รายงานวิจัยฉบับสมบูรณ์

โครงการ “ระบบเชื่อมโยงมาตรฐาน สำหรับ โปรแกรมวิเคราะห์ และ จำลอง การทำงานของวงจรรวมแบบผสม วงจรย่อย อนาลอก ดิจิทัล และ วงจรย่อยระดับสูง”

โดย

ดร. ชูเกียรติ การะเกตุ

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

มีนาคม 2547

สัญญาเลขที่ PDF/27/2541

รายงานวิจัยฉบับสมบูรณ์

โครงการ "ระบบเชื่อมโยงมาตรฐาน สำหรับ โปรแกรมวิเคราะห์ และ จำลอง การทำงานของวงจรรวมแบบผสม วงจรย่อย อนาลอก ดิจิทัล และ วงจรย่อยระดับสูง"

โดย

หัวหน้าโครงการ

ดร. ชูเกียรติ การะเกตุ *คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์*

ที่ปรึกษาโครงการ

Dr. Mark Zwolinski *Department of Electronics and Computer Science*
University of Southampton

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

กิตติกรรมประกาศ

โครงการวิจัย ระบบเชื่อมโยงมาตรฐาน สำหรับ โปรแกรมวิเคราะห์ และ จำลอง การทำงานของวงจรรวมแบบผสมวงจรย่อย อนาลอก ดิจิทัล และวงจรย่อยระดับสูง ได้ดำเนินการประสบความสำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือ และการสนับสนุนจากหลายฝ่าย

ขอขอบคุณ Dr. Mark Zwolinski ผู้ให้คำปรึกษาข้อแนะนำทางด้านเทคนิคและวิชาการ นิสิตปริญญาโทหลายท่านที่ได้ช่วยทดสอบโปรแกรม และค้นหาข้อผิดพลาดเพื่อได้รับการแก้ไขให้ถูกต้อง สำนักงานกองทุนสนับสนุนการวิจัย ผู้ให้การสนับสนุนงบประมาณ และให้คำปรึกษาการดำเนินงาน

ในโอกาสนี้ผู้จัดทำโครงการวิจัยขอขอบพระคุณทุกๆท่านที่สำนักงานกองทุนสนับสนุนการวิจัย มีส่วนช่วยให้งานวิจัยนี้ประสบความสำเร็จได้ด้วยดี

ชูเกียรติ การะเกตุ
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

สัญญาเลขที่ PDF/27/2541

โครงการ "ระบบเชื่อมโยงมาตรฐาน สำหรับ โปรแกรมวิเคราะห์ และ จำลองการทำงานของวงจรรวมแบบผสมวงจรย่อย อนาลอก ดิจิทัล และ วงจรย่อยระดับสูง"

จัดทำโดย

ดร. ชูเกียรติ การะเกตุ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเกษตรศาสตร์ บางเขน กรุงเทพฯ **e-mail** fengcgg@ku.ac.th

Dr. Mark Zwolinski Department of Electronics and Computer Science

University of Southampton England

มิถุนายน 2541 – กรกฎาคม 2543

โครงการวิจัยนี้มีวัตถุประสงค์ที่จะศึกษา และพัฒนาวิธีการเพื่อนำมาใช้ในการวิเคราะห์วงจรรวมที่ประกอบด้วยภาคอนาลอกและดิจิทัลรวมอยู่ในไอซีชิพเดียวกัน วิธีการที่นำเสนอเป็นการขยายขอบเขตของการจำลองวงจรรวมแบบตัวจำลองการทำงานเดี่ยว มาเป็นแบบหลายตัว โดยกำหนดมาตรฐานการเชื่อมต่อที่ใช้ได้กับทั้งตัวจำลองการทำงานแบบอนาลอกและดิจิทัล

แผงเชื่อมโยงตัวจำลองการทำงาน ที่พัฒนาขึ้นในโครงการเป็นสภาพแวดล้อมที่ใช้ประสานการทำงานของตัวจำลองการทำงานแบบต่างๆ เพื่อให้สามารถจำลองการทำงานของวงจรรวมแบบผสม อนาลอก และ ดิจิทัล ในสภาพแวดล้อมของแผงเชื่อมโยงนี้ ตัวจำลองการทำงานของวงจรรวมแต่ละตัว สามารถที่จะทำงานแยกกันเป็นขบวนการ และ อัลกอริธึม ที่อิสระจากกัน โดยอาศัยแผงเชื่อมโยงเป็นตัวส่งผ่านข้อมูลระหว่างกัน แผงเชื่อมโยงจะมีหน้าที่หลักในการควบคุมให้แต่ละตัวจำลองที่มาเชื่อมต่อสามารถทำงานร่วมกันได้ และสามารถที่จะสร้างผลการจำลองการทำงานที่ถูกต้อง โดยตัวจำลองที่นำมาเชื่อมต่อแต่ละตัวจะถูกควบคุมให้ทำงานสอดคล้องกันด้วยวิธี ล็อก-สเตปประยุกต์ ข้อมูลที่แลกเปลี่ยนกันระหว่างวงจรส่วน อนาลอก และ ดิจิทัล จะส่งผ่านการแปลงข้อมูลแบบ เทรชโฮลด์-ฟังก์ชั่น และ บูลีน-คอนโทรล-สวิทช์ ข้อมูลที่ส่งผ่านระหว่างอนาลอก กับ อนาลอก ด้วยกันเอง จะใช้เทคนิคการตรวจสอบ ลาเทนซี่

แผงเชื่อมโยงตัวจำลองการทำงานนี้ประสบความสำเร็จในการเชื่อมโยงตัวจำลองการทำงานทั้งแบบ อนาลอก ดิจิทัล และ แบบบรรยายพฤติกรรมระดับสูง และได้นำไปใช้งานจริงกับวงจรประยุกต์แบบผสมสัญญาณที่มีการป้อนกลับระหว่างวงจรส่วนต่างๆ ซึ่งผลที่ได้จากการทดลองได้แสดงให้เห็นถึงความสามารถ และ ประสิทธิภาพของเทคนิคที่ได้นำเสนอ

ข้อจำกัดสำคัญของการนำไปประยุกต์ใช้คือ ความเร็วในการประมวลผลที่ยังต้องปรับปรุง อย่างไรก็ดี แผงเชื่อมโยงตัวจำลองที่นำเสนอ มีข้อดีคือง่ายในการพัฒนา ให้ผลถูกต้องเชื่อถือได้ เป็นระบบเปิดที่สามารถขยายผลพัฒนาต่อไปได้ และ สามารถปรับประยุกต์ใช้กับปัญหาวิเคราะห์วงจรได้หลากหลายรูปแบบ

คำสำคัญ จำลองการทำงาน วงจรรวม อนาลอก ดิจิทัล ผสมสัญญาณ

Abstract

Contract Number PDF/27/2541

Project " Standard Interface System for Integrated Circuit Simulators of Mixed Analogue Digital and High-level Subcircuits(Simulation Backplane) "

By

Dr. Chugiat Garagate Department of Electrical Engineering Faculty of Engineering Kasetsart University Bangkok 10900 **e-mail** fengcgg@ku.ac.th

Dr. Markk Zwolinski Department of Electronics and Computer Science University of Southampton England

June 1998 - July 2000

The objective of this research project is to study and develop a technique for the simulation of Integrated Circuits(IC) with mixed analogue and digital parts. The technique proposed is to change the simulation methodology that employs a single simulator to that uses multiple simulators. The standard interface machanism is used to combine operations of both analogue and digital circuit simulators.

Simulation Backplane is an integrated simulation environment that combines multiple simulators of different abstraction levels to perform mixed-signal circuit simulation. In the backplane environment, circuit simulators run as separate processes, using their own algorithm and exchanging simulation data through the backplane. A consistent mixed-signal simulation is obtained by using backplane controls. The Applied Lock-step algorithm was adopted to synchronise the simulation processes. Digital and Analogue simulators exchange simulation data by employing a standardised signal conversion scheme, which is based on threshold functions and Boolean-controlled switches. Latency detection technique is used to define analogue events.

Several circuit simulators were integrated into the environment, including a logic simulator, a behavioural simulator and a de-facto analog circuit level simulator. Simulations of practical mixed-signal applications including circuits that contained feedback loops were carried out successfully. The results convincingly showed the capability and flexibility of the approach.

Limitation of this scheme is the computation speed which needs to be improved. However, the advatages of the technique are its ease of development, the reliable simulation results, open systems and also applicable to varities of circuit problems.

Keywords simulation, analogue, digital, mixed-signal, integrated circuits

สัญญาเลขที่ PDF41-ชูเกียรติ การะเกตุ

หน้าสรุปโครงการ (Executive Summary)

ทุนวิจัยหลังปริญญาเอก

ชื่อโครงการ(ภาษาไทย) ระบบเชื่อมโยงมาตรฐาน สำหรับ โปรแกรมวิเคราะห์ และ จำลองการทำงานของวงจรรวมแบบผสมวงจรย่อย อนาลอก ดิจิตอล และ วงจรย่อยระดับสูง

ชื่อโครงการ(ภาษาอังกฤษ) Standard Interface System for Integrated Circuit Simulators of Mixed Analogue-Digital and High-level Subcircuits.(Simulation Backplane)

ชื่อหัวหน้าโครงการ ดร. ชูเกียรติ การะเกตุ

Dr. Chugiat Garagate

สถานที่ติดต่อ :

ที่ทำงาน ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

เขต บางเขน จังหวัด กรุงเทพมหานคร รหัสไปรษณีย์ 10900

โทรศัพท์ 02942-8555 ต่อ 1503 โทรสาร 02942-8555 ต่อ 1550

E-mail fengcgg@ku.ac.th

ชื่อที่ปรึกษาโครงการ ดร. มาร์ค ซอวลินสกี้

Dr. Mark Zwolinski

สถานที่ติดต่อ :

ที่ทำงาน Department of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ ENGLAND

โทรศัพท์ +44 1703 593528 โทรสาร +44 1703 592901

E-mail: mz@ecs.soton.ac.uk

สาขาวิชาที่ทำการวิจัย Very Large Scale Integrated Circuits (VLSI) and Microelectronics

งบประมาณทั้งโครงการ 300,000 บาท

ระยะเวลาดำเนินงาน 2 ปี

ปัญหาที่ทำการวิจัย และความสำคัญของปัญหา

เทคโนโลยีวงจรรวมขนาดใหญ่มาก(Very Large Scale Integrated Circuits, VLSI) มีบทบาทสำคัญอย่างยิ่งใน อุตสาหกรรมการผลิต อุปกรณ์ เครื่องใช้ และ วงจรอิเลกทรอนิกส์ วงจรรวม(Integrated Circuits) หรือ ไอซี(IC) มีการนำไปใช้อย่างแพร่หลายไม่ว่าจะเป็น คอมพิวเตอร์ เครื่องมือสื่อสาร เครื่องยนต์ และอื่นๆ

เทคโนโลยีการออกแบบ และ ผลิตวงจร แบ่งได้เป็นสองกลุ่มคือ เครื่องมือวางแบบวงจร (Circuit Layout Tools) และ เครื่องมือจำลองการทำงานวงจร (Circuit Simulation Tools) ด้วยศักยภาพของงานพัฒนาด้านนี้ ทำให้เกิดการพัฒนาวงจรรวมที่ทำงานซับซ้อนมากขึ้น ประสิทธิภาพสูงขึ้น ในขณะที่ขนาดวงจรเล็กลง จึงนับได้ว่าเครื่องมือออกแบบวงจรเป็นสิ่งที่สำคัญอย่างยิ่งในการพัฒนาวงจรรวมขนาดใหญ่มาก ที่ขนาดความซับซ้อน และความเร็วสูงขึ้น

วงจจรวมในปัจจุบันเพิ่มความซับซ้อนมากยิ่งขึ้น โดยรวมเอาส่วนประกอบวงจรทั้งที่เป็น อนาลอก และดิจิทัลไว้ด้วยกัน ตัวอย่างเช่นวงจรสื่อสารไร้สายต่างๆที่ได้ใช้ส่วนประมวลผลทั้งอนาลอกและดิจิทัลในวงจรรวมเดียวกัน วงจรประเภทนี้จัดอยู่ในกลุ่มวงจรรวมแบบผสมสัญญาณ (Mixed-Signal Integrated Circuits)

ปัญหาสำคัญในการออกแบวงจรรวมแบบผสมสัญญาณคือ การจำลองการทำงานวงจร (Circuit Simulaiton) ซึ่งในปัจจุบันตัวจำลองการทำงานวงจร(Circuit Simulator) ที่ใช้งานถูกออกแบบมาเพียงเพื่อจำลองการทำงานของวงจรดจิทัลล้วน (Digital Only) หรืออนาลอกล้วน (Analogue Only) ทำให้การนำมาใช้งานในวงจรรวมแบบผสมสัญญาณ มีปัญหาประสิทธิภาพต่ำ ผลการจำลองไม่ถูกต้อง ทำงานช้า หรือ ไม่สามารถใช้งานได้

เนื่องจากข้อจำกัดของตัวจำลอง และวงจรที่ต้องการทดสอบในปัจจุบันมีแนวโน้ม ที่จะผสมรวมเอาอนาลอกและดิจิทัลไว้ด้วยกันมากขึ้น จึงเกิดแนวคิดที่จะประสานการทำงานของตัวจำลองต่างชนิดเพื่อทำงานร่วมกัน โดยให้อิสระแก่ตัวจำลองแต่ละตัวที่จะทำงานเฉพาะด้านโดยใช้แบบจำลองวงจร และอัลกอริธึมที่ได้พัฒนาแล้ว ทำงานร่วมไปพร้อมกับตัวจำลองอื่นๆ โดยแลกเปลี่ยนข้อมูลเพื่อรักษาให้ผลการจำลองทั้งระบบให้ถูกต้อง

วัตถุประสงค์ของโครงการ

โครงการนี้นำเสนอวิธีการที่จะขยายขอบเขตของการวิเคราะห์วงจรรวมแบบตัวจำลองการทำงานเดี่ยว(Single Simulator) มาเป็นแบบ หลายตัวจำลองการทำงาน (Multiple Simulators) เชื่อมโยงและทำงานไปพร้อมกัน โดยมีวัตถุประสงค์ที่จะแก้ไข ปัญหาของการจำลองการทำงานของวงจรรวม ซึ่งประกอบไปด้วยส่วนที่เป็น อนาลอก และ ดิจิทัล วิธีการที่นำเสนอ จะประสานการทำงานของตัวจำลองการทำงาน และจะเป็นมาตรฐานสำหรับเชื่อมต่อตัวจำลองการทำงานแบบใดๆก็ได้ โดยโครงการนี้จะพัฒนา เทคนิค และ ระบบเชื่อมโยงมาตรฐาน ซึ่งสามารถรวมเอา โปรแกรมจำลองการทำงานของวงจรไฟฟ้าแบบอนาลอก ดิจิทัล และวงจรย่อยระดับสูง เข้าทำงานด้วยกัน วัตถุประสงค์ของโครงการสามารถแยกเป็นข้อย่อยดังนี้

- ศึกษาและเผยแพร่เทคโนโลยีการทำงาน ของตัวจำลองการทำงานแบบต่างๆ
- ศึกษาและพัฒนาส่วนเชื่อมต่อ(Interface) สำหรับตัวจำลองการทำงานให้สามารถทำงาน ร่วมกันได้
- ศึกษาและวิจัยหาเทคนิคที่จะนำมาใช้เชื่อมโยง ตัวจำลองการทำงานแบบอนาลอกหลายตัว (Multiple Analogue Simulators)
- ปรับปรุงระบบการวิเคราะห์วงจรรวมด้วยตัวจำลองหลายตัวให้มีประสิทธิภาพดี ทำงานรวดเร็ว และ ให้ผลถูกต้องแม่นยำ

ระเบียบวิธีวิจัย

ทฤษฎีการจำลองการทำงานวงจรรวมขนาดใหญ่มาก

ตัวจำลองวงจรสำหรับวงจร VLSI แบ่งได้ตามลักษณะการประยุกต์ใช้งาน และอัลกอริธึมของโปรแกรมเป็น 3 ประเภทคือ

- ตัวจำลองวงจรอนาลอก (Analogue Circuit Simulator)
- ตัวจำลองวงจรดิจิทัล (Digital Circuit Simulator)
- ตัวจำลองวงจรผสมสัญญาณอนาลอกและดิจิทัล (Mixed-Signal Circuit Simulator)

ตัวจำลองวงจรอนาลอก

ตัวจำลองวงจรอนาลอก ใช้วิธีทางคณิตศาสตร์เพื่อแก้ปัญหาวงจร หาคำตอบของ ค่ากระแส ค่าแรงดัน และการตอบสนองของวงจร ซึ่งประกอบด้วยอุปกรณ์ไฟฟ้าและ อิเลกทรอนิกส์พื้นฐานได้แก่ ตัวต้านทาน ตัวเก็บประจุ ไดโอด ทรานซิสเตอร์ชนิดไบโพลาร์ และมอส รวมไปถึงอุปกรณ์เชิงเส้น และไม่เชิงเส้น อีกทั้งแหล่งจ่ายกระแสหรือแรงดันแบบอิสระ และแบบแปรค่ากับตัวแปร อุปกรณ์ไฟฟ้าอิเลกทรอนิกส์เหล่านี้ จะถูกแทนด้วยสมการคุณลักษณะ (Device Characteristic Equation) และสมการของอุปกรณ์ทุกตัวจะถูกรวมนำมาสร้างเป็นสมการระบบ(System Equations) เพื่อใช้ในการวิเคราะห์หาคำตอบที่ต้องการจากวงจรต่อไป

