

บทที่ 1

บทนำ

ในยุคที่เทคโนโลยีก้าวไกลดังปัจจุบัน นอกจากการสร้างสรรค์สิ่งใหม่ๆ แล้วสิ่งที่คู่กันก็คือ การวัดและการตรวจสอบ หรือการวิเคราะห์ผลลัพธ์ ว่าชิ้นงานที่สร้างขึ้นมานี้สามารถใช้งานได้ตรงตามทีออกแบบไว้แค่ไหน ซึ่งในขั้นตอนการวัดหรือการตรวจสอบนี้ อุปกรณ์ตรวจวัดค่านับว่ามีบทบาทสำคัญ เพราะในการวัดค่าบางอย่างเพียงสังเกตทางกายภาพหรือมองด้วยตาเปล่าไม่สามารถทำได้ เช่น การวัดความเร็วของรอบมอเตอร์ การวัดค่าของอุปกรณ์สารกึ่งตัวนำต่างๆ ดังนั้นจึงได้มีการพัฒนาอุปกรณ์เครื่องมือสำหรับตรวจวัดขึ้นมา ในอุปกรณ์ตรวจวัดนี้อาจทำออกมาในรูปแบบของเซ็นเซอร์ตรวจจับ หรือไม่ก็ออกแบบมาเป็นเครื่องที่สามารถใช้งานควบคู่กับเครื่องคอมพิวเตอร์ ซึ่งจะต้องใช้โปรแกรมที่เรียกว่า “โปรแกรม Labview” โดยแต่ละเครื่องที่ออกแบบมาก็จะนำไปใช้งานที่แตกต่างกันออกไป

ในปัจจุบันนี้ก็ได้มีการสร้างเครื่องวัดขึ้นมาหลายรูปแบบที่แตกต่างกันทั้งรูปร่างลักษณะ และแตกต่างกันตรงที่การนำเครื่องวัดไปใช้งานต่างๆ ที่แตกต่างกัน ทั้งนี้เพื่อความแม่นยำและอำนวยความสะดวกต่อการวัด ทำให้การวัดค่าต่างๆ ทำได้อย่างรวดเร็วและแม่นยำยิ่งขึ้น

ในบทความนี้ได้้นำเทคโนโลยีเครื่องวัดที่สามารถใช้ร่วมกับคอมพิวเตอร์ได้ ซึ่งต้องใช้โปรแกรม Labview ในการออกแบบวงจรในการวัดอุปกรณ์สารกึ่งตัวนำที่เราต้องการจะวัด ซึ่งใน Project ครั้งนี้ได้นำอุปกรณ์สารกึ่งตัวนำที่มีชื่อว่า “ไดโอด, มอสเฟต, ทรานซิสเตอร์” มาทดลองวัดกับเครื่องวัด และทำการเขียนวงจรเพื่อที่จะวัดอุปกรณ์สารกึ่งตัวนำทั้ง 3 ชนิดนี้

ไดโอด เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดสองขั้ว ที่ออกแบบและควบคุมทิศทางการไหลของประจุไฟฟ้า มันจะยอมให้กระแสไฟฟ้าไหลในทิศทางเดียว และกั้นการไหลในทิศทางตรงกันข้าม เมื่อกล่าวถึงไดโอด มักจะหมายถึงไดโอดที่ทำมาจากสารกึ่งตัวนำ (Semiconductor diode) ซึ่งก็คือผลึกของสารกึ่งตัวนำที่ต่อกันได้ขั้วทางไฟฟ้าสองขั้ว ส่วนใหญ่เราจะใช้ไดโอดในการยอมให้กระแสไหลในทิศทางเดียว โดยยอมให้กระแสไหลในทางใดทางหนึ่ง ส่วนกระแสที่ไหลทิศ

ทางตรงข้ามกันจะถูกกัน ดังนั้นจึงอาจถือว่าไดโอดเป็นวาล์วตรวจสอบแบบอิเล็กทรอนิกส์อย่างหนึ่ง ซึ่งนับเป็นประโยชน์อย่างมากในวงจรอิเล็กทรอนิกส์

ไดโอดชนิดสารกึ่งตัวนำแบบใหม่ๆ มักจะใช้ผลึกสารกึ่งตัวนำจำพวกซิลิกอนที่ไม่บริสุทธิ์ โดยทำการเจือสารให้เกิดสิ่งลบและสิ่งบวก โดยสิ่งลบจะมีประจุลบคืออิเล็กตรอนมากกว่าเรียกว่า "สารกึ่งตัวนำชนิด n (n-type semiconductor)" ส่วนสิ่งบวกจะมีประจุบวกหรือโฮลเรียกว่า "สารกึ่งตัวนำชนิด p (p-type semiconductor)" โดยไดโอดแบบสารกึ่งตัวนำเกิดมาจากการนำสารกึ่งตัวนำทั้งสองชนิดนี้มาติดด้วย วิธีการพิเศษ โดยส่วนที่สารกึ่งตัวนำทั้งสองชนิดอยู่ติดกันนั้นเรียกว่า "รอยต่อ p-n (p-n junction)" ไดโอดชนิดนี้จะยอมให้อิเล็กตรอนไหลผ่านจากสารกึ่งตัวนำชนิด n ไปยังสารกึ่งตัวนำชนิด p เท่านั้น จึงเรียกฝั่งที่มีสารกึ่งตัวนำชนิด n ว่าแคโทด และฝั่งที่มีสารกึ่งตัวนำชนิด p ว่าแอโนด แต่ถ้าพูดถึงทิศทางของกระแสสมมติที่ไหลสวนทางกับกระแสอิเล็กตรอนนั้น จะเห็นว่ากระแสสมมติจะไหลจากขั้วแอโนดหรือสารกึ่งตัวนำชนิด p ไปยังขั้วแคโทดหรือสารกึ่งตัวนำชนิด n เพียงทิศทางเดียวเท่านั้น

มอสเฟต เป็นทรานซิสเตอร์ ที่ใช้อิทธิพลสนามไฟฟ้าในการควบคุมสัญญาณไฟฟ้า โดยใช้ออกไซด์ของโลหะในการทำส่วน GATE นิยมใช้ในวงจรดิจิทัล โดยนำไปสร้างลอจิกเกตต่างๆ เพราะมีขนาดเล็ก

MOSFET ประกอบด้วยสามส่วน คือ GATE, SOURCE, DRAIN และมอสเฟตมีอยู่ด้วยกัน 2 ประเภท คือ

1. nMOS (negative MOSFET) เป็นทรานซิสเตอร์ประเภท NPN เมื่อมีความต่างศักย์เป็นบวก (สนามไฟฟ้าแรง) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้
2. pMOS (positive MOSFET) เป็นทรานซิสเตอร์ประเภท PNP เมื่อมีความต่างศักย์ต่ำหรือเป็นลบ (สนามไฟฟ้าอ่อน) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้

ทรานซิสเตอร์ เป็นอุปกรณ์สารกึ่งตัวนำที่สามารถทำหน้าที่ ขยายสัญญาณไฟฟ้า เปิด/ปิด สัญญาณไฟฟ้า คงค่าแรงดันไฟฟ้า หรือกล้ำสัญญาณไฟฟ้า (modulate) เป็นต้น การทำงานของทรานซิสเตอร์เปรียบได้กับวาล์วที่ถูกควบคุมด้วยสัญญาณไฟฟ้าขาเข้า เพื่อปรับขนาดกระแสไฟฟ้าขาออกที่มาจากแหล่งจ่ายแรงดัน

ทรานซิสเตอร์แบ่งได้เป็นสองประเภทคือ ทรานซิสเตอร์แบบรอยต่อคู่ (Bipolar Junction Transistor, BJTs) และทรานซิสเตอร์แบบสนามไฟฟ้า (Field Effect Transistors, FETs) ทรานซิสเตอร์จะมีขาเชื่อมต่อสามจุด อธิบายโดยย่อคือเมื่อมีการปรับเพิ่มแรงดันไฟฟ้าที่ขาหนึ่งจะส่งผลให้ความนำไฟฟ้าระหว่างขาที่เหลือสูงขึ้นจนทำให้สามารถควบคุมการไหลของกระแสไฟฟ้าได้ อย่างไรก็ตามหลักทางฟิสิกส์ในการทำงานของทรานซิสเตอร์ทั้งสองแบบ(ชนิดรอยต่อ คู่และชนิดสนามไฟฟ้า)มีความแตกต่างกันอยู่มาก ใน วงจรอนาล็อก นั้นทรานซิสเตอร์จะถูกใช้ขยายสัญญาณต่างๆ เช่น สัญญาณเสียง สัญญาณความถี่วิทยุ หรือควบคุมระดับแรงดัน รวมทั้งเป็นแหล่งจ่ายไฟฟ้าแบบสวิชชิงในเครื่องคอมพิวเตอร์ด้วย ทรานซิสเตอร์ก็ยังถูกใช้ในวงจรดิจิทัล เพียงแต่ใช้งานในลักษณะการเปิด/ปิดเท่านั้น วงจรดิจิทัลเหล่านั้นได้แก่ วงจรตรรกะ (Logic gate), หน่วยความจำแบบสุ่ม (Random Access Memory, RAM) และไมโครโพรเซสเซอร์ เป็นต้น

ในปัจจุบันอุปกรณ์สารกึ่งตัวนำได้มีการนำไปใช้งานอย่างกว้างขวาง เนื่องจากมีขนาดเล็ก น้ำหนักเบา ทำงานที่แรงดันไฟฟ้าต่ำ ใช้กำลังไฟฟ้าน้อย และสามารถทำงานได้อย่างมีประสิทธิภาพ โคนมีความน่าเชื่อถือได้สูง และนำมาประยุกต์ใช้งานให้เข้ากับยุคเทคโนโลยีเพิ่มมากขึ้น เพื่อความสะดวกสบายในการดำเนินชีวิตของมนุษย์เรา

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ไดโอด

ไดโอด (อังกฤษ: diode) เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดสองขั้ว ที่ออกแบบและควบคุมทิศทางการไหลของประจุไฟฟ้า มันจะยอมให้กระแสไฟฟ้าไหลในทิศทางเดียว และกั้นการไหลในทิศทางตรงกันข้าม เมื่อกล่าวถึงไดโอด มักจะหมายถึงไดโอดที่ทำมาจากสารกึ่งตัวนำ (Semiconductor diode) ซึ่งก็คือผลึกของสารกึ่งตัวนำที่ต่อกันได้ขั้วทางไฟฟ้าสองขั้ว ส่วนไดโอดแบบหลอดสุญญากาศ (Vacuum tube diode) ถูกใช้เฉพาะทางในเทคโนโลยีไฟฟ้าแรงสูงบางประเภท เป็นหลอดสุญญากาศที่ประกอบด้วยขั้วอิเล็กโทรดสองขั้ว ซึ่งก็คือแผ่นตัวนำ (plate) และแคโทด (cathode)

ส่วนใหญ่เราจะใช้ไดโอดในการยอมให้กระแสไปในทิศทางเดียว โดยยอมให้กระแสไหลในทางใดทางหนึ่ง ส่วนกระแสที่ไหลทิศทางตรงข้ามกันจะถูกกั้น ดังนั้นจึงอาจถือว่าไดโอดเป็นวาล์วตรวจสอบแบบอิเล็กทรอนิกส์อย่างหนึ่ง ซึ่งนับเป็นประโยชน์อย่างมากในวงจรอิเล็กทรอนิกส์ เช่น ใช้เป็นเรียงกระแสไฟฟ้าในวงจรภาคจ่ายไฟ เป็นต้น

อย่างไรก็ตามไดโอดมีความสามารถมากกว่าการเป็นอุปกรณ์ที่ใช้เปิด-ปิดกระแส ง่ายๆ ไดโอดมีคุณลักษณะทางไฟฟ้าที่ไม่เป็นเชิงเส้น ดังนั้นมันยังสามารถปรับปรุงโดยการปรับเปลี่ยนโครงสร้างของพวกมันที่เรียก ว่าจุดเชื่อมต่อ P-N มันถูกนำไปใช้ประโยชน์ในงานที่มีวัตถุประสงค์พิเศษ นั่นทำให้ไดโอดมีรูปแบบการทำงานได้หลากหลายรูปแบบ ยกตัวอย่างเช่น ซีเนอร์ไดโอดเป็นไดโอดชนิดพิเศษที่ทำหน้าที่รักษาระดับแรงดันให้คงที่ วาริแอกไดโอดใช้ในการปรับแต่งสัญญาณในเครื่องรับวิทยุและโทรทัศน์ ทันเนลไดโอดใช้ในการสร้างสัญญาณความถี่วิทยุ และไดโอดเปล่งแสงเป็นอุปกรณ์ที่สร้างแสงขึ้น ทันเนลไดโอดมีความน่าสนใจตรงที่มันจะมีค่าความต้านทานติดลบ ซึ่งเป็นประโยชน์มากเมื่อใช้ในวงจรบางประเภท

ไดโอดตัวแรกเป็นอุปกรณ์หลอดสุญญากาศ โดยไดโอดแบบสารกึ่งตัวนำตัวแรกถูกค้นพบจากการทดสอบความสามารถในการเรียงกระแสของผลึกโดยคาร์ล เฟอร์ดินานด์ บรอน นักฟิสิกส์ชาวเยอรมัน ในปี พ.ศ. 2417 เรียกว่า cat's whisker diodes และได้ถูกพัฒนาในปี พ.ศ. 2449 โดยทำไดโอดมาจากผลึกแร่กลีนา แต่ทุกวันนี้ไดโอดที่ใช้ทั่วไปผลิตมาจากสารกึ่งตัวนำ เช่น ซิลิกอน หรือเจอร์เมเนียม

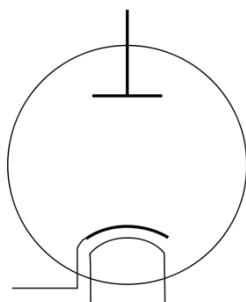
ไดโอดเป็นอุปกรณ์ที่ทำจากสารกึ่งตัวนำ p - n สามารถควบคุมให้กระแสไฟฟ้าจากภายนอกไหลผ่านตัวมันได้ทิศทางเดียว ไดโอดประกอบด้วยขั้ว 2 ขั้ว คือ แอโนด (Anode; A) ซึ่งต่ออยู่กับสารกึ่งตัวนำชนิด p และ แคโทด (Cathode; K) ซึ่งต่ออยู่กับสารกึ่งตัวนำชนิด n

2.1.1 ไดโอดแบบใช้ความร้อนและไดโอดแบบสถานะแก๊ส

ไดโอดแบบใช้ความร้อนเป็นหลอดสุญญากาศ ภายในประกอบไปด้วยขั้วไฟฟ้า (electrode) ล้อมรอบด้วยสุญญากาศภายในหลอดแก้ว คล้ายๆ กับหลอดไส้ (incandescent light bulb)

ในไดโอดแบบใช้ความร้อนนั้น กระแสจะไหลผ่านไส้ความร้อนและให้ความร้อนแก่ขั้วลบหรือขั้วแคโทด ซึ่งขั้วไฟฟ้าจะทำมาจากแบเรียมและสตรอนเชียมออกไซด์ ซึ่งเป็นออกไซด์ของโลหะแอลคาไลน์เอิร์ท ซึ่งมีสภาวะงาน (work function) ต่ำ (บางครั้งจะใช้วิธีให้ความร้อนโดยตรง โดยการใส่ไส้หลอดเป็นทั้งสแตน แทนไส้ความร้อนกับขั้วแคโทด) ความร้อนอันเกิดมาจากการส่งผ่านความร้อนของอิเล็กตรอนไป สู่สุญญากาศ ขึ้นต่อไปคือขั้วบวกหรือขั้วแอโนดที่ล้อมรอบไส้ความร้อนอยู่ จะทำหน้าที่เป็น ประจุบวก นั้นจะเกิดการส่งผ่านอิเล็กตรอนด้วยไฟฟ้าสถิตย์ อย่างไรก็ตามอิเล็กตรอนจะไม่ถูกปล่อยไปโดยง่ายจากขั้วแอโนดเมื่อเราต่อกลับขั้ว เพราะขั้วแอโนดไม่มีความร้อน ดังนั้น ไดโอดแบบใช้ความร้อนจะทำให้อิเล็กตรอนไหลทิศทางเดียว

สำหรับคริสต์ศตวรรษที่ 20 ไดโอดแบบใช้ความร้อนถูกใช้ในสัญญาณอนาล็อก และใช้เรียงกระแสในแหล่งจ่ายกำลังมากมาย ทุกวันนี้ไดโอดที่เป็นหลอดสุญญากาศในกีตาร์ไฟฟ้าและเครื่องขยายเสียงแบบไฮ-เอน รวมทั้งอุปกรณ์ที่ใช้แรงดันสูงๆ



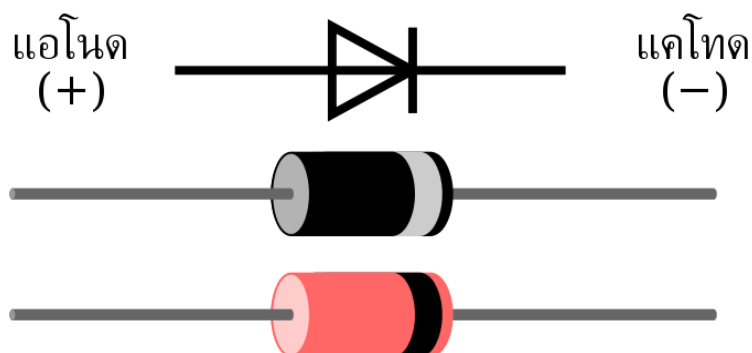
รูปที่ 2.1.1 สัญลักษณ์ของไดโอดแบบใช้ความร้อน จากด้านบนถึงด้านล่างคือขั้วบวก (anode), ขั้วลบ (cathode) และใช้ความร้อน

2.1.2 ไดโอดแบบสารกึ่งตัวนำ

ไดโอดชนิดสารกึ่งตัวนำแบบใหม่ๆ มักจะใช้ผลึกสารกึ่งตัวนำจำพวกซิลิกอนที่ ไม่บริสุทธิ์ โดยทำการเจือสารให้เกิดสิ่งลบและสิ่งบวก โดยสิ่งลบจะมีประจุลบคืออิเล็กตรอนมากกว่าเรียกว่า "สารกึ่งตัวนำชนิด n (n-type semiconductor)" ส่วนสิ่งบวกจะมีประจุบวกหรือโฮลเรียกว่า "สารกึ่งตัวนำชนิด p (p-type semiconductor)" โดยไดโอดแบบสารกึ่งตัวนำเกิดมาจากการนำสารกึ่งตัวนำทั้งสองชนิดนี้มาติดด้วย วิธีการพิเศษ โดยส่วนที่สารกึ่งตัวนำทั้งสองชนิดอยู่ติดกันนั้นเรียกว่า "รอยต่อ p-n (p-n junction)" ไดโอดชนิดนี้จะยอมให้อิเล็กตรอนไหลผ่านจากสารกึ่งตัวนำชนิด n ไปยังสารกึ่งตัวนำชนิด p เท่านั้น จึงเรียกสิ่งที่มีสารกึ่งตัวนำชนิด n ว่าแคโทด และสิ่งที่มีสารกึ่งตัวนำชนิด p ว่าแอโนด แต่ถ้าพูดถึงทิศทางของกระแสสมมติที่ไหลสวนทางกับกระแสอิเล็กตรอนนั้น จะเห็นว่ากระแสสมมติจะไหลจากขั้วแอโนดหรือสารกึ่งตัวนำชนิด p ไปยังขั้วแคโทดหรือสารกึ่งตัวนำชนิด n เพียงทิศทางเดียวเท่านั้น

ในปัจจุบันไดโอดที่ทำมาจากสารกึ่งตัวนำได้มีการนำไปใช้งานอย่างกว้างขวาง เนื่องจากมีขนาดเล็ก น้ำหนักเบา ทำงานที่แรงดันไฟฟ้าต่ำ ใช้กำลังไฟฟ้าน้อย และสามารถทำงานได้อย่างมีประสิทธิภาพ โดนมีความน่าเชื่อถือได้สูง และนำมาประยุกต์ใช้งานให้เข้ากับยุคเทคโนโลยีเพิ่มมากขึ้น เพื่อความสะดวกสบายในการดำเนินชีวิตของมนุษย์เรา

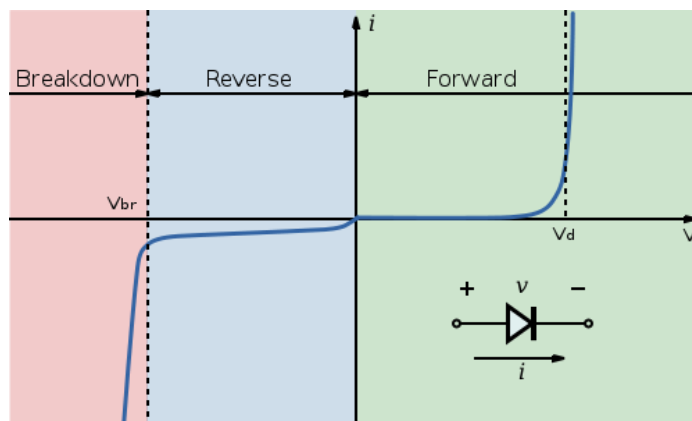
ไดโอดแบบสารกึ่งตัวนำอีกรูปแบบหนึ่งที่สำคัญก็คือ ไดโอดชอททกี (Schottky diode) ซึ่งมีหน้าสัมผัสระหว่างโลหะกับสารกึ่งตัวนำมากกว่ารอยต่อ p-n



รูปที่ 2.1.2 ไดโอดเทียบกับสัญลักษณ์ของไดโอดแบบสารกึ่งตัวนำ (บนสุด) โดยแถบสีดำแสดงฝั่งที่เป็นขั้วแคโทด

2.1.3 คุณสมบัติเฉพาะของกระแสและแรงดัน

พฤติกรรมของไดโอดแบบสารกึ่งตัวนำในวงจรจะก่อให้เกิดคุณสมบัติเฉพาะของกระแสและแรงดัน (current-voltage characteristic) หรือเรียกว่ากราฟ I-V รูปร่างของเส้นโค้งถูกกำหนดจากสัณฐานประจุผ่านเขตปลอดพาหะ (depletion region หรือ depletion layer) ซึ่งอยู่ไยรอยต่อ p-n



รูปที่ 2.1.3 กราฟคุณสมบัติเฉพาะของกระแสและแรงดันของรอยต่อ p-n ของไดโอด

2.1.4 สมการของไดโอดชอททกี

สมการของไดโอดชอททกีในอุดมคติหรือกฎของไดโอดได้ให้สมการที่แสดงถึงกราฟคุณลักษณะเฉพาะของกระแสและแรงดันเอาไว้ว่า

$$I = I_S (e^{V_D/(nV_T)} - 1) ,$$

เมื่อ I คือกระแสที่ไหลผ่านไดโอด

I_S คือกระแสอิ่มตัวเมื่อทำการไบอัสกลับ

V_D คือแรงดันที่ตกคร่อมไดโอด

V_T คือค่าความต่างศักย์อันเนื่องมาจากความร้อน

n คือค่าตัวประกอบอุดมคติ (ideality factor) หรือค่าตัวประกอบคุณภาพ (quality factor) หรือสัมประสิทธิ์การส่งผ่าน (emission coefficient)

ค่าความต่างศักย์อันเนื่องมาจากความร้อน(thermal voltage) V_T มีค่าประมาณ 25.85 mV ที่อุณหภูมิ 300 K ซึ่งเป็นอุณหภูมิห้องปฏิบัติการ แต่เราก็สามารถหาค่าดังกล่าวเมื่ออุณหภูมิอื่นๆได้

$$\text{จากสูตร : } V_T = \frac{kT}{q} ,$$

เมื่อ k คือค่าคงที่ของโบลต์ซมานน์ มีค่าเท่ากับ $1.3806503 \cdot 10^{-23} \times \text{J K}^{-1}$

T คืออุณหภูมิสัมบูรณ์ที่รอบต่อ p-n

q คือประจุของอิเล็กตรอน มีค่าเท่ากับ $1.602176487 \cdot 10^{-19} \times \text{C}$

สมการของไดโอดชอทткиในอุดมคติหรือกฎของไดโอดนั้นเกิดมาจากการอ้างสมมติฐานของกระบวนการเกิดการกระแสไฟฟ้าในไดโอดว่า เป็นการลอยผ่าน, การแพร่, และการรวมความร้อนอีกครั้ง นอกจากนี้ยังสันนิษฐานว่ากระแสจากการรวมตัวอีกครั้ง ณ เขตปลอดพาหะไม่มีนัยสำคัญใดๆ นั่นหมายความว่าสมการของไดโอดชอทткиไม่ต้องคำนวณผลของกระบวนการที่เกี่ยวข้องกับการพังทลายเมื่อกระแสนอนกลับและโฟตอนที่ช่วยให้เกิด R-G

2.1.5 พฤติกรรมของสัญญาณขนาดเล็ก

ในการออกแบบวงจร แบบจำลองของสัญญาณขนาดเล็กจากพฤติกรรมของไดโอดถูกนำมาใช้งานอยู่บ่อยครั้ง

แบบจำลองสัญญาณขนาดเล็ก (Small-signal model) เป็นเทคนิคการวิเคราะห์ทางวิศวกรรมไฟฟ้า ที่อาศัยการประมาณพฤติกรรมของอุปกรณ์ทางไฟฟ้าที่ไม่มีความเป็นเชิงเส้น ด้วยสมการเชิงเส้น ความเป็นเชิงเส้นนี้ขึ้นอยู่กับจุดไบอัสกระแสตรง (DC bias point) ของอุปกรณ์ (นั่นก็คือระดับของ แรงดัน/กระแส ที่แสดงออกเมื่อไม่มีสัญญาณที่ถูกนำมาใช้) และสามารถทำให้ถูกต้องได้ด้วยการมองที่จุดนี้อีกด้วย

2.1.6 ความต้านทาน

เมื่อใช้สมการชอทท์กีไดโอด ค่าความต้านสัญญาณขนาดเล็ก (r_D) ของไดโอดสามารถเข้ามาเกี่ยวกับจุดปฏิบัติการ (Q-point) ที่กระแสไบอัสกระแสตรง (I_Q) และแรงดันใช้งานที่จุดปฏิบัติการ (V_Q)^[9] แรกเริ่มเดิมทีค่าความนำสัญญาณขนาดเล็ก (g_D) ถูกตั้งขึ้น นั่นคือประจุไฟฟ้าในกระแสไฟฟ้าที่ไหลในไดโอดที่เกิดมาจากการเปลี่ยนแปลงเล็กๆ ของแรงดันที่ตกคร่อมไดโอดหารด้วยแรงดันตกคร่อมไดโอดนั้น ดังสมการ

$$g_D = \left. \frac{dI}{dV} \right|_Q = \frac{I_0}{V_T} e^{V_Q/V_T} \approx \frac{I_Q}{V_T}$$

การประมาณค่าเกิดมาจากการอนุมานว่ากระแสไบอัส I_Q นั้นมากพอที่จะทำให้ค่าตัวประกอบ (factor) ของส่วนที่ละลายได้จากสมการชอทท์กีมีค่าเท่า 1 โดยการประมาณนี้มีความถูกต้องแม้แรงดันจะมีค่าต่ำ เพราะแรงดันอันเนื่องมาจากความร้อน (thermal voltage) $V_T \approx 26 \text{ mV}$ ที่อุณหภูมิ 300 เคลวิน (27 องศาเซลเซียส) ดังนั้น V_Q/V_T มีแนวโน้มมากขึ้น หมายความว่าตัวชี้กำลังมีค่าสูงมาก

แต่ไม่ใช่กับค่าความต้านทานสัญญาณขนาดเล็ก r_D ซึ่งเป็นส่วนกลับของค่าความนำสัญญาณขนาดเล็ก ค่าความต้านทานสัญญาณขนาดเล็กไม่ขึ้นอยู่กับไฟฟ้ากระแสสลับ แต่จะขึ้นอยู่กับไฟฟ้ากระแสตรงเท่านั้น ดังสมการ

$$r_D = \frac{V_T}{I_Q}$$

2.1.7 ความเก็บประจุ

ประจุไฟฟ้าจะนำพากระแสไฟฟ้า I_Q ตามสูตร $Q = I_Q \tau_F + Q_J$

เมื่อ τ_F คือเวลาที่ประจุเคลื่อนที่ไป ส่วนแรกคือประจุที่เคลื่อนที่ผ่านไดโอดแล้วเกิดกระแส I_Q ไหลผ่านไดโอด ที่ ส่วนที่สองคือประจุที่เก็บสะสมอยู่ที่รอยต่อ p-n จึงมีคุณลักษณะคล้ายกับตัวต้านทานง่ายๆ มีขั้วทางไฟฟ้าที่มีประจุตรงข้ามกัน ประจุนี้ถูกกักเก็บที่ไดโอดอาศัยแค่แรงดันที่ตกคร่อมตัวมันเพียงแค่นั้น โดยไม่คำนึงถึงกระแสไฟฟ้าที่ไหลผ่าน

การหาค่าความเก็บประจุของไดโอด C_D หาได้จากสมการ

$$C_D = \frac{dQ}{dV_Q} = \frac{dI_Q}{dV_Q} \tau_F + \frac{dQ_J}{dV_Q} \approx \frac{I_Q}{V_T} \tau_F + C_J$$

เมื่อ $C_J = \frac{dQ_J}{dV_Q}$ คือค่าความเก็บประจุที่รอยต่อ p-n โดยประจุในส่วนแรกเรียกว่า ค่าความเก็บประจุแพร่ (diffusion capacitance) เพราะเกี่ยวข้องกับกระแสที่แพร่ตรงรอยต่อของไดโอด

2.1.8 การฟื้นตัวกลับ

ช่วงท้ายของการไบอัสตรงของไดโอดแบบสารกึ่งตัวนำ จะเกิดกระแสไหลไฟฟ้าที่ไหลย้อนกลับในช่วงระยะเวลาสั้นๆ ตัวอุปกรณ์จะยังไม่สามารถป้องกันกระแสไหลย้อนกลับได้เต็มที่จนกระทั่งกระแส ที่เกิดไหลย้อนกลับนั้นได้สิ้นสุดลง

