Glitch

Glitch คือสัญญาณเล็ก ๆ ซึ่งเกิดจากความไม่ตั้งใจที่ถูกเพิ่มเข้ามาในสัญญาณปกติ โดยสัญญาณที่ถูกเพิ่มขึ้นมานี้อาจไปกระตุ้นการทำงานของวงจรที่เชื่อมต่ออยู่ในเวลาที่ไม่ควรเกิดการทำงานทำให้อาจมีผลพวงให้การทำงานในวงจรนั้นผิดไปจากที่ควรจะเป็นได้ การเกิดของ Glitch นั้นมีหลายสาเหตุ การทำ Clock Gating ก็เป็นอีกสาเหตุหนึ่งซึ่งอาจทำให้เกิด Glitch ขึ้นได้ [9] ทั้งนี้นอกจากผลกระทบต่อการทำงานของวงจรแล้วการเกิด Glitch ยังมีผลต่อการใช้พลังงานในวงจร [9] อีกด้วย ดังนั้นในการทำ Clock Gating จึงจำเป็นต้องคำนึงถึงการเกิดและผลของการเกิด Glitch รวมทั้งวิธีแก้ไขหรือป้องการการเกิด Glitch



ก) วงจรแบบ Logic Low Disable



ข) วงจรแบบ Logic High Disable

ภาพที่ 4 การเกิด Glitch อันเนื่องมาจาก Clock Gating

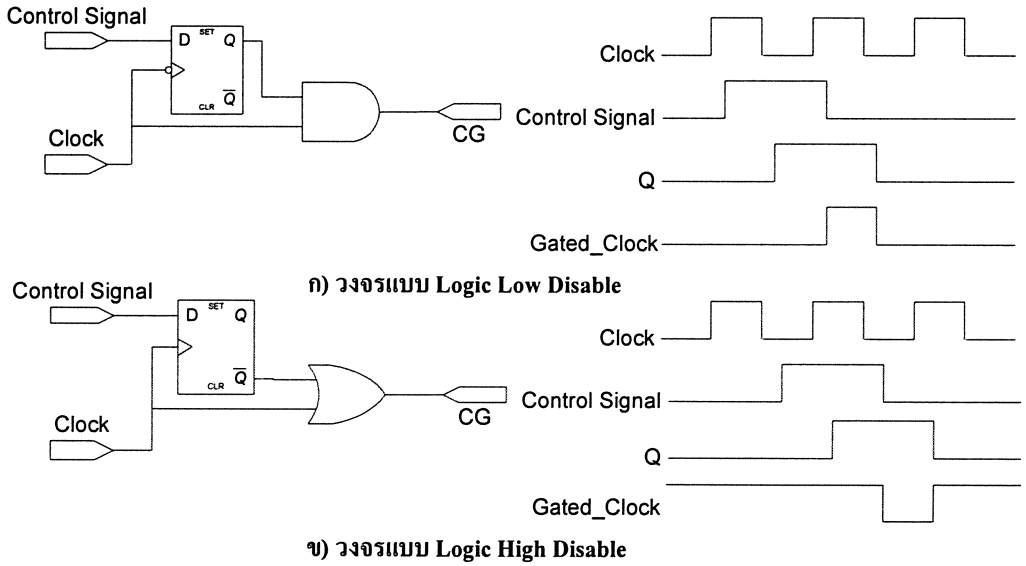
ภาพที่ 6 แสดงตัวอย่างการเกิด Glitch ซึ่งเกิดจากการทำ Clock Gating จะเห็นว่าวงจรในภาพที่ 6 เป็นวงจรเดียวกันกับวงจรในภาพที่ 5 ต่างกันที่สัญญาณ Control Signal ที่ป้อนเข้ามา ถ้าสัญญาณ Control Signal ถูกป้อนเข้ามาดังภาพที่ 6 ก็จะทำให้เกิด Glitch ขึ้น ซึ่งสัญญาณ Control Signal จะเกิดจาก วงจร Combination Logic ที่ถูกกระตุ้นการทำงานโดยสัญญาณนาฬิกาขาขึ้น (ในกรณีเป็นวงจรแบบ Logic Low Disable) ดังนั้น สัญญาณ Control Signal จะเปลี่ยนค่าหลังจากเกิด Clock ขาขึ้นเล็กน้อยอันเนื่องมาจาก Delay ของวงจร Combination Logic ที่สร้างสัญญาณนี้ หากพิจารณาสัญญาณ Control Signal ในช่วงขาลงจะเห็นว่าเมื่อนำมา And กับสัญญาณ Clock จะทำให้เกิดสัญญาณ Gated Clock (CG) เปลี่ยนสถานะเป็น High อีกครั้งใน

ช่วงเวลาสั้น ๆ ก่อนจะกลับมาอยู่ที่ Low สัญญาณที่เกิดนี้เรียกว่า Glitch โดยความกว้างของสัญญาณ Glitch จะขึ้นอยู่กับค่า Delay ของวงจร Combination Logic

4. การป้องกันการเกิด Glitch

การป้องกันการเกิด Glitch ที่เกิดจากการทำ Clock Gating มีด้วยกัน 2 วิธี คือ การควบคุมสัญญาณให้มีการเปลี่ยนแปลงในช่วงที่ไม่ทำให้เกิด Glitch [9, 10] และ การใช้ Latch ช่วยคงสัญญาณควบคุมไว้ในช่วงเวลาที่ต้องการ [10, 11]

4.1 การควบคุมสัญญาณให้มีการเปลี่ยนแปลงในช่วงที่ไม่ทำให้เกิด Glitch วิธีนี้จะใช้ในการทำ Clock Gating โดยใช้วงจรแบบที่เรียกว่า Latch Free Clock Gating Circuit [10] ลักษณะของวงจรจะแสดงได้ดังภาพที่ 5 และภาพที่ 6 จะเห็นว่าทั้งสองรูปแสดงวงจรแบบเดียวกันแต่สัญญาณควบคุมจะต่างกันทำให้เกิด Glitch ขึ้นในภาพที่ 6 ดังที่ได้อธิบายไปแล้ว ส่วนภาพที่ 5 ไม่เกิด Glitch เนื่องจากสัญญาณควบคุมมีการเปลี่ยนค่าในช่วงเวลาที่ไม่ทำให้เกิด Glitch เช่นในกรณีของภาพที่ 5ก) สัญญาณควบคุมจะเปลี่ยนค่าในช่วงสัญญาณขาลง ซึ่งช่วงเวลาในช่วงต่อจากนั้นสัญญาณ Clock จะคงค่าอยู่ที่ Low เมื่อนำสัญญาณควบคุมและสัญญาณ Clock มา And กัน สัญญาณ CG ที่ได้ก็จะยังไม่เปลี่ยนค่าไปตามสัญญาณควบคุมจนกว่าจะถึงช่วงสัญญาณ Clock ในขาขึ้นถัดไป จึงทำให้ไม่มีสัญญาณ Glitch เกิดขึ้น



ก) วงจรแบบ Logic Low Disable

ข) วงจรแบบ Logic High Disable

ภาพที่ 5 วงจรแบบ Latch Based Clock Gating

4.2 การใช้ Latch ช่วยคงสัญญาณควบคุมไว้ในช่วงเวลาที่ต้องการ วิธีนี้จะใช้ในการทำ Clock Gating โดยใช้วงจรแบบที่เรียกว่า Latch Based Clock Gating Circuit [10] วงจร Latch Based Clock Gating นี้จะมีส่วนที่เพิ่มขึ้นจากวงจร Latch Free Clock Gating ขึ้นมาเล็กน้อย คือ จะเพิ่ม Latch เข้ามาในวงจรเพื่อใช้คงค่าสัญญาณควบคุมให้มีค่าคงที่อยู่ในช่วงที่ต้องการ ดังแสดงในภาพที่ 5

จะเห็นว่ารูปแบบการทำ Clock Gating มีด้วยกันหลายรูปแบบ การจะเลือกว่าจะใช้รูปแบบใดนั้นจะขึ้นอยู่กับการทำงานของวงจรที่เราจะนำเทคนิคนี้ไปใช้ นอกจากนี้การทำ Clock Gating แต่ละรูปแบบยังอาจให้ผลทางพลังงานที่ไม่เหมือนกันอีกด้วย [8]

บทที่ 2
ขั้นตอนการดำเนินงานวิจัยและทดลอง

การทดลองในงานวิจัยนี้จะเป็นการทดลองนำเทคนิค Clock Gating รูปแบบต่างๆ ที่ได้รวบรวมจากงานวิจัยอื่นๆ ก่อนหน้านี้มาประยุกต์ใช้ในวงจรที่ได้มีการออกแบบไว้ก่อนแล้ว โดยเขียนอยู่ในรูปภาษา VHDL ซึ่งเป็นภาษาบรรยายพฤติกรรมฮาร์ดแวร์ภาษาหนึ่ง และเพื่อเป็นการประหยัดเวลาที่ใช้ในการทดลองจึงได้นำวงจรที่มีผู้พัฒนาไว้แล้วและมีลักษณะที่เป็น Open Source มาใช้ในการทดลองแทนที่จะสร้างวงจรขึ้นมาใหม่ โดยวงจรที่เลือกใช้ในงานวิจัยนี้คือวงจร Alarm Clock และวงจร T80 ซึ่งจะได้กล่าวถึงต่อไป

1. ขั้นตอนการทดลอง

1.1 ศึกษาวงจรที่จะนำมาทดลอง

ในการที่เราจะนำเทคนิค Clock Gating ไปประยุกต์ใช้ในวงจรใดๆ นั้นจำเป็นจะต้องทำการศึกษาโครงสร้างและการทำงานของวงจรนั้นให้เข้าใจในระดับหนึ่งเสียก่อน เพื่อให้มองเห็นว่าส่วนใดของวงจรที่มีช่วงเวลาที่ไม่จำเป็นต้องทำงาน (Idle time) เหมาะที่จะนำเทคนิค Clock Gating มาประยุกต์ใช้ และสามารถหา Control Signal ซึ่งสามารถควบคุมการเปิดปิดสัญญาณนาฬิกาให้สอดคล้องกับการทำงานของวงจรโดยไม่ส่งผลกระทบต่อการทำงานโดยรวมมากเกินกว่าที่สามารถรับได้เมื่อเทียบกับการทำงานของวงจรเดิม โดยรายละเอียดของการศึกษาวงจร Alarm Clock และวงจร T80 ซึ่งจะนำมาใช้ทดลองในงานวิจัยนี้นั้นจะกล่าวถึงในหัวข้อถัดไป

1.2 การประยุกต์ใช้เทคนิค Clock Gating ในวงจร

การทำ Clock Gating นั้นไม่ได้มีวิธีเดียวแต่สามารถทำได้หลายวิธีซึ่งสามารถเลือกใช้ได้ตามความเหมาะสม ดังที่ได้ศึกษาไว้ในบทที่ 1 โดยสามารถแยกวิธีการทำ Clock Gating ออกเป็นกลุ่มทางเลือกได้ 2 กลุ่มดังนี้

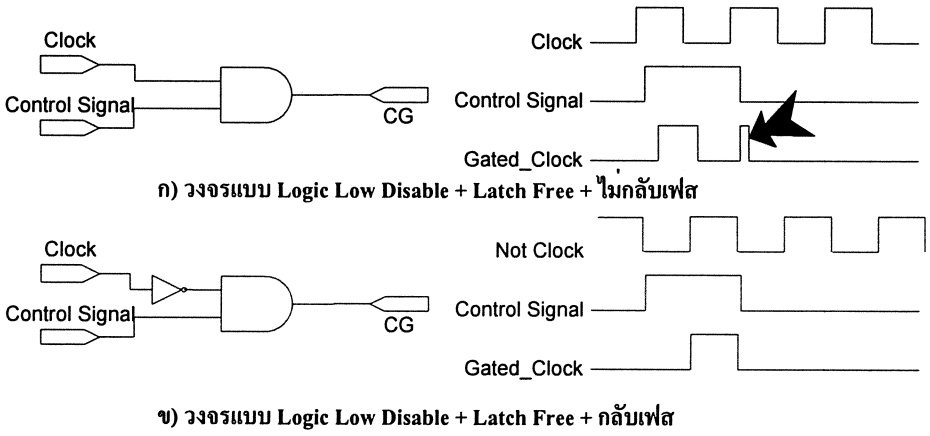
(1) Logic Low Disable หรือ Logic High Disable