ตัวจำลองวงจรระดับเกทลอจิก (Gate-Level Simulation)

สำหรับวงจรดิจิทัลที่ประกอบขึ้นจากเกทลอจิก เช่น แอนด์ (AND) ออร์ (OR) แนนด์ (NAND) นอร์ (NOR) การจำลองการทำงานจะใช้กระบวนการที่ต่างจากการจำลองวงจรอิเล็กทรอนิกส์ระดับทรานซิสเตอร์ โปรแกรมจำลองวงจรลอจิก(Logic Simulator) เป็นโปรแกรมคอมพิวเตอร์ที่ใช้ตรวจสอบการทำงานของวงจรดิจิทัลโดยเฉพาะ วิธีการจำลองระดับเกทจะคำนวณสถานะลอจิกของวงจรและควบเวลาของสัญญาณต่างๆในวงจร เมื่อวิเคราะห์ในระดับเกท อุปกรณ์ต่างๆในวงจรจะทำงานกับสัญญาณไฟฟ้าที่แบ่งเป็นระดับลอจิกเท่านั้น

ตัวจำลองแบบผสมอนาลอก และดิจิทัล (Mixed-Signal Simulator)

วงจรที่มีส่วนประกอบทั้งอนาลอก และดิจิทัลอยู่รวมกันต้องอาศัยตัวจำลองแบบผสมสัญญาณ เพื่อทดสอบการทำงานของวงจรประเภทนี้ ตัวจำลองแบบผสมสัญญาณ มีข้อดีในด้านความถูกต้องของผลลัพธ์ ความสามารถครอบคลุมวงจรหลากหลาย และความเร็วในการประมวลผล

ในการพัฒนาตัวจำลองแบบผสมสัญญาณ มีวิธีการที่ใช้อยู่ 3 แบบ คือ

1. Core Mixed Mode Simulator
2. Unified Mixed Mode Simulator
3. Glued Mixed mode Simulator

ระบบเชื่อมโยงตัวจำลองการทำงานแบบผสมสัญญาณ

(Mixed-Signal Simulation Backplane)

วิธีการที่นำเสนอในงานวิจัยนี้คือ แผงเชื่อมตัวจำลอง(Simulation Backplane) ซึ่งใช้หลักการเดียวกันกับ Glued Mixed-Mode Simulator โดยพัฒนาโปรแกรมเป็นส่วนเชื่อมโยงมาตรฐาน (Standard Interface) ที่ไม่จำกัดชนิด หรืออัลกอริธึม หรือแม้แต่คุณสมบัติของตัวจำลองที่จะเชื่อมรวมกันเข้าหาตัวระบบ

แผงเชื่อมตัวจำลองนี้พัฒนาขึ้นเป็นโปรแกรมสำเร็จ และได้กำหนดมาตรฐานการติดต่อระหว่างตัวจำลอง ซึ่งตัวจำลองใดๆก็ตาม ไม่ว่าจะเป็น อนาลอก ดิจิทัล จากบริษัทใดๆ จะสามารถนำมาเชื่อมต่อเข้ากับระบบ และทำงานร่วมกันกับตัวจำลองอื่นๆ ได้

โครงสร้างของแผงเชื่อมต่อตัวจำลอง

แผงเชื่อมต่อตัวจำลอง ประกอบไปด้วยชุดโปรแกรม ซึ่งสามารถจะเชื่อมโยง เข้ากับตัวจำลองใดๆ โดยผ่านทาง Application Programing Interface (API) ชุดของคำสั่ง API ที่พัฒนาขึ้นนี้ สามารถติดต่อและควบคุมการทำงานของตัวจำลองที่เข้ามาเชื่อมต่อ ไม่ว่าจะเป็นอนาลอก ดิจิทัล หรืออื่นๆ ส่วนประกอบสำคัญของแผงเชื่อมโยงตัวจำลองมีดังนี้

a. ***User Interface*** เป็นส่วนสั่งการที่ผู้ใช้จะกำหนดรูปแบบวงจร
b. ***Simulation Backplane*** เป็นแผงเชื่อมต่อและส่วนแกนของระบบ
c. ***Waveform Tool*** เป็นส่วนการนำเสนอสัญญาณที่เป็นผลลัพธ์
d. ***Socket Interface*** เป็นช่องทางการส่งต่อข้อมูลสัญญาณระหว่างตัวจำลอง
e. ***Spice3e2 : Logic Simulation และ: Behavioural Simulator*** เป็นตัวจำลองการทำงานวงจรอิเลกทรอนิกส์ ที่เข้ามาเชื่อมต่อกันเป็นระบบ

ตัวจำลองแต่ละตัวในระบบจะดำเนินการจำลองไปตามช่วงเวลากำหนดโดยอัลกอริธึมภายในตัวเอง โดยหากไม่มีการส่งข้อมูลระหว่างกัน ก็จะวิ่งตัวจำลองไปอย่างต่อเนื่อง โดยมีแผงเชื่อมโยงทำหน้าที่ให้คำสั่ง แต่หากเมื่อใดก็ตามที่มีข้อมูลแลกเปลี่ยนระหว่างกัน แผงควบคุมตัวจำลองจะสั่งการหยุดทำงาน และส่งถ่ายข้อมูลให้มีข้อมูลสัญญาณตรงกันก่อนที่จะดำเนินการทำงานต่อไป

เนื้อหางานวิจัย

บทนำ

เทคโนโลยีวงจรรวมขนาดใหญ่มาก(Very Large Scale Integrated Circuits, VLSI)[1] มีบทบาทสำคัญอย่างยิ่งใน อุตสาหกรรมการผลิต อุปกรณ์ เครื่องใช้ และ วงจรอิเลกทรอนิกส์ วงจรรวม(IC, Integrated Circuits) หรือ ไอซี มีการนำไปใช้อย่างแพร่หลายไม่ว่าจะเป็น คอมพิวเตอร์ เครื่องมือสื่อสาร เครื่องยนต์ และอื่นๆ เทคโนโลยีการออกแบบ และ ผลิตวงจรรวม เป็นหัวใจสำคัญที่สร้างให้เกิดการพัฒนาวงจรรวมสำหรับ การประยุกต์ใช้งานรูปแบบต่างๆ โดยเฉพาะด้านการออกแบบวงจรรวมในปัจจุบัน ได้นำเครื่องมือช่วยออกแบบด้วยคอมพิวเตอร์ (CAD, Computer Aids Design Tools) เข้ามาใช้งานเพื่อความสะดวก รวดเร็ว และ ออกแบบงานที่มีความซับซ้อนสูงได้อย่างถูกต้องแม่นยำมากขึ้น

CAD ที่ใช้ในการออกแบบวงจรรวม แบ่งได้เป็นสองกลุ่มคือ เครื่องมือวางแบบวงจร (Circuit Layout Tools) และ เครื่องมือจำลองการทำงานวงจร (Circuit Simulation Tools) ด้วยศักยภาพของงานพัฒนาด้าน CAD นี้ ทำให้เกิดการพัฒนาวงจรรวมที่ทำงานซับซ้อนมากขึ้น ประสิทธิภาพสูงขึ้น ในขณะที่ขนาดอุปกรณ์เล็กลง จึงนับได้ว่า CAD เป็นสิ่งที่สำคัญอย่างยิ่งในการพัฒนาวงจรรวมขนาดใหญ่มาก ที่ความซับซ้อน และความเร็วสูงขึ้น

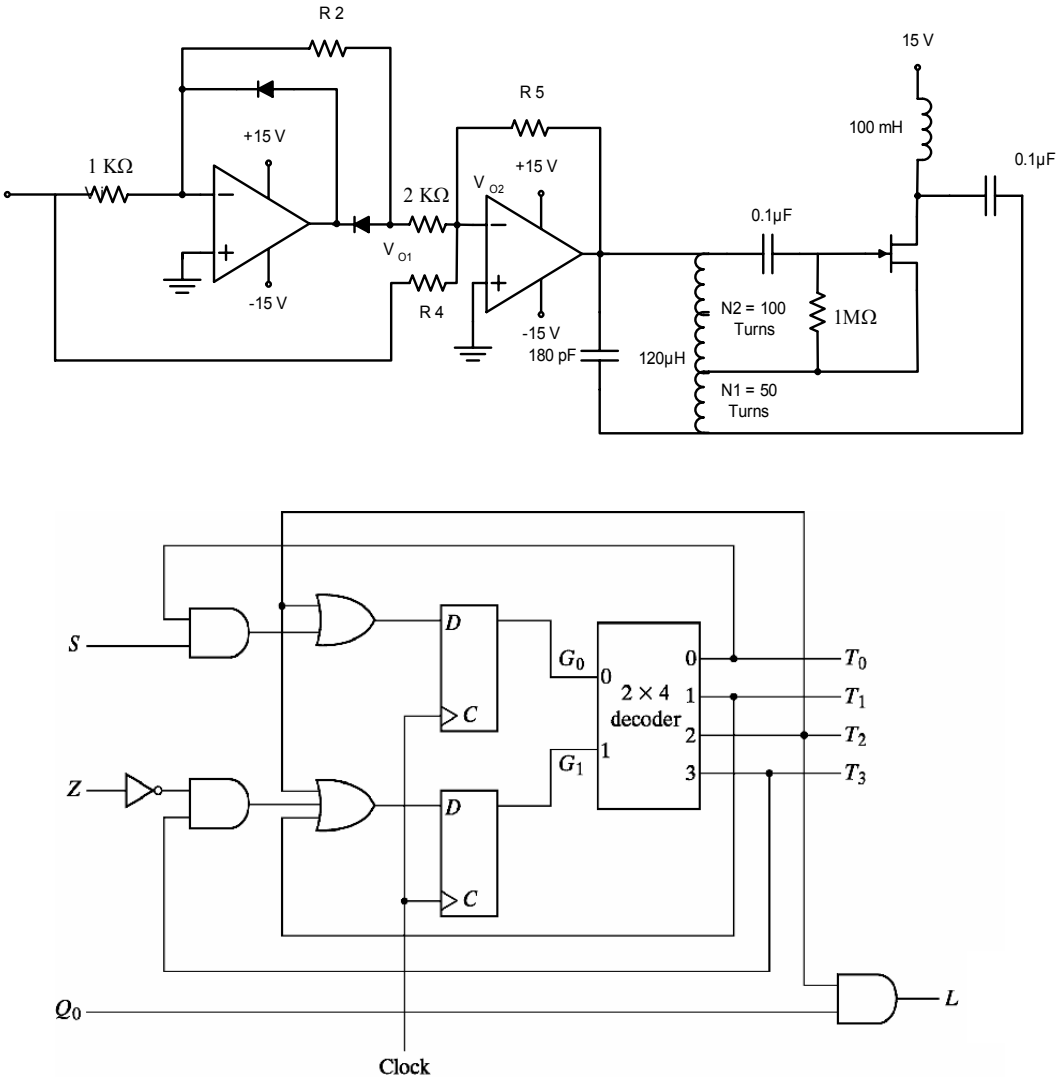
ความสำคัญและที่มาของปัญหา

วงจจรวมในปัจจุบันเพิ่มความซับซ้อนมากยิ่งขึ้น โดยรวมเอาส่วนประกอบวงจรทั้งที่เป็นอนาลอก และดิจิทัลไว้บนแผ่นฐาน(Substrate)เดียวกัน[3] เช่นไมโครคอนโทรลเลอร์ หน่วยประมวลผลสัญญาณดิจิทัล ก็ได้มีการรวมเอาวงจรแปลงสัญญาณ อนาลอกเป็นดิจิทัลรวมไว้ในชิพ(Chip)เดียวกัน หรือวงจรสื่อสารไร้สายต่างๆที่ได้ใช้ส่วนประมวลผลทั้งอนาลอกและดิจิทัลในวงจรรวมเดียวกัน ทั้งนี้เพื่อความสะดวกในการใช้งาน ลดต้นทุนการผลิต และ เพิ่มประสิทธิภาพการทำงานของวงจรให้ดียิ่งขึ้น วงจรประเภทนี้จัดอยู่ในกลุ่มวงจรรวมแบบผสมสัญญาณ (Mixed-Signal Integrated Circuits)

ปัญหาสำคัญในการออกแบวงจรรวมแบบผสมสัญญาณคือ การจำลองการทำงานวงจร (Circuit Simulaiton) ซึ่งในปัจจุบันตัวจำลองการทำงานวงจร(Circuit Simulator) ที่ใช้งานถูกออกแบบมาเพียงเพื่อจำลองการทำงานของวงจรดจิทัลล้วน(Digital Only) หรืออนาลอกล้วน

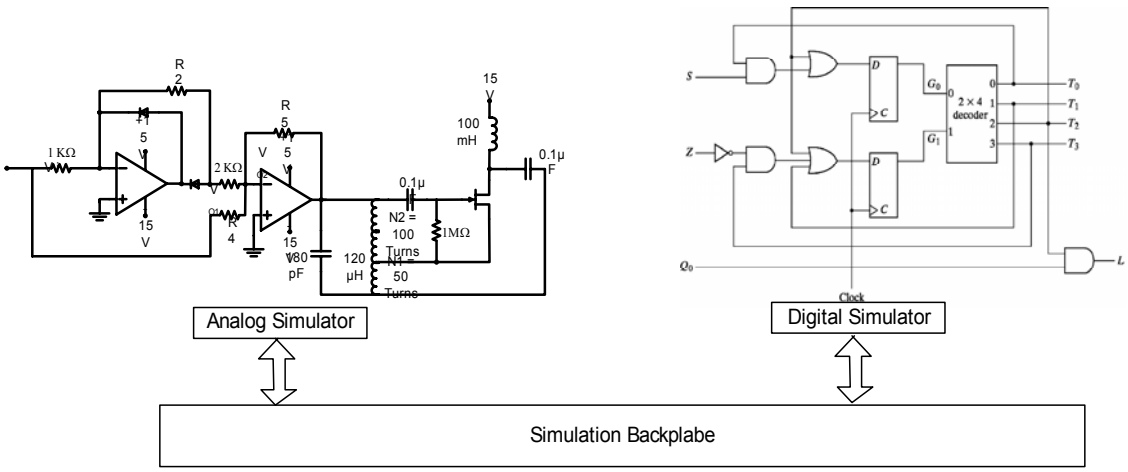
(Analogue Only) ทำให้การนำมาใช้งานในวงจรรวมแบบผสมสัญญาณ จะมีปัญหาคือประสิทธิภาพต่ำ ผลการจำลองไม่ถูกต้อง ทำงานช้า หรือ ไม่สามารถใช้งานได้

ตัวจำลองวงจรดิจิทัลถูกออกแบบมาเพื่อทดสอบวงจรที่สร้างขึ้นด้วยเกทลอจิก และสัญญาณดิจิทัล 0 และ 1 ที่เป็นแบบทิศทางเดียว(Uni-directional) จึงไม่สามารถทำงานร่วมกับอุปกรณ์ไม่เชิงเส้นอย่างเช่นทรานซิสเตอร์ชนิดไบโพลาร์(Bipolar Transistors) หรือมอส(MOS, Metal Oxide Semiconductor) อีกทั้งยังไม่สามารถจัดการกับสัญญาณอนาลอกที่มีค่าต่างๆ และคุณสมบัติเป็นสองทิศทาง(Bi-directional)[8] ในทางกลับกัน ตัวจำลองวงจรอนาลอกถูกออกแบบมาให้มีอัลกอริธึมที่สามารถทดสอบวงจรซึ่งประกอบไปด้วย อุปกรณ์อิเลกทรอนิกส์ที่หลากหลายกว่าดิจิทัล แต่หากต้องใช้กับวงจรรวมขนาด VLSI จะสิ้นเปลืองทรัพยากร และเวลาในการประมวลผลมาก รูปที่ 1 แสดงตัวอย่างวงจรดิจิทัล และอนาลอกที่ต้องใช้ตัวจำลองต่างกัน



รูปที่ 1 วงจรอนาลอก และ ดิจิทัล ซึ่งต้องวิเคราะห์โดยใช้ตัวจำลองต่างชนิดกัน

เนื่องจากข้อจำกัดของตัวจำลอง และวงจรที่ต้องการทดสอบในปัจจุบันมีแนวโน้ม ที่จะผสมรวมเอา อนาลอกและดิจิทัลไว้ด้วยกันมากขึ้น จึงเกิดแนวคิดที่จะประสานการทำงานของตัวจำลองต่างชนิดเพื่อทำงานร่วมกัน โดยให้อิสระแก่ตัวจำลองแต่ละตัวที่จะทำงานเฉพาะด้าน โดยใช้แบบจำลองวงจร และอัลกอริธึมที่ได้พัฒนาแล้ว ทำงานร่วมไปพร้อมกับตัวจำลองอื่นๆ โดยแลกเปลี่ยนข้อมูลเพื่อรักษาให้ผลการจำลองทั้งระบบให้ถูกต้อง ดังรูปที่ 2 แสดงการเชื่อมต่อตัวจำลอง เพื่อจำลองวงจรรวมขนาดใหญ่ที่มีทั้งส่วนอนาลอก และดิจิทัล



<u>รูปที่ 2 เชื่อมต่อตัวจำลองต่างชนิด เพื่อใช้ในงานจำลองวงจรรวมขนาดใหญ่ที่มีทั้งส่วนอนาลอกและดิจิทัล</u>