ผลกระทบที่เกิดขึ้นนั้นมีความสำคัญเมื่อมีการสวิตซ์ (switching) ของกระแสที่สูงและรวดเร็วมาก (di/dt มีค่า 100 A/ μ s หรือมากกว่านั้น) ค่าที่แน่นอนของ "เวลาฟื้นตัวกลับ (reverse recovery time)" t_r (อยู่ในช่วงเวลาเป็นนาโนวินาที) อาจจะเคลื่อนย้าย "ประจุฟื้นตัวกลับ (reverse recovery charge)" Q_r (อยู่ในช่วงนาโนคูลอมป์) ออกจากไดโอด ในระยะเวลาระหว่างฟื้นตัวนี้ ไดโอดจะสามารถทำงานในทิศทางตรงข้ามได้ แน่นอนว่าในความเป็นจริงผลกระทบนี้มีความสำคัญในการพิจารณาความสูญเสียที่ เกิดขึ้นอันเนื่องมาจากไดโอดไม่เป็นอุดมคติ อย่างไรก็ตามอัตราการแกว่ง (slew rate) ของกระแสไฟฟ้านั้นรุนแรงมาก (di/dt มีค่า 10 A/ μ s หรืออน้อยกว่านั้น) ผลกระทบนี้ยังอยู่ในเกณฑ์ที่ปลอดภัยจึงละเลยไว้ได้ ในการใช้งานไดโอดส่วนใหญ่จึงไม่มีผลกระทบที่สำคัญมากนัก

กรณีที่กระแสไหลย้อนกลับอย่างฉับพลันเมื่อประจุไฟฟ้าที่ถูกเก็บสะสมปลด พาหะแล้ว จะนำไปใช้ประโยชน์ในขั้นตอนการฟื้นตัวของไดโอดสำหรับกำเนิดสัญญาณ พัลส์สั้นโดยเฉพาะ

2.1.9 ประเภทของไดโอดแบบสารกึ่งตัวนำ

แบ่งออกได้เป็น 5 ประเภท คือ

1. ไดโอดเปล่งแสงหรือแอลอีดี (Light Emitting Diode ; LED)

LED เป็นไดโอดที่ใช้สารประเภทแกเลียมอาร์เซไนด์ฟอสไฟด์ (Gallium Arsenide Phosphide ; GaAsP) หรือสารแกเลียมฟอสไฟด์ (Gallium Phosphide ; GaP) มาทำเป็นสารกึ่งตัวนำชนิด p และ n แทนสาร Si และ Ge สารเหล่านี้มีคุณลักษณะพิเศษ คือ สามารถเรืองแสงได้เมื่อได้รับไบอัสตรง การเกิดแสงที่ตัว LED นี้เราเรียกว่า อิเล็กโทรลูมิเนสเซนซ์ (Electroluminescence) ปัจจุบันนิยมใช้ LED แสดงผลในเครื่องมืออิเล็กทรอนิกส์ เช่น เครื่องคิดเลข, นาฬิกา เป็นต้น

2. โฟโตไดโอด (Photo Diode)

โฟโตไดโอด เป็นไดโอดที่อาศัยแสงจากภายนอกผ่านเลนส์ ซึ่งฝังตัวอยู่ระหว่างรอยต่อ p-n เพื่อกระตุ้นให้ไดโอดทำงาน การต่อโฟโตไดโอดเพื่อใช้งานจะเป็นแบบไบอัสกลับ ทั้งนี้เพราะไม่ต้องการให้โฟโตไดโอดทำงานในทันทีทันใด แต่ต้องการให้ไดโอดทำงานเฉพาะเมื่อมีปริมาณแสงสว่างมากพอตามที่กำหนดเสีย ก่อน กล่าวคือ เมื่อเลนส์ของโฟโตไดโอดได้รับแสงสว่างจะเกิดกระแสรั่วไหล ปริมาณกระแสรั่วไหลนี้เพิ่มขึ้นตามความเข้มของแสง

3. ไดโอดกำลัง (Power Diode)

ไดโอดกำลัง เป็นไดโอดที่ออกแบบให้บริเวณรอยต่อมีช่วงกว้างมากกว่าไดโอดทั่วไป เพื่อนำไปใช้กับงานที่มีกำลังไฟฟ้าสูง กระแสสูงและทนต่ออุณหภูมิสูงได้ เช่น ประกอบเป็นวงจรเรียงกระแส ในอิเล็กทรอนิกส์กำลัง เป็นต้น จะเห็นได้ว่าเมื่อพิกัดกระแสไฟฟ้ามีค่าหลายร้อยแอมป์ ทำให้ไดโอดมีอุณหภูมิขณะทำงานสูง โดยทั่วไปจึงนิยมใช้ร่วมกับตัวระบายความร้อน (Heat Sinks) เพื่อเพิ่มพื้นที่ระบายความร้อนภายในตัวไดโอดกำลัง

4. ไดโอดวาร์กเตอร์หรือวาร์แคป (Varactor or Varicap Diode)

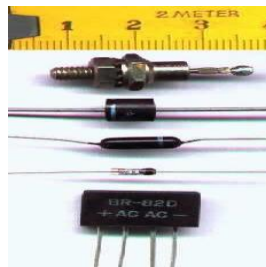
ไดโอดวาร์กเตอร์หรือวาร์แคปเป็นไดโอดที่มีลักษณะพิเศษ คือ สามารถปรับค่าคาปาซิแตนซ์เชื่อมต่อ (Ct) ได้โดยการปรับค่าแรงดันไบอัสกลับ ไดโอดประเภทนี้มีโครงสร้างเหมือนกับ

ไดโอดทั่วไป ขณะแรงดันไบอัสกลับ (Reverse Bias Voltage ; V_r) มีค่าต่ำ Depletion Region จะแคบลงทำให้ C_t ครอบรอบมีค่าสูง แต่ในทางตรงข้ามถ้าเราปรับ V_r ให้สูงขึ้น Depletion Region จะขยายกว้างขึ้น ทำให้ C_t มีค่าต่ำ จากลักษณะดังกล่าว เราจึงนำวารีแคปไปใช้ในวงจรปรับความถี่ เช่น วงจรจูนความถี่อัตโนมัติ (Automatic Fine Tuning ; AFC) และวงจรกรองความถี่ซึ่งปรับช่วงความถี่ได้ตามต้องการ (Variable Bandpass Filter) เป็นต้น

5. ซีเนอร์ไดโอด (Zener Diode)

ซีเนอร์ไดโอดเป็นอุปกรณ์สารกึ่งตัวนำที่นำกระแสได้เมื่อได้รับไบอัสกลับ และระดับแรงดันไบอัสกลับที่นำซีเนอร์ไดโอดไปใช้งานได้เรียกว่า ระดับแรงดันพังทลายซีเนอร์ (Zener Breakdown Voltage ; V_z) ซีเนอร์ไดโอดจะมีแรงดันไบอัสกลับ (V_r) น้อยกว่า V_z เล็กน้อย ไดโอดประเภทนี้เหมาะ

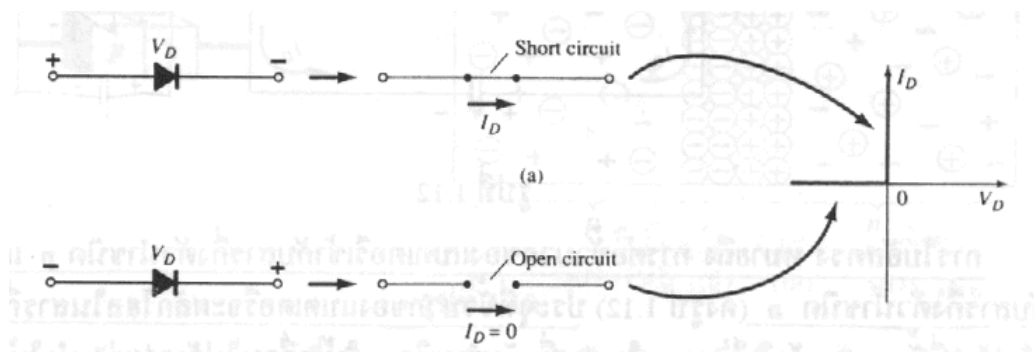
ที่จะนำไปใช้ควบคุมแรงดันที่โหลดหรือวงจรที่ต้องการแรงดันคงที่ เช่น ประกอบอยู่ในแหล่งจ่ายไฟเลี้ยง หรือโวลเทจเรกูเลเตอร์



รูปที่ 2.1.4 ไดโอดชนิดต่างๆ

2.1.10 ไดโอดในทางอุดมคติ

ไดโอดในอุดมคติ มีลักษณะเหมือนสวิตช์ที่สามารถนำกระแสไหลผ่านได้ในทิศทางเดียว ถ้าต่อขั้วแบตเตอรี่ให้เป็นแบบไบอัสตรงไดโอดจะเปรียบเสมือนกับสวิตช์ที่ปิด หรือไดโอดลัดวงจร I_d ไหลผ่านไดโอดได้ แต่ถ้าต่อขั้วแบตเตอรี่แบบไบอัสกลับ ไดโอดจะเปรียบเสมือนสวิตช์เปิด หรือเปิดวงจรทำให้ I_d เท่ากับศูนย์

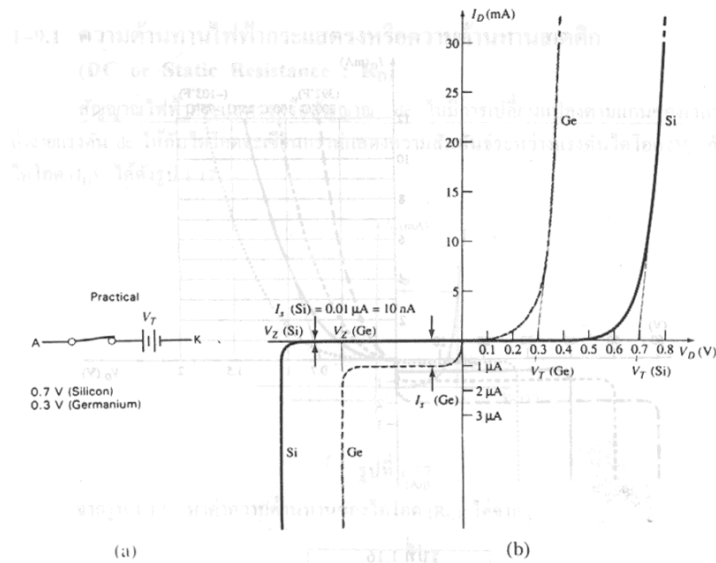


รูปที่ 2.1.5 ไดโอดในทางอุดมคติ

2.1.11 ไดโอดในทางปฏิบัติ

ไดโอดในทางปฏิบัติ มีการแพร่กระจายของพาหะส่วนน้อยที่บริเวณรอยต่ออยู่จำนวนหนึ่ง ดังนั้น ถ้าต่อไบอัสตรงให้กับไดโอดในทางปฏิบัติก็จะเกิด แรงดันเสมือน ($V_{Ge} \approx 0.3V$; $V_{Si} \approx 0.7V$) ซึ่งต้านแรงดันไฟฟ้าที่จ่ายเพื่อการไบอัสตรง ขนาดของแรงดันเสมือนจึงเป็นตัวบอกจุดทำงาน ดังนั้น จึงเรียก "แรงดันเสมือน" อีกอย่างหนึ่งว่า "แรงดันในการเปิด" (Turn-on Voltage ; V_t)

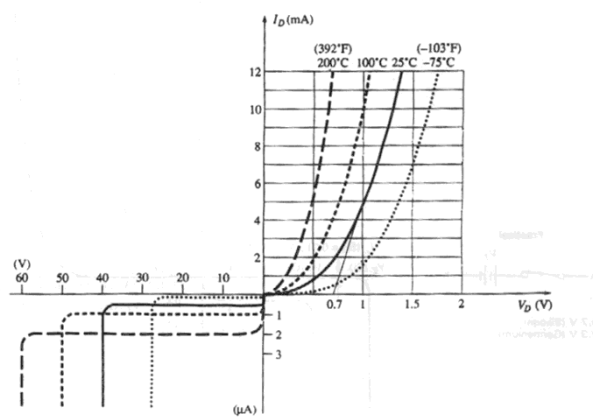
กรณีไบอัสกลับ เราทราบว่า Depletion Region จะขยายกว้างขึ้น แต่ก็ยังมีพาหะข้างน้อยแพร่กระจายที่รอยต่ออยู่จำนวนหนึ่ง แต่ก็ยังมีกระแสรั่วไหลอยู่จำนวนหนึ่ง เรียกว่า กระแสรั่วไหล เมื่อเพิ่มแรงดันไฟฟ้าขึ้นเรื่อยๆ กระแสรั่วไหลจะเพิ่มขึ้นจนถึงจุดที่ไดโอดนำกระแสเพิ่มขึ้นมาก ระดับกระแสที่จุดนี้ เรียกว่า "กระแสอิ่มตัวย้อนกลับ" (Reverse Saturation Current ; I_s) แรงดันไฟฟ้าที่จุดนี้ เรียกว่า แรงดันพังทลาย (Breakdown Voltage) และถ้าแรงดันไบอัสสูงขึ้นไปจนถึงจุดสูงสุดที่ไดโอดทนได้ เราเรียกว่า "แรงดันพังทลายซีเนอร์" (Zener Breakdown Voltage ; V_z) ถ้าแรงดันไบอัสกลับสูงกว่า V_z จะเกิดความร้อนอย่างมากที่รอยต่อของไดโอด ส่งผลให้ไดโอดเสียหายหรือพังได้ แรงดันไฟฟ้าที่จุดนี้เราเรียกว่า แรงดันพังทลายอวาเลนซ์ (Avalanche Breakdown Voltage) ดังนั้น การนำไดโอดไปใช้งานจึงใช้กับการไบอัสตรงเท่านั้น



รูปที่ 2.1.6 ไดโอดในทางปฏิบัติ

2.1.12 ผลกระทบของอุณหภูมิ (Temperature Effects)

จากการทดลองพบว่า I_s ของ Si จะมีค่าเพิ่มขึ้นเกือบ 2 เท่า ทุกๆ ครั้งที่อุณหภูมิเพิ่มขึ้น 10 องศาเซลเซียส ขณะที่ Ge มีค่า I_s เป็น 1 หรือ 2 micro-amp ที่ 25 องศาเซลเซียส แต่ที่ 100 องศาเซลเซียสจะมีค่า I_s เพิ่มขึ้นเป็น 100 micro-amp ระดับกระแสไฟฟ้าขนาดนี้จะเป็นปัญหาต่อการเปิดวงจรเนื่องจากได้รับการไบอัสกลับ เพราะแทนที่ I_d จะมีค่าใกล้เคียงศูนย์ แต่กลับนำกระแสได้จำนวนหนึ่งตามอุณหภูมิที่เพิ่มขึ้น



รูปที่ 2.1.7 ไดโอดที่มีผลกระทบของอุณหภูมิ

2.2 ทรานซิสเตอร์

ทรานซิสเตอร์ เป็นอุปกรณ์สารกึ่งตัวนำที่สามารถทำหน้าที่ ขยายสัญญาณไฟฟ้า เปิด/ปิด สัญญาณไฟฟ้า คงค่าแรงดันไฟฟ้า หรือกล้ำสัญญาณไฟฟ้า (modulate) เป็นต้น การทำงานของทรานซิสเตอร์เปรียบได้กับวาล์วที่ถูกควบคุมด้วยสัญญาณไฟฟ้าเข้า เพื่อปรับขนาดกระแสไฟฟ้าขาออกที่มาจากแหล่งจ่ายแรงดัน

ทรานซิสเตอร์แบ่งได้เป็นสองประเภทคือ ทรานซิสเตอร์แบบรอยต่อคู่ (Bipolar Junction Transistor, BJTs) และทรานซิสเตอร์แบบสนามไฟฟ้า (Field Effect Transistors, FETs) ทรานซิสเตอร์จะมีขาเชื่อมต่อสามจุด อธิบายโดยย่อคือเมื่อมีการปรับเพิ่มแรงดันไฟฟ้าที่ขาหนึ่งจะส่งผลให้ความนำไฟฟ้าระหว่างขาที่เหลือสูงขึ้นจนทำให้สามารถควบคุมการไหลของกระแสไฟฟ้าได้ อย่างไรก็ตามหลักทางฟิสิกส์ในการทำงานของทรานซิสเตอร์ทั้งสองแบบ(ชนิดรอยต่อ คู่และชนิดสนามไฟฟ้า)มีความแตกต่างกันอยู่มาก ใน วงจรอนาล็อก นั้นทรานซิสเตอร์จะถูกใช้ขยายสัญญาณต่างๆ เช่น สัญญาณเสียง สัญญาณความถี่วิทยุ หรือควบคุมระดับแรงดัน รวมทั้งเป็นแหล่งจ่ายไฟฟ้าแบบสวิชชิงในเครื่องคอมพิวเตอร์ด้วย ทรานซิสเตอร์ก็ยังถูกใช้ในวงจรดิจิทัล เพียงแต่ใช้งานในลักษณะการเปิด/ปิดเท่านั้น วงจรดิจิทัลเหล่านั้นได้แก่ วงจรตรรกะ (Logic gate), หน่วยความจำแบบสุ่ม (Random Access Memory, RAM) และไมโครโพรเซสเซอร์ เป็นต้น

2.2.1 ประวัติความเป็นมาของทรานซิสเตอร์

ในปี ค.ศ. 1928 สิทธิบัตรใบแรกของหลักการทำงานของทรานซิสเตอร์ได้ถูกจดทะเบียนโดย Julius Edgar Lilienfeld ในประเทศเยอรมนี ต่อมาในปี ค.ศ. 1934 นักฟิสิกส์ชาวเยอรมันชื่อ Dr. Oskar Heil ได้ขึ้นทะเบียนหลักการทำงานของทรานซิสเตอร์แบบสนามไฟฟ้า และในปี 1947 นักวิจัยชื่อ John Bardeen, Walter Brattain และ William Shockley ก็ประสบความสำเร็จในการสร้างทรานซิสเตอร์ที่เบลแล็บ พร้อมทั้งส่งเข้าสู่สายการผลิตที่ เวสเทอร์นอิเล็กทรอนิกส์ ออลเดนทาวน์ รัฐเพนซิลวาเนีย เพียงสองทศวรรษต่อจากนั้น ทรานซิสเตอร์ก็ได้เข้าไปทดแทนเทคโนโลยีหลอดสุญญากาศในเครื่องใช้อิเล็กทรอนิกส์แทบทุกชนิดและก่อให้เกิดอุปกรณ์ชนิดใหม่ๆ มากมาย เช่น วงจรรวม และ เครื่องคอมพิวเตอร์ส่วนบุคคล เป็นต้น

“คำว่าทรานซิสเตอร์เกิดจากการรวมคำว่า transconductance หรือ transfer และคำว่า resistor เพราะตัวอุปกรณ์นั้นทำงานคล้ายวาไรสเตอร์ (Varistor) คือสามารถเปลี่ยนค่าความต้านทานได้ ดังนั้นการตั้งชื่อจึงสามารถบรรยายการทำงานในเบื้องต้นได้”

ในเดือนสิงหาคม ค.ศ. 1948 นักฟิสิกส์ชาวเยอรมันชื่อ Herbert F. Mataré และ Heinrich Walker ซึ่งทำงานกับบริษัทลูกของ Westinghouse Corporation ในเมืองปารีส ฝรั่งเศส ได้จดสิทธิบัตรด้วยขยาสัญญาแบบโซลิดสเตตในชื่อว่า ทรานซิสตรอน เพราะว่าทางเบลล์ไม่ได้เปิดเผยการค้นพบจนปี ค.ศ. 1948 ทรานซิสตรอนจึงถือเป็นการพัฒนาที่เกิดขึ้นโดยไม่เกี่ยวข้องกับทรานซิสเตอร์ ในขณะนั้น การค้นพบทรานซิสตรอนของ Mataré เกิดจากการสังเกตการเปลี่ยนความนำไฟฟ้าในเจมส์นิเยมไดโอดจากอุปกรณ์เรดาร์ในสงครามโลกครั้งที่สอง ทรานซิสตรอนนี้ถูกนำมาขายและใช้งานในบริษัทโทรศัพท์และทางทหารของประเทศ ฝรั่งเศส และในปี ค.ศ. 1953 เครื่องรับวิทยุต้นแบบที่ทำงานจากทรานซิสตรอน 4 ตัวถูกนำมาแสดงในงาน Dusseldorf Radio Fair.

2.2.2 ความสำคัญของทรานซิสเตอร์

ทรานซิสเตอร์ถือว่าเป็นหนึ่งในการประดิษฐ์ที่สำคัญในประวัติศาสตร์ยุคใหม่ เฉกเช่น การพิมพ์รถยนต์ และโทรศัพท์ ทรานซิสเตอร์ถือว่าเป็นอุปกรณ์แบบแอ็กทีฟหลักในอุปกรณ์อิเล็กทรอนิกส์สมัยใหม่ ความสำคัญของทรานซิสเตอร์ในทุกวันนี้เกิดจากการที่มันสามารถถูกผลิตขึ้นด้วย กระบวนการอัดโน้มติในจำนวนมากๆ (fabrication) ในราคาต่อชิ้นที่ต่ำ

แม้ว่าทรานซิสเตอร์แบบตัวเดียว (Discrete Transistor) หลายล้านตัวยังถูกใช้อยู่แต่ ทรานซิสเตอร์ส่วนใหญ่ในปัจจุบันถูก สร้างขึ้นบนไมโครชิป (Micro chip) หรือเรียกว่าวงจรรวม พร้อมกับไดโอด ตัวต้านทานและตัวเก็บประจุเพื่อประกอบกันเป็นวงจรอิเล็กทรอนิกส์ วงจรอนาล็อก ดิจิทัล หรือวงจรสัญญาณผสม (Mixed Signal) ถูกสร้างขึ้นบนชิปตัวเดียวกัน ต้นทุนการออกแบบและพัฒนาวงจรรวมที่ซับซ้อนนั้นสูงมากแต่เนื่องจากการผลิตที่ ะมาากๆ ในระดับล้านตัวทำให้ราคาต่อหน่วยของวงจรรวมนั้นต่ำ วงจรตรรกะ (Logic Gate) ประกอบด้วยทรานซิสเตอร์ประมาณ 20 ตัว ในขณะที่หน่วยประมวลผล(Microprocessor) ล่าสุดของปี ค.ศ. 2005 ประกอบด้วยทรานซิสเตอร์ราว 289 ล้านตัว เนื่องด้วยราคาที่ถูกลง ความยืดหยุ่นในและความเชื่อถือได้ในการทำงาน ทรานซิสเตอร์จึงเปรียบเหมือนอุปกรณ์ครอบจักรวาลในงานที่ไม่ใช้งานกล เช่น คอมพิวเตอร์แบบดิจิทัล เป็นต้น วงจรที่ทำงานด้วยทรานซิสเตอร์ยังได้เข้ามาทดแทนอุปกรณ์

เชิงกล-ไฟฟ้า (Electromechanical) สำหรับงานควบคุมเครื่องมือเครื่องใช้ และเครื่องจักรต่างๆ เพราะมันมีราคาถูกกว่าและการใช้วงจรรวมสำเร็จรูปรวมกับการเขียนโปรแกรม คอมพิวเตอร์นั้นมีประสิทธิภาพในการใช้งานเป็นระบบควบคุมดีกว่าการใช้อุปกรณ์เชิงกล เนื่องด้วยราคาที่ถูกลงของทรานซิสเตอร์และการใช้งานคอมพิวเตอร์แบบดิจิทัล ที่เกิดขึ้นต่อมาก่อให้เกิดแนวโน้มการสร้างข้อมูลในเชิงเลข (Digitize information) ด้วยเครื่องคอมพิวเตอร์ที่มากด้วยความสามารถในการค้นหา จัดเรียงและประมวลผลข้อมูลเชิงเลข และทำให้มีความพยายามมากมายเพื่อผลักดันให้เกิดการสร้างข้อมูลแบบดิจิทัล สื่อหลายๆ ประเภทในปัจจุบันถูกส่งผ่านรูปแบบของดิจิทัลโดยนำมาแปลงและนำเสนอในรูปแบบอนาล็อกด้วยเครื่องคอมพิวเตอร์ การปฏิวัติทางดิจิทัลเช่นนี้ส่งผลกระทบสื่อเช่น โทรทัศน์ วิทยุ และหนังสือพิมพ์

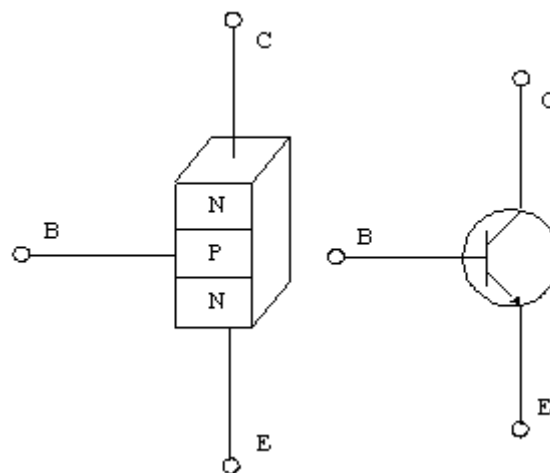
2.2.3 ชนิดและโครงสร้างของทรานซิสเตอร์ แบ่งออกได้ 2 ชนิดใหญ่ๆ คือ

1. ทรานซิสเตอร์แบบรอยต่อคู่ (Bipolar junction transistor)

ทรานซิสเตอร์แบบรอยต่อคู่ (BJT) เป็นทรานซิสเตอร์ชนิดหนึ่ง มันเป็นอุปกรณ์สามขั้วต่อถูกสร้างขึ้นโดยวัสดุสารกึ่งตัวนำที่มีการเจือสาร และอาจจะมีการใช้ในการขยายสัญญาณหรืออุปกรณ์สวิตชิง ทรานซิสเตอร์แบบรอยต่อคู่ถูกตั้งขึ้นมาตามชื่อของมันเนื่องจากช่องการนำสัญญาณหลักมีการใช้ทั้งอิเล็กตรอนและโฮลเพื่อนำกระแสไฟฟ้าหลัก โดยแบ่งออกได้อีก 2 ชนิดคือ ชนิดเอ็นพีเอ็น (NPN) และชนิดพีเอ็นพี (PNP) ตามลักษณะของการประกบสารกึ่งตัวนำ

1.1 ทรานซิสเตอร์ชนิด NPN

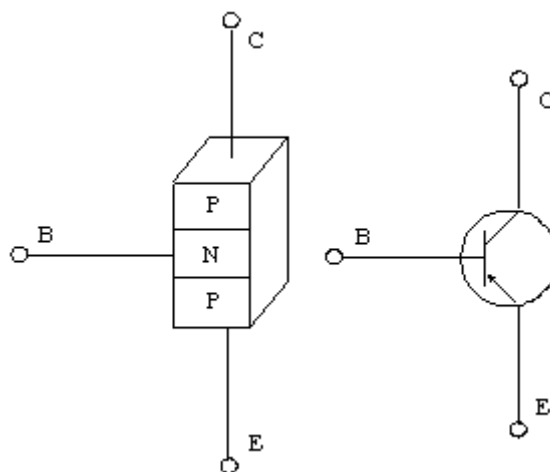
เป็นทรานซิสเตอร์ที่สร้างจากสารกึ่งตัวนำชนิด N ชนิด P และชนิด N มาต่อเรียงกันตามลำดับ แล้วต่อสายออกมา 3 สาย เพื่อเป็นขาต่อกับวงจรสารกึ่งตัวนำชนิด P ซึ่งอยู่ตรงกลางจะเป็นจุดร่วม สารกึ่งตัวนำชนิด N จะทำหน้าที่จ่ายอิเล็กตรอนซึ่งจะไหลเป็นกระแสในวงจรส่วนนี้เราเรียกว่า อิมิตเตอร์ อิเล็กตรอนจะเคลื่อนที่ผ่านสารกึ่งตัวนำชนิด P ซึ่งเราเรียกว่าเบสส่วนเบสนี้จะเป็นตัวคอยควบคุมอิเล็กตรอนให้ไหลไปยังสารกึ่งตัวนำชนิด N ถัดไปได้มากหรือน้อยอิเล็กตรอนส่วนที่ผ่านเบสมาก็จะเคลื่อนที่มายังสารกึ่งตัวนำชนิด N ซึ่งเราเรียกว่า คอลเลกเตอร์ และกลายเป็นกระแสไหลในวงจรภายนอกต่อไป



รูปที่ 2.2.1 โครงสร้างและสัญลักษณ์ของทรานซิสเตอร์ชนิด NPN

1.2 ทรานซิสเตอร์ชนิด PNP

คือทรานซิสเตอร์ที่สร้างจากสารกึ่งตัวนำชนิดพี ชนิดเอ็น และชนิดพี มาเรียงกันตามลำดับ แล้วต่อสายจากแต่ละชั้นส่วนออกมาเป็น 3 สายเพื่อต่อกับวงจรสารกึ่งตัวนำเอ็นจะเป็นจุดร่วม



รูปที่ 2.2.2 โครงสร้างและสัญลักษณ์ของทรานซิสเตอร์ชนิด PNP

2. ทรานซิสเตอร์แบบสนามไฟฟ้า (Field-effect transistor)

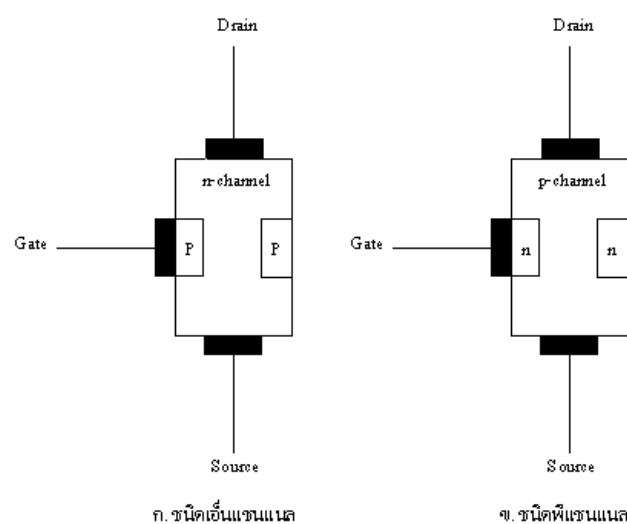
ทรานซิสเตอร์แบบสนามไฟฟ้า(FET) มีขาต่อสามขา คือ ขา เดรน(drain) เกท(gate) ซอร์ส (source) หลักการทำงานแตกต่างจากทรานซิสเตอร์แบบห้วต่อไบโพลาร์(BJT) นั่นคืออาศัย สนามไฟฟ้าในการสร้างช่องนำกระแส(channel) เพื่อให้เกิดการนำกระแสของตัวทรานซิสเตอร์ ใน

แง่ของการนำกระแส ทρανซิสเตอร์แบบสนามไฟฟ้าและแบบหัวต่อไบโพลาร์มีลักษณะของกระแสไฟฟ้าที่ไหล ผ่านอุปกรณ์ที่แตกต่างกัน นั่นคือกระแสในทรานซิสเตอร์แบบหัวต่อไบโพลาร์จะเป็นกระแสที่เกิดจากพาหะส่วน น้อย(minor carrier) แต่กระแสในทรานซิสเตอร์สนามไฟฟ้าจะเกิดจากพาหะส่วนมาก(major carrier)

ทรานซิสเตอร์แบบสนามไฟฟ้าแบ่งเป็น 2 ประเภทหลักๆ คือ

1. เจเฟต (J-FET) เป็นทรานซิสเตอร์ที่ประกอบมาจากหัวต่อ P-N การทำงานของเฟตชนิดนี้จะใช้หลักการพองตัวของสนามไฟฟ้าในรูปของดีพลีชันตรงช่วงรอยต่อ พี-เอ็น

เจเฟต (Junction Field Effect Transistor) เมื่อพิจารณาคำโครงสร้างดังรูปที่ 1 จะพบว่า เจเฟตมี 2 ชนิด คือ เจเฟตชนิดเอ็นแชนแนล ดังรูปที่ 1 ก. และชนิด พีแชนแนล ดังรูปที่ 1 ข. เจเฟต นั้นมีขาต่อใช้งาน 3 ขา คือ ขาเดรน (D) ขาเกต (G) และขาซอร์ส (S) เจเฟตชนิดเอ็นแชนแนล ขึ้นสารชนิดเอ็นจะเป็นขาเดรนและขาซอร์ส สำหรับขาเกตจะเป็นขึ้นสารชนิดพี ดังรูปที่ 1 ก. ส่วน เจเฟตชนิดพีแชนแนลนั้น ขาเดรนและขาซอร์สจะเป็นขึ้นสารชนิดพี สำหรับขาเกตจะเป็นขึ้นสารชนิดเอ็น ดังรูปที่ 1 ข.