(2) Latch Free Clock Gating หรือ Latch Base Clock Gating

และเมื่อนำกลุ่มทางเลือกในการทำ Clock Gating ทั้ง 2 กลุ่มมารวมกันก็จะได้การทำ Clock Gating ถึง 4 รูปแบบ และถ้ารวมกับการที่สามารถกำหนดการทำงานของวงจรให้เป็นแบบ Active High หรือ Active Low ได้ก็จะทำให้มีทางเลือกในการใช้เทคนิค Clock Gating ได้ถึง 8 รูปแบบโดยเป็นการเลือกระหว่าง Logic Low หรือ High Disable, Latch Free หรือ Latch Base และ Active High หรือ Low

ถึงแม้ว่าจะเลือกรูปแบบการทำ Clock Gating ได้ถึง 8 รูปแบบแต่ว่าในบางรูปแบบอาจทำให้เกิด Glitch ขึ้นได้ตามที่ได้อธิบายไว้ในบทที่ 1 เช่นกรณีที่ 1 และ 2 (Logic Low Disable ร่วมกับ Latch Free Clock Gating) หากนำไปใช้ในวงจรที่มีการทำงานอยู่ในช่วงสัญญาณนาฬิกาขาขึ้น สัญญาณควบคุมซึ่งเป็นสัญญาณซึ่งเกิดจากการทำงานภายในก็จะมีการเปลี่ยนแปลงในช่วงสัญญาณนาฬิกาขาขึ้นเช่นกันและเป็นเหตุให้เกิด Glitch ขึ้นดังแสดงในภาพที่ 6ก) แต่ปัญญานี้สามารถแก้ไขได้โดยการกลับเฟสสัญญาณนาฬิกาก่อนที่จะทำ Clock Gating เพื่อให้สัญญาณ

ควบคุมนั้นเกิดการเปลี่ยนแปลงในช่วงสัญญาณนาฬิกาขาลงซึ่งจะทำให้เป็นไปตามหลักการการทำ Clock Gating และทำให้ไม่เกิด Glitch ดังแสดงในภาพที่ 6ข)

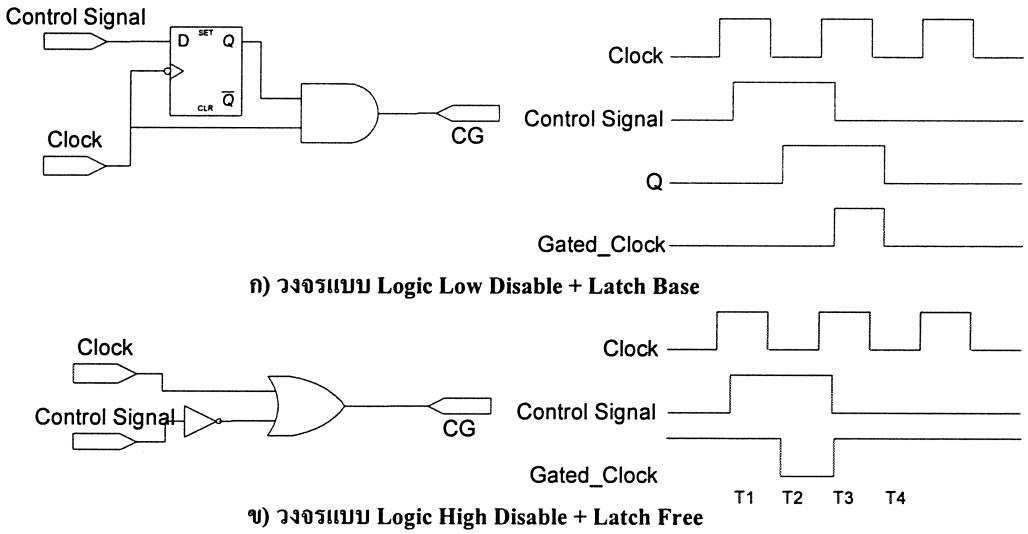


ก) วงจรแบบ Logic Low Disable + Latch Free + ไม่กลับเฟส

ข) วงจรแบบ Logic Low Disable + Latch Free + กลับเฟส

ภาพที่ 6 เปรียบเทียบสัญญาณวงจร Clock Gating ที่ต้องมีการกลับเฟสสัญญาณนาฬิกา

แต่อย่างไรก็ตามในบางวงจรก็ไม่สามารถเลือกใช้รูปแบบการทำ Clock Gating ได้ทุกรูปแบบ ทั้งนี้เนื่องจากการทำ Clock Gating ในแต่ละแบบนั้นจะให้สัญญาณนาฬิกาที่ผ่านการทำ Clock Gating (Gated Clock) ที่มีลักษณะและเวลาเกิดที่แตกต่างกันดังเช่นตัวอย่างที่แสดงในภาพที่ 7



ก) วงจรแบบ Logic Low Disable + Latch Base

ข) วงจรแบบ Logic High Disable + Latch Free

ภาพที่ 7 เปรียบเทียบสัญญาณ Gated Clock จากวงจรที่ต่างกัน

จะเห็นว่าที่ ณ เวลาเดียวกัน Control Signal ได้ทำงานพร้อมกันในช่วงเวลา T1 แต่ใช้วงจรในการทำ Clock Gating ต่างวิธีกันทำให้สัญญาณ Gated Clock ที่ได้เกิดขึ้นในช่วงเวลาที่แตกต่างกันและให้ลักษณะการกระตุ้นการทำงานที่ไม่เหมือนกัน (การเลือกใช้ Active Low และ Active High) โดยสัญญาณ Gated Clock ของวงจรในภาพที่ 7ก) จะสามารถใช้กระตุ้นการ

ทำงานได้ในช่วงเวลา T3 (Active High) และช่วงเวลา T4 (Active Low) ส่วนวงจรในภาพที่ 7 ข) จะสามารถใช้กระตุ้นการทำงานได้ในช่วงเวลา T2 (Active Low) และช่วงเวลา T3 (Active High) รวมแล้วจากทั้ง 2 วงจรจะทำให้เราสามารถเลือกวิธีการทำ Clock Gating ได้ถึง 4 วิธี แต่ทว่าในการนำเทคนิคนี้ไปใช้จริงในบางวงจรที่มี Stage การทำงานที่ค่อนข้างจะต่อเนื่องและมีข้อจำกัดทางเวลาที่ค่อนข้างมาก หรือในวงจรที่ปรับเปลี่ยนลักษณะการกระตุ้นการทำงานได้ยาก การทำ Clock Gating ในบางวิธีอาจมีผลทำให้การทำงานโดยรวมของทั้งวงจรผิดพลาดหรือไม่สามารถทำงานได้ ดังนั้นการทดลองประยุกต์ใช้เทคนิค Clock Gating ในแต่ละวงจรจึงอาจจะไม่สามารถทดลองได้ครบทั้ง 8 วิธี

1.3 สังเคราะห์วงจรและจำลองการทำงาน (Synthesis และ Simulate)

ทำการสังเคราะห์วงจรและจำลองการทำงานโดยใช้โปรแกรม ISE 6.3i และ ModelSim SE 6.1 โดยการจำลองการทำงานจะใช้โปรแกรม Benchmark ซึ่งได้ออกแบบขึ้นเองเพื่อทดสอบการทำงานว่าทำงานได้ถูกต้องหรือไม่เมื่อเทียบกับวงจรเดิม โดยมีรายละเอียดการทดลองของแต่ละวงจรดังนี้

1.3.1 วงจร Alarm Clock จะใช้วิธีจำลองการทำงานโดยการสมมุติให้มีการใช้งานวงจรนี้เหมือนการใช้นาฬิกาปลุกโดยทั่วๆ ไป โดยจะสมมุติการทำงานเหมือนการใช้งานนาฬิกาปลุกในรอบ 1 สัปดาห์ ในระหว่างนั้นจะมีการตั้งเวลาปลุก 2 ครั้งคือในช่วงวันธรรมดา (จันทร์ - ศุกร์) จะตั้งเวลาปลุกไว้ที่ 6.00 น. ส่วนวันหยุด (เสาร์ - อาทิตย์) จะตั้งเวลาไว้ที่ 9.00 น. และจะตั้งเวลาปัจจุบันใหม่ 1 ครั้งในช่วงต้นสัปดาห์ก่อนที่จะมีการตั้งเวลาปลุกในช่วงวันธรรมดา แต่อย่างไรก็ตามเนื่องจากมีข้อจำกัดของโปรแกรมที่ใช้ในการจำลองการทำงานจึงจำเป็นต้องใช้วิธีสมมุติความเร็วของการทำงานให้เร็วขึ้นโดยกำหนดให้เวลา 1 นาที เป็นการจำลองการทำงานโดยใช้เวลา 100 ns

1.3.2 วงจร T80 เนื่องจากเป็นวงจรที่เป็นตัวประมวลผล (Processor) ซึ่งเลียนแบบการทำงานของ Z80 Microprocessor ดังนั้นในการจำลองการทำงานจึงใช้วิธีรันโปรแกรม Benchmark ซึ่งมีจุดประสงค์ในการทำงานต่างๆ กัน จากนั้นดูผลลัพธ์ที่ได้ว่าผลการทำงานถูกต้องหรือไม่ สำหรับโปรแกรม Benchmark ที่ใช้ในการทดลองจะเป็นโปรแกรมซึ่งผู้วิจัยได้เขียนขึ้นมาเอง 3 โปรแกรม ดังนี้

(1) โปรแกรมหาค่า Factorial

(2) โปรแกรมเรียงลำดับตัวเลขโดยใช้วิธีจัดเรียงค่าแบบ Bubble Sort

(3) โปรแกรมหาตัวเลข Pascal's Triangle

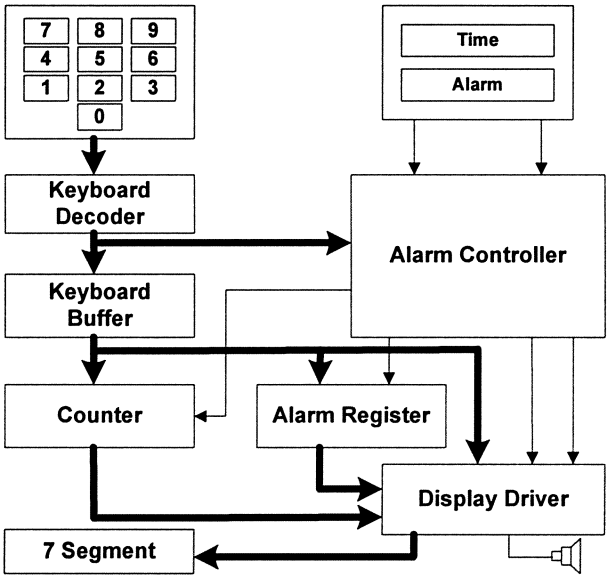
1.4 วิเคราะห์การใช้พลังงาน บันทึกผลและอธิบายผลการทดลอง

บันทึกผลการสังเคราะห์วงจรและเปรียบการเปลี่ยนแปลงของวงจรก่อนและหลังการประยุกต์ใช้เทคนิค Clock Gating ในวงจร บันทึกผลการจำลองการทำงานเปรียบเทียบและวิเคราะห์ผลการทำงานของวงจรก่อนและหลังการประยุกต์ใช้ Clock Gating จากนั้นใช้ผลการจำลองการทำงานไปวิเคราะห์การใช้พลังงานโดยใช้ XPower ซึ่งเป็นเครื่องมือที่ใช้วิเคราะห์การใช้

พลังงานที่มีอยู่ในโปรแกรม ISE 6.3i จากนั้นเปรียบเทียบและวิเคราะห์ผลทางพลังงานที่ได้ สุดท้ายนำข้อมูลที่ได้จากการมาสรุปผลการทดลอง