วัตถุประสงค์ของโครงการ

โครงการนี้นำเสนอวิธีการที่จะขยายขอบเขตของการวิเคราะห์วงจรรวมแบบตัวจำลองการทำงานเดี่ยว (Single Simulator) มาเป็นแบบ หลายตัวจำลองการทำงาน (Multiple Simulators) เชื่อมโยงและทำงานไปพร้อมกัน โดยมีวัตถุประสงค์ที่จะแก้ไข ปัญหาของการจำลองการทำงานของวงจรรวม ซึ่งประกอบไปด้วยส่วนที่เป็น อนาลอก และ ดิจิทัล วิธีการที่นำเสนอ จะประสานการทำงานของตัวจำลองการทำงาน และจะเป็นมาตรฐานสำหรับเชื่อมต่อตัวจำลองการทำงานแบบใดๆก็ได้ โดยโครงการนี้จะพัฒนา เทคนิค และ ระบบเชื่อมโยงมาตรฐาน ซึ่งสามารถรวมเอา โปรแกรมจำลองการทำงานของวงจรไฟฟ้าแบบอนาลอก ดิจิทัล และ ระดับสูง เข้าทำงานด้วยกัน วัตถุประสงค์ของโครงการสามารถแยกเป็นข้อย่อยดังนี้

- ศึกษาและเผยแพร่เทคโนโลยีการทำงาน ของตัวจำลองการทำงานแบบต่างๆ
- ศึกษาและพัฒนาส่วนเชื่อมต่อ(Interface) สำหรับตัวจำลองการทำงานให้สามารถทำงาน ร่วมกันได้
- ศึกษาและวิจัยหาเทคนิคที่จะนำมาใช้เชื่อมโยง ตัวจำลองการทำงานแบบอนาลอกหลายตัว (Multiple Analogue Simulators)
- ปรับปรุงระบบการวิเคราะห์วงจรรวมด้วยตัวจำลองหลายตัวให้มีประสิทธิภาพดี ทำงานรวดเร็ว และ ให้ผลถูกต้องแม่นยำ

ทฤษฎีการจำลองการทำงานวงจรรวมขนาดใหญ่มาก

ตัวจำลองวงจรสำหรับวงจร VLSI แบ่งได้ตามลักษณะการประยุกต์ใช้งาน และอัลกอริธึมของโปรแกรมเป็น 3 ประเภทคือ

- ตัวจำลองวงจรอนาลอก (Analogue Circuit Simulator)
- ตัวจำลองวงจรดิจิทัล (Digital Circuit Simulator)
- ตัวจำลองวงจรผสมสัญญาณอนาลอกและดิจิทัล (Mixed-Signal Circuit Simulator)

ทั้งสามประเภทมีเทคนิควิธีการ แบบจำลองที่ใช้ และการประยุกต์ใช้งานที่แตกต่างกัน การที่จะนำ ตัวจำลองมาทำงานร่วมกันได้จึงต้องเข้าใจวิธีการทำงานภายในของแต่ละตัวก่อน

ตัวจำลองวงจรอนาลอก

ตัวจำลองวงจรอนาลอก ใช้วิธีทางคณิตศาสตร์เพื่อแก้ปัญหาวงจร หาคำตอบของ ค่ากระแส ค่าแรงดัน และการตอบสนองของวงจร ซึ่งประกอบด้วยอุปกรณ์ไฟฟ้าและอิเลกทรอนิกส์พื้นฐานได้แก่ ตัวต้านทาน ตัวเก็บประจุ ไดโอด ทรานซิสเตอร์ชนิดไบโพลาร์ และ มอส รวมไปถึงอุปกรณ์เชิงเส้น และไม่เชิงเส้น อีกทั้งแหล่งจ่ายกระแสหรือแรงดันแบบอิสระ และแบบแปรค่ากับตัวแปร อุปกรณ์ไฟฟ้าอิเลกทรอนิกส์เหล่านี้ จะถูกแทนด้วยสมการคุณลักษณะ (Device Characteristic Equation) และสมการของอุปกรณ์ทุกตัวจะถูกรวมนำมาสร้างเป็นสมการระบบ(System Equations) เพื่อใช้ในการวิเคราะห์หาคำตอบที่ต้องการจากวงจรต่อไป

การสร้างสมการระบบ(Network Equation Formulation) [5]

อุปกรณ์ไฟฟ้าอิเลกทรอนิกส์เมื่อถูกแทนด้วยสมการคุณลักษณะ และอุปกรณ์ทุกตัวถูกรวมนำมาสร้างเป็นสมการระบบ(System Equations) เพื่อใช้ในการวิเคราะห์หาคำตอบที่ต้องการจากวงจร การสร้างสมการระบบ มีหลายวิธีการ ขึ้นอยู่กับลักษณะวงจร และคำตอบที่ต้องการหา กล่าวโดยสรุปดังนี้

- ***Nodal Admittance Formulation*** ใช้สมการไฟฟ้าพื้นฐาน คือ กฎกระแสของเคอร์ซอฟ (Kirchoff's Current Law, KCL) กล่าวว่า ผลรวมกระแสที่ปมใดๆ ในวงจรมีค่าเป็นศูนย์เมื่อประยุกต์ KCL กับวงจรที่มีจำนวนปมเป็น N จะได้ผลลัพธ์เป็นสมการระบบ ขนาด $N \times N$ ดังสมการที่ 1

$$\begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{bmatrix} \quad \ldots\ldots\ldots (1)$$

เมื่อค่า Yi เป็นเมตริกซ์ของแอดมิตแตนซ์ Vi เป็นแรงดันปม Ji เป็นเวกเตอร์ของแหล่งจ่ายกระแส ข้อเสียของ Nodal Admittance Formulation คือได้สมการระบบเป็นเมตริกซ์ขนาดใหญ่ ทำให้เปลืองทรัพยากรในการเก็บค่าตัวแปร และการคำนวณหาคำตอบจะใช้เวลานาน

- ***Modified Nodal Formulation*** เป็นวิธีการสร้างสมการที่แปลงรูปมาจาก Nodal Formulation วิธีของ Modified Nodal จะใช้สมการ 3 ชุด คือ กฎกระแสของเคอร์ซอฟ (KCL) กฎแรงดันของเคอร์ซอฟ (KVL) และสมการคุณลักษณะของอุปกรณ์ (Device Constitutive Equation) วิธีนี้จะทำให้การสร้างสมการระบบสามารถรวมเอาอุปกรณ์อิเล็กทรอนิกส์ต่างๆ และตัวแปรคำตอบที่ต้องการบรรจุลงในเมตริกซ์ และจัดให้อยู่ในรูปกระชับ (Compact Form) ซึ่งจะไม่ใช้ทรัพยากรในการประมวลผลสูงเกินไป วิธีการของ Modified Nodal จะตัดตัวแปร และค่าพารามิตเตอร์อิมพีแดนซ์ แอคมิตแตนซ์บางตัวออกเหลือไว้เฉพาะส่วนที่ต้องการประมวลผลหาคำตอบ ดังสมการที่ 2

$$\begin{bmatrix} A_1 Y_1 A_1^t & A_2 \\ Y_2 A_2^t & Z_2 \end{bmatrix} \begin{bmatrix} V_n \\ I_2 \end{bmatrix} = \begin{bmatrix} -A_3 J \\ W_2 \end{bmatrix} \quad \ldots\ldots (2)$$

เมื่อค่า Yi และ Zi เป็นเมตริกซ์ของแอดมิตแตนซ์ และอิมพีแดนซ์ Vi เป็นแรงดันปม Ii เป็นกระแส J เป็นเวกเตอร์ของแหล่งจ่ายกระแส Wi เป็นแหล่งกระแสและแรงดันแบบแปรค่า และจะรวมค่าเริ่มต้นของกระแสและแรงดันในตัวเก็บประจุและตัวเหนี่ยวนำ

วิธีอินทิเกรทเชิงตัวเลข (Numerical Integration Methods) [4]

วิธี Numerical Integration เป็นการจัดรูปสมการดิฟเฟอเรนเชียล ให้อยู่ในรูปสมการผลต่าง ซึ่งสามารถใช้วิธีทางคณิตศาสตร์หาคำตอบโดยการใช้ประมวลผลคอมพิวเตอร์ วิธีนิวเมอริคัล แบ่งได้เป็น 3 วิธี คือ

1. Forward Euler Formula เป็นการประมาณค่าดิฟเฟอเรนเชียลของ $x(t)$ ที่ $t = t_n$ ซึ่งมีค่าดิฟเฟอเรนเชียลเป็น x'_n โดยประมาณค่า x_{n+1} ซึ่งเป็นค่าคำตอบที่ $t = t_{n+1}$ ตามสมการ Forward Euler Formula ในสมการที่ 3

$$x_{n+1} = x_n + hx'_n \qquad \ldots\ldots \text{(3)}$$

เมื่อ $h = t_{n+1} - t_n$

2. Backword Euler Formula ในทางตรงกันข้ามจะใช้ค่า x'_{n+1} ซึ่งเป็นค่าดิฟเฟอเรนเชียลที่ $t = t_{n+1}$ เพื่อหาค่าคำตอบ x_{n+1} ดังสมการที่ 4

$$x_{n+1} = x_n + hx'_{n+1} \qquad \ldots\ldots \text{(4)}$$

3. Trapizoidal Rule Formula จะเป็นวิธีรวมเอง Forward และ Backward Euler เพื่อหาค่าเฉลี่ยของดิฟเฟอเรนเชียล และคำนวณคำตอบ ดังสมการที่ 5

$$x_{n+1} = x_n + \frac{h}{2}\left(x'_n + x'_{n+1}\right) \qquad \ldots\ldots \text{(5)}$$

Newton – Raphson Algorithin [5]

ขบวนการทางคณิตศาสตร์ ที่ใช้แก้สมการระบบที่ไม่เป็นเชิงเส้น และเป็นฟังก์ชั่นดิฟเฟอเรนเชียล วิธีหนึ่งที่ใช้กันแพร่หลาย คือ วิธี Newton-Raphson โดยอาศัยวิธีการคำนวณซ้ำ (Iterative Method) วิธี Newton-Raphson สามารถแก้สมการระบบแบบไม่เชิงเส้น เพื่อหาคำตอบแบบลู่เข้า (Convergence)

กำหนดให้ระบบสมการแบบไม่เชิงเส้น n สมการ f_i มีตัวแปรที่ต้องการ คือ x_i, $i = 1,....,n$ ดังสมการ (6)

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\cdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \qquad \ldots\ldots$$

(6)

กำหนดให้ค่าคำตอบของสมการเป็น x^{K+1} และแต่ละฟังก์ชั่น f_i สามารถขยายเป็นอนุกรมเทเลอร์ (Taylor Serics) เมื่อคำตอบ x^K ลู่เข้าหา x^{K+1} พจน์ที่ลำดับสูงของอนุกรมเทเลอร์สามารถจะตัดทิ้ง และเปลี่ยนสมการให้อยู่ในรูปเชิงเส้น และแก้สมการโดยวิธีของ Newton-Raphson ดังสมการ (7)

$$\begin{aligned} x^{k+1} &= x^k + \Delta x^k \\ M\Delta x^k &= -f(x^k) \\ \Delta x^k &= x^{k+1} - x^k \end{aligned} \qquad \ldots (7)$$

วิธี Newton-Raphson เป็นวิธีที่ใช้แพร่หลายเพราะสามารถแก้สมการระบบไม่เชิงเส้นได้โดย ทำเป็นโปรแกรมคอมพิวเตอร์

<u>การจำลองวงจรระดับเกทลอจิก (Gate-Level Simulation) [2]</u>

สำหรับวงจรดิจิทัลที่ประกอบขึ้นจากเกทลอจิก เช่น แอนด์ (AND) ออร์ (OR) แนนด์ (NAND) นอร์ (NOR) การจำลองการทำงานจะใช้กระบวนการที่ต่างจากการจำลองวงจรอิเล็กทรอนิกส์ระดับทรานซิสเตอร์ โปรแกรมจำลองวงจรลอจิก(Logic Simulator) เป็นโปรแกรมคอมพิวเตอร์ที่ใช้ตรวจสอบการทำงานของวงจรดิจิทัลโดยเฉพาะ วิธีการจำลองระดับเกทจะคำนวณสถานะลอจิกของวงจรและคาบเวลาของสัญญาณต่างๆในวงจร เมื่อวิเคราะห์ในระดับเกท อุปกรณ์ต่างๆในวงจรจะทำงานกับสัญญาณไฟฟ้าที่แบ่งเป็นระดับลอจิกเท่านั้น ประกอบไปด้วย ลอจิก 0 ลอจิก 1 ลอจิกไม่ทราบค่า (Unknown) และสัญญาณอิมพิแดนซ์สูง (High Impedance)

<u>แบบจำลองอุปกรณ์ลอจิก และค่าหน่วงเวลา</u>

อุปกรณ์ระดับเกท ทำงานเหมือนตัวกระทำการทางตรรกศาสตร์ เกทลอจิกต่างๆ แสดงการกระทำตรรกศาสตร์พื้นฐาน ได้แก่ AND OR XOR และ NOT และอาจรวมไปถึงอุปกรณ์หน่วยความจำ เช่น รีจิสเตอร์ หรือ ฟลิปฟลอบ ลูกคลื่นสัญญาณไฟฟ้า ในวงจรดิจิทัล จะเป็นลักษณะสัญญาณลอจิก 0 - 1 หรือไม่ทราบค่า

เกทลอจิกนอกจากจะแทนตัวกระทำทางตรรกศาสตร์แล้ว ยังต้องรวมแบบจำลองการหน่วงเวลา(Delay time) เวลาหน่วงนี้เป็นค่าแสดงคุณลักษณะของอุปกรณ์ลอจิก ซึ่งเมื่อกระทำการจะต้องใช้เวลาในการสร้างสัญญาณออก หรือทำให้เกิดการเปลี่ยนแปลงสัญญาณ เวลาหน่วงจะนับแบบจำนวนเต็มที่แทนค่าหน่วยเวลาที่เหมาะสมกับวงจรจริง เช่น นาโน หรือไมโครวินาที รูปแบบของที่สำคัญของการหน่วงเวลา แบ่งออกเป็น 3 ประเภท คือ

a. Transport Delay แทนการหน่วงเวลาของเกทลอจิกที่จะแสดงสัญญาณออก เมื่อเกิดการเปลี่ยนแปลงทางด้านสัญญาณเข้า

b. Rise and Fall Transport Delay เป็นการแทนค่าเวลาหน่วงแบบไม่เท่ากัน เมื่อเกิดการหน่วงเวลาขณะสัญญาณเพิ่มขึ้นจาก 0 ไป 1 (Rise time) หรือขณะสัญญาณตกลงจาก 1 ไป 0 (Fall time)

c. Ambiguous transport Delay แสดงช่วงเวลาขณะที่สัญญาณไม่แสดงค่าที่แน่นอน

d. Inertial Delay แสดงความเฉื่อยของเกทลอจิกที่จะต้องรอสัญญาณจากขาเข้าให้มีพลังงานสูงพอที่จะทำให้เกทลอจิกเริ่มทำงาน

<u>อัลกอริธึมขับตามเหตุการณ์ (Event Driven Algorithm) [2]</u>

เกทลอจิกเมื่อต้องจำลองการทำงาน จะถูกขับด้วยการเปลี่ยนแปลงที่สัญญาณเข้า เรียกลักษณะการทำงานแบบนี้ว่า ขับด้วยเหตุการณ์(Event Driven) การเปลี่ยนแปลงของสัญญาณเข้าที่ทุกๆ เกทลอจิก แต่ละเหตุการณ์จะถูกนำมาประมวล เพื่อคำนวณค่าสัญญาณออก และใช้แบบจำลองการหน่วงเวลา เพื่อหาค่าเวลาหน่วงของสัญญาณออก

การประมวลผลตามเหตุการณ์ ใช้หลักการว่าเฉพาะเกทลอจิกที่สัญญาณเข้ามีการเปลี่ยนแปลงเท่านั้น ที่จำเป็นจะต้องถูกประมวล ทำให้การประมวลผลของวงจรดิจิทัลทำได้อย่างรวดเร็ว เพราะจะมีเฉพาะเกทลอจิกที่ถูกกระตุ้นเท่านั้นที่จำเป็นจะต้องนำมาคำนวณ แตกต่างจากการจำลองวงจรอนาลอกที่ทุกๆ อุปกรณ์จะต้องถูกคำนวณใหม่หมดตลอดเวลา

Event Queue เป็นโครงสร้างฐานข้อมูลที่จำเป็นอย่างยิ่งในการประมวลตามเหตุการณ์ ตัวอย่างของการจัดลำดับคิวที่มีใช้กันแพร่หลาย คือ Lineked List และ Time Wheel

<u>การจำลอง แบบผสมอนาลอก และดิจิทัล (Mixed-Signal Simulator) [3]</u>

วงจรที่มีส่วนประกอบทั้งอนาลอก และดิจิทัลอยู่รวมกันต้องอาศัยตัวจำลองแบบผสมสัญญาณ เพื่อทดสอบการทำงานของวงจรประเภทนี้ ตัวจำลองแบบผสมสัญญาณ มีข้อดีในด้าน