รูปที่ 2.2.3 โครงสร้างและสัญลักษณ์ของทรานซิสเตอร์แบบเจเฟต

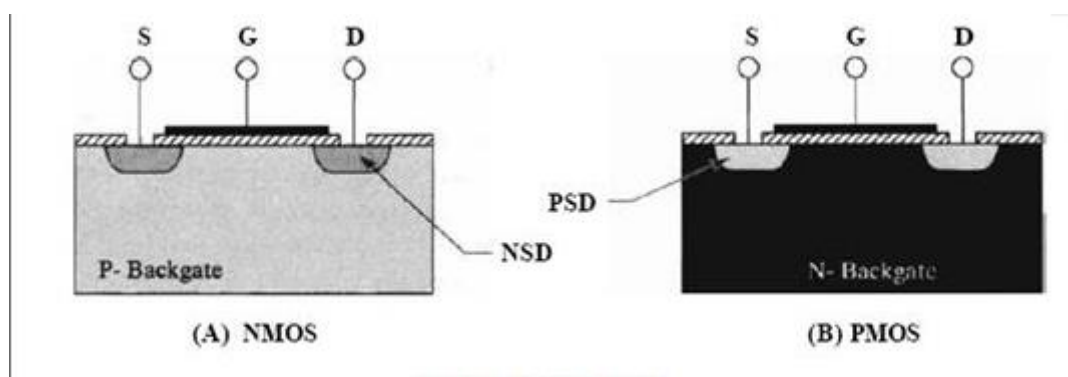
2. มอสเฟต (MOS-FET) เป็นเฟตที่มีโครงสร้างแตกต่างไปจากเจเฟต เพราะที่ขาเกตจะมีฉนวนกั้น การทำงานจะใช้การเหนี่ยวนำของสนามไฟฟ้าแทนที่จะใช้การพองตัวของดีพลีชันโดยตรง

MOSFET ประกอบด้วยสามส่วน คือ

- GATE เป็นส่วนที่ทำมาจากออกไซด์ของโลหะ โดยสร้างให้เกิดความต่างศักย์ตกคร่อมระหว่างแผ่นสองแผ่นเพื่อ สร้างสนามไฟฟ้าเพื่อควบคุมการเข้าออกของสัญญาณไฟฟ้า
- SOURCE เป็นส่วนขาเข้าของสัญญาณ
- DRAIN เป็นส่วนขาออกของสัญญาณ

มอสเฟสมี2ประเภท คือ

1. NMOS (negative MOSFET) เป็นทรานซิสเตอร์ประเภท NPN เมื่อมีความต่างศักย์เป็นบวก (สนามไฟฟ้าแรง) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้
2. PMOS (positive MOSFET) เป็นทรานซิสเตอร์ประเภท PNP เมื่อมีความต่างศักย์ต่ำหรือเป็นลบ (สนามไฟฟ้าอ่อน) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้



รูปที่ 2.2.4 โครงสร้างและสัญลักษณ์ของทรานซิสเตอร์แบบมอสเฟส

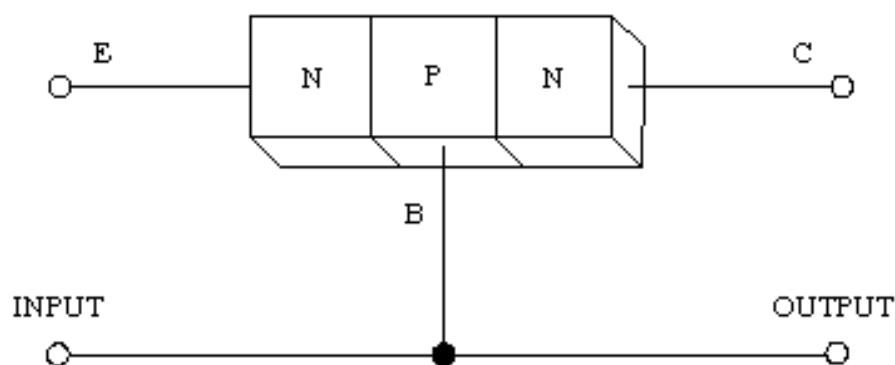
2.2.4 การทำงานของทรานซิสเตอร์

จาก การศึกษาเกี่ยวกับการไหลของกระแสภายในวงจรสารกึ่งตัวนำ การที่เราจะทำให้เกิดการไหลของกระแสหรือให้ทรานซิสเตอร์ทำงานได้นั้น จำเป็นจะต้องให้ไบอัสและกระแสที่ปรากฏทางด้านเอาต์พุตเราต้องสามารถควบคุม ค่าของกระแสได้ด้วยจึงจะทำให้ทรานซิสเตอร์ขยายสัญญาณได้ตามความต้องการ

การอธิบายการทำงานของทรานซิสเตอร์จำเป็นต้องเข้าใจการไหลในรูปของโฮลและอิเล็กตรอน รวมถึงการไบอัสด้วยซึ่งการไบอัสเป็นวิธีการที่จะทำให้ทรานซิสเตอร์พร้อมที่จะทำงานนั่นเอง ในกรณีของทรานซิสเตอร์มี 3 ขา การป้องกันแรงเคลื่อนที่เหมาะสมหรือไบอัสที่ถูกต้องจะทำให้ทรานซิสเตอร์ทำงานได้

เมื่อพิจารณาโครงสร้างของทรานซิสเตอร์แล้วจะสามารถจัดรูปแบบการขยายสัญญาณโดยต้องมีอินพุตและเอาต์พุต เมื่อให้ขาหนึ่งเป็นอินพุตขาหนึ่งเป็นเอาต์พุต ขาที่เหลือก็จะต้องเป็นจุดร่วม (Common) อินพุตกับเอาต์พุต จากหลักการดังกล่าวเรากำหนดให้ระหว่าง B กับ E เป็นอินพุต (Input) และระหว่าง B กับ C เป็นเอาต์พุต (Output) ดังนั้นจะสามารถจัดรูปแบบการขยายได้ 3 แบบ หรือ 3 คอมมอน

เนื่องจากวัตถุประสงค์ของทรานซิสเตอร์สร้างมาจากหลักการที่ต้องการให้กระแสทางด้านอินพุตไปควบคุมกระแสเอาต์พุต ดังนั้นจะต้องไบอัสทางด้านเอาต์พุตเป็นไบอัสแบบย้อนกลับ (Reverse Bias) ถ้าให้ไบอัสตรงจะทำให้ทางด้านเอาต์พุตเป็นอิสระไม่ครบวงจรเอาต์พุตทางด้านอินพุตจะให้ไบอัสตรง (Forward Bias) และแรงเคลื่อนที่มาไบอัสนี้ไม่จำเป็นจะต้องเป็นแรงเคลื่อนไฟฟ้าที่มีค่าสูงแต่อย่างไร เพราะถ้าให้กระแสนินพุตสูงเกินไปจะทำให้กระแสเอาต์พุตเกิดการอิ่มตัว



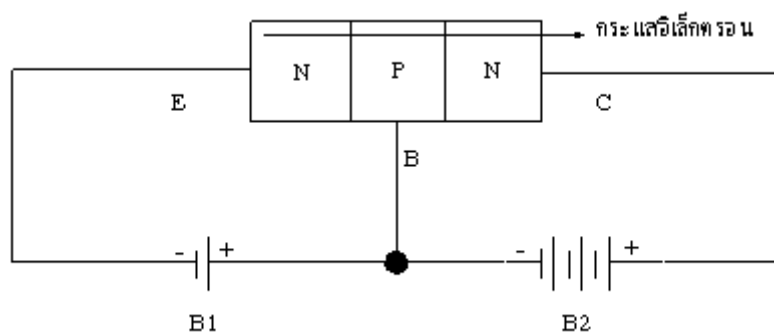
รูปที่ 2.2.5 การจัดอินพุตและเอาต์พุต

2.2.5 การให้ไบอัสทรานซิสเตอร์

ทรานซิสเตอร์ทั้งชนิด NPN และ PNP เมื่อนำไปใช้งานไม่ว่าจะใช้ในวงจรขยายสัญญาณหรือทำงานเป็นสวิตช์จะต้องทำการจัดแรงไฟให้เหมาะสมหรือเรียกว่าการให้ไบอัส (Bias) ให้ทรานซิสเตอร์ก่อน ทรานซิสเตอร์จึงจะทำงานได้โดยใช้หลักการไบอัสดังนี้

1. ไบอัสตรง (Forward Bias) ให้กับรอยต่อระหว่างอิมิตเตอร์กับเบส
2. ไบอัสกลับ (Reverse Bias) ให้กับรอยต่อระหว่างคอลเลคเตอร์กับเบส

การให้ไบอัสทรานซิสเตอร์ชนิด NPN



รูปที่ 2.2.6 การให้ไบอัสของทรานซิสเตอร์ชนิด NPN

B_1 และ B_2 เป็นแรงดันไฟฟ้าป้อนเป็นไบอัสให้ขาต่างๆ ของทรานซิสเตอร์ NPN

B_1 เป็นไบอัสตรงระหว่าง B-E ความต้านทานระหว่าง B-E จะต่ำมาก

B_2 เป็นไบอัสกลับระหว่าง B-C ความต้านทานระหว่าง B-C จะสูงมาก

ฉะนั้นกระแสอิเล็กตรอนจะไหลในวงจร B-E ครอบคลุมและอิเล็กตรอนในสารกึ่งตัวนำชนิด N จะเข้าไปรวมในสาร P ทันทีทันใด จึงเกิดแรงดันผลักดันให้อิเล็กตรอนเคลื่อนตัวไปขา C กระแสอิเล็กตรอนจะผ่าน B_1 - B_2 ครอบคลุมทางขา E

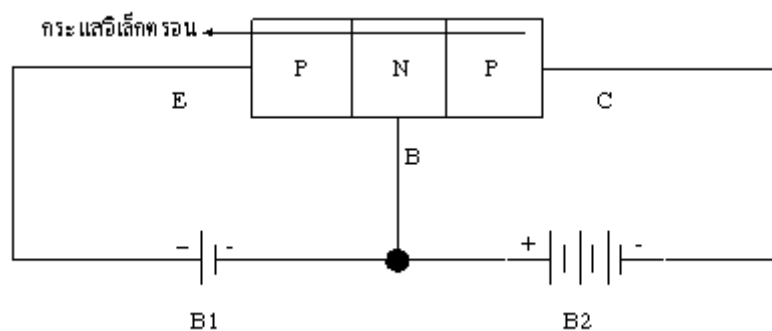
จากรูปที่ 2.2.6 สำหรับทรานซิสเตอร์ชนิด NPN ก็เช่นกัน การให้ไบอัสทางด้านอินพุตเป็นแบบฟอร์เวิร์ดไบอัสให้ไบอัสทางเอาต์พุตเป็นแบบ รีเวิร์สไบอัส ในลักษณะเช่นนี้ถ้าใช้กระแสนิยมเป็นก็จะได้ทิศทางการไหลของกระแสทางอินพุต จากขาเบสไปยังอิมิตเตอร์ อย่างไรก็ตามเรา

ได้ทราบแล้วว่าที่ขาเบสนั้นถูกโด้ไปไว้แคบมากจึงทำให้ประจุ ส่วนใหญ่ไม่สามารถที่จะไหลไปได้ จะต้องให้รีเวิร์สไบอัสระหว่างเบสกับคอลเลคเตอร์เป็นบวกมากๆ เพื่อผลักประจุเหล่านี้ให้เคลื่อนที่ไปยังอีมิเตอร์ เช่นเดียวกันถ้าดูจากทิศทางการไหลของกระแสแล้วสามารถสรุปได้ว่า

$$I_E = I_B + I_C$$

การไล่ทิศทางการไหลของกระแสอีกแบบหนึ่งซึ่งนิยมไล่จากขั้วลบไปหาขั้วบวกซึ่งเราเรียกว่า การไล่แบบกระแสอิเล็กทรอนิกส์นั้น ทิศทางของกระแสจะทวนหัวลูกศรของสัญลักษณ์ทรานซิสเตอร์ แต่ผลที่ออกมาจะเหมือนกันทุกประการทั้งนี้เพราะการไหลของกระแสไบโพลาร์ก็คือการแลกเปลี่ยนประจุกันระหว่างโฮลกับอิเล็กตรอนนั่นเอง

การให้ไบอัสทรานซิสเตอร์ชนิด PNP



รูปที่ 2.2.7 การให้ไบอัสทรานซิสเตอร์ชนิด PNP

จากวงจร

B_1 และ B_2 เป็นแรงดันไฟฟ้าป้อนเป็นไบอัสให้ขาต่างๆ ของทรานซิสเตอร์ PNP

B_1 เป็นไบอัสตรงระหว่าง B-E ความต้านทานระหว่าง B-E จะต่ำมาก

B_2 เป็นไบอัสกลับระหว่าง B-C ความต้านทานระหว่าง B-C จะสูงมาก

ฉะนั้นกระแสอิเล็กตรอนจะไหลในวงจร B ไป E สนามไฟฟ้าลบใน N จึงลดต่ำลง อิเล็กตรอนจากขา C จึงผ่าน B ไป E ผ่าน B_2 - B_1 ครบวงจร ลักษณะการทำงานในตัวของ ทรานซิสเตอร์ PNP คล้ายกับหลอดหรือโฮลทางขา E วิ่งผ่าน B ไปยังขา C รับเอาอิเล็กตรอนหรือโฟลตจาก B_1 จึงทำให้เกิดกระแสอิเล็กตรอนไหลครบวงจร

จากรูปที่ 2.2.7 เมื่อให้เบสกับอิมิตเตอร์ได้รับไบอัสแบบฟอร์เวิร์ด ทำให้มีกระแสไหลจากอิมิตเตอร์ไปยังเบส (ตามทิศทางหัวลูกศร) ซึ่งเราเรียกว่า กระแสเบส ย่อด้วย I_B กระแสที่ไหลมีค่าประมาณ 2-5 % ของค่ากระแสทั้งหมด เนื่องจากที่ขาเบสนั้นสารที่ไดโปกมีพื้นที่น้อยมากจึงทำให้ประจุจำนวนมากของโฮลมารออยู่ที่ขาเบส ถ้าให้ไบอัสระหว่างเบสกับคอลเลคเตอร์แบบรีเวิร์สมากๆ จะทำให้มีกระแสไหลจากคอลเลคเตอร์ไปยังอิมิตเตอร์ได้ เรียกว่า กระแสคอลเลคเตอร์ ย่อด้วย I_C กระแสที่ไหลจะมีค่าประมาณ 95-98 % ของกระแสทั้งหมดซึ่งสามารถสรุปได้ว่า

$$I_E = I_C + I_B$$

I_E คือ กระแสอิมิตเตอร์มีค่าเท่ากับ 100 %

I_C คือ กระแสคอลเลคเตอร์ มีค่าเท่ากับ 95-98 %

I_B คือ กระแสเบส มีค่าเท่ากับ 2-5 %

V_{BE} คือ แรงไฟไบอัสหรือศักย์ตกคร่อมระหว่างเบสกับอิมิตเตอร์ในขณะที่ทรานซิสเตอร์ทำงานปกติ

Bias ของเยอรมันเนียมทรานซิสเตอร์มีค่า 0.2-0.3 โวลต์

Bias ของซิลิคอนทรานซิสเตอร์ มีค่า 0.6-0.7 โวลต์

2.3 มอสเฟต

มอสเฟต เป็นทรานซิสเตอร์ ที่ใช้อิทธิพลสนามไฟฟ้าในการควบคุมสัญญาณไฟฟ้า โดยใช้ ออกไซด์ของโลหะในการทำส่วน GATE นิยมใช้ในวงจรดิจิทัล โดยนำไปสร้างลอจิกเกตต่างๆ เพราะมีขนาดเล็ก

2.3.1 โครงสร้างของ MOSFET

MOSFET ประกอบด้วยสามส่วน คือ

- GATE เป็นส่วนที่ทำมาจากออกไซด์ของโลหะ โดยสร้างให้เกิดความต่างศักย์ตกคร่อมระหว่างแผ่นสองแผ่นเพื่อ สร้างสนามไฟฟ้าเพื่อควบคุมการเข้าออกของสัญญาณไฟฟ้า
- SOURCE เป็นส่วนขาเข้าของสัญญาณ

- DRAIN เป็นส่วนขาออกของสัญญาณ

2.3.2 ประเภทของ MOSFET

- nMOS (negative MOSFET) เป็นทรานซิสเตอร์ประเภท NPN เมื่อมีความต่างศักย์เป็นบวก (สนามไฟฟ้าแรง) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้
- pMOS (positive MOSFET) เป็นทรานซิสเตอร์ประเภท PNP เมื่อมีความต่างศักย์ต่ำหรือเป็นลบ (สนามไฟฟ้าอ่อน) สัญญาณไฟฟ้าจึงจะไหลจาก source ไป drain ได้

2.3.3 สัญลักษณ์แทน MOSFET

สัญลักษณ์ในทางดิจิทัล

MOSFET ในทางดิจิทัลถูก มองว่าเป็นสวิตช์ โดย nMOS จะเป็นสวิตช์ที่เมื่อสัญญาณเข้าเป็น "1" สวิตช์ก็จะปิด ถ้าไม่สวิตช์ก็ยังคงเปิดอยู่ (normal opened switch) ส่วน pMOS จะเป็นสวิตช์ที่เมื่อสัญญาณเข้าเป็น "1" สวิตช์ก็จะเปิด ถ้าไม่สวิตช์ก็จะปิดอยู่ (normal closed switch) และสัญลักษณ์ทั่วไปจะมีสามขา ขากลางเป็น gate ส่วนอีกสองขาชื่อ source และ drain โดยใช้ใน nMOS เป็นหลักเพื่อสื่อสัญลักษณ์เดียวกับทรานซิสเตอร์ทั่วไปคือ ไฟขา base ไหล ขา Collector จะต่อกับ Emitter ส่วน pMOS ก็จะใส่ bubble ที่ขา gate

2.3.4 การทำงานของ MOSFET

- nMOS เมื่อปล่อยความต่างศักย์สูง จะเกิดสนามไฟฟ้าใน ทิศลงอย่างแรง โฮลใน p-type จะถูกผลักลงมาอยู่ด้านล่าง (ตามรูปที่ประกอบข้างบน) ประกอบกับมีอิเล็กตรอนอิสระบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีอิเล็กตรอนอิสระมากจนเป็น n-type ได้เรียกว่า channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง channel นี้ซึ่งเป็น n-type เหมือนกับ drain และ source ได้โดยใช้อิเล็กตรอนอิสระเป็นพาหะ
- pMOS จะทำงานกลับกับ nMOS โดยเมื่อปล่อยความต่างศักย์ต่ำ (โดยมากมักจะติดลบ) จะเกิดสนามไฟฟ้าใน ทิศขึ้นอย่างแรง อิเล็กตรอนอิสระใน n-type จะถูกผลักลงมาอยู่ด้านล่าง ประกอบกับมีโฮลบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีโฮลมากจนเป็น

p-type ได้เรียกว่า channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง channel นี้ซึ่งเป็น p-type เหมือนกับ drain และ source ได้โดยใช้โฮลเป็นพาหะ

2.4 Labview

LabVIEW เป็นโปรแกรมคอมพิวเตอร์ที่สร้างเพื่อนำมาใช้ในด้านการวัดและเครื่องมือวัด สำหรับงานทางวิศวกรรม LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่งหมายความว่า เป็นโปรแกรมที่สร้าง เครื่องมือวัดเสมือนจริงในห้องปฏิบัติการทาง วิศวกรรม ดังนั้นจุดประสงค์หลักของการทำงานของโปรแกรมนี้ก็คือการจัดการในด้านการวัดและ เครื่องมือวัด อย่างมีประสิทธิภาพ และในตัวของโปรแกรมจะประกอบไปด้วยฟังก์ชันที่ช่วยย่น การวัดมากมายและแน่นอนที่สุด โปรแกรมนี้จะมีประโยชน์อย่างสูงเมื่อใช้ร่วมกับเครื่องมือวัดทาง วิศวกรรมต่างๆ

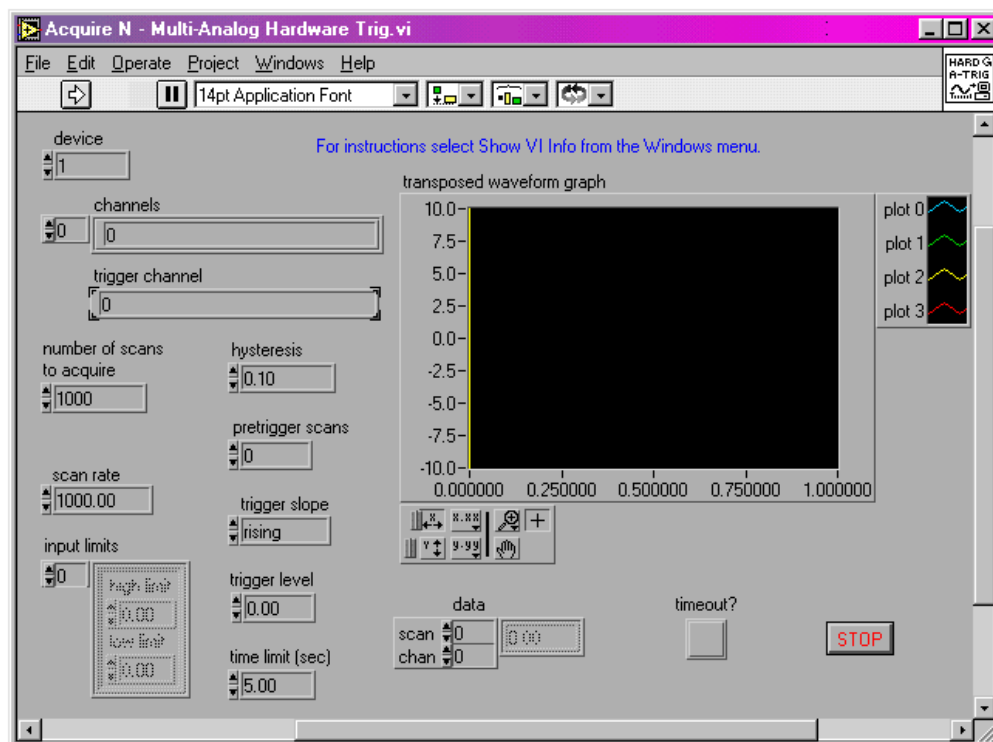
สิ่งที่ LabVIEW แตกต่างจากโปรแกรมอื่นอย่างเห็นได้ชัดที่สุดก็คือ LabVIEW นี้เป็น โปรแกรมประเภท GUI (Graphic User Interface) โดยสมบูรณ์ นั่นคือเราไม่จำเป็นต้องเขียน code หรือคำสั่งใดๆ ทั้งสิ้น และที่สำคัญลักษณะภาษาที่ใช้ในโปรแกรมนี้เราจะเรียกว่าเป็น ภาษารูปภาพ หรือเรียกอีกอย่างว่าภาษา G (Graphical Language) ซึ่งจะแทนการเขียนโปรแกรมเป็นบรรทัดอย่าง ที่เรารู้เคยกับภาษาพื้นฐาน เช่น C, BASIC หรือ FORTRAN ด้วยรูปภาพหรือสัญลักษณ์ทั้งหมด ซึ่งแม้ว่าในเบื้องต้นเราอาจจะสับสนกับการจัดเรียงหรือเขียนโปรแกรมบ้าง แต่เมื่อเรารู้เคยกับ การใช้โปรแกรมนี้แล้วเราจะพบว่า LabVIEW นี้มีความสะดวกและสามารถลดเวลาในการเขียน โปรแกรมลงไปได้มาก โดยเฉพาะในงานเขียนโปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อกับอุปกรณ์อื่นๆ เพื่อใช้ในการวัดและการควบคุม

2.4.1 ส่วนประกอบของ LabVIEW

1. Front Panel

Front Panel หรือ หน้าปัทม์คือส่วนที่ผู้ใช้จะใช้ติดต่อกับโปรแกรม ในขณะที่เราให้ VI ทำงานอยู่นั้นหน้าปัทม์นี้จะต้องทำงานร่วมอยู่ด้วย เพื่อให้ผู้ใช้หรือผู้ควบคุมสามารถให้ข้อมูลเข้าสู่

โปรแกรม และเมื่อข้อมูลได้รับการประมวลผลแล้วก็จะแสดงออกมาทาง Front Panel นี้ ดังนั้นหากจะเปรียบกับโปรแกรมสำเร็จรูปอื่นๆ และ Front Panel นี้ก็คือ Graphic User Interface (GUI) ของ LabVIEW นั่นเอง ตัวอย่างของลักษณะของ Front Panel ใน LabVIEW เป็นไปตามรูป ซึ่งในขั้นแรกนี้ผู้ที่ยังไม่มีความคุ้นเคยกับโปรแกรมนี้อาจมองดูว่าการสร้างองค์ประกอบต่างๆ คงจะมีความยุ่งยาก แต่ถ้าหากเราเริ่มทำความเข้าใจกับ LabVIEW แล้วเราจะพบว่า การเขียน Front Panel ในลักษณะในรูปนี้ไม่ใช่สิ่งที่ยุ่งยากหรือสิ้นเปลืองเวลาในการเขียนเลย



รูปที่ 2.4.1 Front Panel

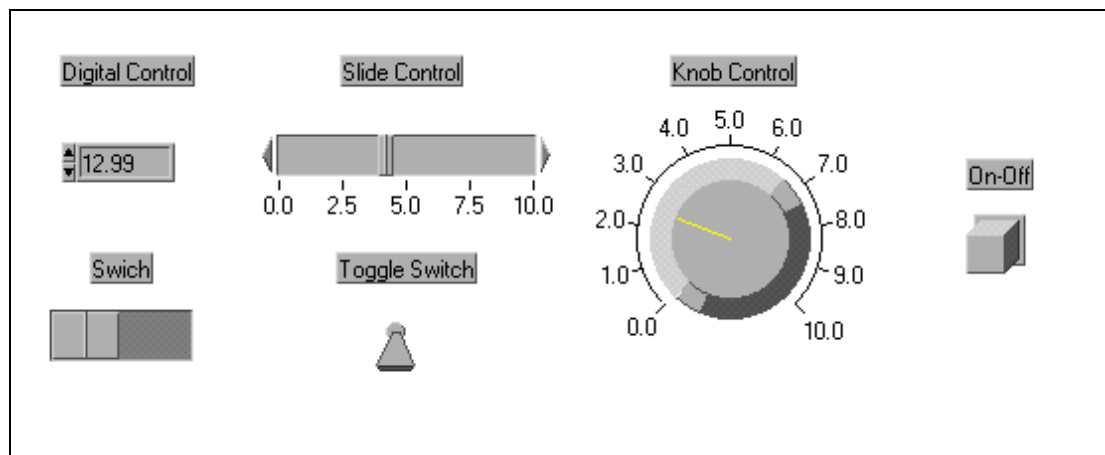
หากเราสังเกตจากรูปนี้เราจะพบว่าบนหน้าปัทม์ Front Panel ของ LabVIEW จะมีส่วนประกอบที่สำคัญ 2 แบบ คือ ตัวควบคุม (Controlled) และ ตัวแสดงผล (Indicator) ซึ่งส่วนประกอบทั้ง 2 จะมีการทำงานต่างกันและหน้าที่ตรงกันข้ามกัน ดังมีรายละเอียดต่อไปนี้

2. Controls

Controls มีหน้าที่เป็นตัวควบคุมคือให้ค่าหรือ Input จากผู้ใช้ ลักษณะของ Controls เช่น ปุ่มปรับค่า, สะพานปิด - เปิดไฟ, แท่งเลื่อนเพื่อปรับค่า, การให้ค่าด้วยตัวเลข Digital หรืออื่นๆ ดังนั้นจากหลักการของ Controls ก็หมายความว่า เป็นการกำหนดค่าหรือแหล่ง (source) ของข้อมูลโดยปกติเรา