2. วงจร Alarm Clock

วงจร Alarm Clock เป็นวงจรนาฬิกาปลุกอย่างง่ายมีการทำงานไม่ซับซ้อนนัก โดยจะทำงานตามช่วงจังหวะสัญญาณนาฬิกา วงจรนาฬิกาปลุกนี้สามารถแสดงเวลาได้ 4 หลักโดยบอกชั่วโมงและนาทีซึ่งตัวเลขจะเพิ่มขึ้นทุก ๆ นาที สามารถตั้งเวลาปัจจุบันและเวลาปลุกได้ ส่วนประกอบวงจร Alarm Clock มีรายละเอียดดังภาพที่ 8



ภาพที่ 8 โครงสร้างวงจร Alarm Clock

การกดปุ่มตัวเลขหรือปุ่มควบคุมใด ๆ (ปุ่ม Time และ ปุ่ม Alarm) จะต้องกดค้างไว้จนกระทั่งถึงสัญญาณนาฬิกาในช่วงขาขึ้นจึงจะมีผลให้เกิดการกระตุ้นการทำงานของวงจรเพื่อตอบสนองต่อการกดปุ่มนั้น ๆ การกดปุ่มใด ๆ แล้วปล่อยโดยไม่อยู่ในช่วงที่เกิดสัญญาณนาฬิกาขาขึ้นจะไม่ก่อให้เกิดผลใด ๆ ทั้งสิ้น ซึ่งการทำงานของวงจรนาฬิกานี้จะถูกแบ่งออกเป็นโหมด ต่าง ๆ ได้ 5 โหมด ตามผลของการกระตุ้นจากการกดปุ่มต่าง ๆ ดังนี้

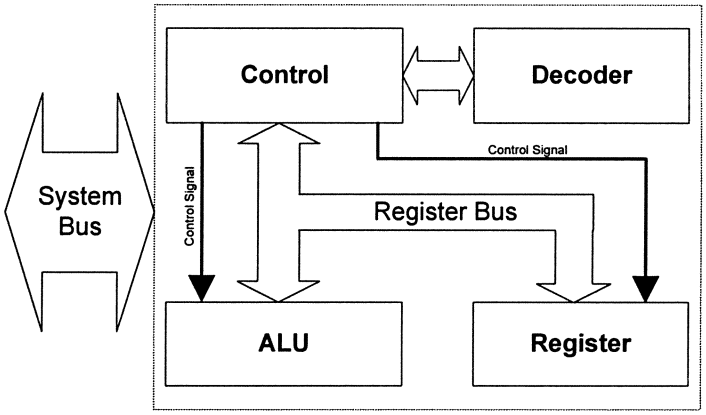
1. โหมดการทำงานปกติ
2. โหมดการทำงานเมื่อมีการกดปุ่ม
3. โหมดการตั้งเวลา
4. โหมดการตั้งเวลาปัจจุบัน
5. โหมดแสดงเวลาปลุก

6.

3. วงจร T80

T80 เป็นโพรเซสเซอร์ขนาด 8 บิต ถูกพัฒนาขึ้นมาโดย Daniel Wallner [19] โดยใช้ภาษา VHDL ในการพัฒนา และมีลักษณะเป็น Open Core ซึ่งมีคุณสมบัติและลักษณะการทำงาน

เลียนแบบ Z80 (Z80 Compatible) และมีโครงสร้างดังแสดงในภาพที่ 9 สามารถ Download ตัว Source Code มาใช้ทดลองได้ที่ http://www.opencores.org/ โดยไม่เสียค่าใช้จ่ายใด ๆ แต่อย่างไรก็ตาม Source Code ที่ได้มาจะต้องมีการปรับปรุงเพิ่มเติมบางส่วนเพื่อให้สามารถใช้งานร่วมกับโปรแกรมที่ใช้ในการสังเคราะห์



ภาพที่ 9 โครงสร้างของ T80

โครงสร้างการทำงานของ T80 สามารถแยกการทำงานออกได้เป็น 4 ส่วนหลักคือ Control, Decoder, ALU และ Register โดยมีส่วน Control เป็นส่วนควบคุมการทำงานของส่วนอื่น ๆ ให้ทำงานสอบคล้องกันตามจังหวะสัญญาณนาฬิกาดังนี้

3.1 Control เป็นส่วนที่กำหนด State การทำงานและควบคุมการทำงานของส่วนอื่น ๆ รวมทั้งการรับและส่งข้อมูลจากภายนอกด้วย

3.2 Decoder เป็นส่วนที่ใช้ถอดรหัสชุดคำสั่งว่าอยู่กลุ่มคำสั่งชนิดใด ใช้ State การทำงานแบบไหน และต้องใช้ทรัพยากรใดบ้างในการประมวลผลคำสั่ง

3.3 ALU เป็นส่วนที่ใช้คำนวณตรรกะ และชุดคำสั่งทางคณิตศาสตร์ทั้งหมด การทำงานจะทำงานในทันทีที่อินพุตมีการเปลี่ยนค่าโดยไม่ต้องรอจังหวะสัญญาณนาฬิกา สัญญาณอินพุตที่ใช้ในการคำนวณของ ALU

3.4 Register เป็นหน่วยความจำภายในที่ใช้รับค่าจากภายนอก หรือรับค่าผลลัพธ์จากการคำนวณภายในเพื่อรอส่งออกสู่ภายนอก

บทที่ 3
ผลการศึกษาและทดลอง

การทดลองประยุกต์ใช้งานเทคนิค Clock Gating ในงานวิจัยนี้จะทดลองใน 2 ระบบหลักดังที่ได้อธิบายไว้แล้วในบทที่ 4 คือ

- วงจร Alarm Clock โดยแบ่งเป็น 4 ชุดการทดลอง ดังนี้
  - ในส่วน Alarm Controller (ALC)
  - ในส่วนAlarm Register (ARL)
  - ในส่วน Keyboard Buffer (ALK)
  - รวมทุกส่วนของวงจร Alarm Clock (ALL)
- วงจร T80 Microprocessor (T80)

1. ผลการทดลอง Alarm Clock

เมื่อได้พิจารณาโครงสร้างและการทำงานโดยรวมของวงจร Alarm Clock แล้วพบว่ามีส่วนที่น่าจะสามารถนำเทคนิค Clock Gating เข้าไปประยุกต์ใช้เพื่อลดอัตราการใช้พลังงานได้ใน 3 ส่วนหลักคือ ในส่วนของ Alarm Controller ในส่วนAlarm Register และในส่วน Keyboard Buffer

การทดลองของวงจร Alarm Clock นี้จึงแบ่งเป็น 4 ชุดการทดลองโดย 3 ชุดแรกได้มาจากการทดลองของทั้งสามส่วนหลักของวงจรดังที่ได้กล่าวมา และชุดการทดลองสุดท้ายจะเป็นการทดลองรวมทุกส่วนของ Alarm Clock

ผลทางพลังงานเมื่อใช้โปรแกรม XPower วิเคราะห์การใช้พลังงานเมื่อได้จำลองการทำงานตามที่กำหนดไว้ในบทที่ 3 คือ 1.52 mW โดยมีรายละเอียดดังตารางที่ 1

ตารางที่ 1 ผลการใช้โปรแกรม XPower วิเคราะห์การใช้พลังงานกับวงจรต้นแบบ

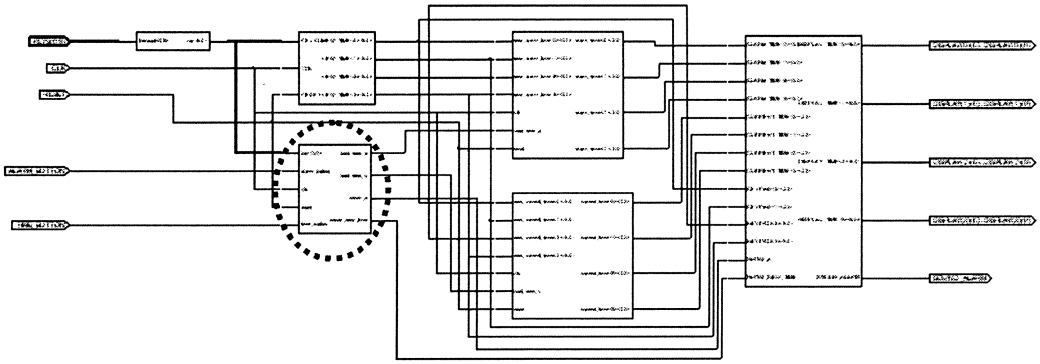
ส่วนประกอบ	พลังงาน (mW)	คำอธิบาย
Clock Power	1.36	พลังงานที่ถูกใช้โดยโครงข่ายสัญญาณนาฬิกา
Inputs Power	0.15	พลังงานที่ถูกใช้โดย Input Buffer
Logic Power	0.00	พลังงานที่ถูกใช้โดยกระบวนการทางตรรกะ
Outputs Power	0.00	พลังงานที่ถูกใช้โดย Output Buffer
Signals Power	0.00	พลังงานที่ถูกใช้โดยโครงข่ายสัญญาณภายใน

1.1 การทดลองประยุกต์ใช้เทคนิค Clock Gating ในส่วน Alarm Controller

เมื่อพิจารณาพฤติกรรมการทำงานจากโครงสร้างเดิมของวงจร Alarm Clock ซึ่งได้อธิบายไว้แล้วในบทที่ 3 จะสังเกตได้ว่าการทำงานในส่วนของ Alarm Controller นั้นจะมีการทำงานก็ต่อเมื่อมีการกดปุ่ม Alarm หรือปุ่ม Time หรือปุ่มตัวเลขใด ๆ เท่านั้น และจะทำงานต่อเนื่องไปตามขั้นตอนในโหมดการทำงานที่แตกต่างกันซึ่งขึ้นอยู่กับลำดับในการกดปุ่มจนกระทั่ง

ได้ทำงานตามที่ต้องการแล้ว Alarm Controller ก็จะหยุดการทำงานอีกครั้ง หรือจะพูดอีกนัยหนึ่งก็คือกลับเข้าสู่การทำงานในโหมดปกติเหมือนก่อนที่จะมีการกดปุ่มกระตุ้นการทำงานใด ๆ ดังนั้นจุดนี้จึงเป็นโอกาสในการทดลองลดอัตราการใช้พลังงานโดยการประยุกต์ใช้เทคนิค Clock Gating แบบต่าง ๆ เข้าไปเพื่อตัดสัญญาณนาฬิกาที่จะไปกระตุ้นการทำงานในส่วนของ Alarm Controller นี้ในช่วงโหมดการทำงานปกติซึ่ง Alarm Controller ไม่มีการทำงานและจะปล่อยสัญญาณนาฬิกาเข้าไปก็ต่อเมื่อมีการกดปุ่มกระตุ้นการทำงานใด ๆ ดังที่ได้อธิบายไปแล้วเท่านั้น

ภาพที่ 10 แสดงถึงโครงสร้างวงจร Alarm Clock เดิมก่อนที่จะมีการทดลองซึ่งถูกสังเคราะห์ขึ้นมา โดยส่วนที่เป็น Alarm Controller คือส่วนที่ถูกเน้นในกรอบ



ภาพที่ 10 วงจร Alarm Clock เดิม (Alarm Controller)

ในการทดลองแต่ละชุดการทดลองนั้น มีการประยุกต์ใช้ Clock Gating (GClk_C) แทนสัญญาณนาฬิกา (Clk) เดิม โดยปรับ Code ใน Alarm_Clock.vhd ของแต่ละชุดการทดลองดังนี้

```
U_ALARM_CONTROL: ALARM_CONTROLLER
port map(   Clk          =>    GClk_C,      -- replace Clk with GClk_C
            Reset                =>    Reset,
            Key                  =>    Key,
            Alarm_Button         =>    Alarm_Button,
            Time_Button          =>    Time_Button,
            Load_Alarm_Time      =>    Load_Alarm_Time,
            Show_Alarm_Time      =>    Show_Alarm_Time,
            Show_New_Time        =>    Show_New_Time,
            Load_New_Time        =>    Load_New_Time
        );
```