ความถูกต้องของผลลัพธ์ ความสามารถครอบคลุมวงจรหลากหลาย และความเร็วในการประมวลผล

ในการพัฒนาตัวจำลองแบบผสมสัญญาณ มีวิธีการที่ใช้อยู่ 3 แบบ คือ

1. Core Mixed Mode Simulator
2. Unified Mixed Mode Simulator
3. Glued Mixed mode Simulator

Core Mixed Mode Simulator

วิธี Core จะใช้แกนหลักของระบบตัวจำลองเป็น อนาลอก หรือดิจิทัล ที่ขยายเพิ่มความสามารถให้รองรับกับวงจรส่วนเพิ่มที่ต่างชนิดกัน เช่น หากต้องการจำลองวงจรที่ส่วนประกอบวงจรหลักเป็นอนาลอก และมีส่วนย่อยเป็นดิจิทัล จะสามารถทำได้โดย เพิ่มความสามารถของตัวจำลองวงจรอนาลอกที่ทำหน้าที่แกนหลัก ให้ครอบคลุมการจำลองวงจร ดิจิทัลได้ด้วย ในทางกลับกันก็สามารถก็สามารถที่จะเพิ่มความสามารถ ตัวจำลองวงจรดิจิทัล เป็นแกน ให้ครอบคลุมวงจรอนาลอก

Unified Mixed-Mode Simulator

วิธี Unified เป็นการพัฒนาเทคนิคและแบบจำลองขึ้นใหม่ทั้งหมด เพื่อสร้างอัลกอริธึมและแบบจำลองที่สามารถทำงานกับวงจรผสมอนาลอกและดิจิทัลไว้ในวงจรเดียวกัน วิธี Unified จะสามารถทำงานกับวงจรที่ประกอบไปด้วยอุปกรณ์อนาลอกและดิจิทัล โดยอาศัยห้องสมุดอุปกรณ์ที่มีอุปกรณ์ครบทั้งสองชนิด

ข้อจำกัดของวิธี Unified คือจะต้องเสียทรัพยากรมากในการพัฒนาโปรแกรมขึ้นมาใหม่ และยังพบปัญหาความเข้ากันได้กับโปรแกรมจำลองการทำงานอื่นๆ จึงไม่เป็นที่นิยม

Glued Mixed-Mode Simulator

วิธี Glued จะอาศัยเทคนิค ที่รวมเอาตัวจำลองการทำงานที่ได้มีการพัฒนาขึ้นแล้วเชื่อมโยงเข้าทำงานร่วมกัน วิธี Glued ไม่จำเป็นต้องพัฒนาตัวจำลอง หรือ อัลกอริธึมขึ้นใหม่ ตัวจำลองเดิมจะสามารถใช้อัลกอริธึมเฉพาะของตัวเอง รวมไปถึงแบบจำลองอุปกรณ์เดิมที่พัฒนาไว้ตรงกับคุณสมบัติของอุปกรณ์ และสัญญาณ ไม่ว่าจะเป็นอนาลอกหรือดิจิทัล

อย่างไรก็ดีสิ่งที่สำคัญในวิธี Glued คือ การกำหนดเทคนิค หรือวิธีการที่จะทำให้ตัวจำลองต่างชนิดกัน สามารถทำงานร่วมกันได้ และให้ผลลัพธ์คำตอบที่ถูกต้องแม่นยำ

ระบบเชื่อมโยงตัวจำลองการทำงานแบบผสมสัญญาณ

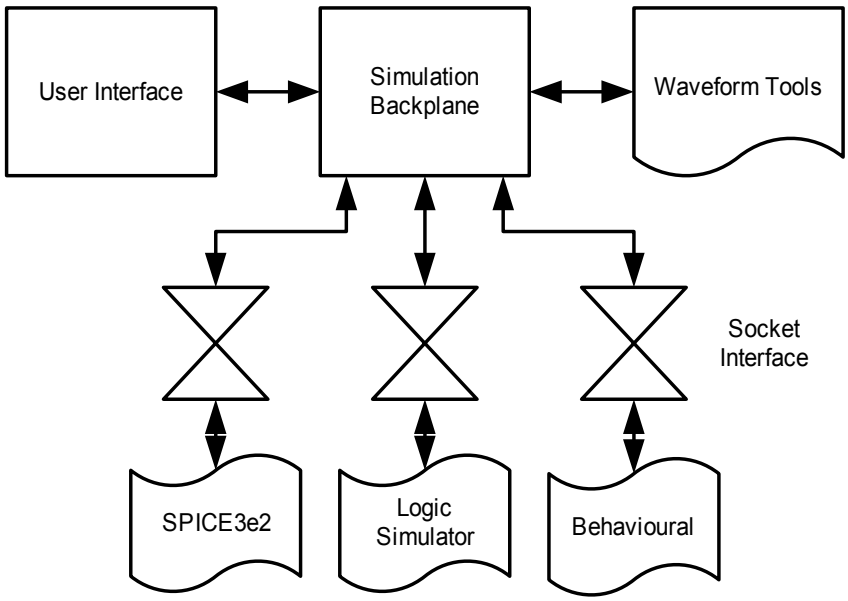
(Mixed-Signal Simulation Backplane)

วิธีการที่นำเสนอในงานวิจัยนี้คือ แผงเชื่อมตัวจำลอง(Simulation Backplane) [1,7,9] ซึ่งใช้หลักการเดียวกันกับ Glued Mixed-Mode Simulator โดยพัฒนาโปรแกรมเป็นส่วนเชื่อมโยงมาตรฐาน (Standard Interface) ที่ไม่จำกัดชนิด หรืออัลกอริธึม หรือแม้แต่คุณสมบัติของตัวจำลองที่จะเชื่อมรวมกันเข้าหาตัวระบบ

แผงเชื่อมตัวจำลองนี้พัฒนาขึ้นเป็นโปรแกรม และได้กำหนดมาตรฐานการติดต่อระหว่างตัวจำลอง ซึ่งตัวจำลองใดๆก็ตาม ไม่ว่าจะเป็น อนาลอก ดิจิทัล จากบริษัทใดๆ จะสามารถนำมาเชื่อมต่อเข้ากับระบบ และทำงานร่วมกันกับตัวจำลองอื่นๆ ได้

โครงสร้างของแผงเชื่อมต่อตัวจำลอง

แผงเชื่อมต่อตัวจำลอง ประกอบไปด้วยชุดโปรแกรม ซึ่งสามารถจะเชื่อมโยง เข้ากับตัวจำลองใดๆ โดยผ่านทาง Application Programing Interface (API) ชุดของคำสั่ง API ที่พัฒนาขึ้นนี้ สามารถติดต่อและควบคุมการทำงานของตัวจำลองที่เข้ามาเชื่อมต่อ ไม่ว่าจะเป็นอนาลอก ดิจิทัล หรืออื่นๆ โครงสร้างการเชื่อมต่อของแผงควบคุมกับตัวจำลองแสดงดังรูปที่ 3



รูปที่ 3 แสดงโครงสร้างระบบแผงเชื่อมโยงตัวจำลอง

ส่วนประกอบสำคัญของแผงเชื่อมโยงตัวจำลองมีดังนี้

a. User Interface เป็นส่วนสั่งการที่ผู้ใช้จะกำหนดรูปแบบวงจร ชนิด ประเภทวงจร และตัวจำลอง ที่ต้องการใช้ในระบบ

b. Simulation Backplane เป็นแผงเชื่อมต่อและส่วนแกนของระบบ ทำหน้าที่ควบคุมการทำงาน ตั้งแต่รับคำสั่งจากผู้ใช้สั่งการ เชื่อมต่อตัวจำลองเข้าหาระบบ รอรับผล และส่งผลการทำงานต่อไปยังตัวจำลองต่างๆ ในระบบ

c. Waveform Tool เป็นส่วนการนำเสนอสัญญาณที่เป็นผลลัพธ์จากการทำงานออกทางหน้าจอ

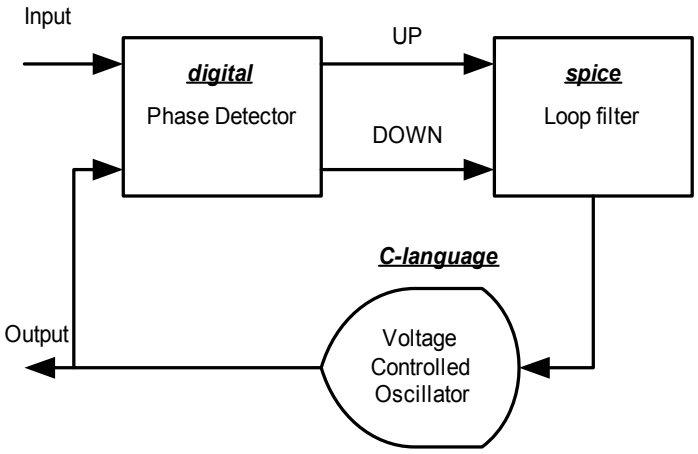
d. Socket Interface เป็นช่องทางการส่งต่อข้อมูลสัญญาณระหว่างตัวจำลอง[11] และแผงควบคุม เพื่อให้ทุกส่วนของระบบทำงานร่วมกันได้อย่างสอดคล้อง

e. Spice3e2 : Logic Simulation และ: Behavioural Simulator เป็นตัวจำลองการทำงานวงจรอิเลกทรอนิกส์ ที่เข้ามาเชื่อมต่อกันเป็นระบบ สามารถมีได้ไม่จำกัดจำนวน ขึ้นกับการประยุกต์ใช้งาน

<u>การแบ่งภาควงจร (Circuit Partitioning)</u>

วงจรที่จะนำมาใช้ในระบบ จะต้องกำหนดภาคต่างๆ ของวงจร โดยแบ่งแยกตามชนิด เช่นภาควงจรอนาลอกหรือภาควงจรดิจิทัล ที่ภาครับของแผงเชื่อมจะมีส่วน User Interface ซึ่งพัฒนาขึ้นด้วยภาษา C ส่วน User Interface นี้ทำหน้าที่รับ Configuration File ซึ่งบรรจุโครงข่ายวงจรรวมที่แบ่งเป็นภาคต่างๆไว้แล้ว โดยจะจัดให้อยู่ในรูปแบบข้อมูล ที่สามารถสื่อสารกันได้กับแผงควบคุมการจำลองการทำงาน

รูปแบบของ Configuration File จะระบุภาควงจรที่จะใช้ในระบบ ตัวจำลองที่จะใช้กับภาควงจรนั้นๆ และกำหนดส่วนของวงจรที่เชื่อมโยงเข้าหากัน รูปที่ 4 แสดงตัวอย่างวงจรที่ใช้ในการศึกษา ซึ่งประกอบไปด้วย 3 ภาควงจร คือ Phase Detector: Loop Filter และ Voltage Controlled Oscillator (VCO) ซึ่งภาควงจรเหล่านี้ จะมีตัวจำลองที่ใช้ระบุไว้ด้วย เช่น Phase Detector จะจำลองการทำงานด้วย ตัวจำลองดิจิทัล ส่วน Loop Filter จำลองโดยใช้ SPICE และ VCO เป็นแบบจำลองสร้างจากภาษา C



รูปที่ 4 แสดงตัวอย่างวงจรที่ใช้ในการศึกษา

วงจรมี Configuration File เป็นดังนี้

```
***** PLL backplane configuration *****
digital PhaseDectector
  global portinOut    is Output
  global portoutUp    is Up
  global portoutDown  is Down
spice   Filter
  global portinxin1   is Up
  global portinxin22   is Down
  global portoutxout1 is Control
c-lang Vco
  global portinVcontrol is Control
  global portoutOut   is Output
time 200000
```

การแบ่งภาควงจรเป็นขั้นตอนที่สำคัญมากในการใช้งานโปรแกรม การแบ่งภาควงจรอย่างไม่เหมาะสม หรือการเลือกตัวจำลองที่ไม่ตรงกับการทำงานของวงจร จะทำให้ผลการจำลองการทำงานผิดพลาด และระบบไม่สามารถทำงานได้

Global Events

ขณะดำเนินการจำลองการทำงาน ตัวจำลองของทุกภาควงจรจะทำงานไปพร้อมกัน ดังเช่นตัวอย่างในรูปที่ 4 ตัวจำลองดิจิทัล SPICE และ ภาษา C จะดำเนินการทำงานไปพร้อมกัน อย่างไรก็ดี ในการจำลองระบบ ขนาดใหญ่ ที่มีการเชื่อมต่อแต่ละภาควงจรเข้าหากันนั้น จะมีการส่งสัญญาณเชื่อมต่อระหว่างกัน การเปลี่ยนแปลงสัญญาณที่เปลี่ยนแปลงในภาควงจรใดๆ ที่ส่งผลเปลี่ยนแปลงต่อไปยังภาควงจรอื่น จะเรียกว่า Glabal Events การเกิด Global Events เป็นส่วนสำคัญที่จะทำให้การดำเนินการจำลองการทำงาน ของวงจรเป็นไปอย่างถูกต้องเมื่อใดก็ตามที่มี Global Events เกิดขึ้น แผงเชื่อมต่อตัวจำลองจะเป็นตัวจัดการส่ง Global Events จากภาคส่ง ไปยังภาครับโดยถูกต้อง

การส่ง/รับ Global Events ระหว่างภาควงจร ที่ใช้ลักษณะ Event เหมือนกัน เช่น ภาควงจรส่งเป็นดิจิทัลและภาควงจรรับเป็นดิจิทัล จะสามารถส่งผ่านกันได้ทันที แต่หากต้องส่งผ่านระหว่างอนาลอกไปดิจิทัล หรือกลับกัน สัญญาณ Event เหล่านี้จะต้องถูกแปลงให้อยู่ในรูปสัญญาณที่ถูกต้องเสียก่อนจึงค่อยส่งไป

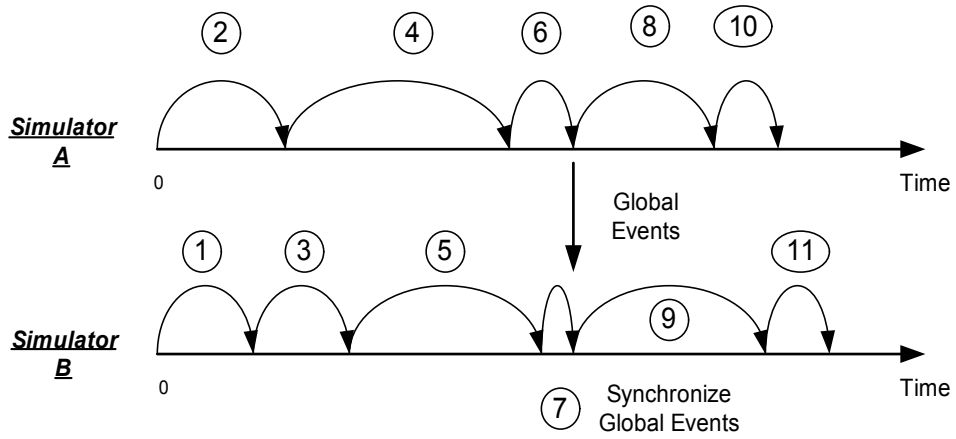
หลักการแปลง Event ได้กำหนดเป็นสัญญาณแปลงมาตรฐาน ที่จะใช้กับภาควงจรต่างๆ โดยสามารถเลือกการแปลงให้เกิดขึ้นที่ภาครับ หรือภาคส่งก็ได้ โดยหลักการคือการแปลงสัญญาณควรกระทำที่ภาควงจรที่มีข้อมูลสัญญาณละเอียดแม่นยำสูง โดยมากจึงใช้ภาคอนาลอกเป็นตัวดำเนินการแปลงสัญญาณ

อัลกอริธึมของแผงเชื่อมต่อ(Simulation Backplane Algorithm)

อัลกอริธึมที่ใช้ในการควบคุมการจำลองวงจรของแผงเชื่อมต่อเป็นวิธี Applied Lock-Step[2] วิธีนี้ถูกนำมาใช้เพราะเป็นวิธีที่เหมาะสมกับวงจรที่มีการเชื่อมต่อแบบเหนียวแน่น (Strong Coupling) อย่างเช่นวงจรดิจิทัลที่มีการเปลี่ยนแปลงสัญญาณตลอดเวลา หรือวงจรอนาลอกที่ตอบสนองต่อการเปลี่ยนแปลงสัญญาณอย่างรวดเร็ว

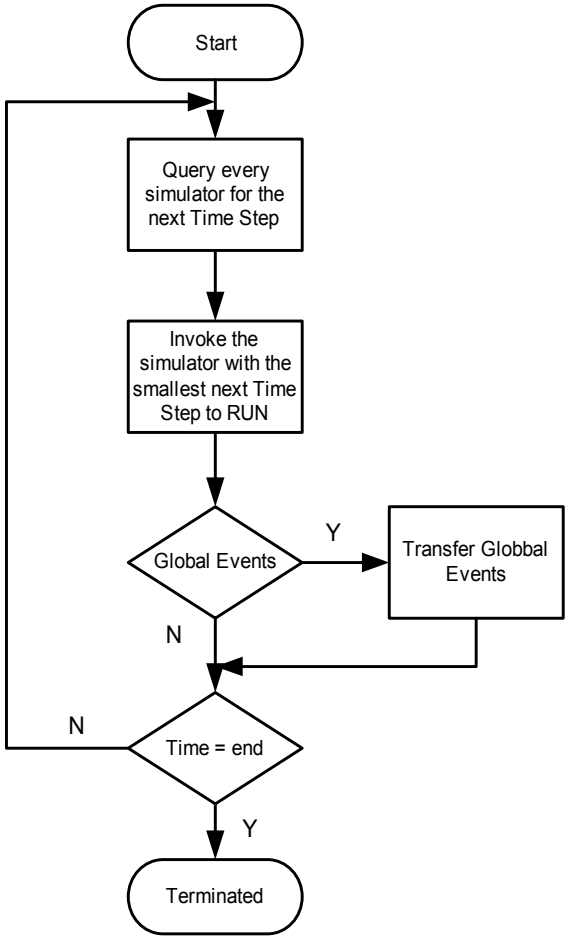
วิธีการ Lock Step สามารถอธิบายได้ดังรูปที่ 5 ตัวจำลอง A และ B จะดำเนินการจำลองไปตามช่วงเวลากำหนดโดยอัลกอริธึมภายในตัวเอง การทำงานเป็นไปตามลำดับช่วงเวลา 1, 2, 3, . . . โดยหากไม่มีการส่ง Global Event ระหว่างกัน ก็จะวิ่งตัวจำลองไปอย่างต่อเนื่อง โดยมีแผงเชื่อมต่อทำหน้าที่กำหนดจังหวะให้ แต่ละตัวจำลองทำงานตามคำสั่ง แต่หากเมื่อใดก็ตามที่เกิด Global Event ระหว่าง A และ B เช่นในตำแหน่งที่ 6 ดังรูปที่ 5 ตัวจำลอง A