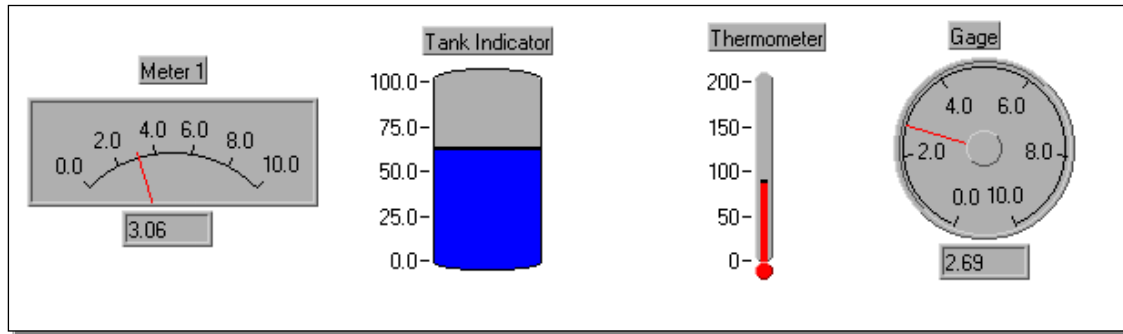
จะไม่สามารถนำข้อมูลมาแสดงผลที่ Controls ได้ และถ้าหากเราพยายามที่จะให้ Control แสดงผล ข้อมูลก็จะเกิดความผิดพลาดขึ้นใน VI ของเราทันที

ตัวอย่างของ Object ที่ปกติแล้วจะทำหน้าที่เป็น Controls บน Front Panel บางประเภท จะแสดงในรูปต่อไปนี้ เราจะสังเกตเห็นว่าหากเปรียบเทียบกับในอุปกรณ์เครื่องมือวัดจริงแล้ว อุปกรณ์เหล่านี้จะได้รับการกำหนดค่าจากผู้ใช้งาน ดังนั้นจะเห็นว่า LabVIEW พยายามทำให้เรารู้สึกว่าใช้งานกับเครื่องมือจริงๆอยู่



รูปที่ 2.4.2 Controls

3. Indicators มีหน้าที่เป็นตัวแสดงผลเพียงอย่างเดียวโดยจะรับค่าที่ได้จากแหล่งข้อมูลมาแสดงผลซึ่งอาจปรากฏในรูปของกราฟ, เข็มชี้, ระดับของเหลว หรืออื่นๆ Indicator นี้เปรียบเสมือน output เพื่อให้ผู้ใช้งานได้ทราบค่าสิ่งที่เรากำลังวิเคราะห์อยู่ และผู้ใช้งานไม่สามารถปรับค่าบน indicator ได้โดยตรงแต่จะต้องมีแหล่งข้อมูลที่ส่งให้กับ Indicators เหล่านี้ ดังนั้นเราอาจมอง Indicator ว่าเป็นเหมือน Sink ของข้อมูล ตัวอย่างของ object ที่ปกติแล้วจะมีเป็น Indicator บางชนิดได้แสดงในรูปต่อไปนี้

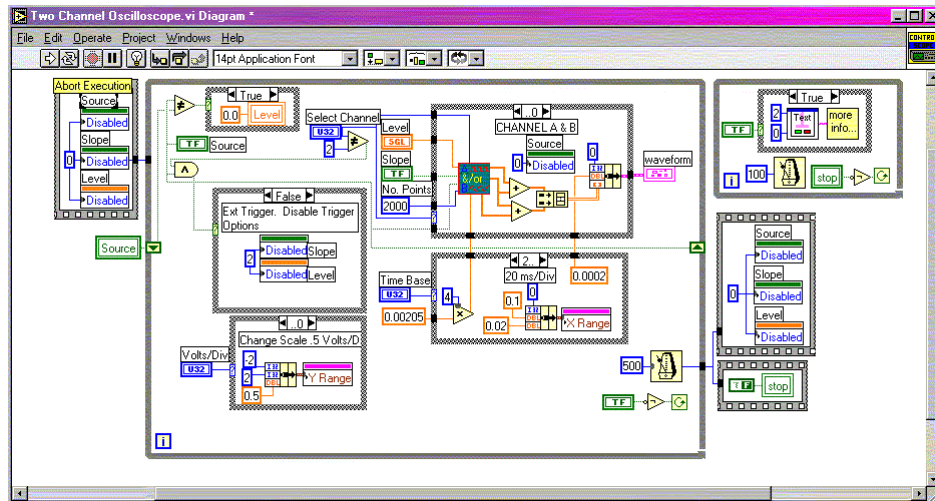


รูปที่ 2.4.3 Indicators

ในการเขียน VI อันดับแรกคือการเขียนหน้าปัทม์ซึ่งผู้ใช้จะต้องออกแบบส่วนนี้จะจัดวางให้เหมาะสม ซึ่งจะกล่าวถึงในบทต่อไป สำหรับในขั้นนี้เราเพียงแต่เน้นว่าบนหน้าปัทม์ Front Panel จะประกอบด้วยสองส่วนและการที่เราจะเลือก Controls และ Indicators เป็นเรื่องสำคัญเพราะทั้งสองนี้ไม่สามารถแทนกันได้นั่นคือเราไม่สามารถกำหนดค่าให้ Controls แสดงค่าได้และไม่สามารถนำค่าจาก Indicators ออกไปเป็นข้อมูลของระบบได้

4. Block Diagrams

ที่ผ่านมาเราจะพบว่าเราสามารถสร้าง Front Panel ได้ให้เป็นไปตามต้องการของเรา ซึ่งไม่ใช่สิ่งที่ยากมากนักสำหรับ LabVIEW แต่สิ่งที่จะยุ่งยากมากกว่าคือการกำหนดให้สิ่งต่างๆ หรือที่เราเรียกว่า Object นั้นให้มีขั้นตอนหรือมีกระบวนการของการวิเคราะห์ต่างๆ ตามที่เราต้องการ เพราะเราจะต้องกำหนดการทำงานที่เกิดขึ้นหลังจาก Front Panel เหล่านั้น นั่นคือหลังจากการที่เราออกแบบ GUI เรียบร้อยแล้วขั้นต่อไปก็คือการกำหนดการทำงานของ GUI เหล่านั้นนั่นเอง และส่วนที่มีหน้าที่นั้นคือ Block Diagram ในเบื้องต้นนี้เราอาจมอง Block Diagram ว่าเป็น Data Flow Chart และตัวโปรแกรมหรือ code ของ LabVIEW ก็ได้ การเขียน Block Diagram ก็คือการเขียน code ในภาษา G นั่นเอง ซึ่งก็เหมือนกับการเขียน code ในภาษา C หรือ FORTRAN นั่นเอง ความแตกต่างที่สำคัญระหว่าง code ในภาษา C หรือ FORTRAN กับ Block Diagram ใน LabVIEW ก็คือ Block Diagram นั้นพร้อมที่จะ execute หรือทำการประมวลผลตลอดเวลา นั่นคือในระหว่างที่เราสร้าง Block Diagram อยู่ LabVIEW จะตรวจสอบการทำงานของ VI อยู่ตลอดเวลา ตัวอย่างของ Block Diagram เป็นไปตามรูป



รูปที่ 2.4.4 Block Diagram

ถ้าหากเราพิจารณาองค์ประกอบใน Block Diagram เราจะพบว่าในส่วนของ Block Diagram จะมีส่วนประกอบที่สำคัญ 3 ส่วน คือ

- ▶▶ Terminal
- ▶▶ Node
- ▶▶ Wire

ทั้งสามส่วนจะมีหน้าที่หลักคือการควบคุมการส่งผ่านหรือเราอาจเรียกว่าการไหลของข้อมูล (Data Flow) และกำหนดถึงวิธีการประมวลผลข้อมูล

5. Terminal

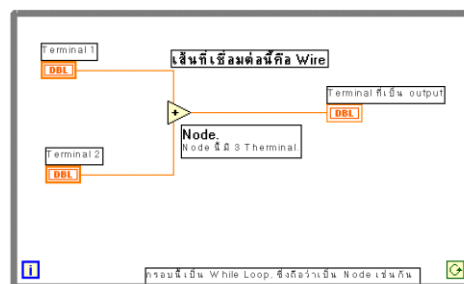
ทุกครั้งที่เราสร้าง Control หรือ Indicator บน Front Panel ใน Window ของ Block Diagram จะปรากฏ Terminal ขึ้น Terminal ก็คล้ายกับสถานีของข้อมูลคือจะเป็นทั้งสถานีต้นทางของข้อมูลถ้า Terminal นั้นเป็น Terminal ของ Controls และขณะเดียวกันจะเป็นทั้งสถานีปลายทางของข้อมูลถ้า Terminal นั้นเป็น Terminal ของ Indicator

ข้อที่ควรเข้าใจอย่างหนึ่งก็คือ Object นี้เกิดขึ้นจากการเขียนขึ้นบน Front Panel ดังนั้นเมื่อเราจะไม่สามารถลบ Terminal ออกจาก Block Diagram ได้ และถ้าหากเราจะลบ Control หรือ Indicator นั้นออกไปจาก Front Panel แล้ว Terminal เหล่านี้ก็จะหายไปจาก Block Diagram เช่นกัน สำหรับลักษณะของตัวแปรต่างๆเราจะกล่าวในบทต่อไป

6. Node

Node เป็นคำที่ใช้เรียก object ที่ทำกรรมวิธีใดๆ เพื่อประมวลข้อมูลใน Block Diagram เช่นเดียวกับที่เราเขียน Flow Chart แล้วใช้สัญลักษณ์ต่างๆ แทนวิธีการวิเคราะห์ข้อมูล เมื่อมีข้อมูลเข้าสู่ node สิ่งที่เกิดขึ้นภายใน node ก็จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไบนั้น จะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร หาค่าเฉลี่ย หรือเป็นประเภทการเปรียบเทียบข้อมูล ว่ามากหรือน้อยกว่า หรืออื่นๆ ซึ่งจะเป็นการประมวลผลทางคณิตศาสตร์ทั่วไป นอกเหนือจากนั้น node นี้จะมีส่วนที่เรียกว่า Function แบบต่างๆ ซึ่งจะเหมือนกับ Function สำเร็จรูปเช่น sine, cosine, log เป็นต้น ซึ่งก็จะเหมือนกับในภาษาที่เป็นตัวอักษรทุกๆ ไป

รูปต่อไปนี้จะแสดงถึงลักษณะของ Node และ Terminal ที่บรรจุอยู่ภายใน Block Diagram ของ LabVIEW



รูปที่ 2.4.5 Node and Terminal ที่บรรจุอยู่ภายใน Block Diagram

นอกเหนือจากการประมวลผลทางคณิตศาสตร์แล้ว เรายังมี node ประเภท Structure หรือ Control Flow อีกด้วย (ในภาษาตัวอักษร Structure Command จะเป็น คำสั่งจำพวก IF...THEN, FOR..., WHILE... เป็นต้น)

7. Wires

ขณะที่เรามีที่มาของข้อมูล ส่วนประมวลหรือปรับแต่งข้อมูล และส่วนแสดงผลข้อมูล เรียบร้อยแล้ว ขั้นตอนต่อไปก็จะต้องสามารถควบคุมการส่งผ่านข้อมูลให้เป็นไปตามที่เราต้องการ อุปกรณ์ที่ใช้ใน LabVIEW ก็คือ การต่อสาย หรือ Wires ซึ่งจะเป็นการเชื่อมการส่งข้อมูลระหว่าง terminal หรือ node ต่างๆที่มีใน Block Diagram นี้เข้าด้วยกัน โดย wires นี้จะเป็นการกำหนด

เส้นทางของข้อมูลว่าเมื่อออกจาก terminal หนึ่งแล้ว จะกำหนดการไหลไปที่ node ใดบ้าง มีลำดับเป็นอย่างไร และสุดท้ายจะให้แสดงผลที่ terminal ใดนั่นเอง ซึ่งการเชื่อมต่อสายนี้จะทำให้เราเข้าใจถึงหลักการของ Data Flow Programming ได้ดีขึ้น

เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษร หรือค่าจริง-เท็จ (Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบ LabVIEW จึงได้กำหนดให้ลักษณะของ wires สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน นอกจากนั้นข้อมูลแต่ละแบบดังกล่าวยังอาจมีลักษณะเป็น scalars, 1-D array, 2-D array ได้ซึ่งลักษณะของเส้นของข้อมูลแต่ละแบบก็จะแตกต่างออกไปอีก

ตารางต่อไปนี้แสดงถึงลักษณะของเส้น wire เมื่อใช้เชื่อมต่อระหว่างข้อมูลแต่ละชนิด

ตารางที่ 2.4.1 แสดงลักษณะเส้น Wire ของข้อมูลแต่ละชนิด

	Scalars	1-D Array	2-D Array	สี
เลขทศนิยม				ส้ม
เลขจำนวนเต็ม				น้ำเงิน
Boolean				เขียว
ตัวอักษร				ชมพู

โดยปกติการต่อข้อมูลระหว่าง Node หรือ Terminal จะกำหนดแน่นอนว่าต้องการข้อมูลลักษณะใด เช่นถ้า Node ต้องการตัวเลข เราจำเป็นต้องต่อสายตัวเลขเข้ากับ Node นั้น ถ้าหากเราต่อสายจากตัวอักษรเข้าใน terminal ที่ต้องการตัวเลข สายต่อนั้นจะกลายเป็นสายต่อเสีย หรือ Bad Wire และโปรแกรมก็จะไม่สามารถทำงานต่อไปได้

เราจะเข้าใจหลักการของ Data Flow Programming ได้ในขณะที่เราต่อสายระหว่าง terminal และ Node ต่างๆ เข้าด้วยกัน ซึ่งหลักการทำงานของ Data Flow Programming จะต่างจากการเขียนโปรแกรมโดยใช้ตัวอักษร เพราะในโปรแกรมตัวอักษรคำสั่งจะถูกส่งเข้าสู่ส่วนประมวลผลทีละบรรทัด เพื่อการคำนวณตามลำดับบรรทัด แต่ใน Data Flow นั้น โปรแกรมจะคำนวณเมื่อมีข้อมูลส่งเข้ามาถึง Input ของ Node นั้นๆ ครบ และเมื่อข้อมูลส่งครบเข้าถึง Node นั้นแล้ว จะมีการประมวลผลและส่งค่าที่ได้ออกไปตามการต่อเชื่อมสาย การคำนวณจะเสร็จสิ้นในแต่ละรอบเมื่อข้อมูลส่งเข้ามาไปถึง Terminal สุดท้าย การประมวลผลไม่ได้เป็นไปตามลำดับการจัดวางคือไม่ได้ทำจากซ้ายไปขวาหรือบนลงล่าง แต่เป็นไปตามขั้นตอนการเดินทางของข้อมูลซึ่งในการเขียนโปรแกรมในเบื้องต้นผู้ที่คุ้นเคยกับภาษาตัวหนังสือโดยทั่วไปอาจจะต้องใช้เวลาสักครู่เพื่อที่จะทำความเข้าใจการทำงานของ LabVIEW

8. Icon และ Connector

ถ้าเรารู้คุ้นเคยกับการเขียนโปรแกรมเป็นตัวอักษร เราคงทราบว่าเราสามารถเขียนโปรแกรมย่อยขึ้นมาเพื่อใช้ร่วมกับโปรแกรมหลัก โดยการเขียนโปรแกรมย่อยหรือ Subroutine นี้จะมีประโยชน์มากในกรณีที่จะต้องทำการประมวลผลบ่อย ในภาษารูปภาพก็เช่นกันเราสามารถที่จะสร้าง Subroutine ขึ้นมาได้ ซึ่งเราจะเรียกว่า subVI โดย สำหรับข้อดีของการเขียนด้วยภาษารูปภาพนี้ก็คือ ทุก VI ที่เราเขียนขึ้นมาสามารถทำหน้าที่เป็น subVI ได้ แต่เนื่องจากเราจำเป็นต้องกำหนดลักษณะของ subVI ให้เป็นรูปภาพ และมีช่องที่จะต้องส่งเข้าสู่ subVI นั้น เราจึงได้กำหนด Icon และ Connector ขึ้น

หากเราจะกล่าวคร่าวๆ icon นี้ก็อาจจะมองว่าเป็น node ในอีกรูปแบบหนึ่งก็ได้ โดย Icon จะหมายถึง Node ของ subVI ในทุกครั้งที่เราเขียน VI เราจะพบว่า LabVIEW จะให้ VI นั้นสามารถทำงานเป็น subVI ได้ถ้าหากเราต้องการ โดยที่ LabVIEW จะกำหนด icon ให้กับทุก VI ที่เขียนขึ้น ซึ่งเราสามารถเปลี่ยนแปลงรูปแบบของ Icon ที่ LabVIEW เขียนขึ้นได้

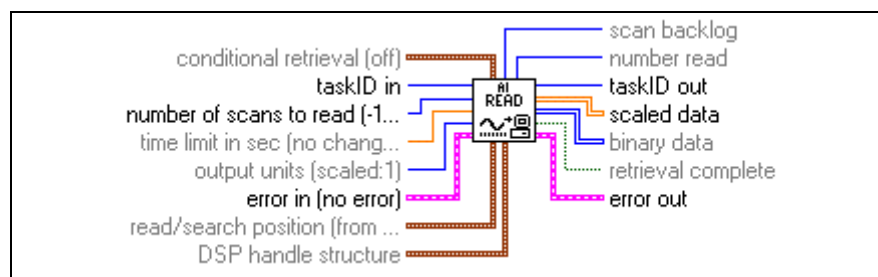


หากเราเปิด LabVIEW ใหม่ขึ้นมาเราจะสามารถสังเกตรูปของ Icon ที่ LabVIEW กำหนดขึ้นมาได้ โดยรูปของ icon จะปรากฏอยู่ที่มุมบนด้านซ้ายได้ Title Bar ของหน้าต่างต่าง Block Diagram และ Front Panel

เมื่อเรานำ VI นี้ ไปใช้เพื่อเป็น subVI เราสามารถที่จะกำหนดให้มี Input และ Output ที่จะเข้าและออกจาก VI ของเราได้ ซึ่งการเข้าและออกของข้อมูลสู่ subVI นั้นจะผ่านทาง Connector ซึ่งหากจะเปรียบเทียบกับภาษาตัวอักษรที่เราคุ้นเคยแล้ว การใช้ Connector ก็เหมือนกับการใช้คำสั่ง parameter ในภาษา C หรือการใช้คำสั่ง Function ในภาษา FORTRAN นั่นคือการกำหนดว่าข้อมูลใดเป็นข้อมูลที่ส่งไปสู่ Subroutine และข้อมูลใดเป็นข้อมูลที่จะรับกลับออกมาจาก Subroutine ทำนองเดียวกันกับใน LabVIEW คือเราจะส่งข้อมูลเข้าสู่ Icon หรือ subVI โดยผ่านทาง Input connector เมื่อข้อมูลได้รับการประมวลใน subVI แล้วก็ส่งกลับมาจาก Output Connector Terminal

โดยปกติแล้ว Connector จะถูกบังอยู่ด้านหลังของรูป Icon เราสามารถแสดงให้เห็น Connector ได้โดยการใช้คำสั่ง Show Connector ซึ่งรายละเอียดเหล่านี้จะกล่าวในภายหลัง

ในรูปต่อไปนี้เป็นารแสดง Icon และ Connector ของ VI หนึ่งซึ่งเป็นส่วนที่มาพร้อมกับ LabVIEW ส่วนที่เป็นรูปตรงกลางเราเรียก Icon และส่วนที่เป็นสายต่อต่างๆ เราเรียก Connector



รูปที่ 2.4.6 การแสดง Icon และ Connector ของ VI

สำหรับรายละเอียดเกี่ยวกับการสร้าง Icon และการกำหนด Connector นี้จะกล่าวถึงในบทต่อไป หลังจากที่เราได้ทำความคุ้นเคยกับการเขียนโปรแกรมด้วย LabVIEW แล้ว

9. Data Operation

เมนูนี้เป็นเมนูที่จะปรากฏขึ้นเฉพาะขณะที่ VI อยู่ในโหมดการทำงานเท่านั้น ซึ่งจะใช้กับ Control และ Indicator โดยจะมีเมนูย่อยต่อไปนี้

- ▶▶ Reinitialize to Default จะเป็นการกำหนดค่าของ Object ให้กลับไปหาค่าเริ่มต้น (Default) ที่ LabVIEW กำหนดมาให้ครั้งแรก

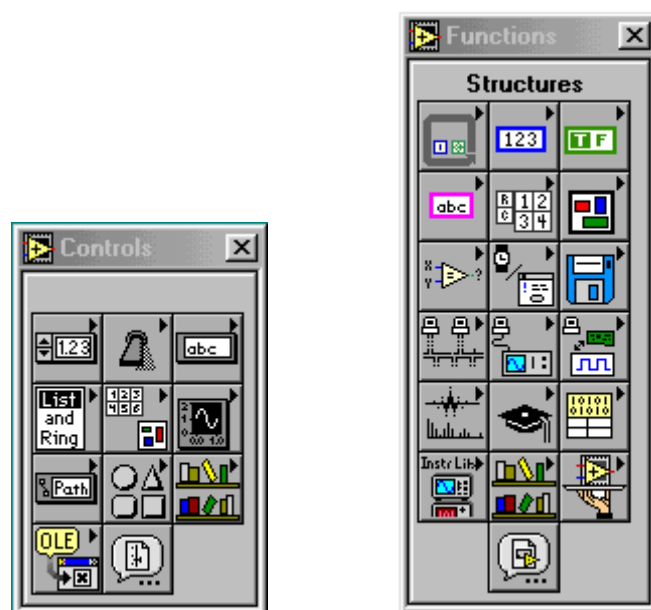
- ▶▶ Make Current Value Default จะกำหนดค่าในขณะนั้นให้เป็นค่าเริ่มต้นเมื่อเริ่มให้ VI ทำงาน
- ▶▶ Cut Data, Copy Data และ Paste Data เป็นการตัด, คัดลอก หรือใส่ข้อมูล ออกจากหรือลงใน Control หรือ Indicator

10. Controls Palettes

จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อ Front Panel อยู่ในสภาพพร้อมทำงานอยู่เท่านั้น ถ้าหาก Front Panel ไม่อยู่ในสภาพพร้อมทำงานส่วนของ Controls Palettes จะหายไป หรือเราอาจปิดเฉพาะ Controls palettes โดยใช้เมาส์กดที่ Exit Button บน Title bar ก็ได้

11. Function Palettes

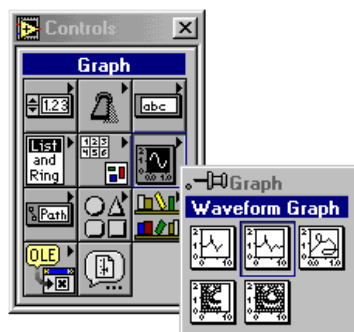
จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อ Block Diagram อยู่ในสภาพพร้อมทำงานอยู่เท่านั้นถ้าหาก Block Diagram ไม่อยู่ในสภาพพร้อมทำงาน ส่วนของ Tools Palettes นี้จะหายไป หรือเราปิด Palette นี้เอง



รูปที่ 2.4.7 ลักษณะของ Controls และ Functions Palette

Palettes ทั้งสองจะมี Subpalette บรรจุอยู่ถ้าหากว่าเราเลื่อนลูกศรของเมาส์ไปที่ปุ่มของ Subpalette ในชื่อของ Subpalette นั้นจะปรากฏขึ้น และหากเรากดลงไปทีปุ่มของ Subpalette ได้

และกดค้างไว้เราจะพบว่าจะมีการแสดงส่วนประกอบของ Subpalette นั้นต่อออกไปอีก และในหลายกรณีใน Subpalette ก็จะถูกแยกย่อยเป็น Subpalette ต่อไปอีกดังที่แสดงต่อไปในรูปแบบ



รูปที่ 2.4.8 Palettes ก็จะถูกแยกย่อยเป็น Subpalette

โดยปกติเราจะแสดง Subpalette ได้ปรากฏขึ้นได้ก็ต่อเมื่อเรากดเมาส์ปุ่มซ้ายค้างไว้เท่านั้น อย่างไรก็ตามบนทุก subpalette ที่มุมบนด้านขวาจะมีรูปที่ติดกระดากตามด้วยชื่อของ Subpalette นั้น ที่ติดกระดากหรือ Thumbtack นี้เป็นตัวช่วยให้เราสามารถแสดง subpalette นั้นตลอดไปโดยไม่ต้องเลือกจาก palette หลัก ซึ่งจะมีประโยชน์ในกรณีที่เรากำลังใช้ subpalette นั้นบ่อยๆ วิธีการใช้ก็คือให้ click ที่ตำแหน่งที่ติดกระดากนั้น Subpalette ก็จะติดอยู่บนจอต่อไปแม้ว่าเราจะเลื่อนเมาส์ออกไปแล้วก็ตาม และเราสามารถปิด subpalette นั้นได้เมื่อเราต้องการเหมือนกับการปิดหน้าต่างอื่นๆ นอกเหนือจากนั้นเราสามารถที่จะตกแต่งและเปลี่ยนแปลงลักษณะและสิ่งที่บรรจุอยู่ใน Palette หรือ Subpalette ได้ตามที่เรต้องการโดยใช้คำสั่ง Edit Control and Function Palettes จาก Edit menu ซึ่งจะกล่าวต่อไปในภายหลัง

12. Tools Palette

Tools คือหน้าที่พิเศษของตัวชี้ของเมาส์ เราจะใช้ Tools เพื่อจะให้เราทำงานในการแก้ไขหรือปฏิกิริยาหน้าที่ซึ่งเราต้องการ คล้ายกันกับที่เราใช้ในโปรแกรมการวาดรูปต่างๆ ไป ใน Tools Palette จะประกอบด้วยปุ่มที่มีหน้าที่ต่างๆ ดังนี้



รูปที่ 2.4.9 Tools Palette

สำหรับหน้าที่ของอุปกรณ์ต่างๆ บน Tools Palette นี้สรุปได้ตามตารางต่อไปนี้

ตารางที่ 2.4.2 แสดงลักษณะและหน้าที่ของอุปกรณ์ต่างๆบน Tools Palette

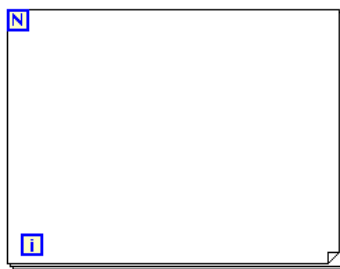
ลักษณะปุ่ม	ชื่อ	หน้าที่
	Operating Tool	ช่วยให้เราเปลี่ยนค่าของ control บน front panel ขณะที่ VI ทำงานหรือเปลี่ยนค่าอื่นๆ ในขณะที่ VI อยู่ในโหมดแก้ไข
	Positioning Tool	ช่วยปรับขนาด, เคลื่อนย้ายที่ หรือเลือก object
	Labeling Tool	สร้างหรือแก้ไข Text
	Wiring Tool	ใช้ในการต่อเชื่อมสายใน block diagram เข้าด้วยกัน
	Pop-up Menu Tool	ถ้าตัวชี้ชี้ไปที่ object ใด ก็จะทำให้เกิด Pop-up Menu ของ object นั้นขึ้น ใช้แทนการกดเมาส์ปุ่มขวาที่ Object ได้
	Scroll Tool	เลื่อนภาพบนหน้าต่างที่กำลัง active อยู่ไปในทิศทางที่ต้องการ
	Breakpoint Tool	เป็นการใส่ตำแหน่งหยุดลงใน block diagram เมื่อข้อมูลเดินทางมาถึงจุดนี้ การประมวลผลจะชะลอชั่วคราว เพื่อให้

ลักษณะปุ่ม	ชื่อ	หน้าที่
เราตรวจสอบ และแก้ไขการทำงาน		
	Probe Tool	สร้างเครื่องวัดลงบนเส้นเชื่อมเพื่อแสดงค่าข้อมูลในขณะที่ผ่านเครื่องวัดนั้นๆ
	Color Copy Tool	ใช้ในการคัดลอกสีจาก object ที่เราต้องการเพื่อสามารถปรับแก้สีที่ object อื่นให้เหมือน object นั้น
	Color Tool	ใช้ในการปรับแต่งสีของ VI ให้เป็นไปตามต้องการ

ในการเขียน Block Diagram ใน LabVIEW มีอยู่บ่อยครั้งที่เราจำเป็นต้องมีการควบคุมการเดินทางของข้อมูลให้เดินทางไปเฉพาะในทิศทางที่เราต้องการ เพราะในบางกรณีเราอาจต้องการให้ข้อมูลเดินทางไปในทิศทางหนึ่งเพราะสภาพข้อมูลเป็นไปในลักษณะหนึ่ง และอาจต้องการให้เดินทางไปในทิศทางอื่นเมื่อสภาพข้อมูลเปลี่ยนแปลงไป หรือในบางกรณีเราต้องการให้มีการทำส่วนใดหรือส่วนหนึ่งของข้อมูลซ้ำกันไปเรื่อยๆ จนกว่าเราต้องการให้ข้อมูลเดินทางออกไปสู่ส่วนอื่นต่อไป หรือให้สิ้นสุดการทำงาน

13. For Loop

For Loop เป็นการให้โปรแกรมทำงานภายในข้อกำหนดที่จะให้โปรแกรมทำงานซ้ำไปเรื่อยๆ เป็นจำนวนเท่ากับที่ผู้ใช้งานต้องการ ลักษณะของ For Loop ที่ปรากฏอยู่ใน Block Diagram จะเป็นไปตามรูป