เมื่อพิจารณาถึงการทำงานในส่วน Alarm Controller พบว่าในการปรับสัญญาณนาฬิกาโดยใช้เทคนิค Clock Gating นี้ เพื่อไปใช้แทนที่สัญญาณนาฬิกาเดิม จะเกี่ยวข้องกับสัญญาณควบคุมซึ่งสามารถแบ่งออกได้เป็น 2 กลุ่ม คือ สัญญาณควบคุมจากภายนอกและสัญญาณควบคุมจากภายใน สัญญาณควบคุมจากภายนอกจะประกอบไปด้วยสัญญาณจากปุ่มตัวเลขต่าง ๆ (Key) และสัญญาณ Alarm_Button ส่วนสัญญาณควบคุมจากภายในจะประกอบไปด้วยสัญญาณ Load_Alarm_Time สัญญาณ Show_Alarm_Time สัญญาณ Show_New_Time และสัญญาณ Load_New_Time ซึ่งเป็นสัญญาณควบคุมที่เกิดจาก Alarm Controller เอง ทั้งนี้สัญญาณควบคุม

การทำงานจากภายในโดยปกติจะเป็นสัญญาณที่ใช้ควบคุมการทำงานของวงจรส่วนอื่น ๆ เพื่อให้เป็นไปตามการทำงานโหมดต่าง ๆ

ส่วนสัญญาณ Time_Button นั้นไม่ได้นำมาใช้เพราะสัญญาณนี้จะไม่ก่อให้เกิดการกระตุ้นการทำงานของ Alarm Controller เลยเมื่ออยู่ในโหมดการทำงานปกติ (สัญญาณ Time_Button จะกระตุ้นการทำงานของ Alarm Controller ก็ต่อเมื่อมีการกดปุ่มตัวเลขใด ๆ ไปก่อนนี้แล้วเท่านั้น)

เนื่องจากชุดการทดลองแต่ละชุด (ALC-01 ถึง ALC-08) มีการประยุกต์ใช้ Clock Gating ที่แตกต่างกันไป ทำให้ Code ของการได้มาซึ่งสัญญาณ GClk_C ใน Alarm_Controller.vhd ของแต่ละชุดการทดลองจึงมีการปรับที่แตกต่างกันไปดังนี้

ในกรณีของ Active High (กรณีที่เป็นเลขคี่) สัญญาณนาฬิกาจะทำงานที่ขอบขาขึ้น (Rising Edge) แต่สำหรับกรณี Active Low (กรณีที่เป็นเลขคู่) จะต้องมีการปรับจากสัญญาณนาฬิกาที่เคยทำงานที่ขอบขาขึ้น ให้ทำงานที่ขอบขาลง (Falling Edge) ดังนี้

```
elsif falling_edge(clk) then  -- from rising to falling
```

จากนั้น Code ที่เหลือของแต่ละคู่ชุดทดลอง (คู่ 01 และ 02 คู่ 03 และ 04 คู่ 05 และ 06 และคู่ 07 และ 08) จะมีลักษณะเหมือนกัน

ส่วนข้อแตกต่างของ Code ที่มีลักษณะ Logic Low Disable และ Logic High Disable คือ ในช่วงที่สัญญาณ Clock Gating ไม่ทำงานจะมีค่าเป็น 0 และ 1 ตามลำดับ ซึ่งหมายถึงการเป็นนิเสธซึ่งกันและกันนั่นเอง ดังนี้

```
GClk_C <= Clk WHEN ..... ELSE '0';
```

และ

```
GClk_C <= Clk WHEN ..... ELSE '1';
```

ตามลำดับ

และสุดท้ายคือข้อแตกต่างของ Code ที่มีลักษณะ Latch Free และ Latch Base คือ การปรับ Code ในส่วนของ GClk_C ให้มี Latch หรือ ไม่มี Latch นั่นเอง Code ที่ถูกปรับแล้วจะมีลักษณะที่แตกต่างกันดังนี้

สำหรับแบบ Latch Free เมื่อเป็นแบบ Logic Low Disable คือ

```
GClk_C <= not Clk WHEN Key /= 10 or Alarm_Button = '1' or
          Load_Alarm_Time = '1' or Show_Alarm_Time = '1' or
          Show_New_Time = '1' or Load_New_Time = '1' ELSE '0';
```

และเมื่อเป็นแบบ Logic High Disable คือ

```
GClk_C <= Clk WHEN Key /= 10 or Alarm_Button = '1' or
          Load_Alarm_Time = '1' or Show_Alarm_Time = '1' or
          Show_New_Time = '1' or Load_New_Time = '1' ELSE '1';
```

และสำหรับแบบ Latch Base เมื่อเป็นแบบ Logic Low Disable คือ

```
GClk_C <= Clk and GSignal2;
```

```
GSignal1 <= '1' WHEN Key /= 10 or Alarm_Button = '1' or
            Load_Alarm_Time = '1' or Show_Alarm_Time = '1' or
            Show_New_Time = '1' or Load_New_Time = '1' ELSE '0';

      control1 : process (CLK,RESET)
      begin
            if RESET = '1' then
                  GSIGNAL2 <= '0';
            elsif falling_edge(CLK) then
                  GSIGNAL2 <= GSIGNAL1;
            end if;
      end process control1;
```

และเมื่อเป็นแบบ Logic High Disable คือ

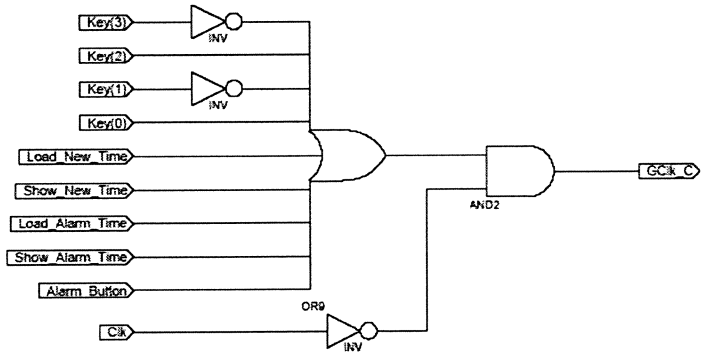
```
GClk_C <= not Clk and GSignal2;
GSignal1 <= '1' WHEN Key /= 10 or Alarm_Button = '1' or
            Load_Alarm_Time = '1' or Show_Alarm_Time = '1' or
            Show_New_Time = '1' or Load_New_Time = '1' ELSE '1';

      control1 : process (CLK,RESET)
      begin
            if RESET = '1' then
                  GSIGNAL2 <= '0';
            elsif falling_edge(CLK) then
                  GSIGNAL2 <= GSIGNAL1;
            end if;
      end process control1;
```

ผลการทดลองแสดงให้เห็นว่าสัญญาณ Gated_Clock (GClk_C) ซึ่งเป็นสัญญาณนาฬิกาใหม่ที่เข้าไปกระตุ้นการทำงานของวงจร Alarm Controller มีลักษณะแตกต่างจากสัญญาณนาฬิกาปกติก่อนที่จะทำ Clock Gating อย่างชัดเจน

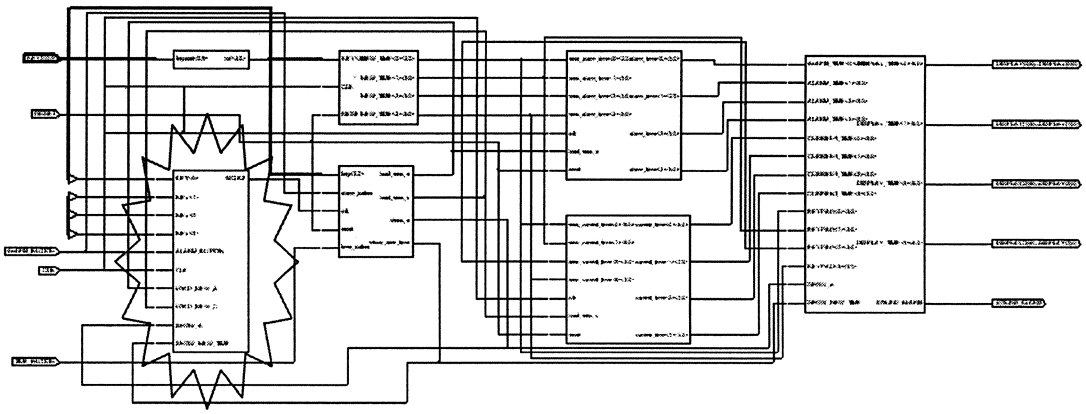
รายละเอียดของผลการทดลองของแต่ละชุดการทดลองเป็นดังนี้

<u>การทดลอง ALC-01 (Logic Low Disable + Latch Free + Active High)</u>

เมื่อทำการสังเคราะห์วงจรแล้วจะได้ผลดังแสดงในภาพที่ 12 วงจรที่ถูกล้อมด้วยกรอบคือวงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดภายในดังแสดงในภาพที่ 11



ภาพที่ 11 รายละเอียดวงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-01



ภาพที่ 12 วงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-01

ในการทดลองนี้มีผลทำให้สัญญาณ Show_New_Time ซึ่งเป็นสัญญาณที่ใช้ควบคุมการแสดงผลทำงานเร็วขึ้นในทุกกรณีทำให้การแสดงผลตัวเลขที่มีการกดแสดงเร็วกว่าปกติก่อนเวลาที่ควรจะเป็นดังที่ได้แสดงให้เห็นในภาพที่ 19 ถึง 21 และยังมีผลทำให้สัญญาณ Load_Alarm_Time และสัญญาณ Show_Alarm_Time ทำงานผิดปกติ โดยถ้ามีการกดปุ่ม Alarm Button ในช่วงสัญญาณนาฬิกาขาขึ้นจะทำให้ทั้งสองสัญญาณทำงานเร็วขึ้นกว่าปกติครึ่งรอบสัญญาณนาฬิกาและก่อให้เกิดการทำงานที่ผิดปกติด้วยดังจะสังเกตได้จากการเก็บค่าเวลาปลุก (Alarm Time) จะเกิดขึ้นเร็วกว่าปกติ แต่ถ้ากดปุ่ม Alarm Button ในช่วงสัญญาณนาฬิกาขาลงการทำงานของทั้งสองสัญญาณจะทำงานช้ากว่าปกติครึ่งรอบสัญญาณนาฬิกา

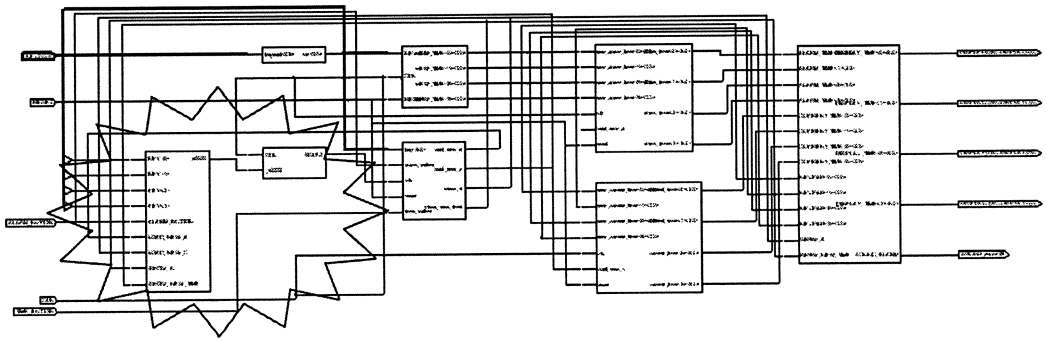
การทำ Clock Gating ในการทดลองนี้ได้ทำให้สัญญาณนาฬิกาที่ได้มี Glitch เกิดขึ้นแต่ Glitch ที่เกิดขึ้นนี้ไม่มีผลให้เกิดการทำงานที่ผิดพลาดใด ๆ นอกจากนี้วงจรการทำ Clock Gating ในการทดลองนี้ก็ยังมีผลทำให้สัญญาณ Show_New_Time ซึ่งเป็นสัญญาณที่ใช้ควบคุมการแสดงผลทำงานเร็วขึ้นในกรณีที่มีการกดปุ่มตัวเลขครั้งแรกในช่วงสัญญาณนาฬิกาขาขึ้นทำให้การแสดงผลตัวเลขที่มีการกด แสดงเร็วกว่าปกติก่อนเวลาที่ควรจะเป็น แต่ผลการทำงานที่เปลี่ยนไปนี้ก็ไม่ก่อให้เกิดผลเสียหายต่อการทำงานโดยรวมแต่ประการใด ส่วนกรณีกดปุ่มตัวเลขครั้งแรกในช่วงสัญญาณนาฬิกาขาลงการทำงานจะเป็นปกติไม่ก่อให้เกิดผลการทำงานที่เร็วขึ้น ดังที่ได้กล่าวไว้ข้างต้น