สร้าง Global Event ส่งไปยัง B แผงควบคุมตัวจำลองจะสั่งการหยุดทำงาน และส่งถ่ายข้อมูลระหว่าง A และ B ให้มีข้อมูลสัญญาณตรงกันก่อนที่จะดำเนินการทำงานต่อไป



รูปที่ 5 รูปแสดง Applied Lock Step

อัลกอริธึมของแผงควบคุมดังกล่าว สามารถแสดงได้ดังโฟลว์ชาร์ตดังรูปที่ 6



รูปที่ 6 Flowchart แสดงการทำงานของแผงควบคุม

การทำงานมีขั้นตอนดังนี้ เริ่มจากแผงควบคุมส่งคำสั่งไปตรวจสอบค่าเวลาก้าว (time step) ของตัวจำลองทุกตัวในระบบจากนั้น ตัวจำลองที่มีค่าเวลาน้อยที่สุด จะเป็นตัวเริ่มทำงานก่อน โดยจะวิ่งไปให้ทันตัวที่มีก้าวเวลาสูงถัดไป จากนั้นแผงควบคุมก็จะสั่งการ ให้ตัวถัดไปเริ่มทำงานโดยมีขั้นตอนการทำงานเหมือนกัน คือจะต้องวิ่งตามให้ทันตัวที่สูงขึ้นไปเรื่อยๆ เพราะฉะนั้นทุกๆ ตัวจำลองจะถูกสั่งให้วิ่งไปข้างหน้า สลับกันไปมา จนถึงเวลาสุดท้ายที่เป็นการจบการทำงาน

อย่างไรก็ดี ขณะจำลองการทำงานในสภาวะที่เป็นสัญญาณผสม จะมีการส่งผ่านข้อมูลเชื่อมต่อระหว่างภาควงจรที่แตกต่างกันเสมอ นั่นคือ การเกิด Global Events นั้นเอง เมื่อใดก็ตามที่เกิด Global Events หมายความว่า มีข้อมูลที่จะต้องแลกเปลี่ยนระหว่างภาควงจร ขณะนี้ตัวควบคุมจะสั่งหยุดการทำงาน ของตัวจำลองทุกตัว และทำการส่งถ่ายข้อมูลระหว่างกัน ก่อนที่จะเริ่มขบวนการจำลองต่อไป

สิ่งที่สำคัญในขบวนการ จำลองวงจรผสมสัญญาณก็คือ การเชื่อมต่อข้อมูลเข้าหากันระหว่างตัวจำลองการทำงานนั่นเอง หากการส่งข้อมูลระหว่างตัวจำลองผิดพลาด ผลการจำลองจะขาดความน่าเชื่อถือได้ ค่าตัวแปรที่สำคัญในการจำลองการทำงานอีกค่าหนึ่งคือ ค่าช่วงก้าวเวลา ของตัวจำลองแต่ละตัว โดยค่านี้จะแตกต่างกันตามลักษณะวงจรคือ สำหรับวงจรดิจิทัล ค่าช่วงก้าวเวลา จะได้มาจาก ค่าก้าวเวลาแรกในรายการ Event Queue แต่สำหรับวงจรอนาลอก ค่าก้าวเวลา จะได้มาจากเวลาปัจจุบันที่ได้คำนวณหาค่าคำตอบไว้ล่วงหน้าแล้วนั่นคือ ตัวจำลองวงจรอนาลอก จะต้องจำลองผลการทำงานของวงจร ณ เวลาปัจจุบันให้เสร็จสิ้น แล้วจึงนำค่าเวลานั้นระบุเป็นค่าก้าวเวลา

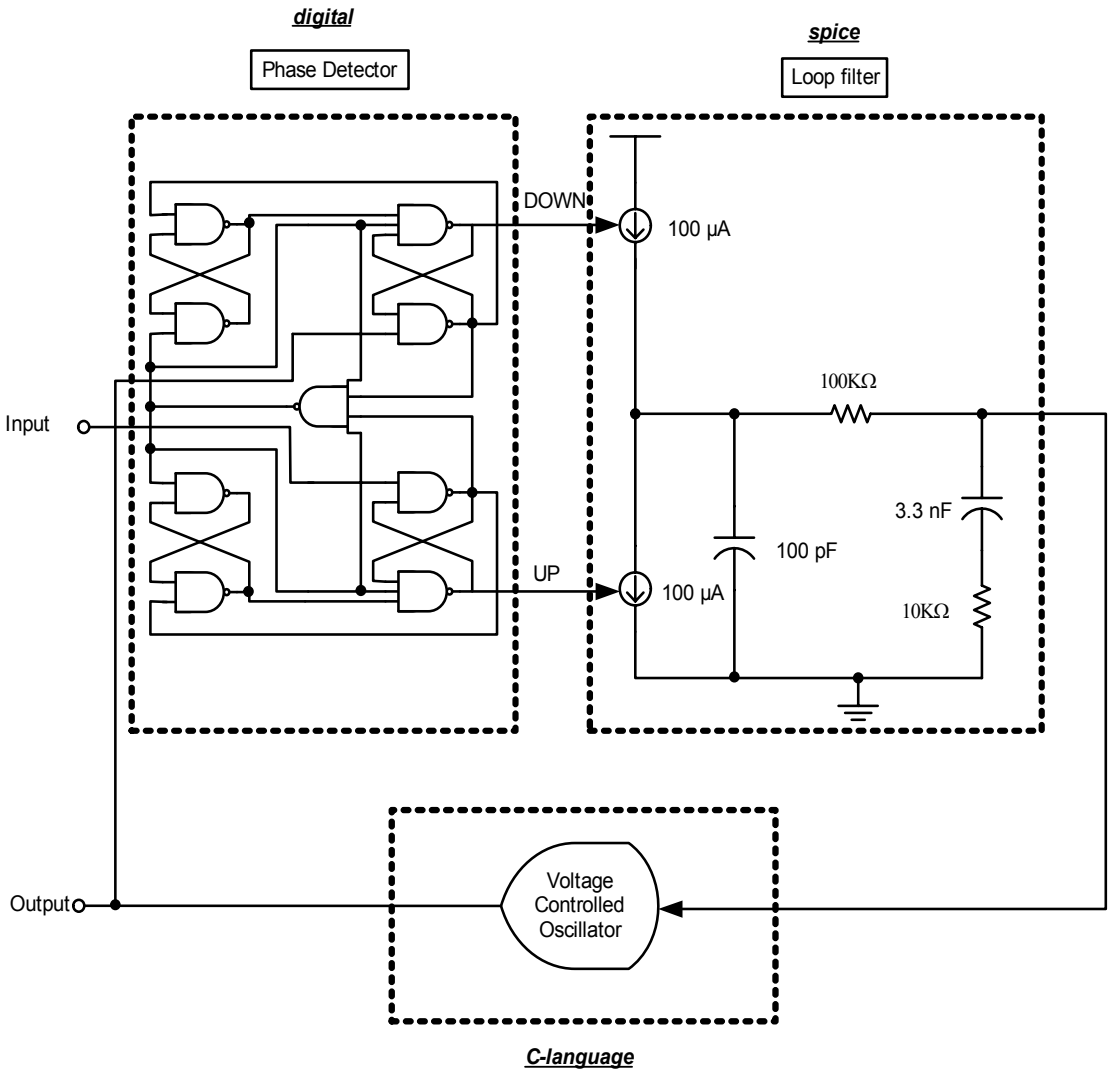
การที่วงจรอนาลอก จะต้องประมวลผลหาคำตอบก่อนที่จะระบุค่าก้าวเวลาเพราะ ค่าคำตอบของวงจรอนาลอก อาจมีการเลื่อนค่าได้ หากการคำนวณไม่พบคำตอบ การที่ให้ประมวลผลเสร็จสิ้นจนได้คำตอบแล้ว จึงประกันได้ว่า ค่าก้าวเวลาจะได้ถูกต้องเป็นปัจจุบัน

ผลการทดลอง

วงจรตัวอย่างที่นำมาทดสอบระบบที่พัฒนาขึ้นนี้ เป็นวงจร (PLL) Phase Locked Loop ซึ่งเป็นวงจรรวมแบบผสมสัญญาณอนาลอก และดิจิทัล โดยเฉพาะ PLL มีคุณสมบัติสำคัญที่เหมาะในการทดสอบ อีกทั้งยังมีการป้อนกลับ ซึ่งเป็นปัญหาสำคัญในการประมวลผลวงจรรวมผสมสัญญาณ

ส่วนประกอบวงจร Phase Locked Loop

วงจร PLL ประกอบไปด้วย 3 ส่วนหลัก คือ วงจร Phase Detector ; Voltage Controlled Oscillator และ Filer ดังแสดงในรูปที่ 7 วงจร PLL ใช้หลักการควบคุมแบบป้อนกลับ เปรียบเทียบ Phase ของสัญญาณเข้า และสัญญาณออก และปรับความถี่จนกระทั่งความถี่ และเฟสของสัญญาณขาเข้า และขาออกมีค่าเท่ากัน



รูปที่ 7 วงจร Phase Locked Loop

วงจร Phase Detector ทำหน้าที่เปรียบเทียบ Phase ของสัญญาณเข้า และสัญญาณออกที่ได้มาจาก VCO หาสัญญาณเข้าและออก มีค่าความถี่ไม่ตรงกัน วงจร Phase Detector จะส่งสัญญาณควบคุมไปสั่ง VCO ให้เพิ่มหรือลดความถี่ จนกระทั่งความถี่ และเฟสของสองสัญญาณจะตรงกัน

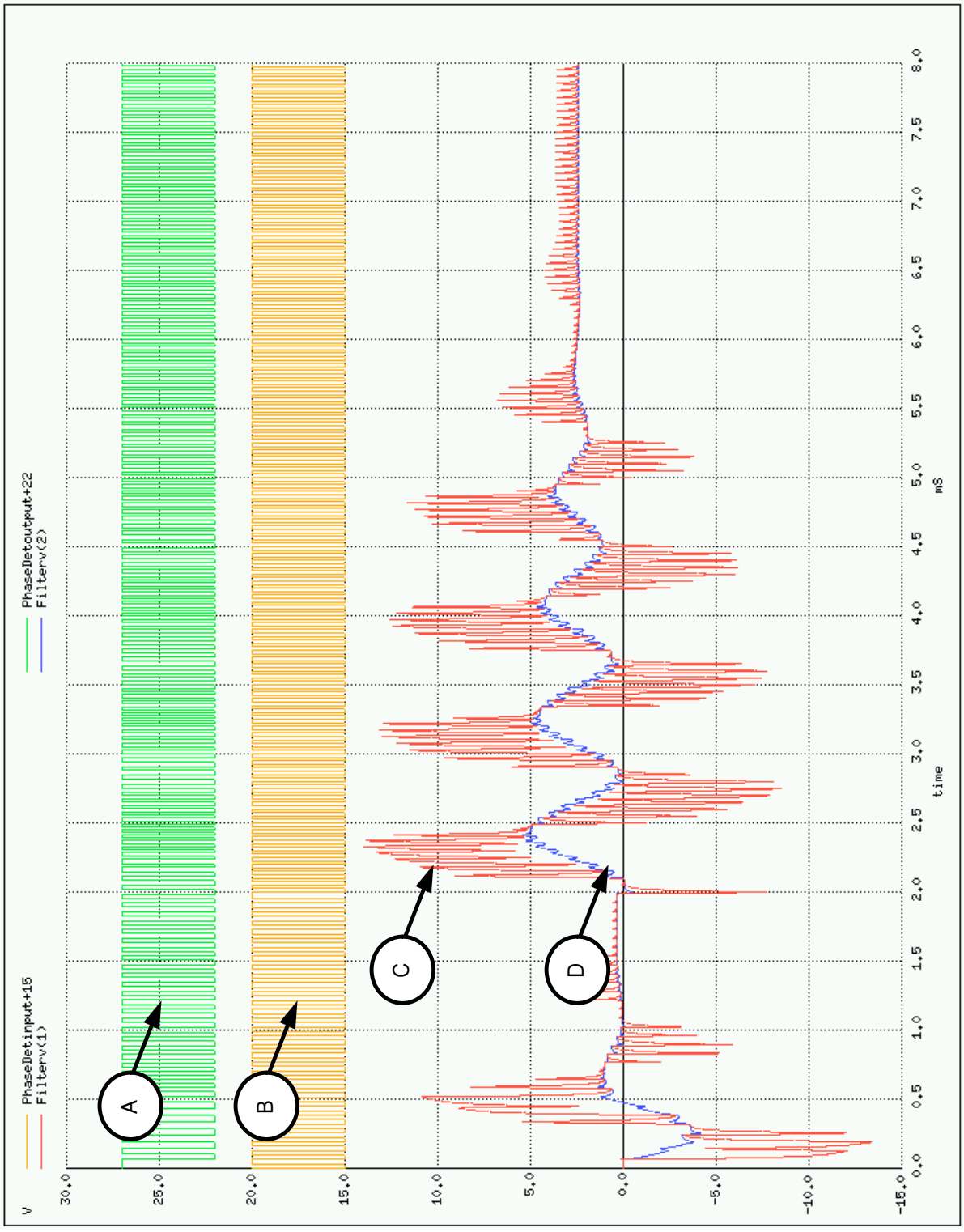
สัญญาณควบคุมที่ได้จากวงจร Phase Detector จะต้องผ่านวงจร Filter เสียก่อน เพื่อกรอกความถี่สูงทิ้งให้สัญญาณมีลักษณะเรียบขึ้น ป้องกันการเกิด Oscillation เมื่อ PLL ทำงานจบสมบูรณ์ สัญญาณออกที่ได้จาก VCO จะมีความถี่และเฟสตรงกับสัญญาณเข้า

ในการทดลองนี้ วงจร VCO พัฒนาด้วยภาษา C และ Filter สร้างเป็นภาควงจรอนาลอกโดยใช้โปรแกรม SPICE [13] ส่วนวงจร phase Detector ทำหน้าที่เปรียบเทียบเฟส อยู่ในภาควงจรดิจิทัล สัญญาณความถี่ขาเข้า ได้ตั้งไว้ที่เริ่มต้น 15.5 kHz และเปลี่ยนเป็น 20 kHz และมีความไวในการปรับความถี่กลางเป็น 15 kHz และมีความไวในการปรับความถี่เท่ากับ 2 kHz/Volt มีผลการจำลองการทำงานดังรูป 8

ผลการจำลองวงจร PLL

กราฟ A เป็นค่าความถี่สัญญาณเข้า ตั้งไว้ที่ 15.5 kHz และเปลี่ยนเป็น 20 kHz ที่เวลา 2 m Sec กราฟ B เป็นสัญญาณออกที่ได้จากวงจร จะสังเกตุเห็นว่า สัญญาณออกจะมีการปรับเปลี่ยนความถี่ เพื่อจะให้ตรงกับสัญญาณเข้า จากการทดลองนี้พบว่าวงจรจะทำการ ล็อกสัญญาณ 15.5 kHz ได้ภายในเวลา 1.5 mSec กราฟ C และ D แสดงการตอบสนองของวงจร Filter ที่กรองแรงดันควบคุม VCO ซึ่งจะปรับเปลี่ยนจนกระทั่งได้ค่าที่ถูกต้องตรงกับความถี่ต้องการ

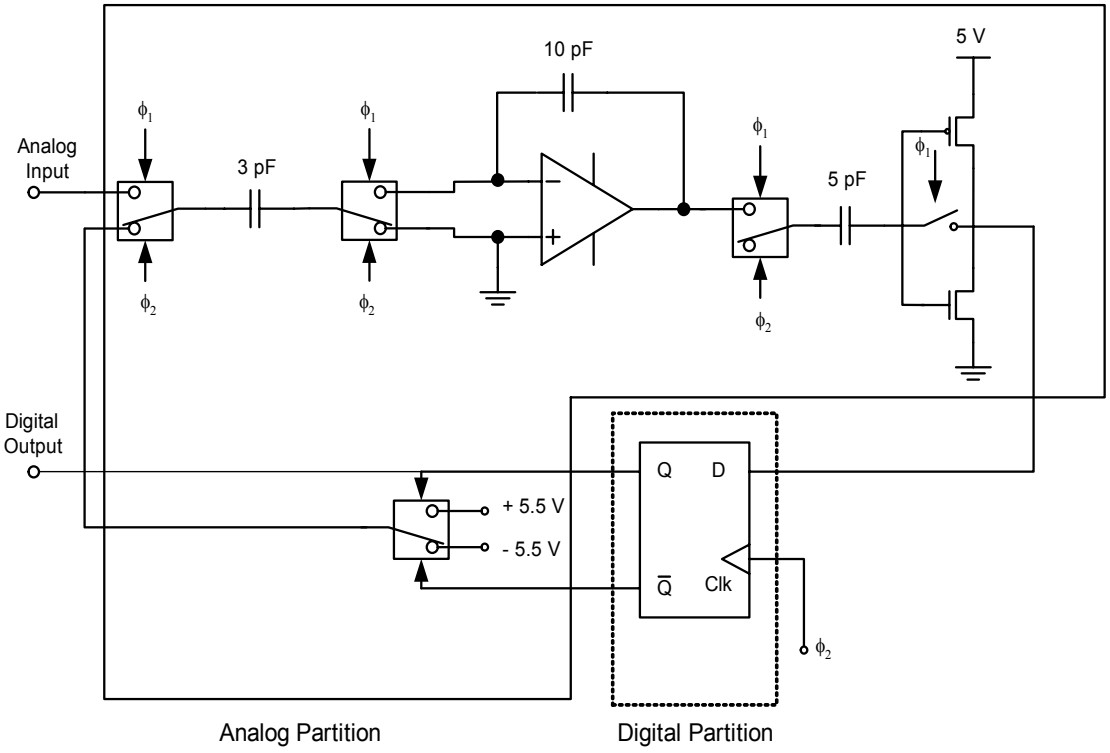
ในการทดลองที่ 20 kHz เวลาที่ใช้ในการล็อกสัญญาณสูงขึ้นเป็น 4 m Sec เนื่องจากช่างความถี่เปลี่ยนสูงขึ้น วงจรใช้เวลานานมากขึ้นในการหา ความถี่ที่ตรงกับสัญญาณเข้า