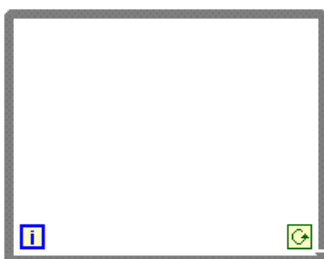


รูปที่ 2.4.10 ลักษณะของ For Loop ที่ปรากฏอยู่ใน Block Diagram

จากรูปจะเห็นว่าภายในโครงสร้างนี้จะมี terminal อยู่ 2 terminal ได้แก่ N ซึ่งเป็น Count Terminal ซึ่งต้องการ Input ว่าต้องการให้ loop นี้ ทำซ้ำเป็นจำนวนกี่ครั้ง และ i เป็น Iteration Terminal ซึ่งเป็น Output ที่บอกจำนวนครั้งที่จะมีการทำซ้ำ โดยค่าจะเริ่มจาก 0 สำหรับการทำครั้งแรกและ N-1 สำหรับครั้งสุดท้ายเมื่อ N เป็นค่าที่เรากำหนดให้กับ Count Terminal

14. While Loop

While Loop จะเป็นการทำงานของ loop ซ้ำกันไปเรื่อยๆ จนกว่าหากว่าผู้ใช้ยกเลิกสภาพตามที่กำหนด ส่วนการกำหนดสภาพจะใช้ Boolean ที่ต่อเชื่อมเข้ากับ Conditional Terminal ที่จะเห็นว่าลักษณะเป็นรูปลูกศรวนอยู่ การทำงานของ While Loop จะดำเนินการทำซ้ำไปเรื่อยๆ ตราบใดที่สภาพ Boolean ที่กำหนดให้กับ Conditional Terminal นี้เป็นจริง โดย loop จะหยุดตรวจสอบสภาพที่ terminal นี้หลังจากการทำงานแต่ละครั้ง โดยทำงานก่อนแล้วจึงจะหยุดตรวจสอบ นั่นคือตราบใดที่ Boolean ที่กำหนดให้ Conditional terminal เป็นจริง (TRUE) loop ก็ยังจะทำงานไปเรื่อยๆ ลักษณะของ While Loop เป็นดังรูป



รูปที่ 2.4.11 While Loop

โดย terminal รูปลูกศร นั้นเป็น Conditional Terminal และ i จะเป็น terminal ของจำนวนการทำซ้ำ ซึ่งจะให้ค่าเหมือน For Loop

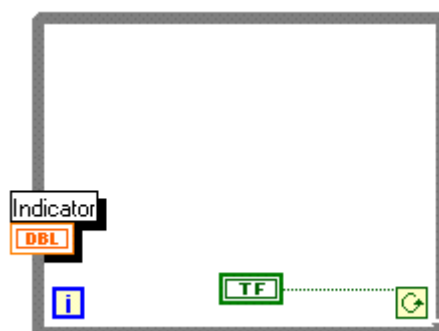
ตามที่ได้กล่าวมาแล้วว่าการทำงานของ While Loop จะทำงานก่อนจึงตรวจสอบสภาพของ Boolean ที่หลัง ดังนั้นหากเรากำหนดค่าเริ่มต้นของ Conditional Terminal เป็นเท็จ loop นี้จะทำงาน 1 ครั้งแล้วหยุด ไม่ใช่ไม่ทำงานเลย

15. การวาง Object ในโครงสร้าง

เมื่อเราเลือกโครงสร้างจาก Structures Subpalette ของ Functions Palette ตัวชี้ของเมาส์จะเปลี่ยนไปตามลักษณะของโครงสร้างที่เลือก จากนั้นเมื่อเรากดเมาส์ค้างไว้ในที่ซึ่งเราต้องการจะวางโครงสร้างใน Block Diagram การที่เรากดเมาส์ค้างไว้ก็เพื่อขยายขนาดของกรอบโครงสร้างให้มีขนาดตามต้องการ และเมื่อเราปล่อยเมาส์เราก็จะได้กรอบของโครงสร้างตามต้องการ

หลังจากที่เราได้โครงสร้างลงใน Block Diagram เราสามารถวาง object ที่ต้องการลงในกรอบนั้น ซึ่งอาจจะเป็นการดึง object จากที่มีอยู่แล้วใน Diagram หรือจะสร้างใหม่ลงไป โครงสร้างนั้นก็ได้ เมื่อเรากด object เข้าสู่โครงสร้างกรอบของโครงสร้างจะเกิดการเน้นสีขึ้นและเมื่อเรากด object ออกจากโครงสร้างกรอบของ Block Diagram จะเกิดการเน้นสี

การวาง object เข้าในโครงสร้างจะต้องวางให้อยู่ภายใน ไม่ใช่การวางคาบหรือวางทับหากว่าการวางเป็นการวางทับ object นั้นจะเกิดเป็นเงาขึ้น ดังแสดงในภาพซึ่งจะไม่ถือว่า object นั้นอยู่ภายในโครงสร้าง



รูปที่ 2.4.12 การวาง Object ในโครงสร้าง

16. พฤติกรรมการทำงานของโครงสร้าง

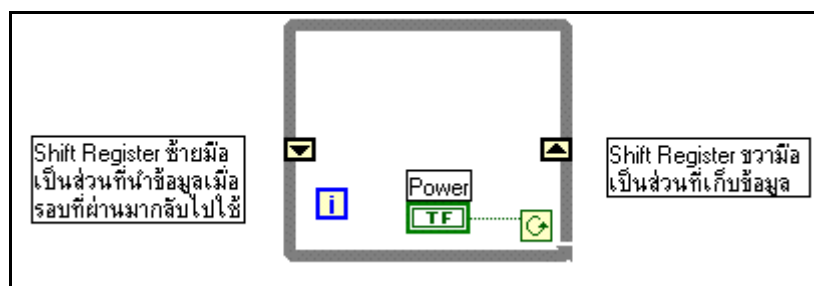
เนื่องจากการทำงานของ LabVIEW เป็นแบบการไหลของข้อมูล ดังนั้นจะต้องมีการส่งข้อมูลเข้าและออกจากโครงสร้าง ข้อมูลจะเข้าและออกจากโครงสร้างทาง อุโมงค์ (Tunnel) ซึ่งจะมีลักษณะเป็นจุดเล็กๆ เพื่อมีการลากสายเข้าหรือออกจากโครงสร้าง หลักการทำงานของโครงสร้างเป็นดังนี้

- ▶▶ ข้อมูลจะส่งเข้าสู่โครงสร้างก่อนที่ loop นั้นจะมีการทำการประมวลซ้ำใน loop เราไม่สามารถนำข้อมูลเข้าสู่ loop อีกจนกว่า loop จะหยุดการทำงาน
- ▶▶ การส่งข้อมูลออกจาก loop จะเกิดขึ้นก็ต่อเมื่อการประมวลผลของ loop นั้นเสร็จสิ้น ลงแล้วอย่างสมบูรณ์เราไม่สามารถนำข้อมูลใดๆ ออกจาก loop ได้ ในระหว่างนั้น ดังนั้นหากว่าเราต้องการให้มีการปรับเปลี่ยนข้อมูลหรือแสดงผลในทุกกรอบการทำงานของ loop นั้นเราจะต้องใส่ object ลงไปใน loop นั้น

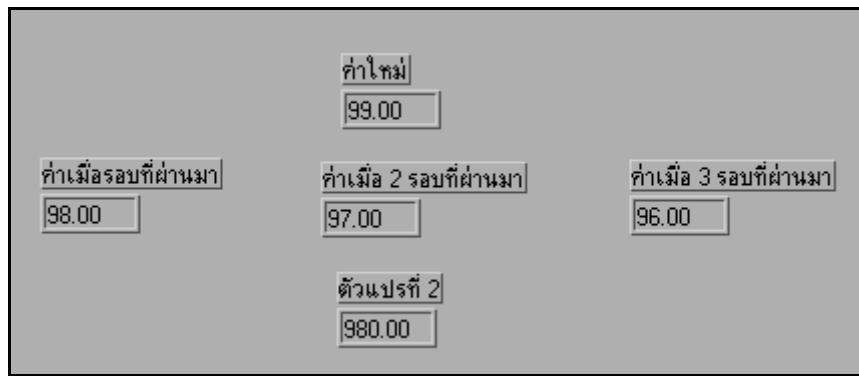
17. Shift Registers

ในการทำ loop ซ้ำกันหลายๆ ครั้งในบางกรณีเราต้องการค่าของตัวแปรบางตัวที่ใช้ในรอบการทำงานที่ผ่านมา เพื่อมาใช้ในการคำนวณสำหรับรอบใหม่ด้วย การเรียกค่าข้อมูลเก่าที่เกิดขึ้นในการทำงานในรอบที่แล้วมาใช้เพื่อนำมาใช้เป็นข้อมูลในการทำงานรอบปัจจุบัน สามารถกระทำได้โดยใช้ Shift Register

การมี Shift Register ก็เหมือนกับการมีที่ฝากค่าตัวแปรบางตัวไว้เพื่อจะนำมาใช้ในการคำนวณรอบต่อไป Shift Register จึงมี 2 ส่วนคือ ส่วน ฝากข้อมูล และส่วนดึงข้อมูล ซึ่งจะมีลักษณะเป็นรูปลูกศร ขึ้น และลง ปรากฏที่ขอบด้านขวาและซ้ายของกรอบโครงสร้าง ดังแสดงในรูป



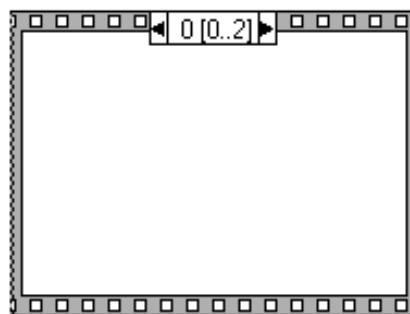
รูปที่ 2.4.13 การ Shift Registers



รูปที่ 2.4.15 รูปแบบการตั้งค่าของการ Shift Registers

18. Sequence Structure

การทำงานของ Sequence Structure จะเป็นในลักษณะกำหนดให้โปรแกรมทำงานทีละขั้น จากขั้น 0, ขั้นที่ 1 ต่อไปจนกระทั่งขั้นสุดท้าย เมื่อโปรแกรมทำงานขั้นตอนสุดท้ายเสร็จแล้ว ข้อมูลก็จะออกไปจากโครงสร้าง ขั้นตอนแต่ละขั้นตอน LabVIEW จะเรียกว่าเฟรม (Frame) ลักษณะของ Sequence Structure แสดงในรูปต่อไป



รูปที่ 2.4.16 ลักษณะของ Sequence Structure

การสร้าง Sequence Structure จะสร้างจาก Structures Subpalette ของ Function Palette และเหมือนกับ Case Structure คือเฟรมจะปรากฏให้เห็นครั้งละ 1 เฟรมเท่านั้น เราต้องกดเมาส์ที่ลูกศรบนกรอบบนของโครงสร้างเพื่อสลับดูสิ่งที่แสดงในเฟรมอื่น หรืออาจจะใช้ Pop-up menu ที่ขอบเขตโครงสร้างแล้วใช้คำสั่ง Show Frame นอกจากนี้การเพิ่มหรือลดเฟรมจะใช้คำสั่ง Add Frame After, Add Frame Before จาก Pop-up menu

ในการส่งข้อมูลของ Sequence Structure ออกสู่ภายนอก เราจะส่งออกที่เฟรมใดก็ได้ และ
 โมเมนต์ที่ส่งข้อมูลออกจะใช้เฉพาะเฟรมนั้นๆ เท่านั้น ไม่ใช้ร่วมกับเฟรมอื่น ซึ่งทำให้แตกต่างจาก
 ในกรณีของการใช้ Case Structure อย่างไรก็ตามข้อมูลจะส่งออกจากโครงสร้างก็ต่อเมื่อโปรแกรม
 ทำการประเมินทุกเฟรมจนครบแล้ว

เราจะใช้ Sequence Structure เพื่อควบคุมลำดับขั้นตอนการทำงานของ node ซึ่งข้อมูลของแต่ละ
 เฟรมไม่เกี่ยวข้องกัน ทำให้ในแต่ละเฟรมจะมีลักษณะคล้ายกับ Block Diagram ที่แยกออกมา
 ต่างหาก อย่างไรก็ตามเราสามารถส่งผ่านข้อมูลข้ามระหว่างเฟรมได้โดยใช้ terminal ที่เรียกว่า
 Sequence Local ซึ่งจะเป็นการฝากและดึงข้อมูลในระหว่างเฟรม ในการจะเพิ่ม Sequence Local นี้
 เราจะได้จากการใช้คำสั่ง Add Sequence Local ภายใต้ Pop-up menu ของ Sequence Structure

2.4.2 การควบคุมเวลา

เราสามารถที่จะควบคุม หรือตรวจสอบเวลาการทำงานของ LabVIEW ได้โดยการเลือกใช้
 ฟังก์ชัน Wait (ms), Wait Until Next ms Multiple และ Tick count (ms) โดยเลือกจาก Time &
 Dialog subpalette ของ Function Palette โดยเครื่องมือควบคุมและตรวจสอบทางด้านเวลาขึ้น
 พื้นฐานมีดังนี้

2.4.2.1 Wait (ms)  เป็นการสั่งให้ LabVIEW หยุดการทำงานชั่วคราวเป็นระยะเวลาตามที่
 กำหนด ก่อนที่จะทำงานต่อไป โดยเวลาที่ใช้จะเป็น millisecond

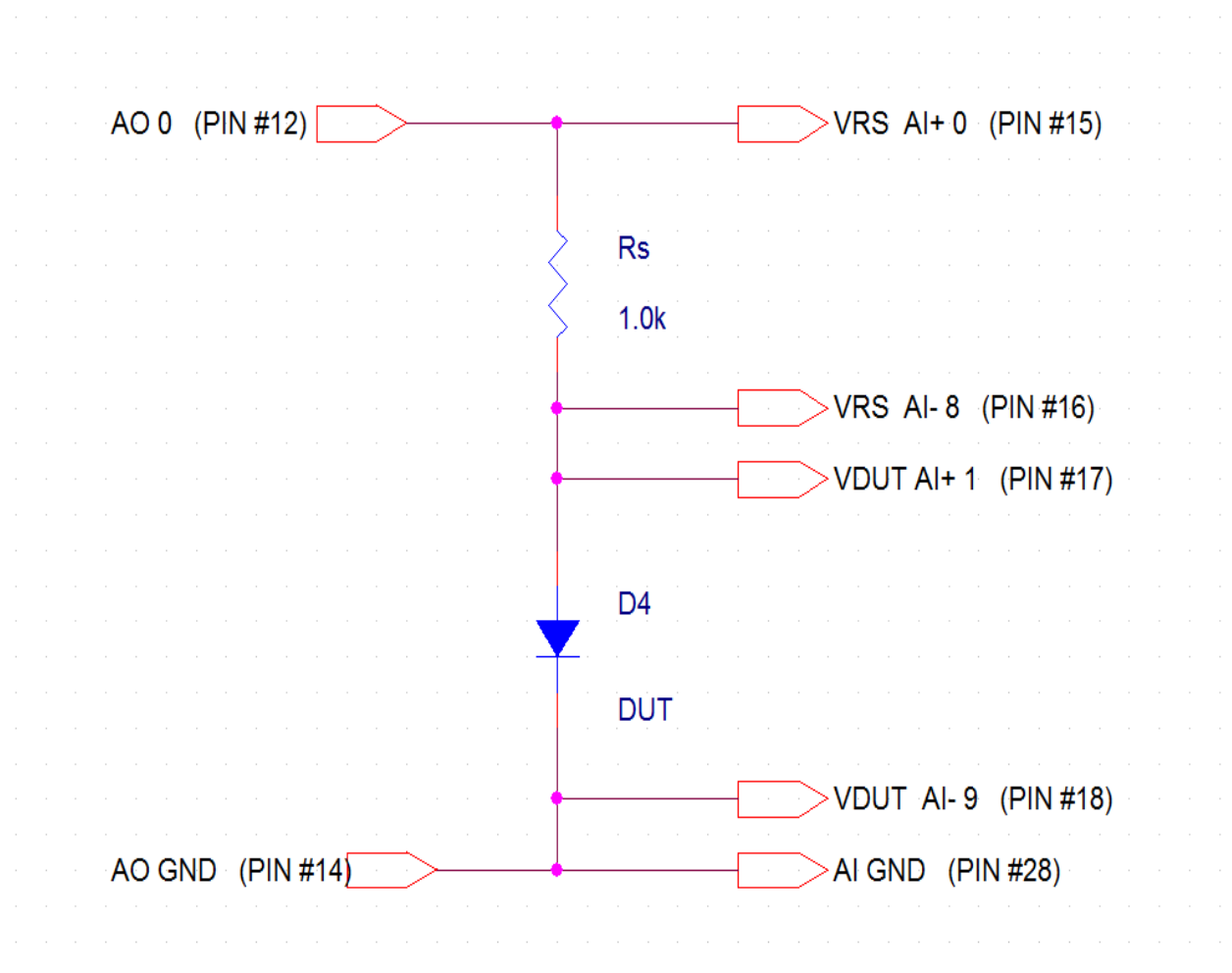
2.4.2.2 Wait Until Next ms Multiple  จะเป็นการสั่งให้ LabVIEW รอจนกว่าเวลาเป็นจำนวน
 เท่าของเวลาที่กำหนดให้ก่อนที่จะทำงานต่อไป

2.4.2.3 Tick count (ms)  เป็นการแจ้งเวลาตามเวลาของนาฬิกาภายในเครื่องคอมพิวเตอร์เป็น
 millisecond

บทที่ 3

การออกแบบ

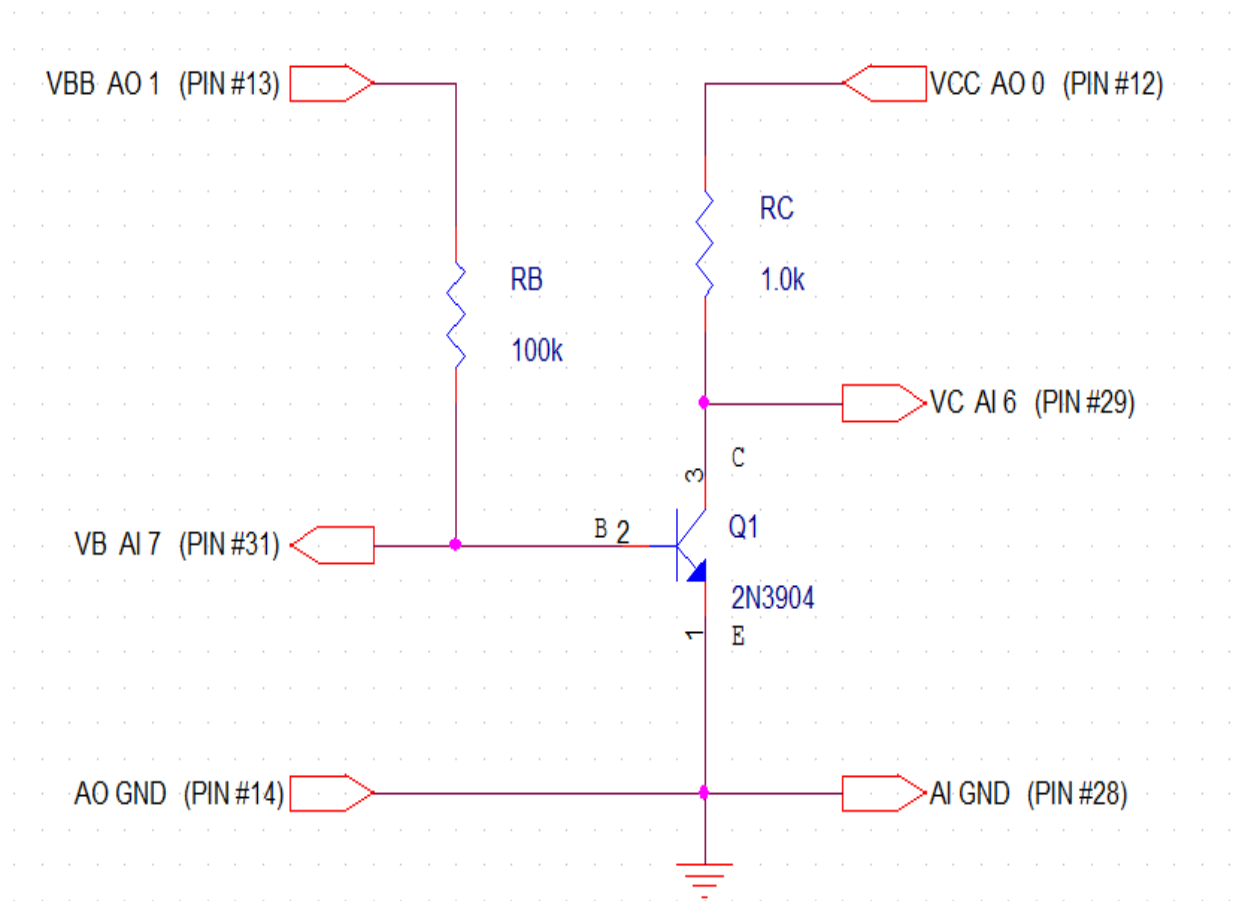
3.1 วงจรไดโอด



รูปที่ 3.1 แบบจำลองการทำงานของวงจรไดโอด

จากรูปที่ 3.1 เป็นการแสดงให้เห็นถึงการทำงานของวงจรไดโอด ซึ่งมี V_1 ทำการจ่ายแรงดันให้ VRS ที่เชื่อม R_s อยู่ และ V_d ที่เชื่อมไดโอดอยู่ ซึ่งค่าที่ได้ก็จะนำมาเก็บไว้หรือแสดงออกมาทางหน้าจอ

3.2 วงจรทรานซิสเตอร์

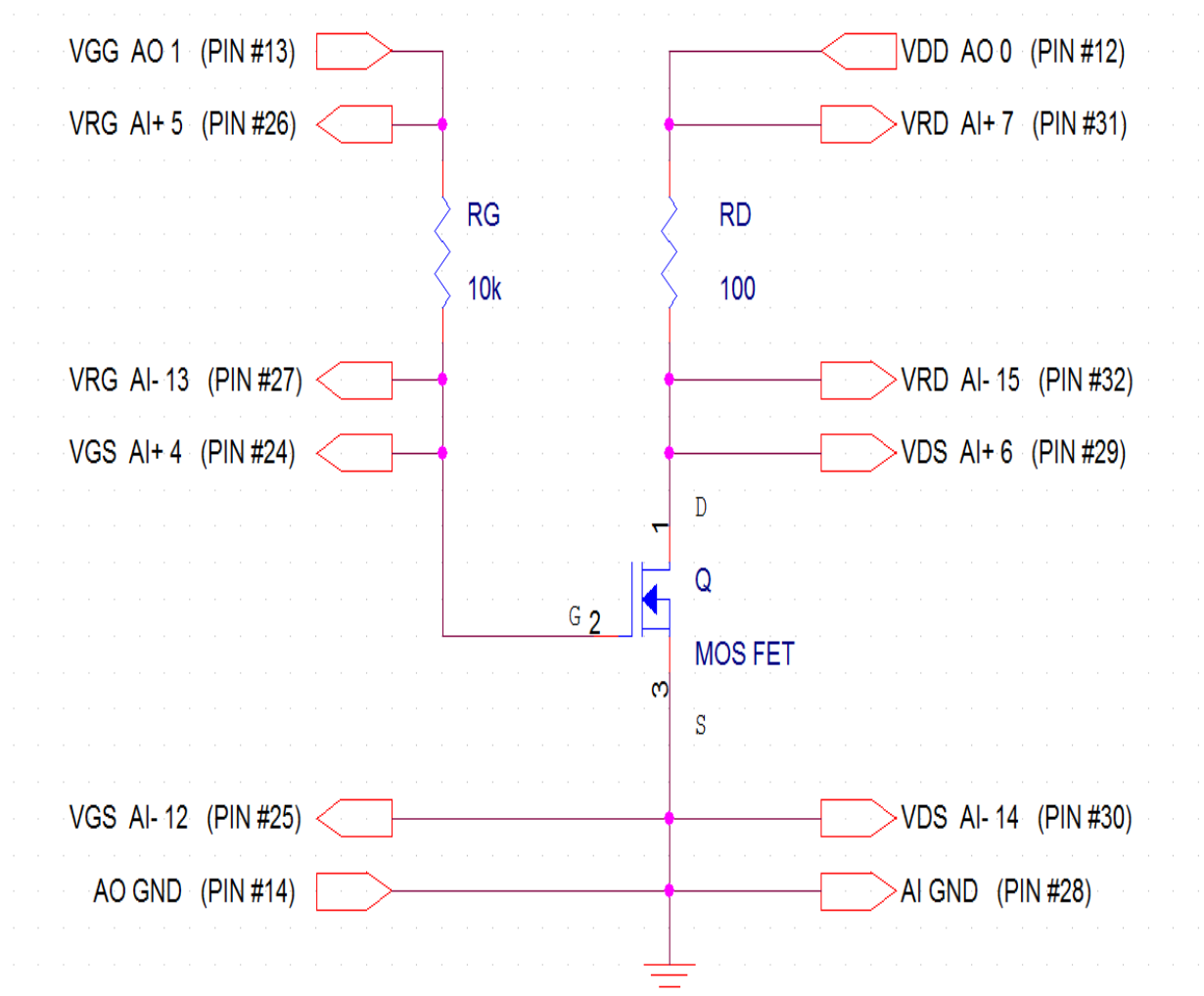


รูปที่ 3.2 แบบจำลองการทำงานของวงจรทรานซิสเตอร์

จากรูปที่ 3.2 เป็นการแสดงให้เห็นถึงการทำงานของวงจรทรานซิสเตอร์ ซึ่งมี VCC ที่จ่ายแรงดันให้กับทั้งขาเบส, ขาอีมิเตอร์ และขาคอลเลกเตอร์ ซึ่งในแต่ละขาก็จะมีแรงดันตกคร่อมของมันเอง

ตามในรูปที่ 3.2

3.3 วงจรมอสเฟต



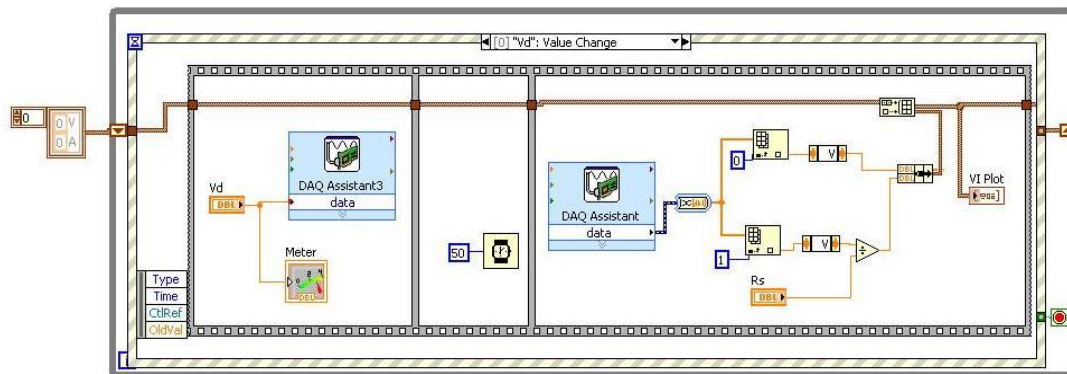
รูปที่ 3.3 แบบจำลองการทำงานของวงจรมอสเฟต

จากรูปที่ 3.3 เป็นการแสดงให้เห็นถึงการทำงานของวงจรมอสเฟต ซึ่งมี VDD ที่จ่ายแรงดันให้กับทั้ง ขาเกต , ขาซอส และขาเดรน ซึ่งในแต่ละขา ก็จะมีแรงดันตกคร่อมของมันเอง ตามในรูปที่ 3.3

บทที่ 4

หลักการและการออกแบบ

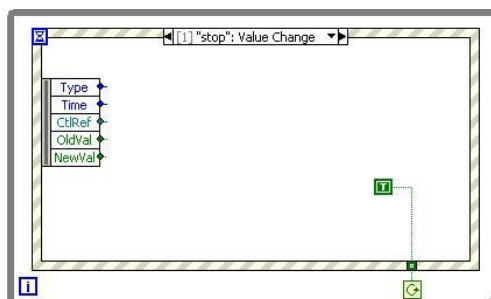
4.1 วงจรไดโอด



รูปที่ 4.1.1 Vd “Value change”

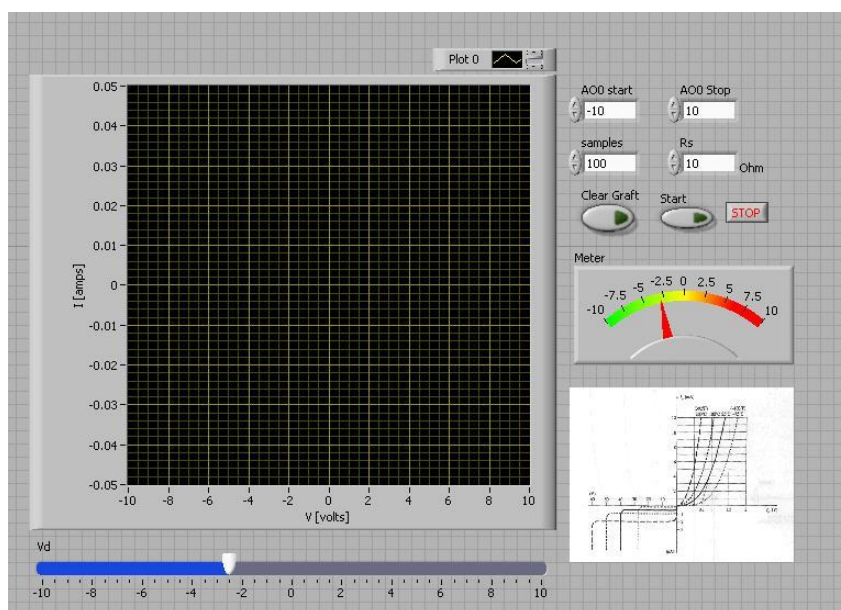
หลักการการทำงานของบล็อกไดโอดแอมป์นี้คือ เมื่อเริ่มการทำงานของโปรแกรมขึ้น ใน event structure Vd บล็อกจะทำการส่งค่าแรงดันขนาด -10 ถึง 10 โวลต์ เข้าสู่ DAQ Assistant3 เพื่อระบุค่าให้กับเอาต์พุตที่กล่อง โดยที่กล่องจะมีไดโอดต่ออนุกรมกับตัวต้านทาน ซึ่งไดโอดนั้นคือไดโอดที่เราต้องการจะวัดค่า และกล่องนี้จะป้อนค่ากลับให้แสดงผลบน Front panel โดยเราจะรู้ระดับแรงดันที่ป้อนให้กล่องได้โดยผ่านทางมิเตอร์ที่อยู่ใน Sequence แรก และใน Sequence ที่สองจะทำการหยุดก่อนเริ่มกดปุ่มสตาร์ทเป็นเวลา 50 มิลลิวินาที เพื่อหน่วงเวลาเริ่มการทำงานของระบบ และเพื่อกันผิดพลาดเล็ก ๆ น้อย ๆ ใน Sequence ที่สามนั้น จะเอาข้อมูลของไดโอดที่ต่ออนุกรมกับตัวต้านทานมาอ่านค่าบนจอคอมพิวเตอร์ หรือ DAQ Assistant นั้น วิ่งเข้าสู่ Index Array ที่กำหนดมิติเป็น 0 กับ 1 ซึ่งในมิติ array เท่ากับ 0 จะไม่มีแรงดันวิ่งผ่าน เหลือ array เท่ากับ 1 ออกมาเป็นแรงดัน -10 ถึง 10 โวลต์ และนำไปหารกับค่าตัวต้านทานที่เราต่ออนุกรมกับไดโอดที่ต้องการวัด โดยเรากำหนดค่าใส่ลงไป ใน Numeric Rs และวิ่งต่อไปเก็บค่าที่ bundle เพื่อเข้าสู่ Build Array ที่เรากำหนดค่าขึ้นมาเอง คือ 0 โวลต์ 0 แอมป์ และนำค่าไปพล็อตกราฟโดยผ่านบล็อก VI Plot เพื่อบันทึกค่าลงสู่กราฟบนหน้าจอคอมพิวเตอร์ โดยจะทำทีละหนึ่งค่า เริ่มจาก -10 โวลต์ ไปจน 10 โวลต์ ภายใน While Loop