<u>การทดลอง ALC-02 (Logic Low Disable + Latch Free + Active Low)</u>

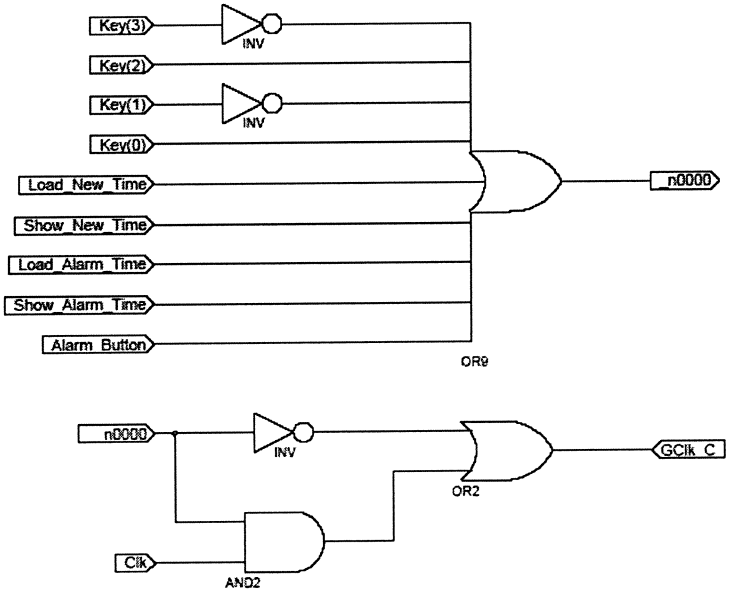
การทดลองนี้จะคล้ายกับชุดทดลองที่ ALC-01 แต่จะมีการกลับเฟสของสัญญาณนาฬิกาที่ได้และเปลี่ยนการทำงานของวงจร Alarm Controller จาก Active High เป็น Active Low มีผลทำให้การทำงานของวงจรใหม่ทำงานได้เหมือนวงจรเดิมทุกประการ อีกทั้งยังทำให้ไม่เกิด Glitch ขึ้นในสัญญาณนาฬิกาใหม่อีกด้วย เมื่อทำการสังเคราะห์วงจรแล้วจะได้วงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดเหมือนกับในการทดลอง ALC-01 ดังแสดงในภาพที่ 11 และ 12

การทดลอง ALC-03 (Logic High Disable + Latch Free + Active High)

เมื่อทำการสังเคราะห์วงจรแล้วจะได้ผลดังแสดงในรูปที่ 13 วงจรที่ถูกล้อมด้วยกรอบคือวงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดภายในดังแสดงในรูปที่ 14



ภาพที่ 13 วงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-03



ภาพที่ 14 รายละเอียดวงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-03

ผลของการทดลองแสดงให้เห็นว่า วงจรมีการทำงานเป็นปกติเช่นเดียวกับก่อนที่จะมีการทำ Clock Gating

การทดลอง ALC-04 (Logic High Disable + Latch Free + Active Low)

การทดลองนี้มีผลทำให้สัญญาณ Show_New_Time ซึ่งเป็นสัญญาณที่ใช้ควบคุมการแสดงผลทำงานเร็วขึ้นในทุกกรณีทำให้การแสดงผลตัวเลขที่มีการกดแสดงเร็วกว่าปกติก่อนเวลาที่ควรจะเป็นดังที่ได้แสดงให้เห็นในภาพที่ 19 ถึง 21 และยังมีผลทำให้สัญญาณ Load_Alarm_Time และสัญญาณ Show_Alarm_Time ทำงานผิดปกติ โดยถ้ามีการกดปุ่ม Alarm Button ในช่วงสัญญาณนาฬิกาขาขึ้นจะทำให้ทั้งสองสัญญาณทำงานเร็วขึ้นกว่าปกติครึ่งรอบสัญญาณนาฬิกาและก่อให้เกิดการทำงานที่ผิดปกติด้วยดังจะสังเกตได้จากการเก็บค่าเวลาปลุก (Alarm Time) จะ

เกิดขึ้นเร็วกว่าปกติ แต่ถ้ากดปุ่ม Alarm_Button ในช่วงสัญญาณนาฬิกาขาลงการทำงานของทั้งสองสัญญาณจะทำงานช้ากว่าปกติครึ่งรอบสัญญาณนาฬิกา

นอกจากนี้วงจรการทำ Clock Gating ยังทำให้สัญญาณนาฬิกาที่ได้มี Glitch เกิดขึ้นแต่ Glitch ที่เกิดขึ้นนี้ไม่ก่อให้เกิดผลการทำงานที่ผิดพลาดใด ๆ เพิ่มขึ้น

เมื่อทำการสังเคราะห์วงจรแล้วจะได้วงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดเหมือนกับในการทดลอง ALC-03 ดังแสดงในภาพที่ 13 และ 14

การทดลอง ALC-05 (Logic Low Disable + Latch Base + Active High)

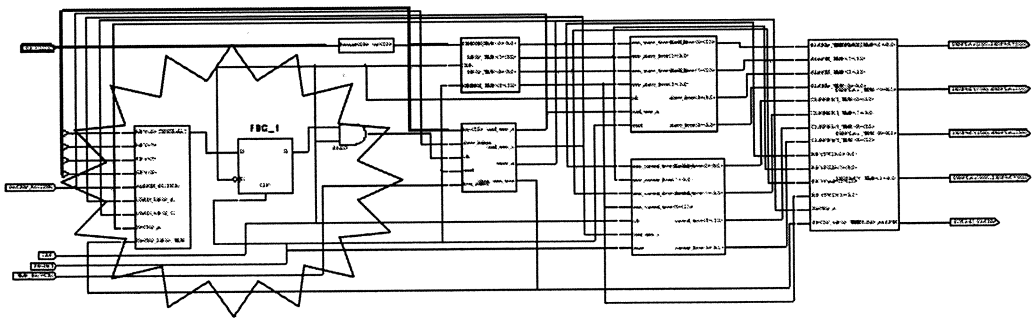
เมื่อทำการสังเคราะห์วงจรแล้วจะได้ผลดังแสดงในภาพที่ 15 วงจรที่ถูกล้อมด้วยกรอบคือวงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดภายในดังแสดงในภาพที่ 16

ผลการทดลองของวงจรในช่วงตั้งเวลาปลุกแสดงเห็นว่า มีการทำงานเป็นปกติเหมือนก่อนการทำ Clock Gating ทุกประการ แต่จริง ๆ แล้วไม่เป็นเช่นนั้นเสมอไปทั้งนี้เนื่องจากต้องมี Latch ในการคงค่าสัญญาณควบคุมในช่วงสัญญาณนาฬิกาขาลงดังนั้นสัญญาณกระตุ้น (การกดปุ่มตัวเลขหรือปุ่ม Alarm) ต้องเกิดในช่วงสัญญาณนาฬิกาขาขึ้นเท่านั้นจึงจะมีการทำงานที่ถูกต้อง แต่ถ้าสัญญาณกระตุ้นเริ่มทำงานในช่วงสัญญาณนาฬิกาขาลงการตอบสนองต่อสัญญาณกระตุ้นก็จะเกิดช้าลง 1 รอบสัญญานาฬิกาซึ่งอาจจะทำให้เกิดการทำงานที่ผิดพลาดขึ้นได้

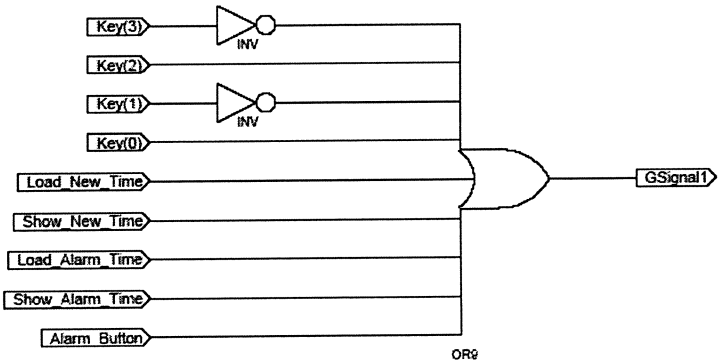
การทดลอง ALC-06 (Logic Low Disable + Latch Base + Active Low)

เมื่อทำการสังเคราะห์วงจรแล้วจะได้วงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดเหมือนกับในการทดลอง ALC-05 ดังแสดงในภาพที่ 15 และ 16

การทำ Clock Gating ในการทดลองนี้ส่งมีผลทำให้สัญญาณ Show_New_Time ซึ่งเป็นสัญญาณที่ใช้ควบคุมการแสดงผลทำงานช้าลงในทุกกรณีทำให้การแสดงผลตัวเลขที่มีการกดแสดงช้ากว่าปกติก่อนเวลาที่ควรจะเป็นดังที่ได้แสดงให้เห็นในรูปที่ ภาพที่ 19 ถึง 21 ทั้งนี้เนื่องจากต้องมี Latch ในการคงค่าสัญญาณควบคุมในช่วงสัญญาณนาฬิกาขาลงดังนั้นสัญญาณกระตุ้น (การกดปุ่มตัวเลขหรือปุ่ม Alarm) ถ้าเกิดในช่วงสัญญาณนาฬิกาขาขึ้นจะทำให้การตอบสนองต่อสัญญาณกระตุ้นช้าลง 1 รอบสัญญาณนาฬิกา และถ้าสัญญาณกระตุ้นเริ่มทำงานในช่วงสัญญาณนาฬิกาขาลงการตอบสนองต่อสัญญาณกระตุ้นก็จะเกิดช้าลง 1 รอบครึ่งสัญญานาฬิกาซึ่งจะทำให้เกิดการทำงานที่ผิดพลาดขึ้นได้ นอกจากนี้การทำ Clock Gating ตามการทดลองนี้ยังมีผลทำให้สัญญาณ Load_Alarm_Time และสัญญาณ Show_Alarm_Time ทำงานผิดปกติ โดยถ้ามีการกดปุ่ม Alarm Button ในช่วงสัญญาณนาฬิกาขาขึ้นจะทำให้ทั้งสองสัญญาณทำงานเร็วขึ้นกว่าปกติครึ่งรอบสัญญาณนาฬิกาและก่อให้เกิดการทำงานที่ผิดปกติด้วย ดังจะสังเกตได้จากการเก็บค่าเวลาปลุก (Alarm Time) ซึ่งจะเกิดขึ้นเร็วกว่าปกติ แต่ถ้ากดปุ่ม Alarm Button ในช่วงสัญญาณนาฬิกาขาลงการทำงานของทั้งสองสัญญาณจะทำงานช้ากว่าปกติครึ่งรอบสัญญาณนาฬิกา



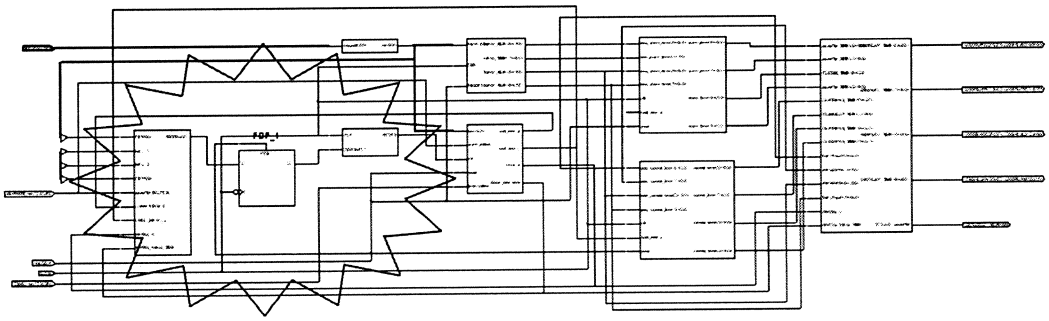
ภาพที่ 15 วงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-05