รูปที่ 8 ผลการทดลองวงจร PLL

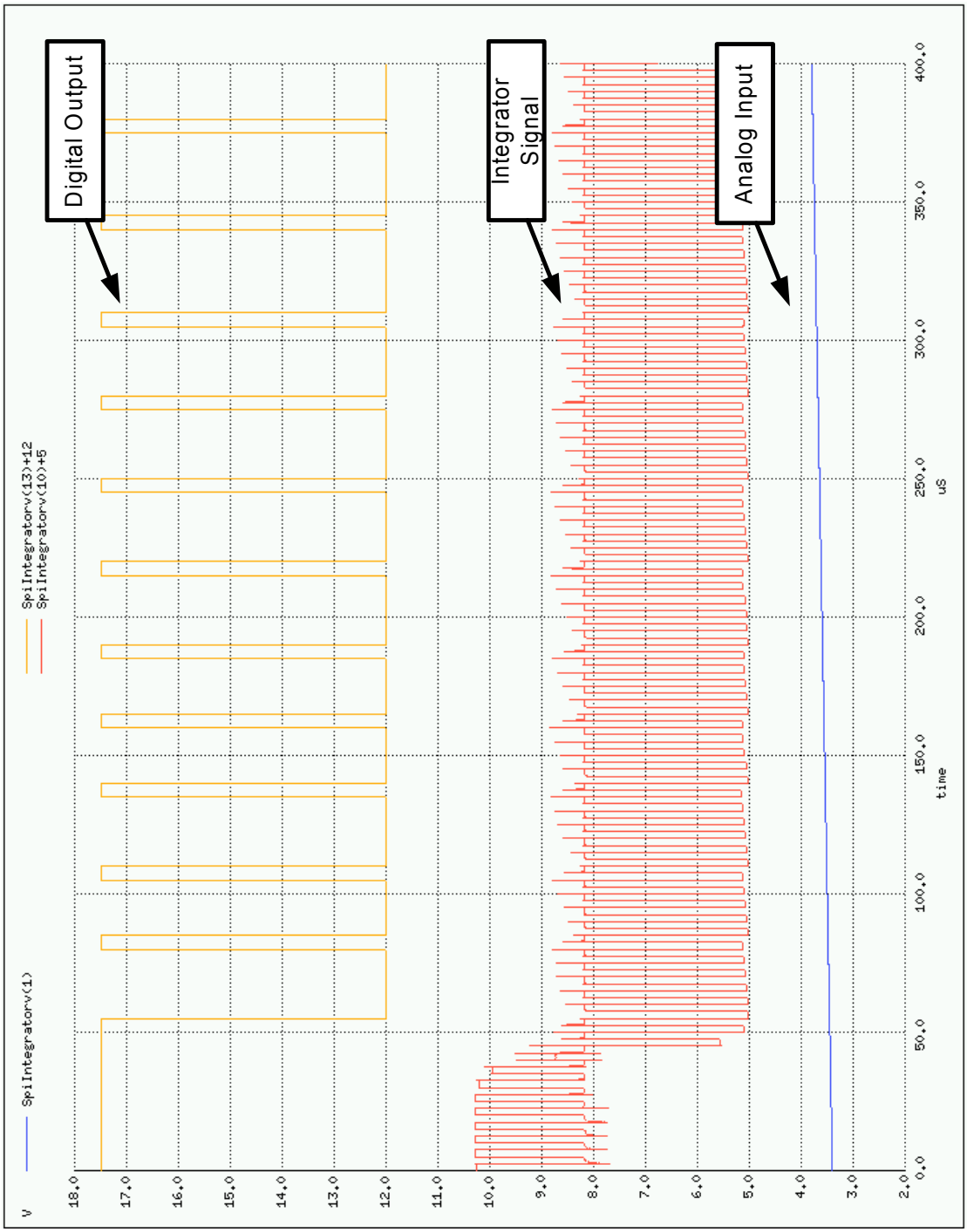
ผลการทดลองกับวงจรอื่น ๆ

ระบบแผงควบคุมตัวจำลอง ได้ทำการทดสอบกับวงจรอื่นๆ ที่เป็นลักษณะผสมสัญญาณ ได้แก่ วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล แบบ 1-Bit Dual Slope และ 3-Bit Flash ซึ่งผลการทดลองที่ได้จากทั้งสองวงจร คงยืนยันความสามารถในการทำงานถูกต้องระบบนี้ ตัวอย่างวงจร และผลที่ได้จากการทดลองแสดงดังรูป 9 , 10 , 11 , 12

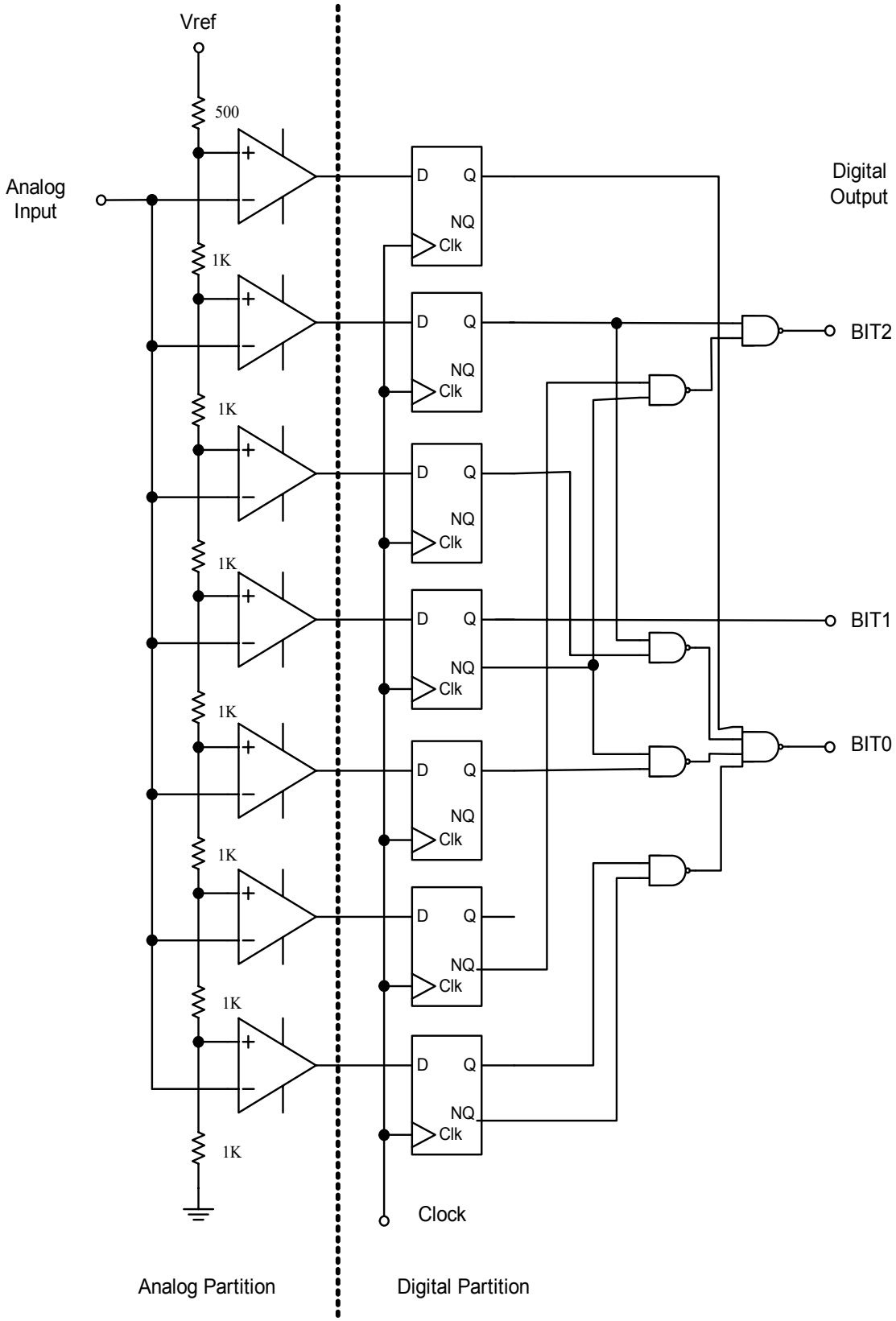


รูปที่ 9 วงจร 1-Bit Dual Slope A to D Convertor

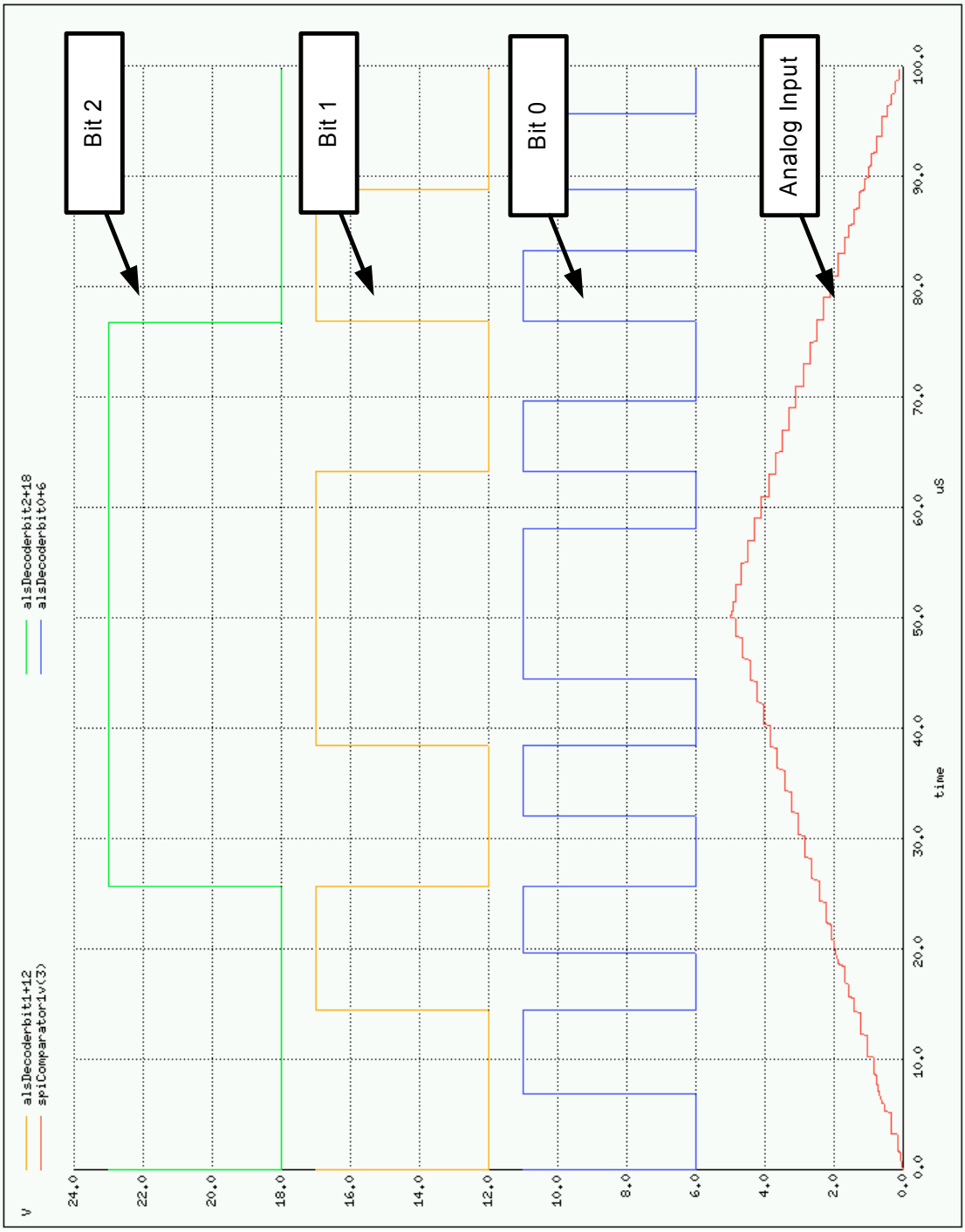
วงจร 1-Bit Dual Slope A to D ประกอบไปด้วยวงจรอินทีเกรเตอร์ สวิทช์คาปาซิเตอร์ และฟลิปฟลอป วงจรจะแปลงสัญญาณเข้าอนาลอก เป็นสัญญาณออกดิจิทัลขนาด 1 bit มีผลจำลองการทำงานดังรูปที่ 10 แสดงค่าสัญญาณดิจิทัล 1 bit ที่มีค่าแตกต่างกันตามค่าสัญญาณเข้าอนาลอก ผลที่ได้ยืนยันควาถูกต้องตามทฤษฎีการทำงานวงจร [10]



รูปที่ 10 ผลการจำลองวงจร 1-Bit Dual Slope A to D Converter



รูปที่ 11 วงจร 3-Bit Flash A to D Converter



รูปที่ 12 ผลการจำลองวงจร วงจร 3-Bit Flash A to D Converter

วงจร Flash A to D ให้สัญญาณออกขนาด 3 bits ที่มีค่าเปลี่ยนแปลงตามสัญญาณเข้าอนาลอก โดยในการทดสอบได้ให้ค่าสัญญาณเข้าเปลี่ยนแปลงเพิ่มขึ้น และลดลง สังเกตุผลของสัญญาณดิจิทัลที่ขาออกได้ค่าถูต้องตามคุณสมบัติวงจร

เวลาในการประมวลผล

เพื่อให้ได้ข้อมูลเวลาการทำงานในการจำลองการทำงานของวงจรทั้งสาม โปรแกรมตรวจสอบเวลาได้ถูกนำมาผนวกเข้ากับชุดแผงจำลองการทำงาน จากการตรวจสอบได้ข้อมูลเวลาในการประมวลผลของวงจรทั้งสามดังตารางที่ 1, 2 และ 3 ดังนี้

ตารางที่ 1 เวลาประมวลผลของวงจร Phase Locked Loop

ขบวนการ และ ตัววงจร	เวลาประมวลผล (seconds)	จำนวน Global Events	จำนวนครั้งที่สั่ง Run
Loop Filter	1164.93	406,937	1,224,581
Phase Detector	3.28	594	3,580
Voltage Controlled Oscillator	1010.46	599	1,221,625
Simulation Backplane	3918.73	-	-
Socket Communication	3027.78	-	-
Total Run Time	9125.18		

ตารางที่ 2 เวลาประมวลผลของวงจร 1-Bit Analog to Digital Dual Slope Converter

ขบวนการ และ ตัววงจร	เวลาประมวลผล (seconds)	จำนวน Global Events	จำนวนครั้งที่สั่ง Run
Analog Switch and Integrator	2.47	23	342
Digital Flip-Flop	0.19	23	343
Simulation Backplane	3.68	-	-
Socket Communication	4.53	-	-
Total Run Time	10.87		

ตารางที่ 3 เวลาประมวลผลของวงจร 3-Bit Flash Analog to Digital Converter

ขบวนการ และ ตัววงจร	เวลาประมวลผล (seconds)	จำนวน Global Events	จำนวนครั้งที่สั่ง Run
Analog Comparators	0.27	14	160
Gate level decoder & Flip-Flop	0.18	0	161
Simulation Backplane	1.18	-	-
Socket Communication	0.94	-	-
Total Run Time	2.57		

จากการทดลองพบว่า เวลาที่สูญเสียไปอย่างมาก เกิดจากการส่งผ่านข้อมูลระหว่างกันของตัวจำลอง ซึ่งถ้านับเฉพาะเวลาในการประมวลผลจริงของตัวจำลองแต่ละตัวจะใช้เวลาลดลงอย่างมาก ดังนั้น พบข้อด้อยประการสำคัญของระบบ คือ เวลาส่วนเพิ่ม (Overhead) ที่เกิดขึ้นจากตัวแผงควบคุมที่คอยสั่งการ และถ่ายโอนข้อมูลระหว่างตัวจำลองในระบบ