ในส่วนของ Front Panel นั้น ปุ่มเริ่มการทำงานและปุ่มหยุดการทำงาน จะถูกควบคุมโดย SubVI ที่สร้างขึ้นมาสำเร็จ บน Event ภายใน While Loop โดยกำหนดความไวของการพล็อตกราฟ โดยบล็อก samples เริ่มโดย AO0 start ด้วย event structure กรณี Start



รูปที่ 4.1.5 Stop “Value change” ของปุ่มคำสั่ง

และระบุให้หยุดกราฟพล็อตโดยบล็อก AO0 Stop ด้วย event structure กรณี Stop ของ event loopตามลำดับ โดยชุดคำสั่งนี้จะรับคำสั่งจาก แรงดันที่ป้อนให้แก่ DAQ Assistant3 จาก ภายใน Shift Register ที่หน่วงเวลาไว้ 100 มิลลิวินาที

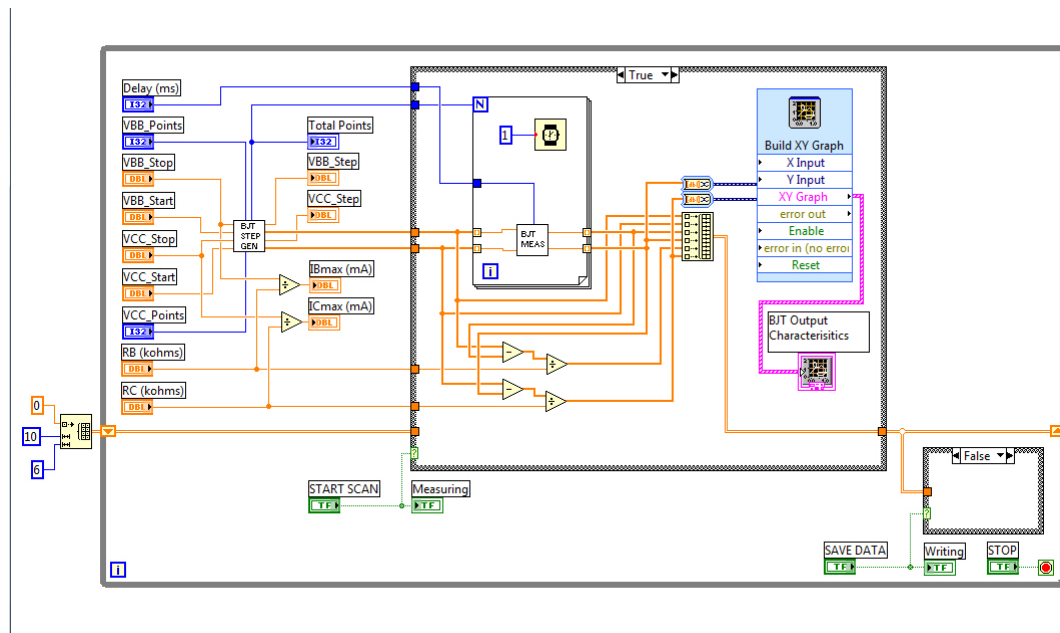


รูปที่ 4.1.6 Front Panel ของชุดคำสั่งวงจรไดโอด

เมื่อชุดคำสั่งทั้งหมดเพียบพร้อมแล้ว เราจะสามารถวัดอุปกรณ์สารกึ่งตัวนำประเภทไดโอด ได้โดยการทำผ่าน Front Panel โดยการต่อไดโอดอนุกรมกับตัวต้านทาน บนเครื่องวัด NI ที่เตรียม

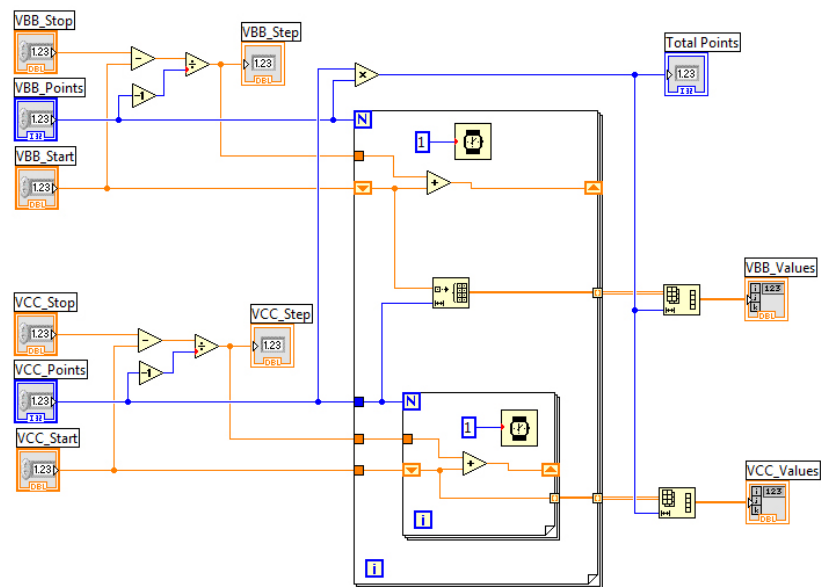
ไว้ในช่องอินพุต และต่อเอาต์พุตเข้ากับเครื่องคอมพิวเตอร์ เพื่อวัดค่าของไดโอดได้ โดยกราฟจะแสดงผลบนกราฟบนหน้าจอ โดยระบุอยู่ในช่วงที่เราหนดไว้

4.2 วงจรทรานซิสเตอร์



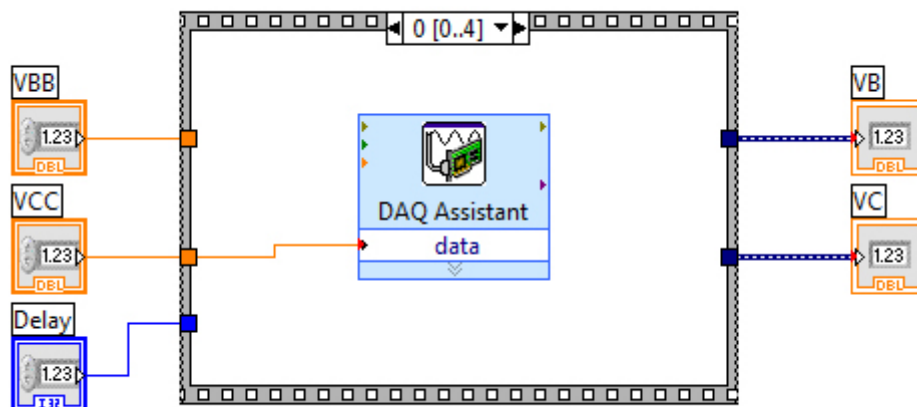
รูปที่ 4.2.1 Block Diagram หลักของทรานซิสเตอร์

จะเริ่มต้นโดยการส่งค่าเริ่มต้น 9 ค่า ส่งไป 2 ส่วน คือเข้าสู่ SubVI BJT STEP GEN เพื่อทำให้เวลาของแรงดันขาเข้ามีค่าไม่เท่ากัน แต่จะเกิดเงื่อนไขการจับแรงดันขึ้น คือเมื่อตัวหนึ่งจับแรงดันครบค่าแล้วอีกตัวจะจับไปแค่หนึ่งค่า และอีกส่วนจะนำไปคำนวณทางคณิตศาสตร์ เมื่อออกจาก SubVI BJT STEP GEN จะไปเข้าสู่ SubVI BJT MEAS เพื่อทำการเก็บค่าข้อมูลเข้าสู่ DAQ และนำข้อมูลไปใช้คำนวณกับข้อมูลที่ได้อีกทางคณิตศาสตร์ตอนแรก เพื่อนำไปพล็อตกราฟและพล็อตข้อมูลลงสู่โปรแกรม Microsoft Excel ต่อไป



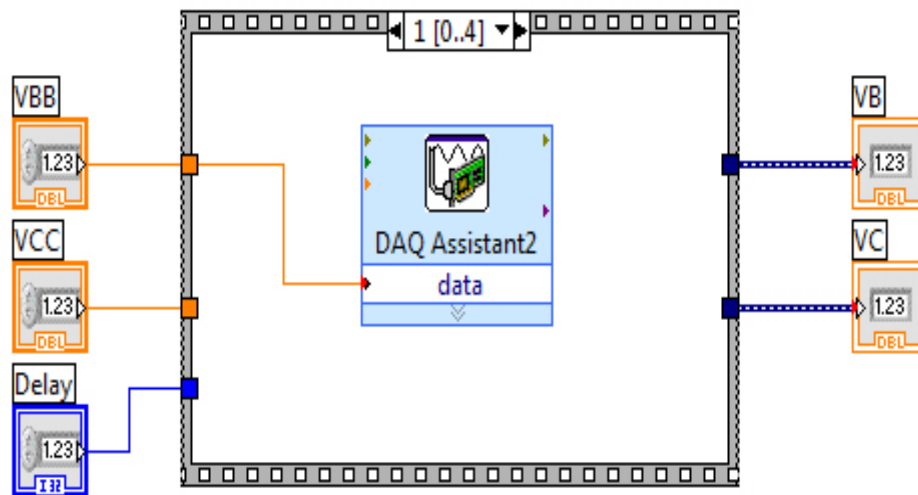
รูปที่ 4.2.2 Block Diagram ของ SubVI BJT STEP GEN

รูปนี้แสดงการหน่วงของเวลา เมื่อรอบสี่เหลี่ยมเล็กวิ่งจนครบตามจำนวน N ที่ฝั่งซ้ายมือบนของกรอบ จะทำให้รอบสี่เหลี่ยมพื้นผ้าใหญ่ขยับหนึ่งค่า และจะเป็นแบบนี้จนกว่าจะครบ N ที่กำหนด



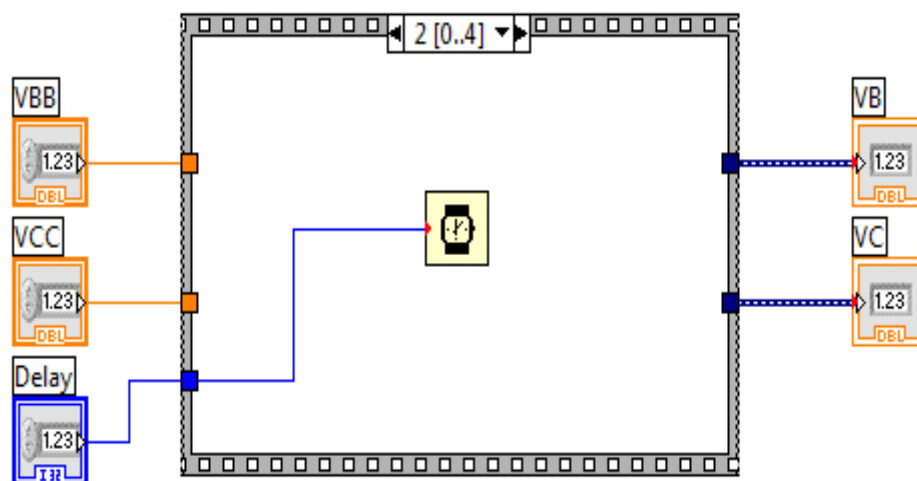
รูปที่ 4.2.3 Block Diagram SubVI BJT MEAS แสดงการนำเข้าค่า V_{cc} สู่ DAQ

รูปนี้แสดงการเก็บค่าของ V_{CC} เข้าไปเก็บไว้ใน DAQ Assistant ซึ่งจะนำค่าที่เก็บไว้ไปใช้ในภายหลัง



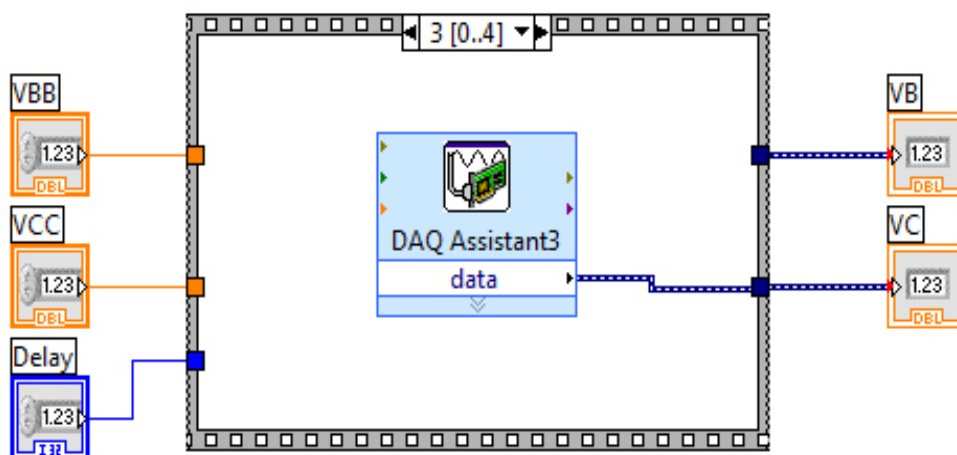
รูปที่ 4.2.4 Block Diagram SubVI BJT MEAS แสดงการนำเข้าค่า Vbb สู่ DAQ

รูปนี้แสดงการเก็บค่าของ VBB เข้าไปเก็บไว้ใน DAQ Assistant2 ซึ่งจะนำค่าที่เก็บไว้ไปใช้ในภายหลัง



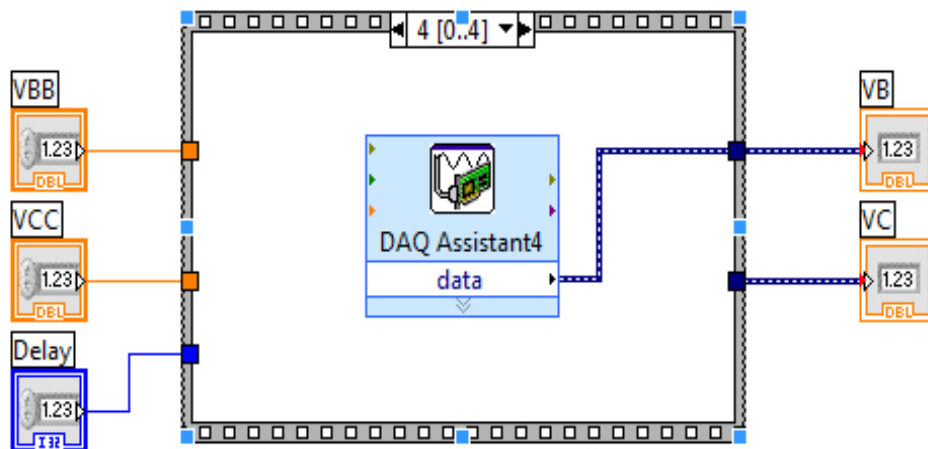
รูปที่ 4.2.5 Block Diagram SubVI BJT MEAS แสดงการนำเข้าค่า Vcc สู่ DAQ

รูปนี้แสดงการหน่วงเวลาของวงจร



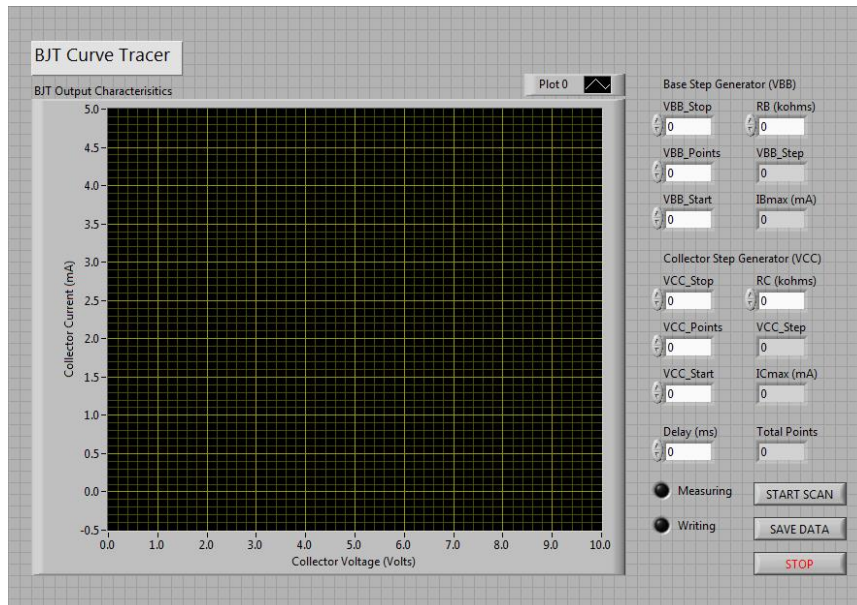
รูปที่ 4.2.6 Block Diagram SubVI BJT MEAS แสดงการส่งค่า Vc จาก DAQ

รูปนี้แสดงการส่งค่าของ VC จาก DAQ assistant3 เพื่อที่จะนำค่าไปใช้ต่อไป



รูปที่ 4.2.7 Block Diagram SubVI BJT MEAS แสดงการส่งค่า Vb จาก DAQ

รูปนี้แสดงการส่งค่าของ VB จาก DAQ assistant4 เพื่อที่จะนำค่าไปใช้ต่อไป

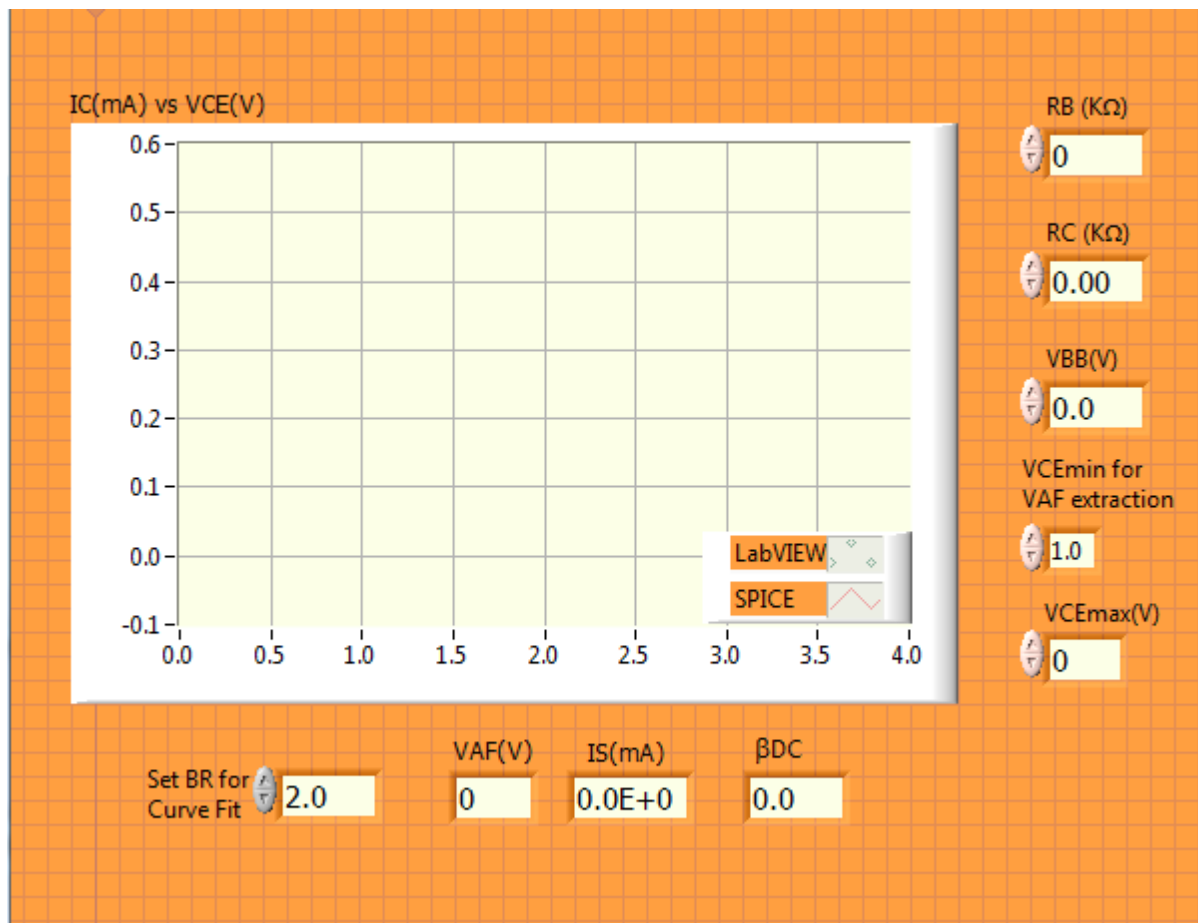


รูปที่ 4.2.8 Front Panel ของชุดคำสั่งวงจรทรานซิสเตอร์

เมื่อชุดคำสั่งทั้งหมดเพียบพร้อมแล้ว เราจะสามารถวัดอุปกรณ์สารกึ่งตัวนำประเภท ทรานซิสเตอร์ได้โดยการทำผ่าน Front Panel โดยการต่อทรานซิสเตอร์กับตัวต้านทาน บนเครื่องวัด NI ที่เตรียมไว้ในช่องอินพุต และต่อเอาต์พุตเข้ากับเครื่องคอมพิวเตอร์ เพื่อวัดค่าของทรานซิสเตอร์ ได้ โดยกราฟจะแสดงผลบนกราฟบนหน้าจอ โดยระบุอยู่ในช่วงที่เรากำหนดไว้

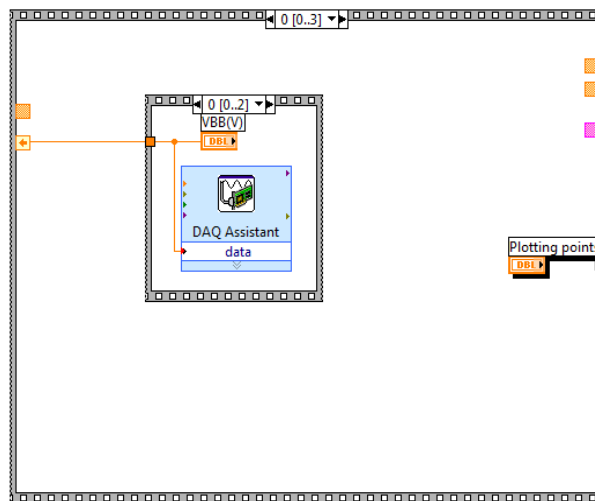
การวัดและวิเคราะห์ทรานซิสเตอร์

หลักการทำงานของเครื่องวัดและวิเคราะห์คุณสมบัติทรานซิสเตอร์จากในรูปที่ 4.2.9 ประกอบด้วยกราฟแสดงผลการวัด โดยแกนในแนวตั้งแสดงค่ากระแส I_c มีหน่วยเป็นมิลลิแอมป์ และแนวแกนนอนแสดงค่า V_{ce} มีหน่วยเป็น โวลต์ โดยในกราฟจะแสดงผลเปรียบเทียบ ระหว่าง การคำนวณ ด้วยแบบจำลอง Spice Model และการวัดค่าจริงเปรียบเทียบกัน จากหน้าจอแสดงผล ก็จะมีช่องให้ได้ ค่า R_b ค่า R_c แรงดัน V_{BB} แรงดัน V_{CEmax} ค่า BR Curve Fitting แรงดัน V_{AF} กระแส I_S และ β_{DC}



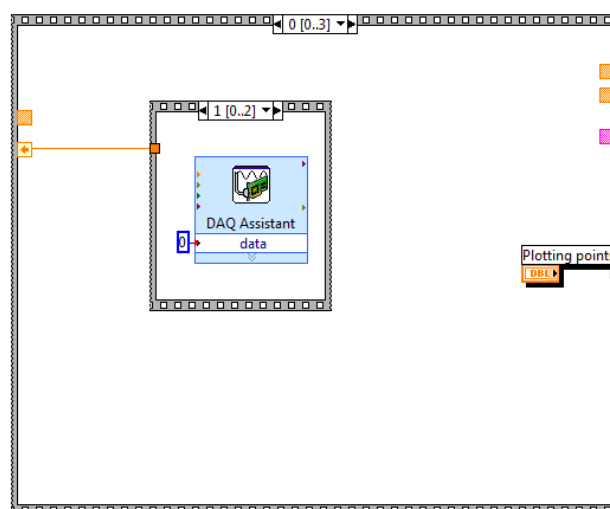
รูปที่ 4.2.9 Front Panel ของ IC_VCE.vi

จาก Block Diagram ในรูปที่ xx2 การทำงานจะเริ่มต้นจากเมื่อเราสั่งให้โปรแกรม IC_VCE.vi ทำงาน โปรแกรม LabVIEW จะนำเอาค่าต่าง ๆ ที่เราป้อนเข้าไปส่งเข้าไปประมวลผล เช่นค่า R_b ค่า R_c แรงดัน V_{BB} แรงดัน V_{CEmax} ค่า BR Curve Fitting แรงดัน V_{AF} กระแส I_S โดยจะมีโปรแกรมย่อยของ LabVIEW ที่ทำงานด้วยคือ IC_VCEsub.vi ซึ่งเป็นโปรแกรมย่อยที่สำคัญในการวัดค่าจริง โดยจะติดต่อกับอุปกรณ์วัดและส่งสัญญาณไฟฟ้าหรือ DAQ หรือ Data Acquisition โปรแกรมย่อยอีกตัวหนึ่งที่สำคัญคือ SPICE_IV เป็นโปรแกรมย่อยที่ทำหน้าที่คำนวณแบบจำลองของทรานซิสเตอร์ เพื่อเทียบกับค่าจริง โปรแกรมย่อย VAFsub.vi ใช้สำหรับหาค่า V_{AF} หรือ Forward Early voltage. เมื่อประมวลผลเสร็จแล้วก็จะได้เป็นค่า Beta ออกมา ส่วนโปรแกรมย่อย XYtoFile.vi สำหรับบันทึกเก็บค่าไว้ใช้งานอื่นต่อไปในลักษณะของตารางคำนวณ



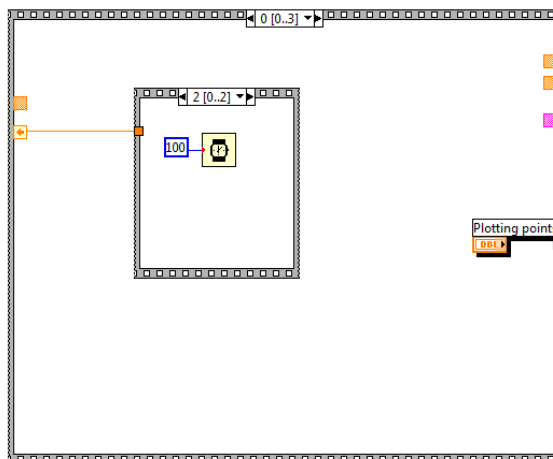
รูปที่ 4.2.12 Block Diagram ของ IC_VCESub.vi ใน Sequence ย่อย

จากรูปที่ xx4 เมื่อ Sequence ใหญ่ภายนอกวิ่งมาตามเงื่อนไขลำดับที่ 0 เริ่มทำงาน Sequence เล็กที่อยู่ภายใน จะตรงกับเงื่อนไขลำดับที่ 0 เช่นกันก็จะส่งค่าออกไปยังอุปกรณ์ DAQ ผ่านทาง Control DAQ Assistant และก็จะทำไปจนหมดตามลำดับของ Sequence เล็กที่อยู่ภายในจาก 0 ถึง 2 นั้น Sequence ใหญ่ที่อยู่ภายนอกก็จะขยับไปทำงานในลำดับถัดมาที่ 1 แล้วก็ทำอย่างนี้ไปเรื่อย ๆ โดยไล่ลำดับใน Sequence เล็กจนหมดแล้วค่อยขยับ Sequence ใหญ่ไปจนครบทุกลำดับใน Sequence แล้วจึงออกจากโปรแกรมย่อยต่อไป โดยในลำดับของ Sequence ต่าง ๆ ที่ทำงานนั้นมีรายละเอียดตั้งแต่ละ Sequence ดังจะอธิบายต่อไป