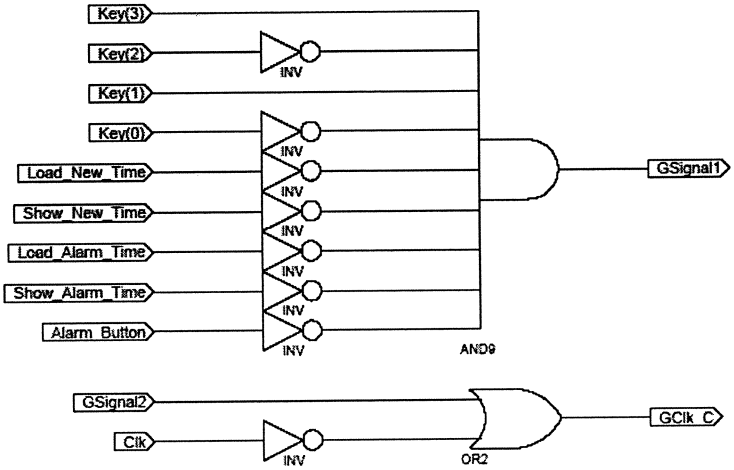
ภาพที่ 16 รายละเอียดวงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-05

<u>การทดลอง ALC-07 (Logic High Disable + Latch Base + Active High)</u>

เมื่อทำการสังเคราะห์วงจรแล้วจะได้ผลดังแสดงในภาพที่ 17 วงจรที่ถูกล้อมด้วยกรอบคือวงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดภายในดังแสดงในภาพที่ 18



ภาพที่ 17 วงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-07



ภาพที่ 18 รายละเอียดวงจรที่เพิ่มเข้ามาใน Alarm Clock ตามการทดลอง ALC-07

ผลการทดลองในชุดการทดลองนี้มีผลการทำงานที่ผิดพลาดด้วยเหตุผลเหมือนกับในชุดการทดลอง ALC-06 ที่ได้กล่าวมาแล้ว

<u>การทดลอง ALC-08 (Logic High Disable + Latch Base + Active Low)</u>

เมื่อทำการสังเคราะห์วงจรแล้วจะได้วงจรควบคุมที่เพิ่มเข้ามาซึ่งมีรายละเอียดเหมือนกับในการทดลอง ALC-07 ดังแสดงในภาพที่ 17 และ 18

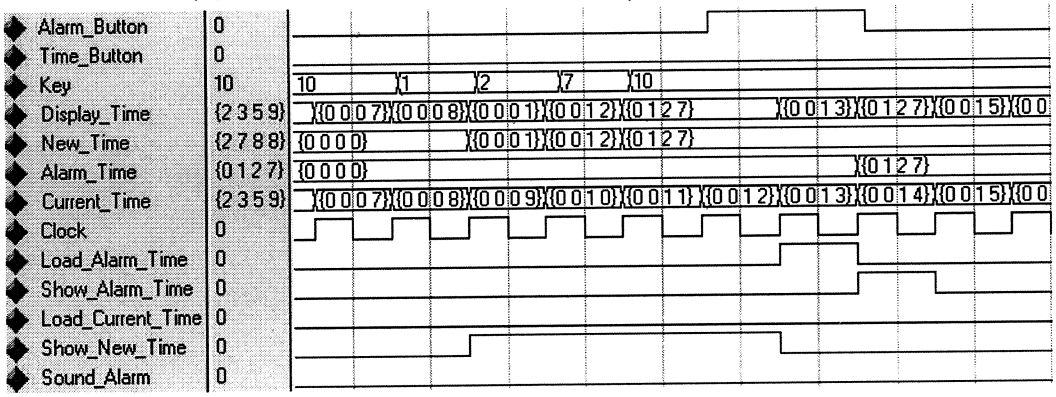
ผลการทดลองในชุดการทดลองนี้มีผลการทำงานที่ผิดพลาดด้วยเหตุผลเหมือนกับในชุดการทดลอง ALC-05 ที่ได้กล่าวมาแล้ว

สรุปผลการทดลองที่ได้จากการจำลองการทำงานทั้งหมด จะแสดงเฉพาะส่วนที่มีความเกี่ยวข้องกับการทดลอง โดยจะแยกเป็นผลจากการทดลองการทำงานในโหมดต่าง ๆ คือ

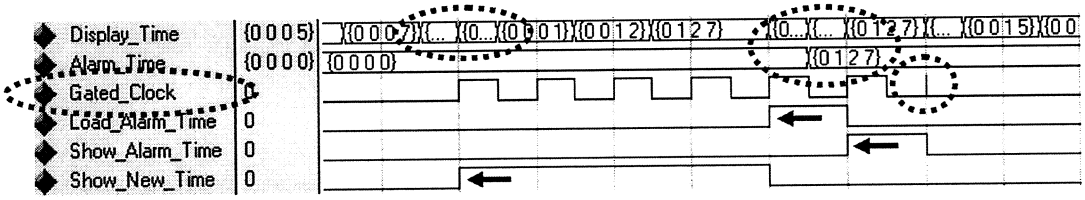
- การทำงานของวงจรในช่วงตั้งเวลาปลุก ดังแสดงในภาพที่ 19
- การทำงานของวงจรในช่วงตั้งเวลาปัจจุบัน ดังแสดงในภาพที่ 20
- การทำงานของวงจรในช่วงที่มีการกดปุ่ม Alarm_Button เพื่อดูเวลาปลุก และช่วงที่มีการกดปุ่มตัวเลขทิ้งไว้ ดังแสดงในภาพที่ 21

ซึ่งมีรายละเอียดดังต่อไปนี้

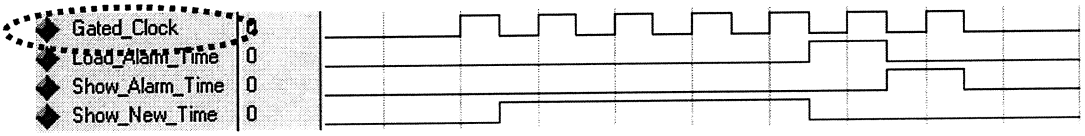
(1) การทำงานของวงจรในช่วงตั้งเวลาปลุก



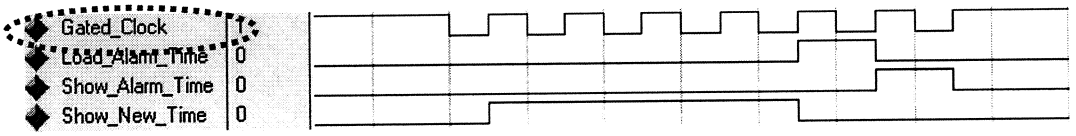
ก) ภาพสัญญาณก่อนทำ Clock Gating



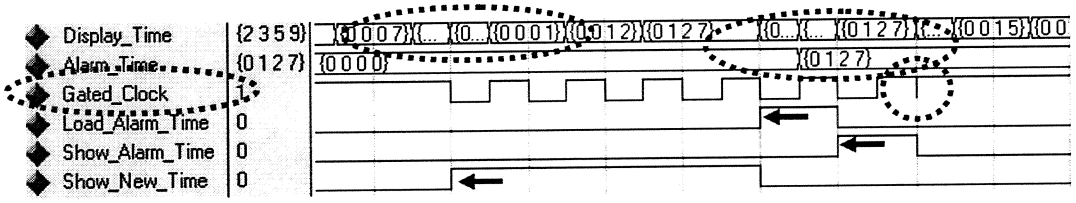
ข) ภาพสัญญาณหลังทำ Clock Gating (ALC-01)



ค) ภาพสัญญาณหลังทำ Clock Gating (ALC-02)

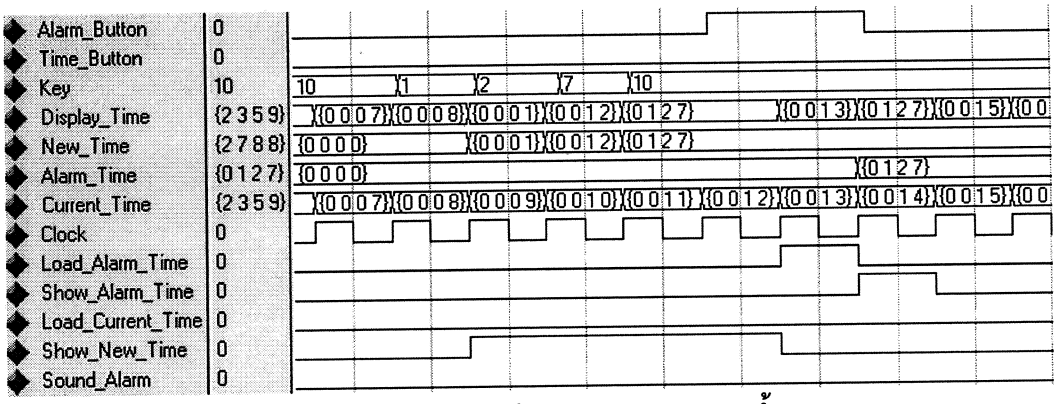


ง) ภาพสัญญาณหลังทำ Clock Gating (ALC-03)

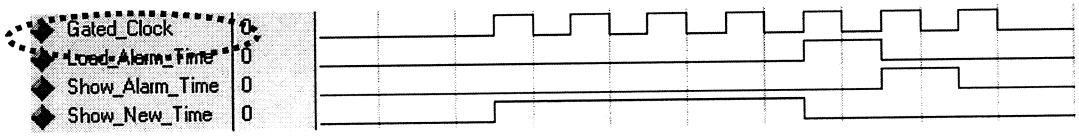


จ) ภาพสัญญาณหลังทำ Clock Gating (ALC-04)

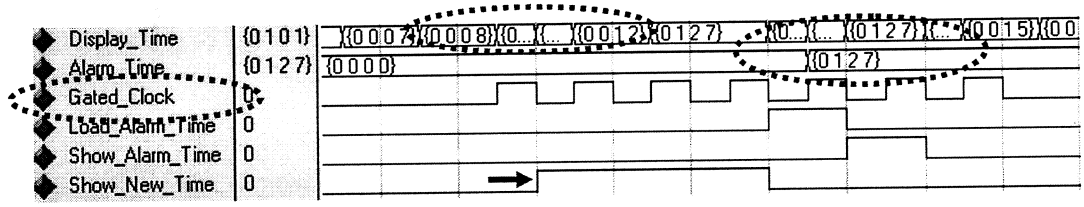
ภาพที่ 19 ผลการทดลองในช่วงตั้งเวลาปลุก



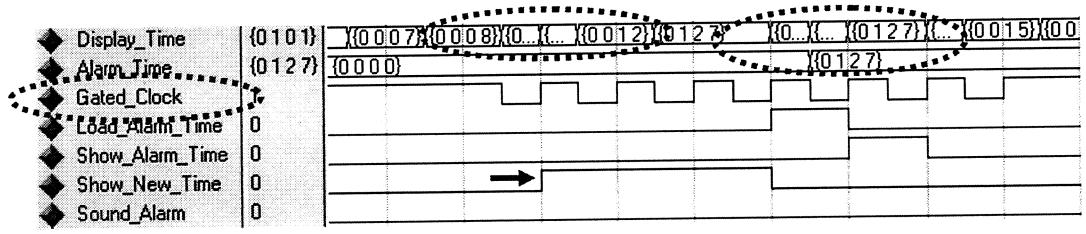
ฉ) ภาพสัญญาณก่อนทำ Clock Gating (ซ้ำ)



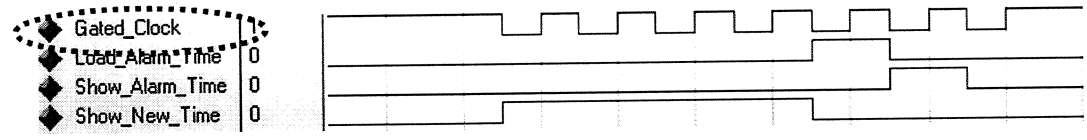
ช) ภาพสัญญาณหลังทำ Clock Gating (ALC-05)



ช) ภาพสัญญาณหลังทำ Clock Gating (ALC-06)