สรุปและวิจารณ์

ระบบแผงควบคุมตัวจำลอง เป็นการเสนอแนวคิดที่จะรวมเอาเครื่องมือที่ใช้ในการจำลองการทำงานของวงจรรวมเพื่อทำงานร่วมกัน โดยมีหลักการสำคัญคือ การควบคุมการทำงานของตัวจำลอง แต่ละตัวที่มาเชื่อมต่อเข้าหากัน ให้มีการทำงานอย่างสอดคล้อง และช่วยกันแก้ปัญหาของแต่ละภาควงจรที่ได้รับมอบหมาย วิธีการควบคุมแบบ Modified Lock-Step สามารถใช้ได้ดีกับวงจรผสมสัญญาณทุกประเภท ดังยืนยันโดยผลการทดลองที่ได้นำเสนอไป ส่วนสำคัญอีกประการหนึ่งคือ การส่งผ่านข้อมูลที่ถูกต้องระหว่างตัวจำลอง

ปัญหาของระบบที่พบในปัจจุบันคือ ความล่าช้าในการทำงาน ถึงแม้ระบบจะหาคำตอบได้อย่างถูกต้อง แม่นยำ แต่เวลาโดยรวมที่ใช้ในการดำเนินการทั้งหมด ใช้เวลาสูงมาก

อย่างไรก็ดี วิธีการของแผงเชื่อมโยงตัวจำลองได้เสนอทางเลือกสำหรับการจำลองวงจรรวมแบบผสมสัญญาณอนาลอก และดิจิทัล ที่ไม่อาจทำได้หากไม่มีการเชื่อมตัวจำลองเหล่านี้เข้าหากัน ปัจจุบันมีวงจรเป็นจำนวนมากที่อยู่ในรูปของผสมสัญญาณ จึงจำต้องอาศัยเทคนิค และวิธีการที่จะจำลองวงจรลักษณะนี้ ซึ่งวิธีการของแผงเชื่อมต่อ เมื่อทำให้เป็นมาตรฐานสากล จะเป็นประโยชน์อย่างยิ่งที่เครื่องมือออกแบบวงจรรวมต่างๆ จะได้แลกเปลี่ยนได้ใช้ความสามารถร่วมกัน เพื่อพัฒนาวงจรรวมที่มีความซับซ้อนสูงยิ่งๆ ขึ้นไป

แนวทางพัฒนาในอนาคต

จากข้อดี และข้อด้อยที่พบในวิธีการเชื่อมต่อตัวจำลอง ผ่านแผงเชื่อมโยงมาตรฐานนี้ ทำให้สามารถกำหนดทิศทางในการพัฒนาเพื่อเพิ่มประสิทธิภาพของระบบในหลายแนงทางดังนี้

1. ลดเวลาที่สูญเสียเนื่องจากการประมวลผลในส่วนของโปรแกรมที่เชื่อมโยงตัวจำลองเข้าหากันโดยประยุกต์เทคนิดเช่น การทำ Parallel Processing หรือ การใช้ Threads ที่มีการใช้งานบนระบบปฏิบัติการ Microsoft Windows
2. เชื่อมโยงตัวจำลองที่มีประสิทธิภาพสูงเข้าสู่ระบบ เพื่อเพิ่มขีดความสามารถให้ครอบคลุมวงจร Mixed-signal ได้หลากหลายมากยิ่งขึ้น
3. จัดทำเอกสารคู่มือการใช้งาน และคู่มือการพัฒนาฟังก์ชั่นส่วนเพิ่ม เพื่อให้ผู้ใช้สามารถปรับปรุงระบบเชื่อมโยงตัวจำลองนี้ให้ตรงตามลักษณะการประยุกต์ใช้งานที่ต้องการ

เอกสารอ้างอิง

[1] CAD Framework Initiative, “Draft Proposal Programming Interface Specification Standard Version 0.6”, Technical Report, CAD Framework Initiative Inc. May 1991

[2] Clair Nauts and Oliver Pribetich, “Digital / Analogue Simulation Synchronisation Dilemma, Leap-frog or Lock-step”, The Europeon Design and Test Conference User-forum, pages 175-179, 1995

[3] Eduardo L. Acuna, James P. Dervenis, Andrew J. Pagones, Fred L. Yang and Resve A. Saleh, “Simulation Techniques for Mixed-Mode Analogue/Digital Circuits”, IEEE Journal of Solid State Circuits, 25(2), pages 353-363, April 1990

[4] Gary D. Hachtel, Robert K. Brayton and Fred G. Gustavson, “The Sparse Tableau Approach to Network Analysis and Design”, IEEE Transactions on Circuit Theory, pages 101-113, January 1971

[5] Jiri Vlach and Kishore Singhal, “Computer Method for Circuit Analysis and Design”, Van Nostrad Reinhold, New York, 1983

[6] JP. Hayes, ‘Digital Simulation with Multiple Logic Values”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pages 274-283, 1986

[7] M. Zwolinski and C. Garagate, “Anatomy of a Simulation Backplane”, IEE Proceedings of Computer and Digital Technology, pages 377-385, 1995

[8] P. Agrawal and V.D. Agrawal, “Can Logic Simulators Handle Bidirectionality and Charge Sharing”, Proceedings of IEEE International Symposium on Circuits and Systems, pages 411-414, 1990

[9] P. Odryna, “A Unified Mixed-mode Digital-Analogue Simulation Environment”, IEEE International Symposium on Circuits and Systems Proceedings, pages 893-896, 1988

[10] P.E. Allen, B.P. Lum Shue Chan and W.M. Zuberek, “Comparison of Mixed Analogue-Digital Simulators”, Proceedings of IEEE International Symposium on Circuits and Systems, pages 101-104, 1990

[11] Paul S. Wang, “An Introduction to Berkeley Unix”, Wadsworth Publishing Company, 1988

[12] R. Beale, R. Chadha, C.F. Chen, A. Posser and K.M. Tham, “ Design Methodology and Simulation Tools for Mixed Analogue-Digital Integrated Circuits”, IEEE International Symposium on Circuits and Systems, pages 1351-1355, 1990

[13] Thomas L. Quarles, A.R. Newton, A.Sangiovanni-Vincentelli and D.O. Pederson, “SPICE3f3 User’s Manual”, Department of Electrical Engineering and Computer Sciences, University of California Berkeley, May 1993

ประโยชน์ที่ได้รับ(Outputs)

ประโยชน์หลักสำคัญที่ได้รับจากงานวิจัยนี้ก็คือ องค์ความรู้ใหม่ที่จะช่วยให้การจำลองการทำงานของวงจรไฟฟ้าแบบผสม อนาลอก ดิจิตอล และ วงจรย่อยระดับสูง เป็นไปได้อย่างมีประสิทธิภาพ เชื่อถือได้ และ ให้ความถูกต้องสูง องค์ความรู้นี้จะส่งเสริมให้ เทคโนโลยีการออกแบบวงจรรวม (IC Technology) มีการพัฒนายิ่งขึ้น

ประโยชน์อีกประการหนึ่งที่ได้รับคือ การเผยแพร่ และถ่ายทอด ความรู้พื้นฐาน และเทคโนโลยีระดับสูง ที่ใช้ในการสร้างโปรแกรมจำลองการทำงานของวงจรไฟฟ้า ให้แก่นิสิต และนักวิจัย ซึ่งโครงการนี้จะมีการแบ่งงานในลักษณะของ โครงงานของปริญญาตรี และ งานวิทยานิพนธ์ของปริญญาโท

ผลงานของโครงการ

- ระบบจำลองการทำงานที่สามารถ เชื่อมโยง SPICE, Logic Simulator และ High Level Simulator ได้พัฒนาขึ้นเป็นผลสำเร็จ
- ได้รับทุนพัฒนาอาจารย์ เพื่อเจรจาความร่วมมือทางวิชาการกับมหาวิทยาลัยในประเทศอังกฤษ โดยได้แลกเปลี่ยนความคิดเห็นกับอาจารย์ และ นักวิจัยของ University of Manchester Institute of Science and Technology (UMIST) และ University of Southampton
- จัดสัมมนาโดยได้เชิญที่ปรึกษาโครงการ Dr. Mark Zwolinski จาก University of Southampton บรรยายในหัวข้อ VLSI Design Automation เป็นการเผยแพร่ความรู้ เสริมสร้างความสัมพันธ์ และความร่วมมืออันดีกับนักวิจัยนานาชาติ
- ตีพิมพ์บทความ เรื่อง ***"Simulation Interface for Mixed Analogue and Digital Circuit Simulators"*** นำเสนอใน ***"IEEE Circuit and Systems Colloquium on Mixed-Signal IC Designs"*** ตุลาคม 2542

ภาคผนวก

บทความวิชาการที่ได้ลงตีพิมพ์

Chugiat Garagate, **"Simulation Interface for Mixed Analogue and Digital Circuit Simulators"**, *The Colloquium on Mixed-Signal IC Designs: Technology & Trends Beyond 2000, IEEE Circuit and Systems Society, Thailand Chapter, Pages 1/7 – 7/7*, **1999**

Simulation Interface for Mixed Analogue and Digital Circuit Simulators

Chugiat Garagate, Mark Zwolinski
Department of Electrical Engineering
Kasetsart University

Abstract

This paper describes an integrated simulation environment that combines multiple simulators of different abstraction levels to perform mixed-signal circuit simulation. The approach to solve mixed-signal simulation problems introduced in the paper is based on a draft standard of the Simulation Backplane, published by CAD Framework Initiative.[1]

In the backplane environment, circuit simulators run as separate processes, using their own algorithm. A consistent mixed-signal simulation is obtained through the backplane controls. The Lock-step[2] algorithm was adopted to synchronise the simulation processes. Digital and analogue simulators exchange simulation data by employing a standardised signal conversion scheme, which is based on threshold functions and Boolean-controlled switches[3]. A signal latency detection technique is employed to define analogue events.

Several circuit simulators were integrated into the environment, including a logic simulator, a behavioural simulatior and a de-facto standard circuit level simulator; SPICE3e2[4]. Simulations of practical mixed-signal applications including circuits that contained feedback loops were carried out successfully. The results convincingly showed the capability and flexibility of the backplane approach.

1. Introduction

Generally, circuit simulators are divided into different levels of abstraction, according to device models and simulation algorithms. Simulators of different abstraction levels are intended for specific types of application. For instance, gate level simulators are intended for logic design simulation, and circuit level simulators are for the simulation of analogue circuits. A single simulator will run a particular algorithm and use a specific set of device models.

There are increasing numbers of electronic design applications that require mixed-signal capabilities. The simulation of a large scale integrated system which consists of analogue and digital devices is an example. A single gate level simulator cannot handle analogue parts, since general logic simulators are not capable to simulate certain device properties, such as charge sharing[5] in analogue MOS circuits. On the other hand, using a single analogue simulator to simulate the whole design, would be too computationally expensive. The individual simulators are constrained by their intrinsic device models and algorithms.

Consequently, it is obvious that an effective model of a mixed-signal simulation system is required. This system should provide a common method to combine simulators of different abstraction levels to work together, and allow simulators to operate using their own algorithm and device models. This system must also provide a flexible solution for various kinds of mixed-signal applications.

2. Conventional Circuit Level Simulation Algorithms[6,7]

Low level device models and various mathematical-numerical analysis routines are combined to create an algorithm for circuit level simulations[6,7]. Circuit level device models are based on the devices' physical and mathematical models of a high accuracy. At the circuit level, electronic systems are described as a network of components which mainly consist of resistors, capacitors, inductors, transistors and sources. To perform the simulation, networks of electronic devices are converted into a system of equation, which will be solved by using mathematical analysis functions.

2.1 Circuit-Level Device Models

At the circuit level, electronic devices are captured by mathematical models that describe the devices' behaviour according to their electrical properties. The basic electronic elements found in circuit level designs are; linear, lumped, time-invariant resistors, capacitors and inductors and the input sources which include independent and dependent sources. These device models and constitutive equations[6] are employed to formulate the system equations. Moreover, these models are also used to model other classes of electronic devices such as, Diodes, Bipolar and MOS Transistors.

2.2 Network Equation Formulation

Network Equation Formulation takes the device models and their constitutive equations[6] to convert networks of electronic devices into system equations.

Nodal Admittance Formulation[6] is based on KCL which states that the algebraic sum of currents entering each circuit node is zero. Considering a circuit design with N nodes, plus one reference or ground node, applying KCL equation to the design will result in a system of N equations:

$$\begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{bmatrix} \quad \ldots (1)$$

On the other hand, the modified Nodal Formulation[6] combines three electrical characteristic including KCL, KVL and devices' Constitutive Equations(CE), to formulate system equations. The designs may contain various sorts of devices, including ones with or without admittance expressions. The modified nodal technique produces system equations in a compact form. It eliminates the unknown branch currents on those branches that have admittance expressions and are not required as solutions, and rearranges the equations into the form where the required solutions can be solved. The final form of a modified nodal equation is:

$$\begin{bmatrix} A_1 Y_1 A_1^t & A_2 \\ Y_2 A_2^t & Z_2 \end{bmatrix} \begin{bmatrix} V_n \\ I_2 \end{bmatrix} = \begin{bmatrix} -A_3 J \\ W_2 \end{bmatrix} \quad \ldots (2)$$

Yi and Zi are admittance and impedance matrices of circuit elements respectively. Vi are node voltages referenced to the ground node, vector Ii contains branch currents, and J is the vector of the independent current sources. Vector Wi includes the independent current and voltage sources, as well as the influence of initial conditions of capacitors and inductors.

2.3 Numerical Integration Methods

Numerical Integration Methods are mathematical techniques which resolve a system of differential equations into a system of algebraic different equations that can be solved by computer.

The Forward Euler Formula approximates the derivative of a linear function $x(t)$ at $t = t_n$ from the values of the function at $t = t_n$ and $t = t_{n+1}$. Consider a linear function $x(t)$, let x_n be the solution obtained at time $t = t_n$, and its derivative be x'_n. Then, x_{n+1} which is the solution at time $t = t_{n+1}$ can be described by the Forward Euler Formula as:

$$x_{n+1} = x_n + hx'_n \quad \ldots (3)$$

where $h = t_{n+1} - t_n$.

The Backward Euler Formula, on the other hand, uses the derivative x'_{n+1} to determine the value of x_{n+1} at $t = t_{n+1}$. The Backward Euler Formula is described by:

$$x_{n+1} = x_n + hx'_{n+1} \quad \ldots (4)$$

The Trapezoidal Rule Formula takes the average of the derivatives between the two time points to form the relation:

$$x_{n+1} = x_n + \frac{h}{2}\left(x'_n + x'_{n+1}\right) \quad \ldots (5)$$

2.4 The Newton-Raphson Algorithm

The Newton-Raphson Algorithm is an iterative algorithm used for solving a system of nonlinear equations. The system equations which represent the circuit network may contain both time derivatives and nonlinear terms. The time derivative terms can be resolved by applying the numerical integration methods. On the other hand, the system of nonlinear equations can be analysed by employing the Newton-Raphson algorithm.

Consider a system of n nonlinear equations; f_i, which contain variables x_i, i = 1,…, n as shown:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\cdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \quad \ldots (6)$$

Assume that the solution of the system is x^{k+1}, each function can be expanded in a Taylor series about x and if x^k is close to x^{k+1}, the higher order terms may be neglected and the system can be written in a linearised form. The final Newton-Raphson equations for solving a system of nonlinear equation are:

$$\begin{aligned} x^{k+1} &= x^k + \Delta x^k \\ M\Delta x^k &= -f(x^k) \quad \ldots (7) \\ \Delta x^k &= x^{k+1} - x^k \end{aligned}$$

The Jacobian matrix M can be obtained by using element stamp[6]. The NR algorithm is an effective method which can solve the system of nonlinear equations. The algorithm is feasible for computer programming.

3. Gate-Level Simulation Techniques[8]

A Gate-Level Simulator is another class of circuit simulator which operates at a higher level of abstraction. Gate-level simulation techniques are concerned with calculating the logic states and defining timing of the output signals. The logic designs are described as networks of Logic Gates, which include “AND”, “OR”, “NAND”, “NOR” and etc. At the gate-level, the device models operate on the input and output signals that are represented as discrete logic states consist of logic Low, High, Unknown and High Impedance.

3.1 Gate-level Device Models and Delay

Gate-level devices can be thought of as discrete mathematical operators. Logic gates represent fundamental binary operations such as, AND, OR, XOR and NOT, and may also include the digital components, such as a Register or Flipflop. The electrical waveform in the gate-level designs is modelled in term of a logic signal. The logic states that are generally used in gate-level simulators are High, Low and Unknown.

Logic gates not only represent binary operations, but also include Delay Models. The delay models are employed to describe timing characteristics of the real devices. The delay times are generally described in multiples of an integer that represents the smallest delay unit of the design. There are several delay modelling schemes, each of which depicts specific properties of the devices' behaviour.

- **The transport delay** is the simplest form of the gate-level delay model. It describes the propagation time that the logic gate takes to respond to an input change.
- **Rise and Fall Transport Delay**; Two different parameters are used to describe the propagation times when the logic gate changes the output logic state to rise from Low to High and, fall from High to Low.
- **Ambiguous Transport Delay**; The ambiguous delay describes the transition period where the output value is unknown. The ambiguous delay consists of two parameters, the minimum delay time; d_m and the maximum delay time; d_M. d_m is the minimum time that the logic gate takes to start the output transition, and d_M indicates the maximum time that it takes to complete the transition. Therefore, the time period in between d_m and d_M is the ambiguous interval.
- **Inertial Delay**; The inertial delay determines the minimum period of time required to produce adequate energy to cause the output to change. For all circuitry, a certain amount of energy is required to switch the output state. The amount of energy delivered is a function of the time period and the signal strength. An input pulse which has an interval less than the inertial delay, is considered as a Spike.