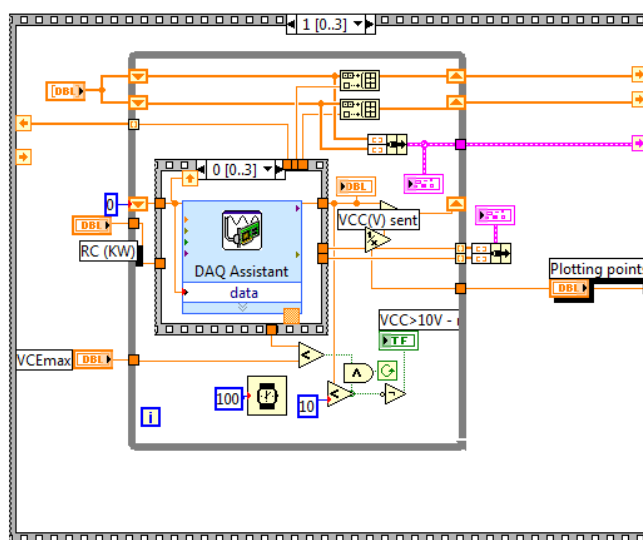
รูปที่ 4.2.13 Block Diagram ของ IC_VCESub.vi ใน Sequence ย่อย

จากรูปที่ 4.2.13 คำสั่งในส่วนนี้จะทำการส่งค่า 0 ออกไปยังอุปกรณ์ DAQ ซึ่งจะปรากฏแรงดันที่วงจรวัด 0 โวลต์นั่นเอง



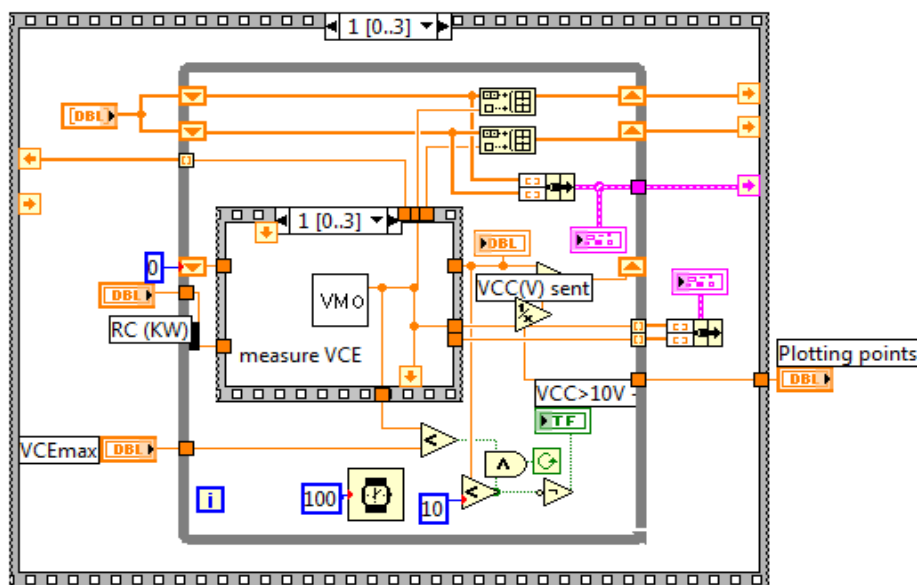
รูปที่ 4.2.14 Block Diagram ของ IC_VCEsub.vi ใน Sequence ย่อย

จากรูปที่ 4.2.14 คำสั่งในส่วนนี้จะทำการหน่วงเวลา 100 มิลลิวินาที เพื่อให้สัญญาณเสถียรพร้อมในการวัด



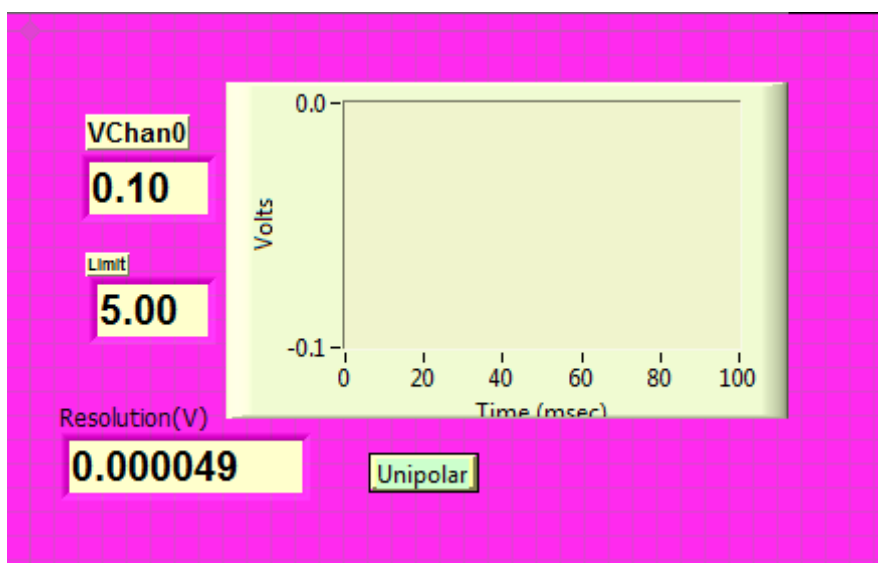
รูปที่ 4.2.15 Block Diagram ของ IC_VCEsub.vi ใน Sequence ย่อย

จากรูปที่ 4.2.15 คำสั่งในส่วนนี้จะทำการส่งค่าแรงดัน ออกไปยังอุปกรณ์ DAQ ซึ่งจะปรากฏแรงดันที่วงจรวัดและทำการไล่ลำดับค่าต่าง ๆ ออกไป ด้วยการวนลูปและ ใช้ Sequence จัดลำดับในการส่งค่าออกไปและเก็บสะสมค่าเข้าไปใน Array



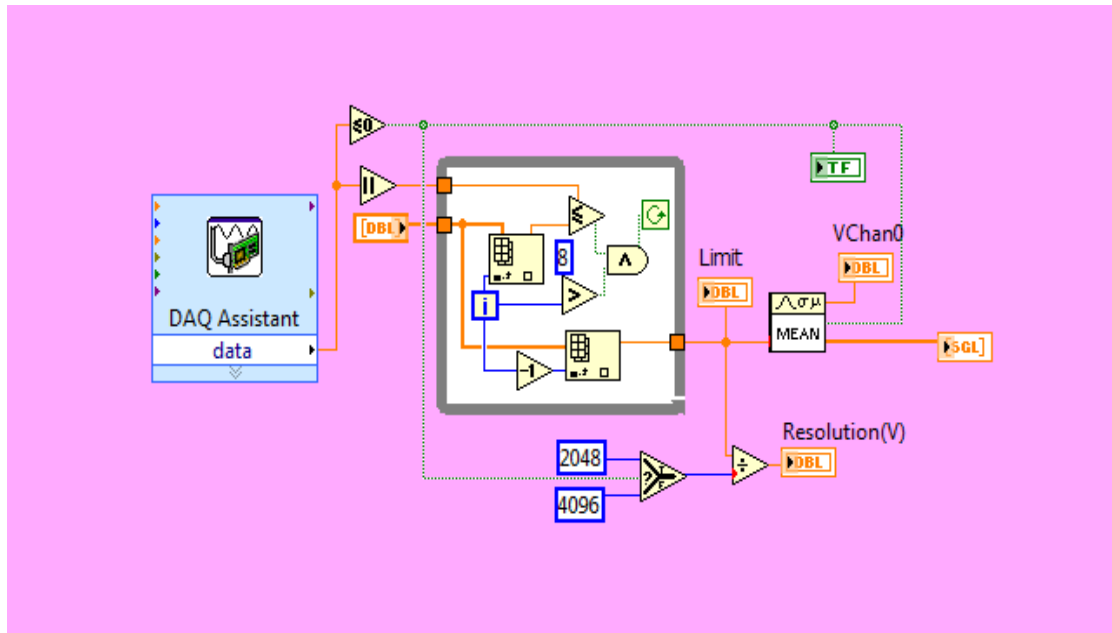
รูปที่ 4.2.16 Block Diagram ของ IC_VCESub.vi ใน Sequence ย่อย

จากรูปที่ 4.2.16 เมื่อเราส่งค่าแรงดันออกไปโดยอาศัยอุปกรณ์ DAQ แล้วเราก็จะทำการเก็บค่าแรงดันกลับมาค่าส่งในส่วนนี้จะทำการวัดออกมาคล้ายกับ Volt Meter โดยจะมีโปรแกรมย่อยทำการวัดอยู่คือ VM0 โดยทำการวัดของ Channel 0 เมื่อวัดสัญญาณมาแล้วจะมาทำการเฉลี่ยค่าที่ได้เนื่องจากในการวัดสัญญาณขนาดเล็กสัญญาณจะแกว่งและมีสัญญาณรบกวนจากนั้นก็จะนำค่าแรงดันที่ได้ไปประมวลผลต่อไปโดยมีรายละเอียดดังรูปที่ xx9 และรูปที่ x10



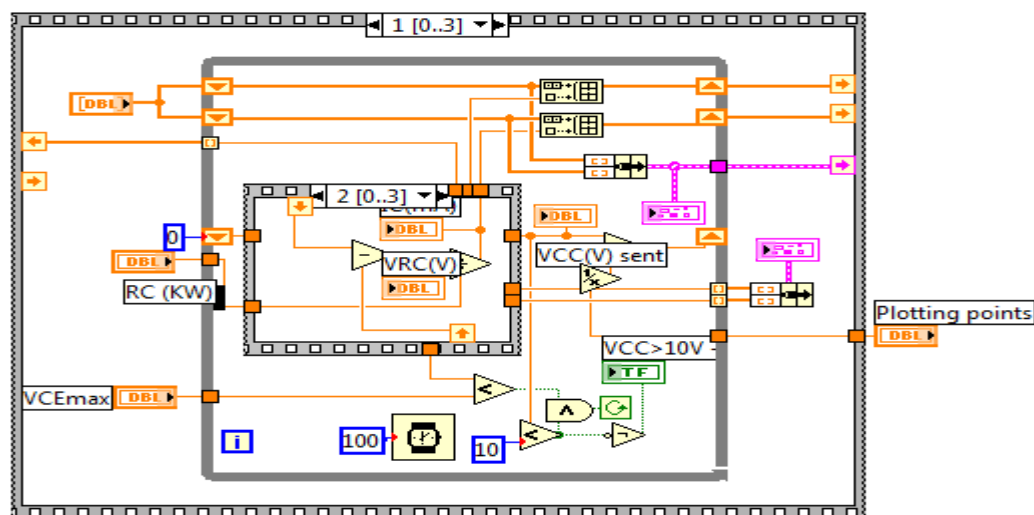
รูปที่ 4.2.17 Front Panel ของ Voltmtr0.vi

Front Panel ของ Voltmtr.vi จะส่งค่า VChan0 limit Voltage Waveform Graphs กลับไปยังโปรแกรมหลัก



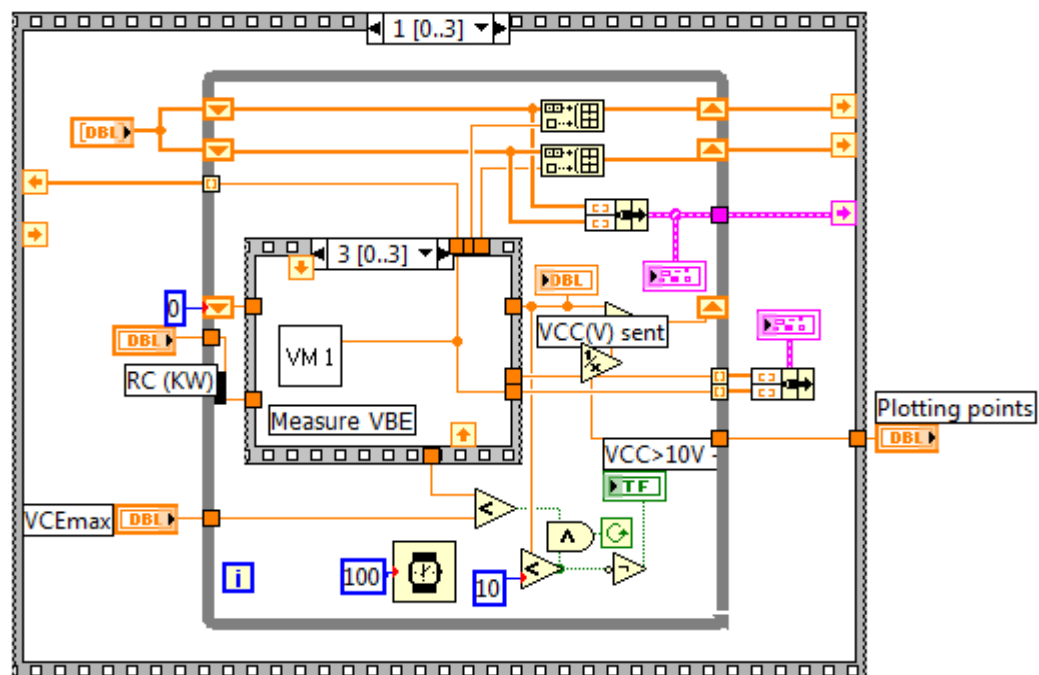
รูปที่ 4.2.18 Block Diagram ของ Voltmtr0.vi

Block Diagram ของ Voltmtr0.vi จะเห็น DAQ Asistant ส่งค่าออกมาแล้วนำค่ามาเฉลี่ย



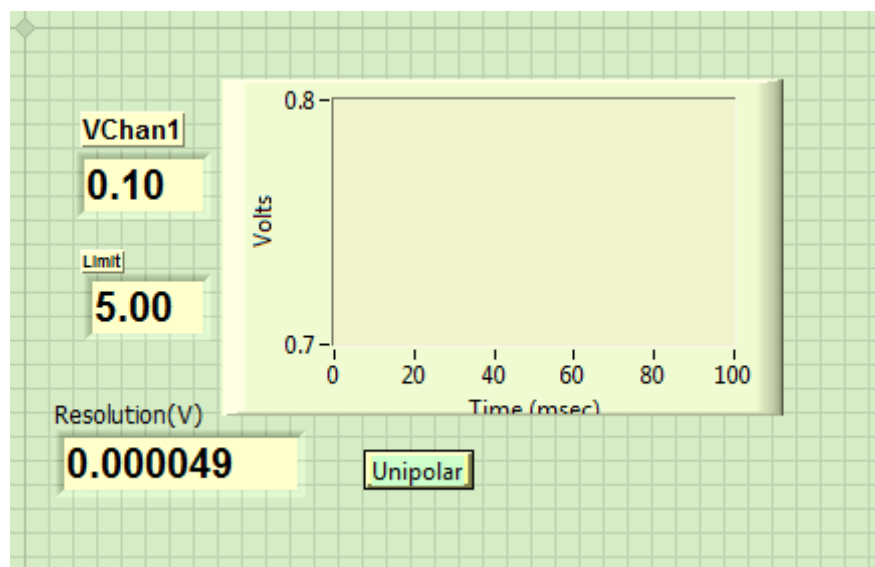
รูปที่ 4.2.19 Block Diagram ของ IC_VCEsub.vi ใน Sequence ย่อย

รูปที่ 4.2.19 ในส่วนนี้จะทำการประมวลผลและเก็บค่าลงใน Array และ Waveform Graphs



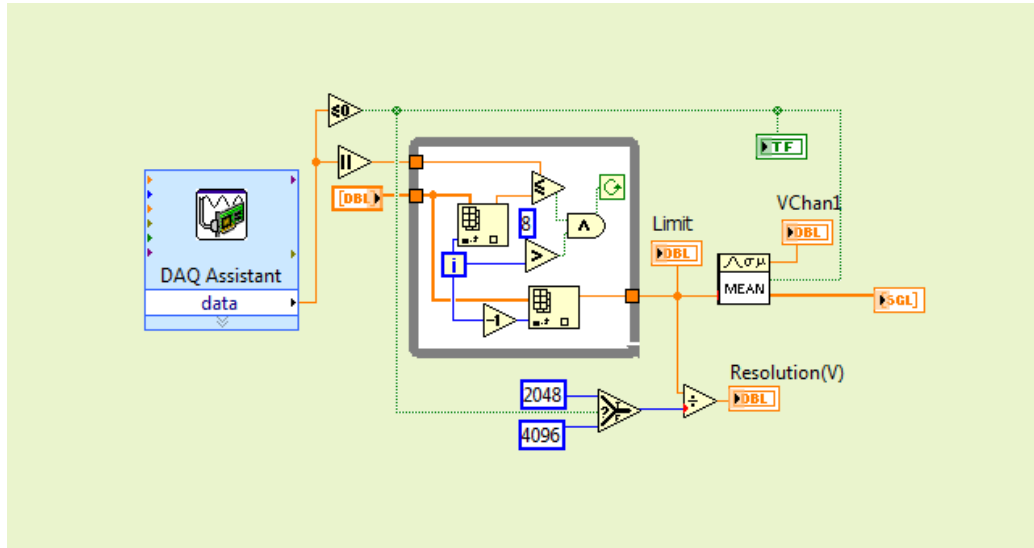
รูปที่ 4.2.20 Block Diagram ของ IC_VCESub.vi ใน Sequence ย่อย

รูปที่ 4.2.20 ในส่วนนี้จะทำการประมวลผลและเก็บค่าลงใน Array และ Waveform Graphs เช่นเดียวกันแต่ รับค่ามาจาก VM1 หรือ โปรแกรมย่อยรับค่า Chanel1



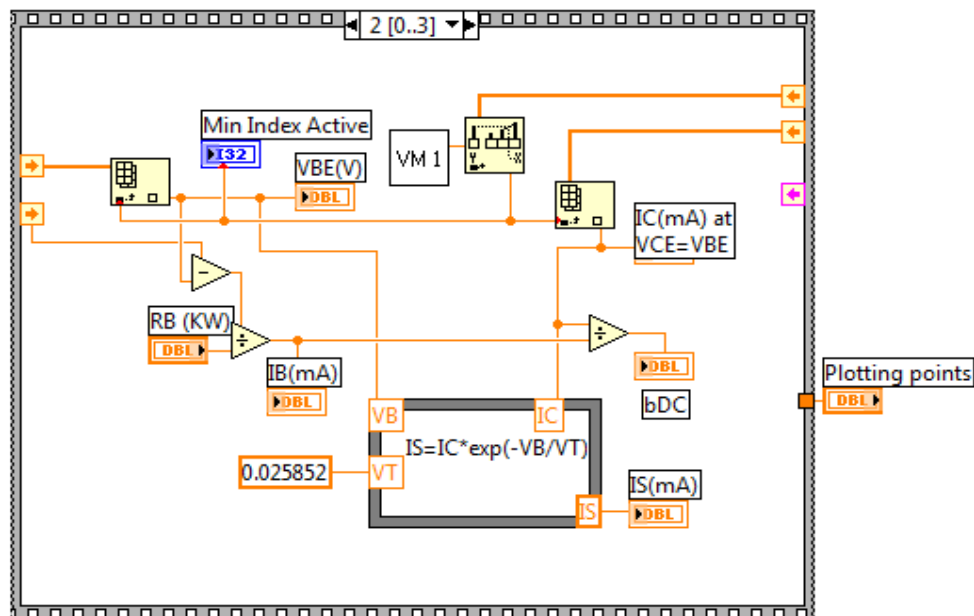
รูปที่ 4.2.21 Front Panel ของ Voltmtr1.vi

จากรูปที่ 4.2.21 Front Panel ของ Voltmtr.vi จะส่งค่า VChan1 limit Voltage Waveform Graphs กลับไปยังโปรแกรมหลัก

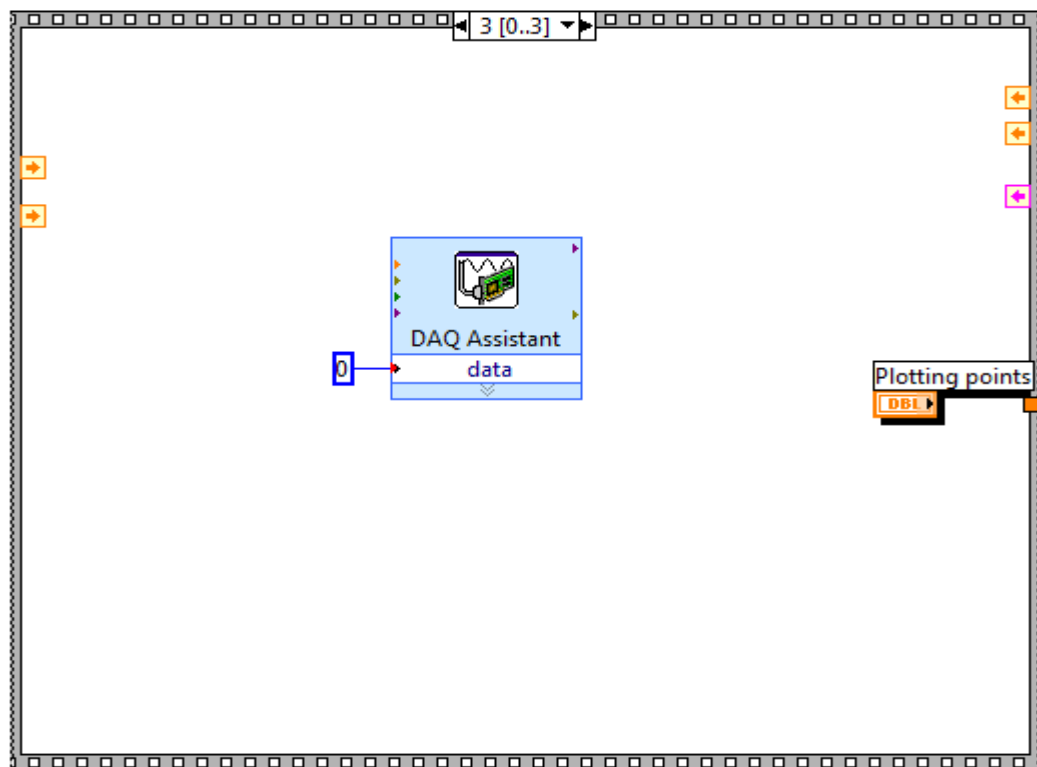


รูปที่ 4.2.22 Block Diagram ของ Voltmtr1.vi

จากรูปที่ 4.2.22 Block Diagram ของ Voltmtr1.vi จะเห็น DAQ Asistent ส่งค่าออกมาแล้วนำค่ามาเฉลี่ย

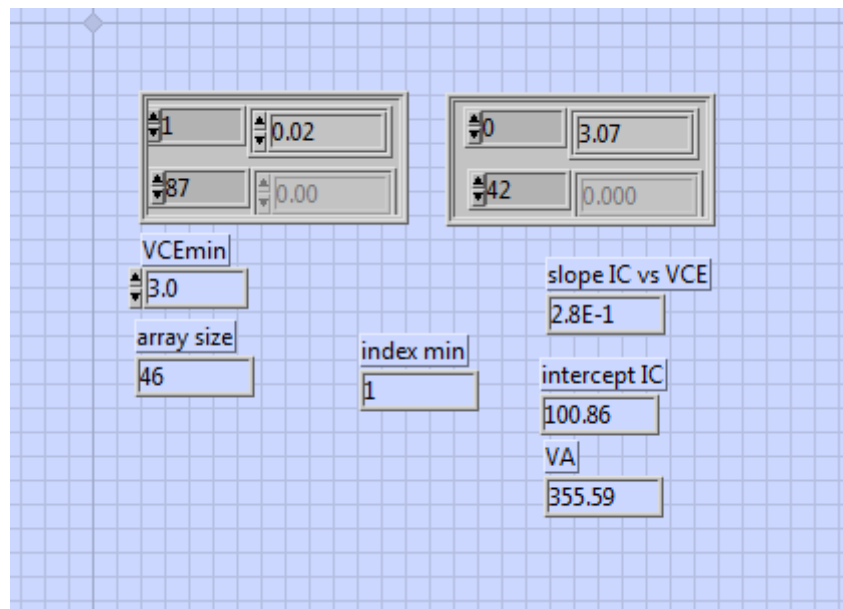


รูปที่ 4.2.23 Block Diagram ของ IC_VCEsub.vi ใน Sequence ย่อย จากรูปที่ xx15 Block Diagram ที่ใช้คำนวณค่าอัตราขยายกระแส หรือค่า Beta โดยใช้ Formula node



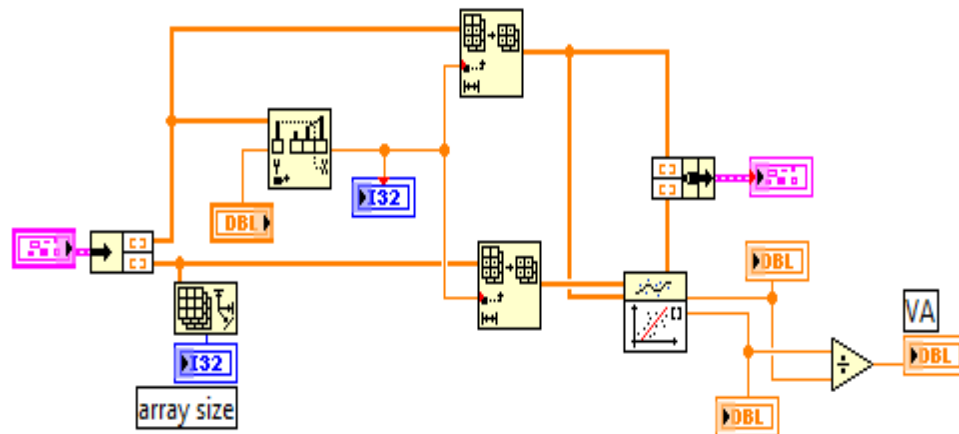
รูปที่ 4.2.24 Block Diagram ของ IC_VCESub.vi ใน Sequence ซ้อย

จากรูปที่ 4.2.24 Block Diagram ที่ทำการล้างค่าทั้งหมดให้เป็น 0



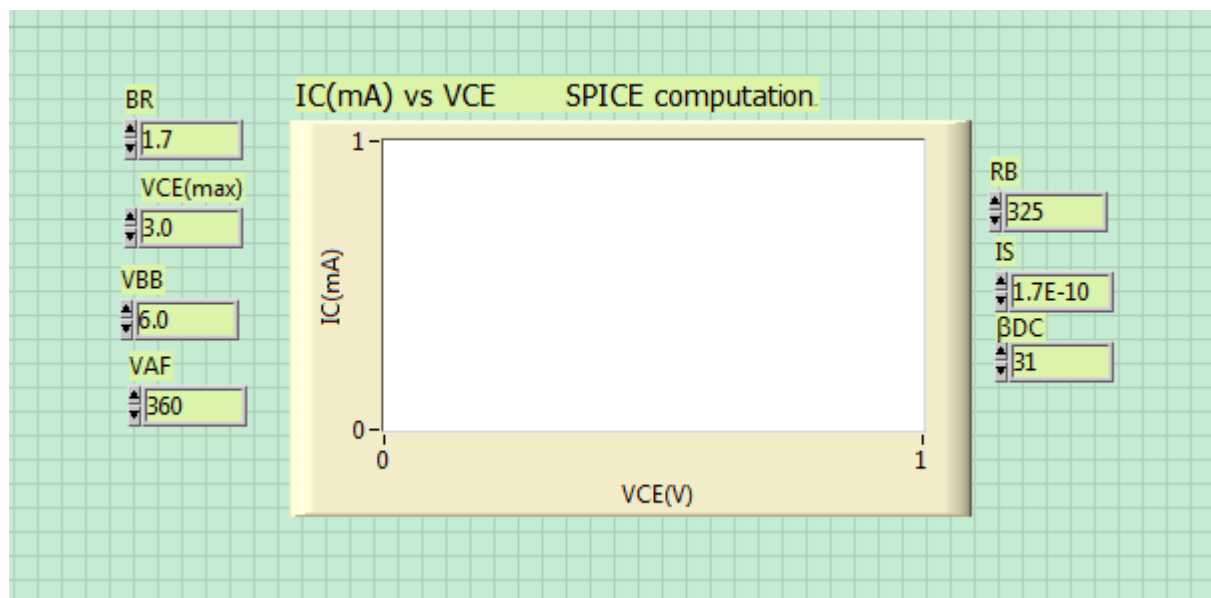
รูปที่ 4.2.25 Front Panel ของ VAFsub.vi

จากรูปที่ 4.2.25 Front Panel ที่ใช้ในการส่งและรับค่า เพื่อหา VAF



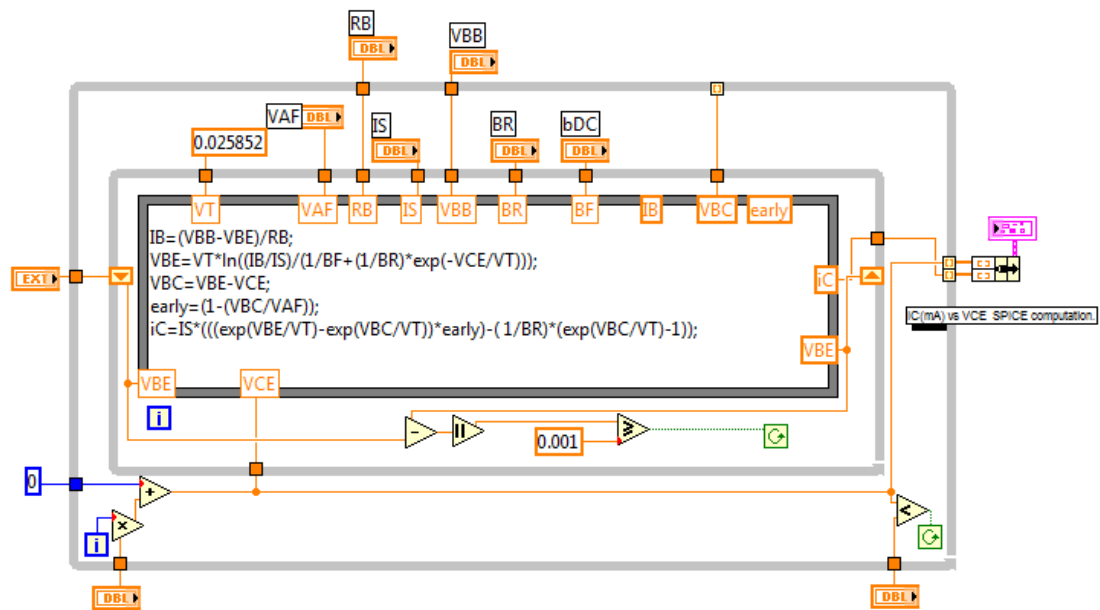
รูปที่ 4.2.26 Block Diagram ของ VAFsub.vi