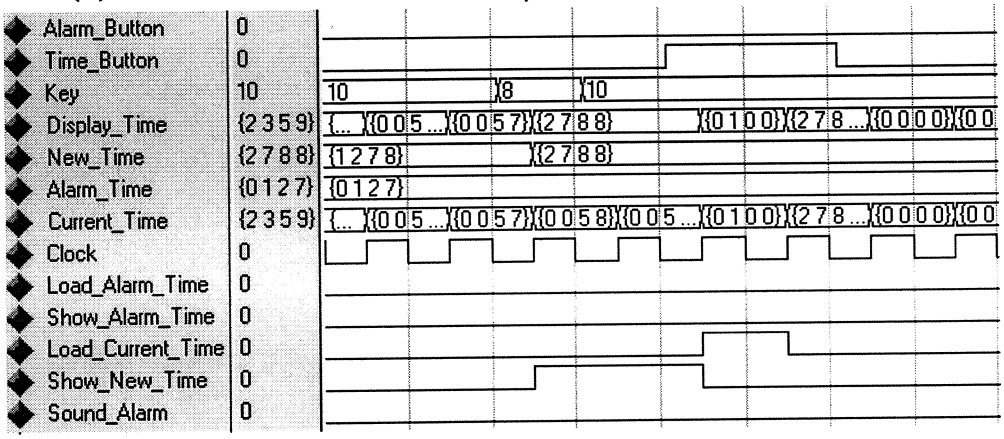
ฌ) ภาพสัญญาณหลังทำ Clock Gating (ALC-07)



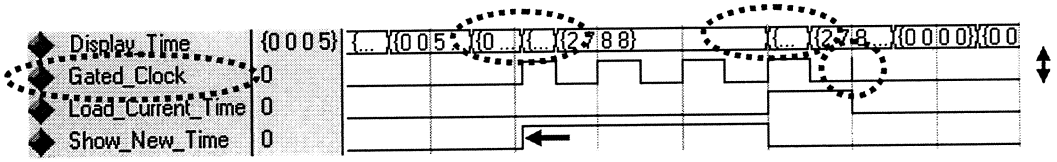
ญ) ภาพสัญญาณหลังทำ Clock Gating (ALC-08)

ภาพที่ 19 ผลการทดลองในช่วงตั้งเวลาปลุก (ต่อ)

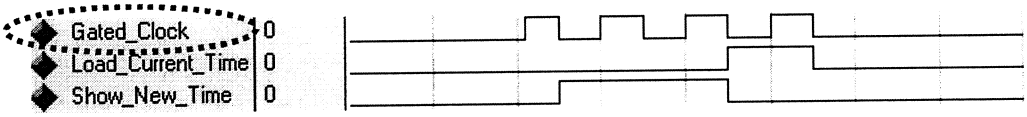
(2) การทำงานของวงจรในช่วงตั้งเวลาปัจจุบัน



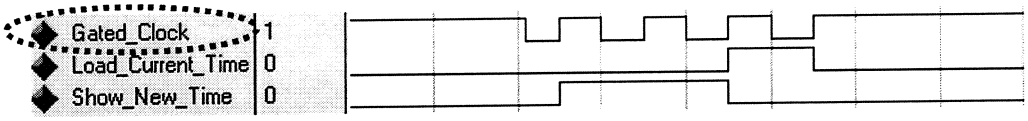
ก) ภาพสัญญาณก่อนทำ Clock Gating



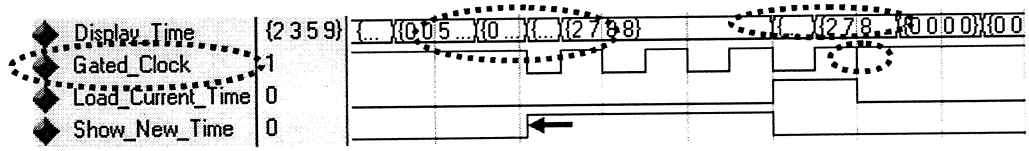
ข) ภาพสัญญาณหลังทำ Clock Gating (ALC-01)



ค) ภาพสัญญาณหลังทำ Clock Gating (ALC-02)

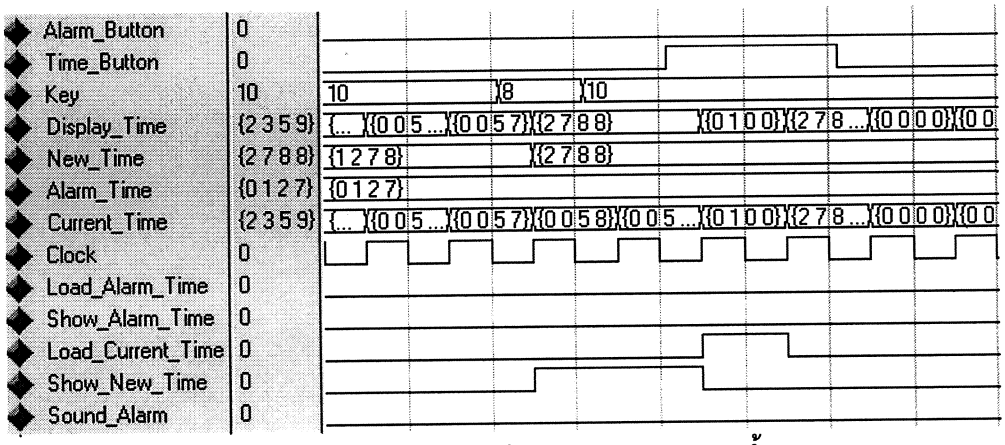


ง) ภาพสัญญาณหลังทำ Clock Gating (ALC-03)

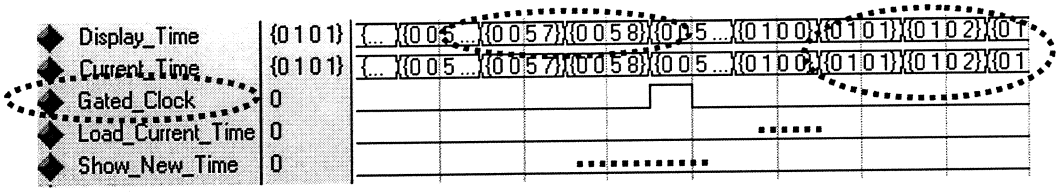


จ) ภาพสัญญาณหลังทำ Clock Gating (ALC-04)

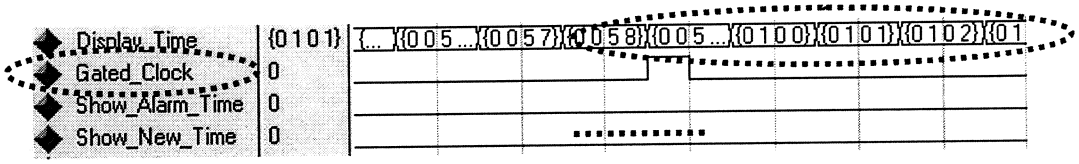
ภาพที่ 20 ผลการทดลองในช่วงตั้งเวลาปัจจุบัน



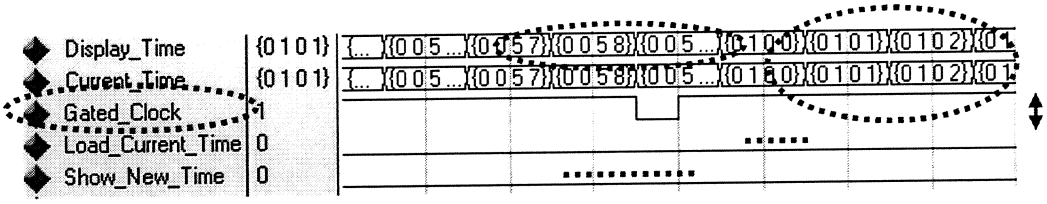
ฉ) ภาพสัญญาณก่อนทำ Clock Gating (ซ้ำ)



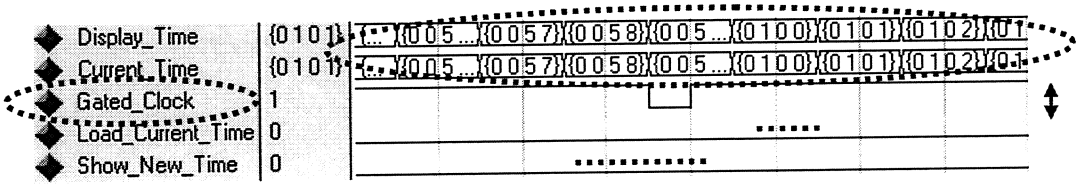
ช) ภาพสัญญาณหลังทำ Clock Gating (ALC-05)



ช) ภาพสัญญาณหลังทำ Clock Gating (ALC-06)



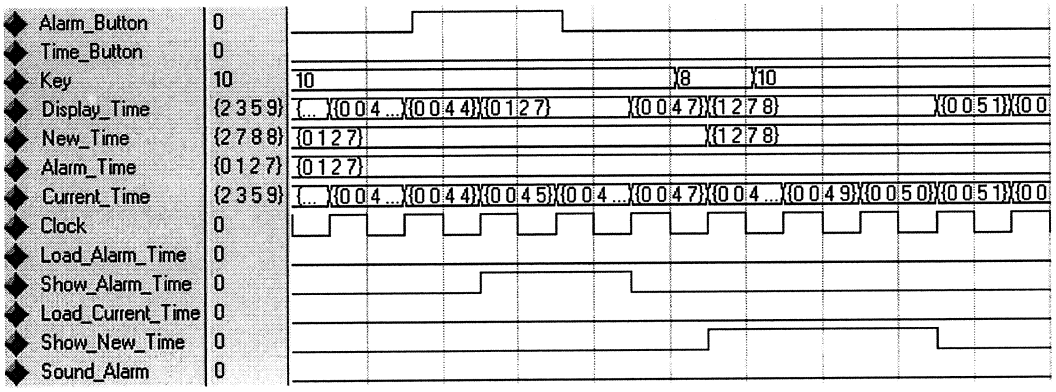
ฌ) ภาพสัญญาณหลังทำ Clock Gating (ALC-07)



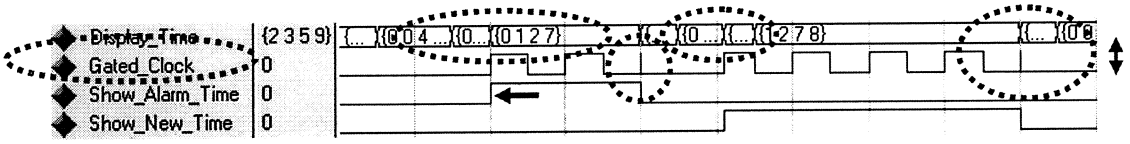
ญ) ภาพสัญญาณหลังทำ Clock Gating (ALC-08)

ภาพที่ 20 ผลการทดลองในช่วงตั้งเวลาปัจจุบัน (ต่อ)

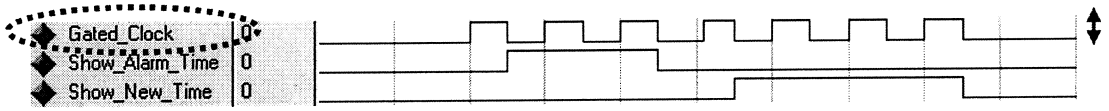
(3) การทำงานของวงจรในช่วงกดปุ่ม Alarm_Button เพื่อดูเวลาปลุก และช่วงที่กดปุ่มตัวเลขทิ้งไว้จนเลิกแสดง



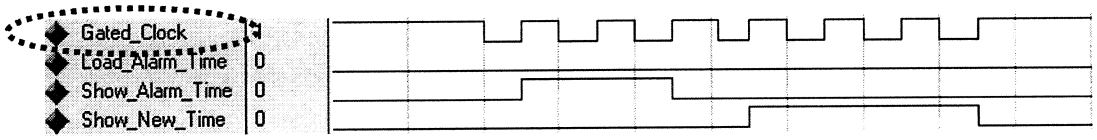
ก) ภาพสัญญาณก่อนทำ Clock Gating



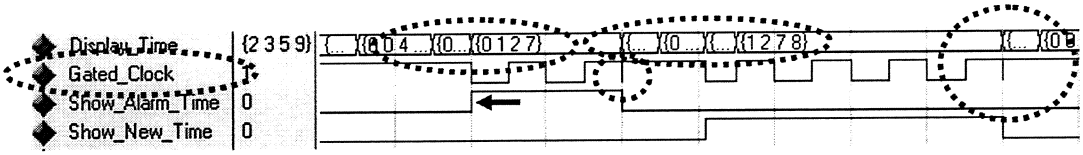
ข) ภาพสัญญาณหลังทำ Clock Gating (ALC-01)