3.2 Event Driven Algorithm

It has been stated that, at the gate-level, the electrical waveforms are considered as a sequence of logic values changing in time. Logic gates evaluate their response to the input signal changes, and generate an output whose timing is defined by the delay models. The logic signal change that occurs at each circuit node is called an Event. Logic gates are therefore,said to be driven by these events to perform their function.

The Event Driven Algorithm is based on the principle that at any single time point, only the logic gates with changing input signals, will require evaluation. This concept is practical for gate level designs because, logic gates will only respond to the signal changes that appear at their inputs. Applying the event driven algorithm results in only the active parts of the logic designs being evaluated at each single time-step.

The algorithm relies on the event queue to provide a list of events occurring at each time point. It also requires a scheduler to allocate new events into the list. Examples of these algorithms are, Linked List and Time Wheel[8].

4. Mixed-Mode Simulation Techniques[9,10,11]

A Mixed-Mode Simulator is a circuit simulator which is capable of simulating circuit designs that contain models at different levels of abstraction. Mixed-mode simulators offer a trade-off between speed and accuracy for the simulation of mixed-signal designs, and designs that contain critical parts. These simulators are an essential tool for the large scale IC design which contains analogue blocks combined with digital functions.

Mixed-mode simulators can be divided into three different groups according to the schemes that are used to develop the simulators. The two essential components of mixed-mode simulators are the “Interface Model” which converts and passes information between the simulators, and the “Synchronisation Method” which maintains the consistent of the whole simulation process.

4.1 Core Mixed-Mode Simulator

A Core Mixed-Mode Simulator is developed by enhancing the capability of a stand-alone simulator to handle models of different abstraction levels. Core mixed-mode simulators are designed for applications that are predominantly digital or analogue. To simulate a design which consists of mostly analogue parts with a few digital blocks, a circuit level simulator is used as the core and circuit level models of logic gates are developed and included into the simulator. These extra models are optimised to represent digital functional blocks, and only take a little extra CPU time to simulate. On the other hand, a functional model of analogue building blocks may be included in a gate level simulator core, to create a mixed-mode simulator for designs that contain mostly digital parts.

4.2 Unified Mixed-Mode Simulator

A new mixed-mode simulator may be entirely developed from the ground up using a new simulation technique and new model library. A Unified Mixed-Mode Simulator is developed specifically for mixed-signal environments. The simulation algorithm is tailored to simulate mixed analogue and digital parts. The new model library will include primitive models for both analogue, digital and interface devices. Mixed-signal circuit designs can be described in a uniform description, and simulated by using a single algorithm.

One approach to achieving a unified mixed-mode simulator is to develop hardware description languages(HDLs) for behavioural level simulations. The new HDL must support circuit descriptions of both analogue and digital design building blocks. The advantages of the unified mixed-mode technique are the flexibility to simulate various mixed-mode configurations and the efficiency of the unique algorithm. However, the drawbacks are the long development time and a high maintenance cost.

4.3 Glued or Combined Mixed-Mode Simulator

A Glued or Combined Simulator incorporates the existing stand-alone simulators to simulate mixed-signal designs. A simulation controller is developed to combine simulators of different abstraction levels and synchronise their operation. The interface models are also built to couple circuit partitions that are described at the circuit, gate and system levels. Each integrated simulator will run with its own algorithm and time-step.

Since the glued mixed-mode simulators are based on existing simulators, the input language and the model libraries are retained. The maintenance of each simulator and the development of the new device models are moderate. However, in order to get different simulators to work together, the issue of synchronisation techniques and interface models must be addressed. In general, glued simulators are employed to simulate circuits with loose coupling between the analogue and digital parts. To simulate tightly coupled circuits, the synchronisation algorithm and the interface models need to be adjusted accordingly.

5. The Simulation Backplane

An economical and effective approach to simulating mixed-mode circuit designs is to combine existing, reliable and widely adopted simulators to work together. One such approach proposed by the CAD Framework Initiative(CFI)[1] is the Simulation Backplane. The simulation backplane is an integrated simulation environment which combines different types of circuit simulators to simulate circuit designs consisting of multiple partitions described at different abstraction levels. The simulation backplane approach aims to develop a standardised Application Programming Interface(API), that allows a large class of circuit simulators and waveform display tools to be integrated by simply writing an interface.

5.1 Structure of The Alfa Backplane

As it has been stated that the backplane communicates with the integrated simulators by using function calls. The API calls can be achieved by using the standard function calls as proposed in the CFI proposal. However, this approach will require that the backplane and the simulators must be compiled and linked to create an executable object running as a single process.

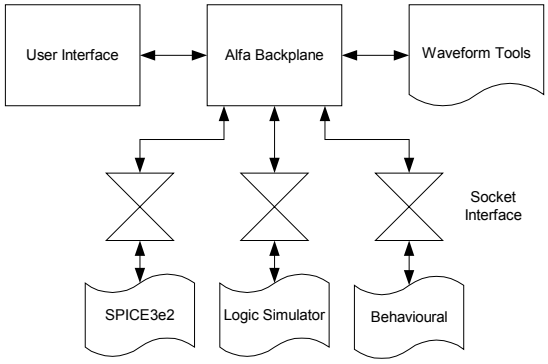


Figure 1 The Structure of Alfa Simulation Backplane

The method proposed and adopted in the Alfa backplane is to separate the backplane and each instance of the simulators into individual processes as shown in Figure 1. This approach will allow the Alfa backplane and the integrated simulators to be developed and maintained separately. A new simulator can be added into the system without the need to recompile the other components. Moreover, since each simulator runs in separate processes, multiple instances of the same simulator can be used at one time.

The Alfa backplane achieves the API calls by employing the interprocess communication utility or Socket[12] which is provided in UNIX systems. The socket and the multiprocessing scheme are transparent to the Alfa backplane and the integrated simulators. The API function calls are translated into a set of parameters sent over the socket by the interface routines. The integrated simulators read these parameters from the socket interfaces, and then interpret them as appropriate function calls.

5.2 Circuit Partitioning

The Alfa backplane requires circuit partitioning. Designers must prepare the designs by partitioning them and assigning the simulators that will be used for each part. Then, the circuit description files must be prepared using the input languages of each simulator. After that, the backplane configuration file written in the Alfa backplane input language must be prepared. The configuration file will contain the list of circuit partitions with their associated simulators and the global port connections. For instance, the Alfa backplane configuration for a Phase Locked Loop (PLL) circuit which is divided in to three partitions is described below[13]:

```
***** PLL backplane configuration *****
als PhaseDectector
  global portinOut   is Output
  global portoutUp   is Up
  global portoutDown  is Down
spi Filter
  global portinxin1   is Up
```

```
  global portinxin22   is Down
  global portoutxout1 is Control
vco Vco
  global portinVcontrol is Control
  global portoutOut   is Output
time 200000
```

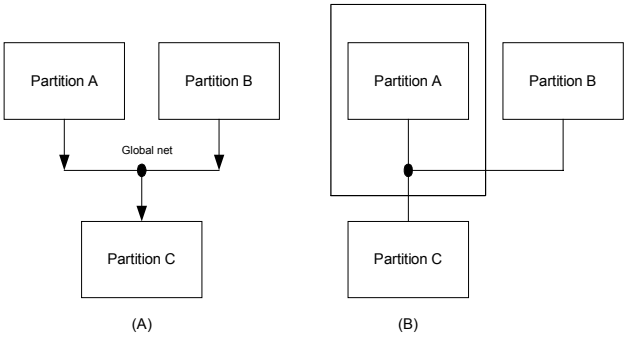


Figure 2 Circuit partitioning in Alfa backplane

A partitioned circuit may have a number of global nets, connecting two or more partitions. In the general case, a partitioning scheme may produce networks with multiply driven nets, as depicted in Figure 2.A. In this example, the global net is driven by outputs of partitions A and B, and its resolved value is driving partition C. The global net is a multiply driven net which will require a solver to resolve the net value. This implies that some forms of resolution function must be included in the backplane. However, including the resolution function would complicate backplane's functionality, since the backplane will not only perform as an integrated environment but itself will become another solver in the system.

To ease the complication of the backplane's functionality, multiply driven nets are not allowed in the design. The method proposed here is to move multiply driven nets into one of the simulators. The multiply driven nets, then can be solved within the simulator. The design configuration in Figure 2.A, then can be resolved by using the configuration depicted in Figure2.B. In Figure 2.B, there are two global nets and neither of which is a multiply driven net. The solution of the multiply driven net is solved in partition A, which now owns the net.

5.3 Global Events

The simulation data are transferred between simulators in the form of Global Events. The global events will contain the event time, the new value and the node identification. The Alfa backplane supports different types of global events, including logic, analogue and integer values. The individual global net must be associated with a unique signal type. The events transferred through this global net, then must be of that particular type. The global net which connects between analogue and digital partitions, for instance, must be assigned to an analogue or a digital type. Then, all the events transferred through this net, must be converted to the corresponding type.

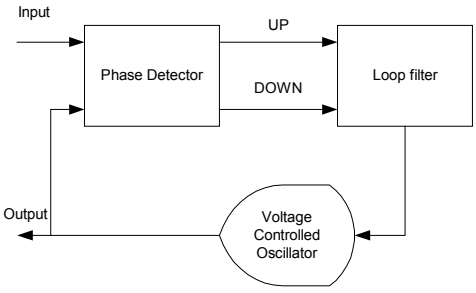
Generally, signal conversion can be done either on the analogue or the digital sides. The two simulators must agree upon the conversion technique that will be used. However, it is reasonable and appropriate in most cases that the conversion should be performed in the partition which contains the more detailed information. In practice, the conversion would be done in the analogue partition, because circuit level simulators offer device models of a higher accuracy, and more accurate signal resolutions.

5.4 The Alfa Backplane Simulation Algorithm

The Alfa simulation backplane employs a hybrid synchronisation technique based on the lock-step algorithm[2]. The lock-step technique has been adopted because, the method is effective to simulate tightly coupled circuits and it does not require backtracking capability. The technique not only allows the backplane to simulate various kinds of mixed-signal applications but also permits a large class of simulators to be used.

The Alfa Backplane Simulation Algorithm

```
while ( time <= end of simulation time)
begin
for ( every simulator )  query for next activity time ;
if ( there is only one smallest next activity time )
  target simulator = the simulator which owns
                     the smallest activity time ;
else
  target simulator = the simulator which has
                     the largest idle count ;
start time = the smallest next activity time ;
target time = the second smallest next activity time ;
invoke the API call to the target simulator to run; update the idle counts ;
if ( there are global events ) transfer global events ;
end ;
```



The Alfa backplane does not have a list of global event to keep all the activities of every partition. On the other hand, it uses the API function to query the next activity time of every partition in each simulation cycle.

The partition's next activity time indicates the next time point that the circuit partitions need to be processed. If the simulator is an event-driven, its next activity time is simply the time of the first event in the event queue. If the event queue is empty, the next activity time is infinity. On the other hand, for circuit

level simulators, their next activity times could not be predicted in advance. Circuit level simulators must evaluate their solutions to convergence and the last solution time point is then used as the circuit's next activity time.

The values of the next activity times are employed to decide which of the simulators will be processed in the current cycle. It is also used to specify the Target Time where the simulator must stop. The selected simulator is allowed to run until it reaches the target time, or it creates global events. If there are global events, the partitions will be synchronised before the next run begins.

6. Simulation Examples

The circuits employed in the simulations are typical real-world mixed-signal applications, which contain mixed analogue and digital parts connected with different degrees of coupling. The problems that will be investigated are: the convergence of simulation of designs with feedback loops; the effects of strong and sensitive coupling ports.

6.1 Phase-Locked Loop Circuit

A Phase-Locked Loop(PLL) is one of the benchmark circuits used for evaluating the performance of mixed-mode circuit simulators. The design contains moderately coupled digital/analogue parts with feedback loop. Simulating a PLL design requires a rather long simulation runtime because in order to get a meaningful result, the design has to be simulated until the PLL is in lock.

The PLL circuit block diagram is shown in Figure[3]. The main components of a PLL design are the Phase Detector, the Loop Filter and the Voltage Controlled Oscillator(VCO). A PLL employs a feedback control technique to synchronise the output clock with the reference input. The phase detector will monitor phase differences between the reference input and the output, and generate UP and DOWN control signals accordingly. The control signals are fed to the loop filter circuit which is chosen to stabilise the feedback loop. The filter's output is used to control the VCO to regulate the phase of the output clock to meet

Figure 3 Phase-locked loop circuit diagram

the input's phase. When, the PLL is in lock, the VCO will generate an output clock with the same phase as the reference input.

The VCO and loop filter must deal with continuous signals, they therefore, will be simulated in the analogue domain. On the other hand, the phase detector which operates with logic values will be simulated at gate level. Three simulators were employed in the simulation. The loop filter was simulated at circuit level, using SPICE3e2, the VCO was simulated at system level using a dedicated simulator written in C, and the phase detector was simulated using a logic simulator. The VCO's centre frequency was set at 15 kHz. It regulates the output clock at 2 kHz/V. The loop filter is a simple RC circuit, it contains two current sources which are controlled by the UP and DOWN signals. The phase detector, on the other hand, is a combinational logic circuit.

In the simulation, the input frequency was set at 15.5kHz, and changed to 20 kHz after 2ms. The 2ms period was chosen to allow the PLL to lock to the first frequency. After the input frequency changed, the simulation was continued until the second frequency was locked. The simulation results are depicted in Figure[4]. The complete simulation took 2 hours and 32 minutes on a Pentium 120 MHz running Linux Operating System. The result showed a succesful simulation of phase-locked operation for both 15.5 and 20 kHz.

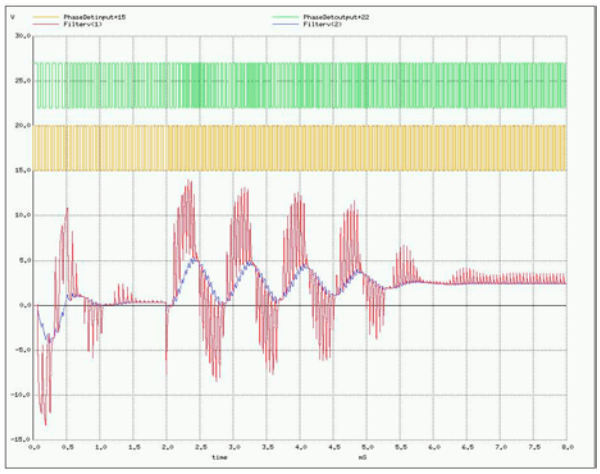


Figure 4 Simulation result of phase-locked loop circuit

7. Conclusion

The simulation backplane approach not only allows designers to perform mixed-signal simulation, but it also provides the basis for CAD tool integration. The simulation backplane has illustrated how different functionalities of different simulators can be combined to do a useful task. IC technology is evolving, new devices are being developed and simulation tools are constantly improving. These different tools will have distinct characteristics, and will be designed for specific purposes. No single tool could handle all kinds of IC design applications. Therefore, a scheme which allows these tools to work together sharing their specific capabilities, will definitely benefit IC designs and developments as a whole. The simulation backplane can be thought of as a step toward the global integration of CAD tools.

References

[1] CAD Framework Initiative, “Draft Proposal Programming Interface Specification Standard Version 0.6”, Technical Report, CAD Framework Initiative Inc. May 1991

[2] Clair Nauts and Oliver Pribetich, “Digital / Analogue Simulation Synchronisation Dilemma, Leap-

frog or Lock-step", The Europeon Design and Test Conference User-forum, pages 175-179, 1995
[3] Eduardo L. Acuna, James P. Dervenis, Andrew J. Pagones, Fred L. Yang and Resve A. Saleh, "Simulation Techniques for Mixed-Mode Analogue/Digital Circuits", IEEE Journal of Solid State Circuits, 25(2), pages 353-363, April 1990
[4] Thomas L. Quarles, A.R. Newton, A.Sangiovanni-Vincentelli and D.O. Pederson, "SPICE3f3 User's Manual", Department of Electrical Engineering and Computer Sciences, University of California Berkeley, May 1993
[5] P. Agrawal and V.D. Agrawal, "Can Logic Simulators Handle Bidirectionality and Charge Sharing", Proceedings of IEEE International Symposium on Circuits and Systems, pages 411-414, 1990
[6] Jiri Vlach and Kishore Singhal, "Computer Method for Circuit Analysis and Design", Van Nostrad Reinhold, New York, 1983
[7] Gary D. Hachtel, Robert K. Brayton and Fred G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design", IEEE Transactions on Circuit Theory, pages 101-113, January 1971
[8] JP. Hayes, 'Digital Simulation with Multiple Logic Values", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pages 274-283, 1986
[9] P.E. Allen, B.P. Lum Shue Chan and W.M. Zuberek, "Comparison of Mixed Analogue-Digital Simulators", Proceedings of IEEE International Symposium on Circuits and Systems, pages 101-104, 1990
[10] R. Beale, R. Chadha, C.F. Chen, A. Posser and K.M. Tham, " Design Methodology and Simulation Tools for Mixed Analogue-Digital Integrated Circuits", IEEE International Symposium on Circuits and Systems, pages 1351-1355, 1990
[11] P. Odryna, "A Unified Mixed-mode Digital-Analogue Simulation Environment", IEEE International Symposium on Circuits and Systems Proceedings, pages 893-896, 1988
[12] Paul S. Wang, "An Introduction to Berkeley Unix", Wadsworth Publishing Company, 1988
[13] M. Zwolinski and C. Garagate, "Anatomy of a Simulation Backplane", IEE Proceedings of Computer and Digital Technology, pages 377-385, 1995