ค) ภาพสัญญาณหลังทำ Clock Gating (ALC-02)

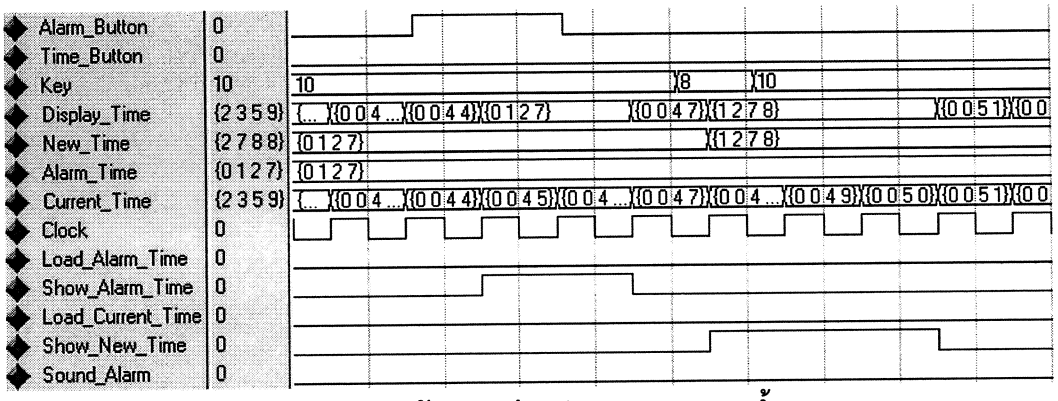


ง) ภาพสัญญาณหลังทำ Clock Gating (ALC-03)

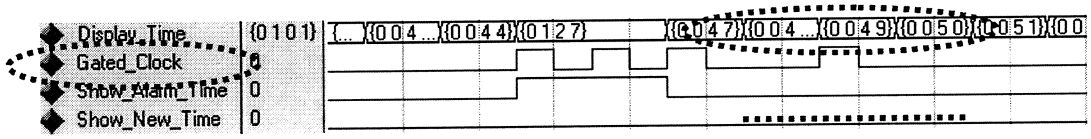


จ) ภาพสัญญาณหลังทำ Clock Gating (ALC-04)

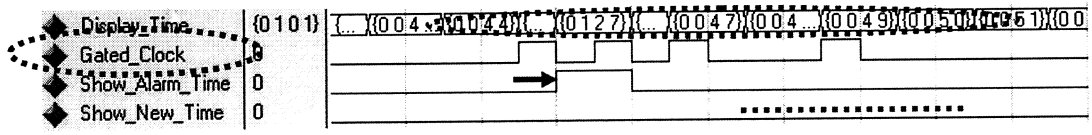
ภาพที่ 21 ผลการทดลองในช่วงกดปุ่ม Alarm_Button เพื่อดูเวลาปลุก และช่วงที่กดปุ่มตัวเลขทิ้งไว้จนเลิกแสดง



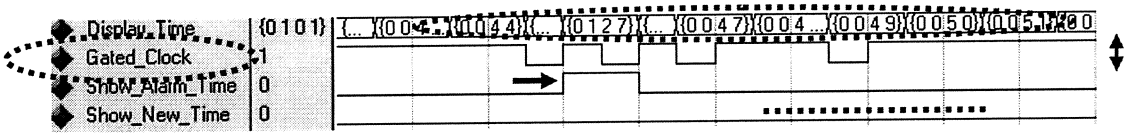
ฉ) ภาพสัญญาณก่อนทำ Clock Gating (ซ้ำ)



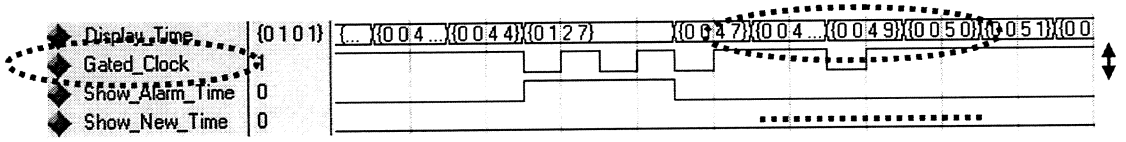
ช) ภาพสัญญาณหลังทำ Clock Gating (ALC-05)



ซ) ภาพสัญญาณหลังทำ Clock Gating (ALC-06)



ฌ) ภาพสัญญาณหลังทำ Clock Gating (ALC-07)



ญ) ภาพสัญญาณหลังทำ Clock Gating (ALC-08)

ภาพที่ 21 ผลการทดลองในช่วงกดปุ่ม Alarm_Button เพื่อดูเวลาปลุก และช่วงที่กดปุ่มตัวเลขทิ้งไว้จนเลิกแสดง (ต่อ)

ผลจากการวิเคราะห์การใช้พลังงานโดยใช้ XPower พบว่าการใช้พลังงานเป็นไปตามที่ได้ตั้งสมมุติฐานไว้โดยมีรายละเอียดดังตารางที่ 2

ตารางที่ 2 ผลการวิเคราะห์การใช้พลังงานโดยใช้ XPower ของ Alarm Controller

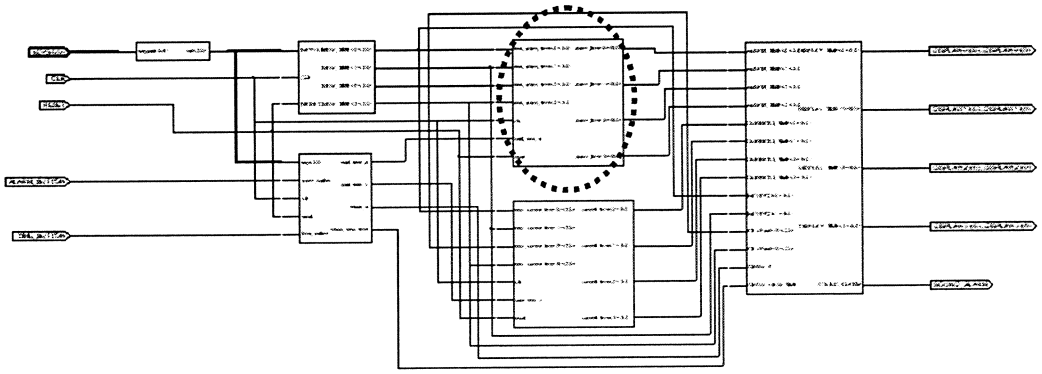
การทดลอง / รายละเอียด	ต้นแบบ	ALC01	ALC02	ALC03	ALC04	ALC05	ALC06	ALC07	ALC08
Clock Power (mW)	1.36	1.33	1.33	1.27	1.27	1.46	1.46	1.33	1.33
Inputs Power (mW)	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
Logic Power (mW)	0.00	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
Outputs Power (mW)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Signals Power (mW)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Summary (mW)	1.52	1.56	1.56	1.49	1.49	1.68	1.68	1.55	1.55
Glitch	✗	✓	✗	✗	✓	✗	✗	✗	✗
การทำงาน	☺	😐B	☺	☺	😐B	☹	☹	☹	☹

**หมายเหตุ : ☺ = ปกติ 😐B = เร็วหรือช้ากว่าปกติ ☹ = ผิดปกติ

1.2 การทดลองประยุกต์ใช้เทคนิค Clock Gating ในส่วน Alarm Register

เมื่อพิจารณาพฤติกรรมการทำงานจากโครงสร้างเดิมของวงจร Alarm Clock ซึ่งได้อธิบายไว้แล้วในบทที่ 2 จะสังเกตเห็นว่าในส่วนของ Alarm Register ซึ่งเป็นส่วนที่ใช้เก็บค่าเวลาปลุกจะมีการทำงาน (เปลี่ยนแปลงค่าเวลาปลุก) เฉพาะในการทำงานในโหมดการตั้งเวลาปลุกเท่านั้น โดยมีสัญญาณ Load_Alarm_Time เป็นสัญญาณควบคุมการทำงานดังแสดงในภาพที่ 22 ดังนั้นจุดนี้จึงเป็นโอกาสที่จะนำเทคนิค Clock Gating เข้ามาใช้ตัดสัญญาณนาฬิกาเพื่อลดการใช้พลังงานในจุดนี้

ภาพที่ 22 แสดงถึงโครงสร้างวงจร Alarm Clock เดิมก่อนที่จะมีการทดลองซึ่งถูกสังเคราะห์ขึ้นมา โดยส่วนที่เป็น Alarm Register คือส่วนที่ถูกเน้นในกรอบ



ภาพที่ 22 วงจร Alarm Clock เดิม (Alarm Register)

ในการทดลองแต่ละชุดการทดลองนั้น มีการประยุกต์ใช้ Clock Gating (GClk_A) แทนสัญญาณนาฬิกา (Clk) เดิม โดยปรับ Code ใน Alarm_Clock.vhd ของแต่ละชุดการทดลอง ดังนี้

```
U_ALARM_REG : ALARM_REG
port map (  Clk   => GClk_A,        -- replace Clk with Clk_A
            New_Alarm_Time   => New_Time,
            Load_Alarm_Time  => Load_Alarm_Time,
            Reset            => Reset,
            Alarm_Time       => Alarm_Time
         );
```

เนื่องจากชุดการทดลองแต่ละชุด (ALR-01 ถึง ALR-08) มีการประยุกต์ใช้ Clock Gating ที่แตกต่างกันไป ทำให้ Code ของการได้มาซึ่งสัญญาณ GClk_A ใน Alarm_Clock.vhd และใน Alarm_Controller.vhd ของแต่ละชุดการทดลองจึงมีการปรับที่แตกต่างกันไปดังนี้

ในกรณีของ Active High (กรณีที่เป็นเลขคี่) สัญญาณนาฬิกาจะทำงานที่ขอบขาขึ้น (Rising Edge) แต่สำหรับกรณี Active Low (กรณีที่เป็นเลขคู่) จะต้องมีการปรับจากสัญญาณนาฬิกาที่เคยทำงานที่ขอบขาขึ้น ให้ทำงานที่ขอบขาลง (Falling Edge) โดยเปลี่ยนแปลง Code ใน Alarm_Controller.vhd ดังนี้

```
elsif falling_edge(clk) and Load_Alarm_Time = '1' then
```

จากนั้น Code ที่เหลือของแต่ละคู่ชุดทดลอง (คู่ 01 และ 02 คู่ 03 และ 04 คู่ 05 และ 06 และคู่ 07 และ 08) จะมีลักษณะเหมือนกัน

ส่วนข้อแตกต่างของ Code ที่มีลักษณะ Logic Low Disable และ Logic High Disable คือ ในช่วงที่สัญญาณ Clock Gating ไม่ทำงานจะมีค่าเป็น 0 และ 1 ตามลำดับ ซึ่งหมายถึงการเป็นนิเสธซึ่งกันและกันนั่นเอง และสุดท้ายคือข้อแตกต่างของ Code ที่มีลักษณะ Latch Free และ Latch Base คือ การปรับ Code ในส่วนของ GClk_A ให้มี Latch หรือ ไม่มี Latch นั่นเอง Code ที่ถูกปรับแล้วจะมีลักษณะที่แตกต่างกันดังนี้

สำหรับแบบ Latch Free เมื่อเป็นแบบ Logic Low Disable คือ

```
GClk_A <= not Clk and Load_Alarm_Time;     -- C'A
```

และเมื่อเป็นแบบ Logic High Disable คือ

```
GClk_A <= Clk or not Load_Alarm_Time;      -- !(C'A) = C+A'
```

และสำหรับแบบ Latch Base เมื่อเป็นแบบ Logic Low Disable คือ

```
GClk_A <= Clk and GSignal0;

control0 : process (Clk,Reset)
begin
      if Reset = '1' then
            GSignal0 <= '0';
      elsif falling_edge(Clk) then
            GSignal0 <= Load_Alarm_Time;
      end if;
end process control0;
```