จากรูปที่ 4.2.26 Block Diagram ที่ใช้คำนวณค่า จะเห็นได้ว่าการใช้ Threshold 1D Array Function ในการปรับสัญญาณ และใช้ VAF Linear Fit VI ทำการ Fit Curve ก่อนจะได้เป็น VA ออกมา



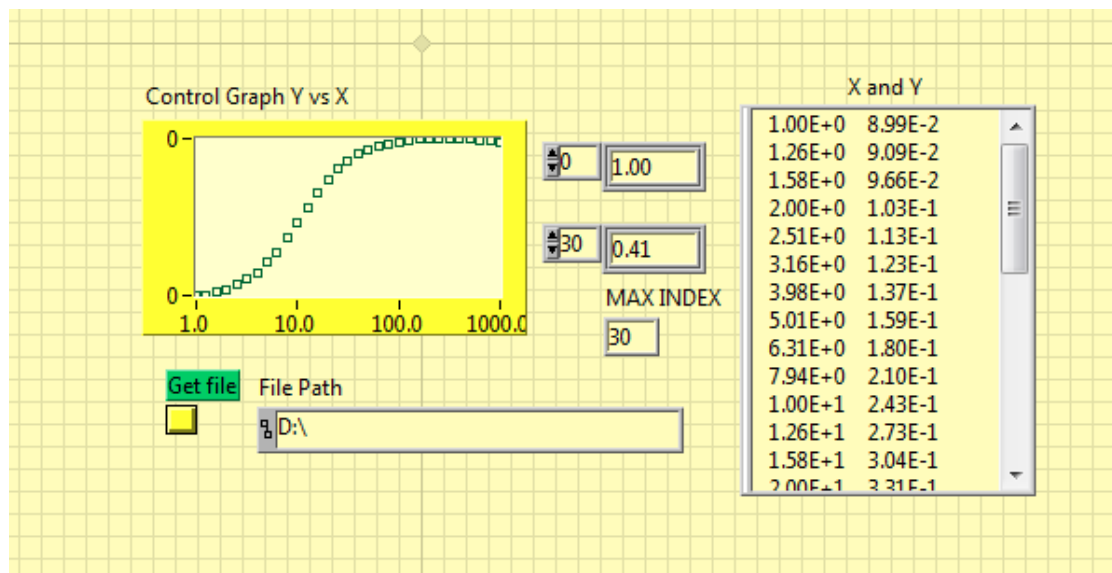
รูปที่ 4.2.27 Front Panel ของ SPICE_VI.vi

จากรูปที่ 4.2.27 Front Panel ที่ใช้ในการส่งและรับค่า เพื่อทำการคำนวณ SPICE Model



รูปที่ 4.2.28 Block Diagram ของ SPICE_VI.vi

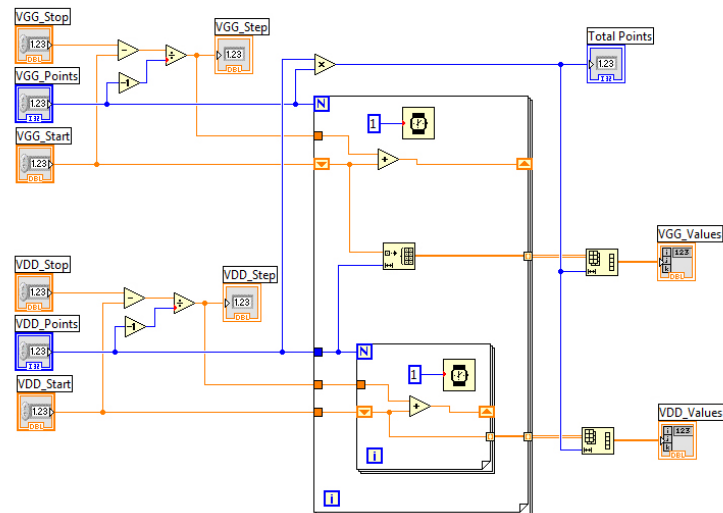
จากรูปที่ 4.2.28 Block Diagram ที่ใช้ในการคำนวณ SPICE Model โดยใช้ Formula Node รายละเอียดการคำนวณดังที่แสดงไว้จากนั้นก็ส่งค่ากลับไปยังโปรแกรมหลักเพื่อแสดงผลในกราฟต่อไป



รูปที่ 4.2.29 Front Panel ของ XYToDataFile.vi

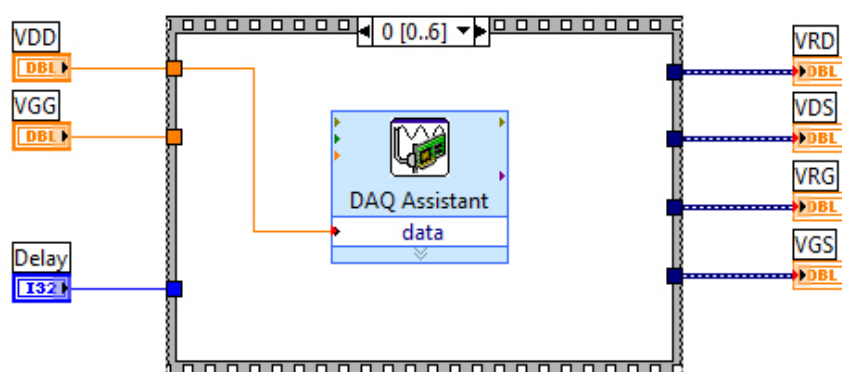
จากรูปที่ 4.2.29 Front Panel ของ XY to File ที่ใช้ในการส่งและรับค่า เพื่อทำการเก็บบันทึกค่าไว้

ส่วนจะนำไปคำนวณทางคณิตศาสตร์ เมื่อออกจาก SubVI FET STEP GEN จะไปเข้าสู่ SubVI FET MEAS เพื่อทำการเก็บค่าข้อมูลเข้าสู่ DAQ และนำข้อมูลไปใช้คำนวณกับข้อมูลที่ได้อีกทาง คณิตศาสตร์ตอนแรก เพื่อนำไปพล็อตกราฟและพล็อตข้อมูลลงสู่โปรแกรม Microsoft Excel ต่อไป



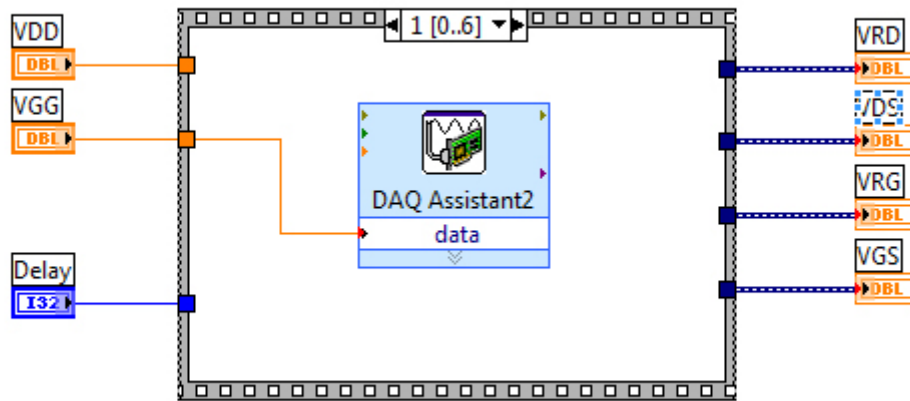
รูปที่ 4.3.2 Block Diagram SubVI ของ SubVI FET STEP GEN

จะเห็นว่าทรานซิสเตอร์และมอสเฟตต่างก็มีรูปแบบ Block Diagram เหมือนกัน ทั้งนี้เพราะทั้งสองชนิด มีค่าใกล้เคียงกัน อุปกรณ์มีสามค่าเท่ากัน มีการอิมิตัวของกระแสเช่นเดียวกัน ทำให้สามารถใช้ไดอะแกรมที่มีความคล้ายคลึงกันได้



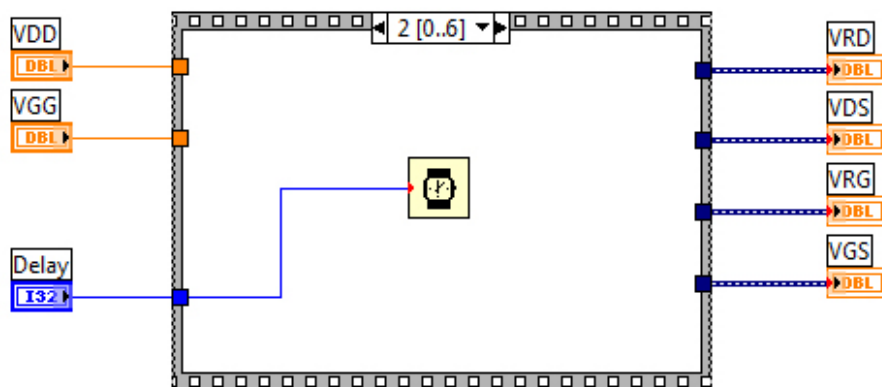
รูปที่ 4.3.3 Block Diagram SubVI FET MEAS แสดงการนำเข้าค่า Vdd สู่ DAQ

รูปนี้แสดงการเก็บค่าของ VBB เข้าไปเก็บไว้ใน DAQ Assistant ซึ่งจะนำค่าที่เก็บไว้ไปใช้ในภายหลัง



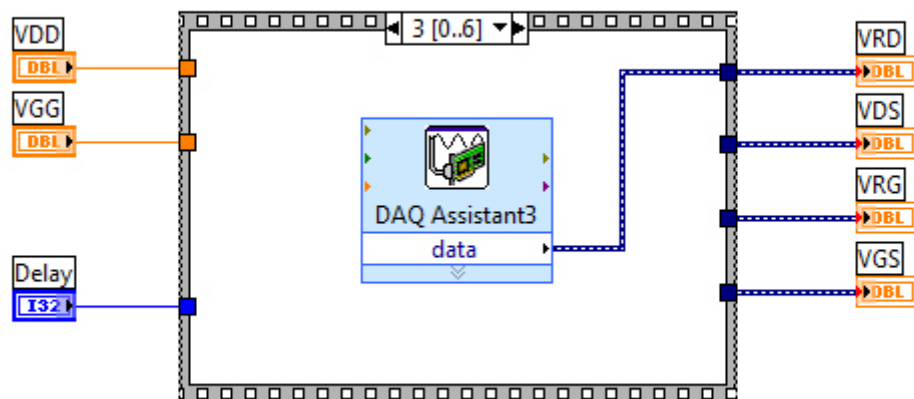
รูปที่ 4.3.4 Block Diagram SubVI FET MEAS แสดงการนำค่า Vgg เข้าสู่ DAQ

รูปนี้แสดงการเก็บค่าของ VGG เข้าไปเก็บไว้ใน DAQ Assistant2 ซึ่งจะนำค่าที่เก็บไว้ไปใช้ในภายหลัง



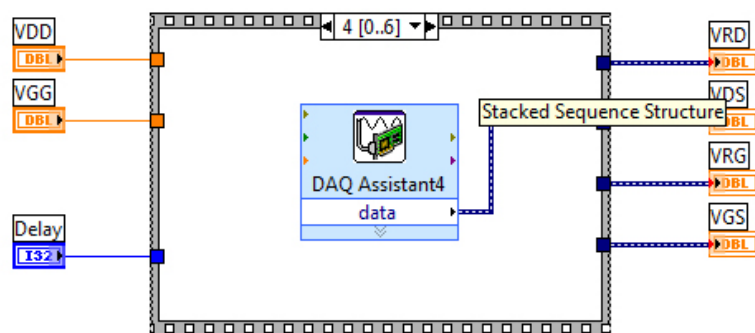
รูปที่ 4.3.5 Block Diagram SubVI FET MEAS แสดงการนำค่า Delay เข้าสู่ DAQ

รูปนี้แสดงการนำค่า Delay เข้าสู่ DAQ assistant



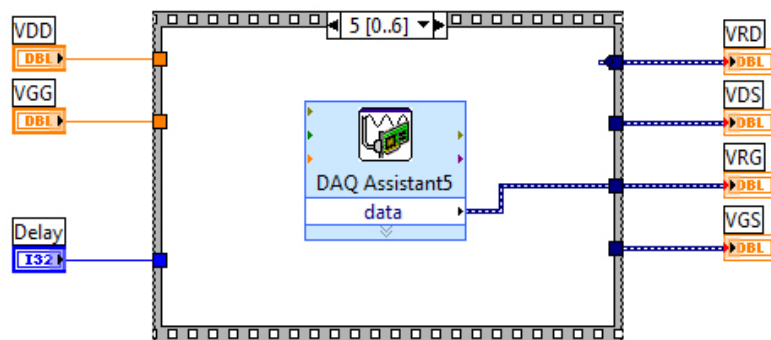
รูปที่ 4.3.6 Block Diagram SubVI FET MEAS แสดงการส่งออกมา Vrd จาก DAQ

รูปนี้แสดงการส่งค่าของ VRD จาก DAQ assistant3 เพื่อที่จะนำค่าไปใช้ต่อไป



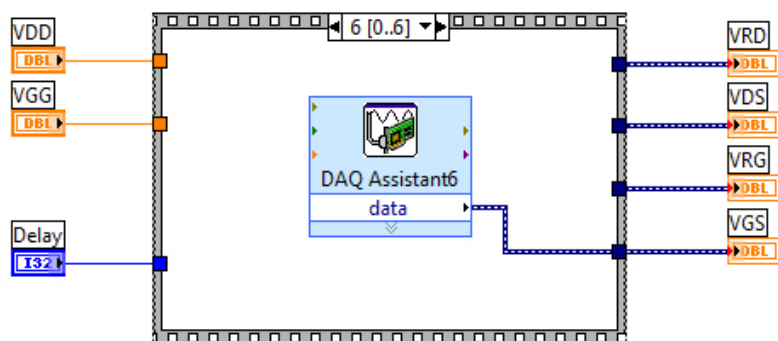
รูปที่ 4.3.7 Block Diagram SubVI FET MEAS แสดงการส่งออกมา Vds จาก DAQ

รูปนี้แสดงการส่งค่าของ VDS จาก DAQ assistant4 เพื่อที่จะนำค่าไปใช้ต่อไป



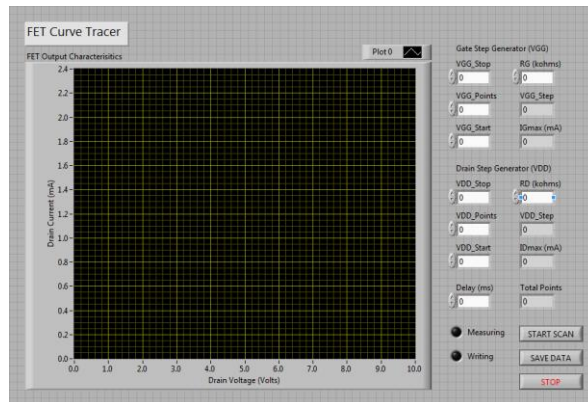
รูปที่ 4.3.8 Block Diagram SubVI FET MEAS แสดงการส่งออกมา Vrg จาก DAQ

รูปนี้แสดงการส่งค่าของ VRG จาก DAQ assistant5 เพื่อที่จะนำค่าไปใช้ต่อไป



รูปที่ 4.3.9 Block Diagram SubVI FET MEAS แสดงการส่งออกมา Vgs จาก DAQ

รูปนี้แสดงการส่งค่าของ VGS จาก DAQ assistant6 เพื่อที่จะนำค่าไปใช้ต่อไป



รูปที่ 4.3.10 Front Panel ของชุดคำสั่งวงจรมอสเฟส

เมื่อชุดคำสั่งทั้งหมดเทียบพร้อมแล้ว เราจะสามารถวัดอุปกรณ์สารกึ่งตัวนำประเภท มอสเฟส ได้โดยการทำผ่าน Front Panel โดยการต่อมอสเฟสกับตัวต้านทาน บนเครื่องวัด NI ที่เตรียมไว้ในช่องอินพุต และต่อเข้าที่พุตเข้ากับเครื่องคอมพิวเตอร์ เพื่อวัดค่าของมอสเฟสได้ โดยกราฟจะแสดงผลบนกราฟบนหน้าจอ โดยระบุอยู่ในช่วงที่เรากำหนดไว้

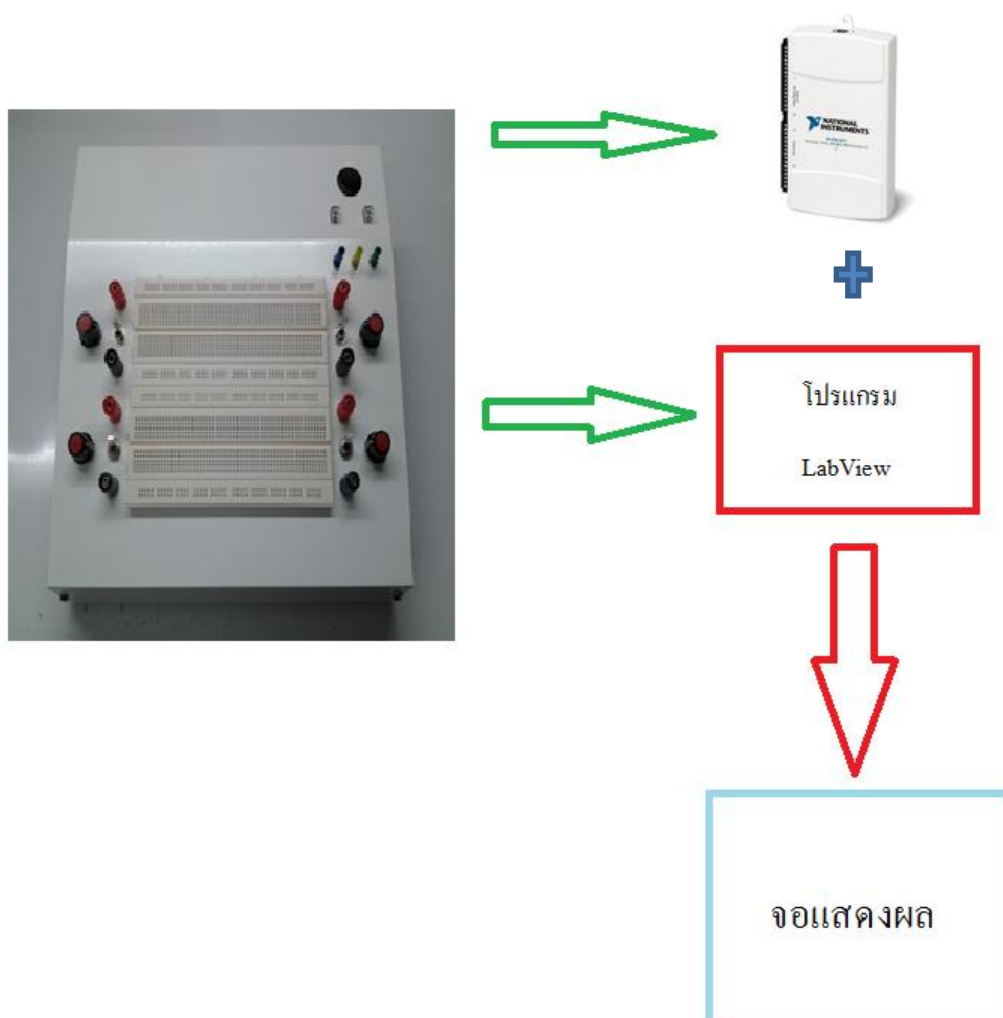
4.4 ชุดตรวจวัดอุปกรณ์สารกึ่งตัวนำ



รูปที่ 4.3.11 ชุดตรวจวัดอุปกรณ์สารกึ่งตัวนำ

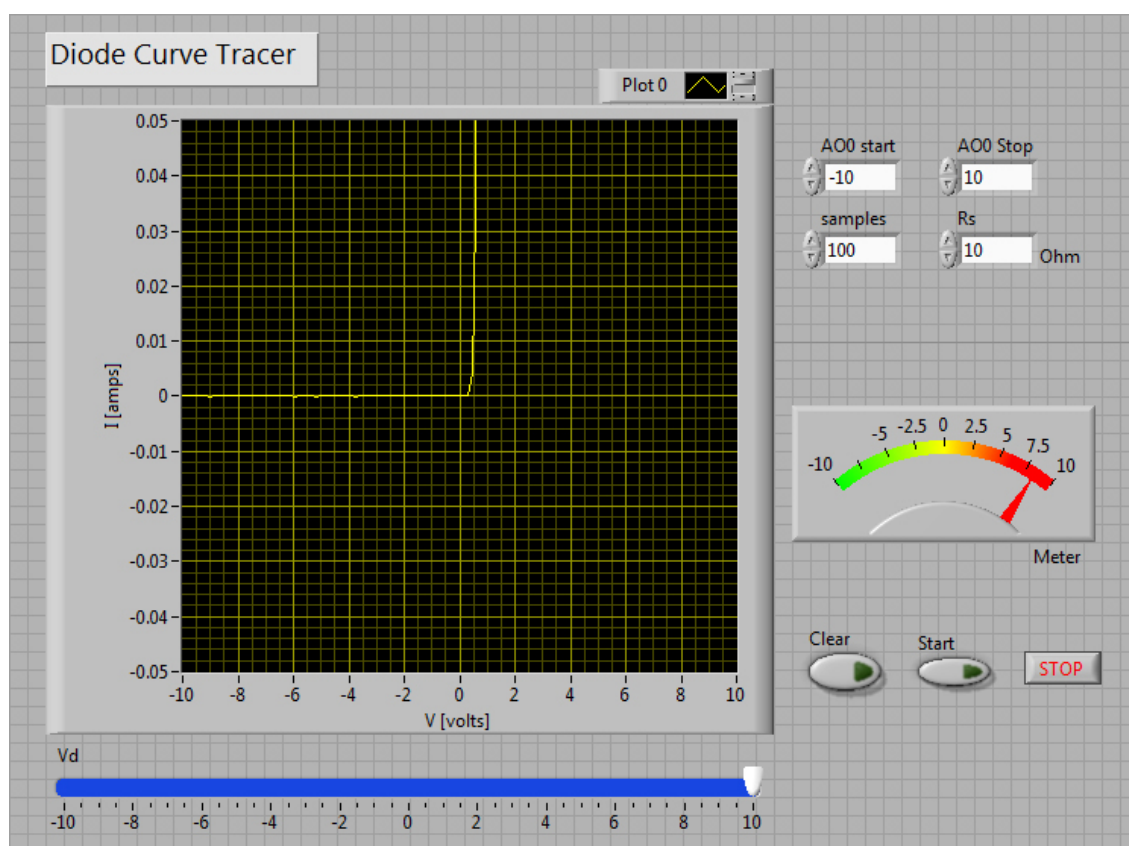
จากรูปที่ 4.1 เป็นชุดตรวจอุปกรณ์สารกึ่งตัวนำ ซึ่งภายนอกประกอบไปด้วย โปรโตบอร์ด มีพื้นที่สำหรับ เสียบอุปกรณ์สารกึ่งตัวนำจะนำมาใช้วัด และประกอบด้วย ขาเสียบเอาต์พุต ซึ่งสามารถต่อเข้ากับ จอแสดงผล ซึ่งแสดงค่าอุปกรณ์สารกึ่งตัวนำที่นำมาใช้วัด ออกทาง จอคอมพิวเตอร์

ส่วนภายในตัวกล่องตรวจวัดอุปกรณ์สารกึ่งตัวนำ ซึ่งจะประกอบไปด้วย โปรแกรม LabView และกล่อง NI USB-6211 ดังรูป 4.3.12



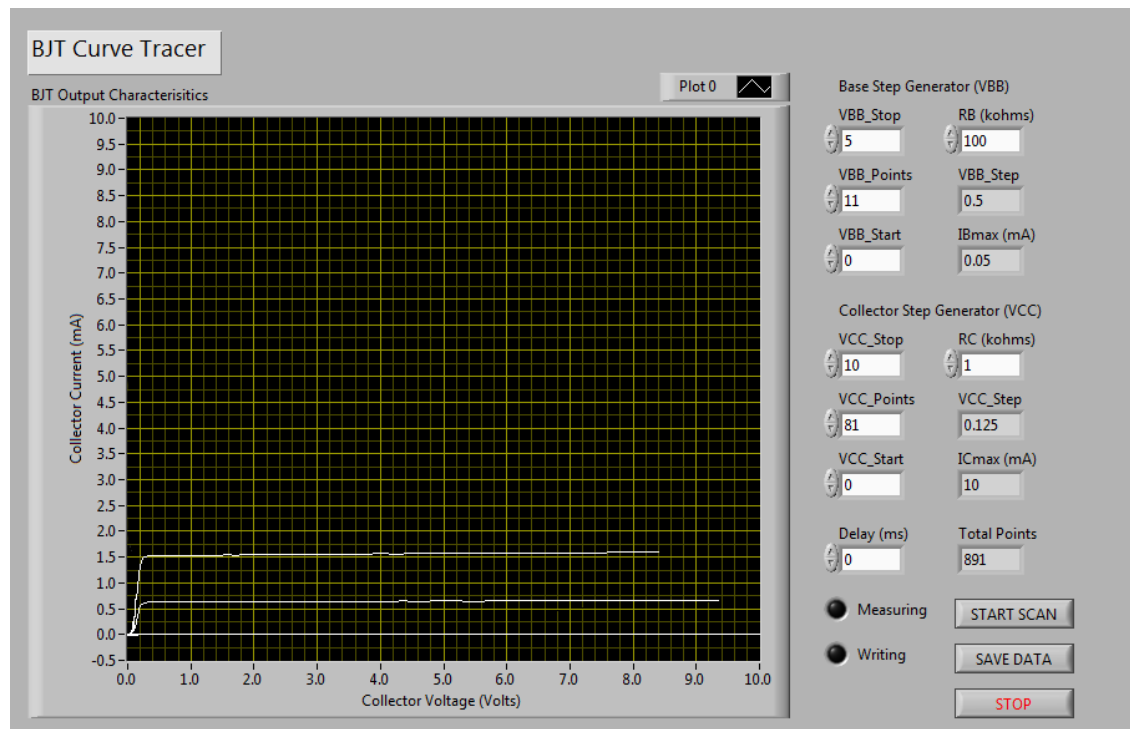
บทที่ 5

ผลการทดลอง



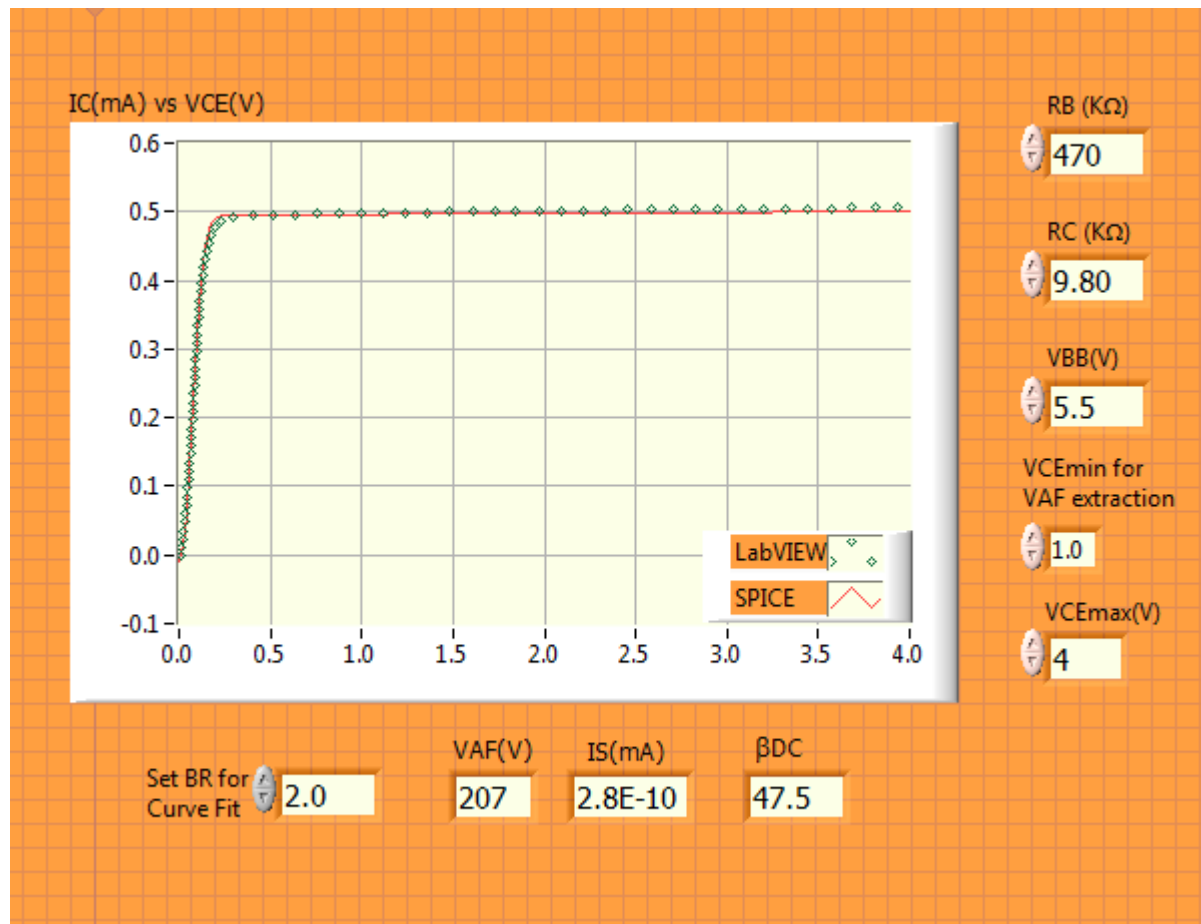
รูปที่ 5.1.1 Front Panel ของชุดคำสั่งวงจรไดโอด

จากผลการทดลองการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำประเภทไดโอด เมื่อทำการจ่ายค่าแรงดันจาก -10 ถึง 10 ให้แก่อิโอด ผลการทดลองก็ออกมาตามรูปที่ 5.1.1 ซึ่งทำให้รู้ว่าไดโอดมีคุณสมบัติในการกรองแรงดันให้มีแรงดันไม่เกิน 0.7 โวลต์



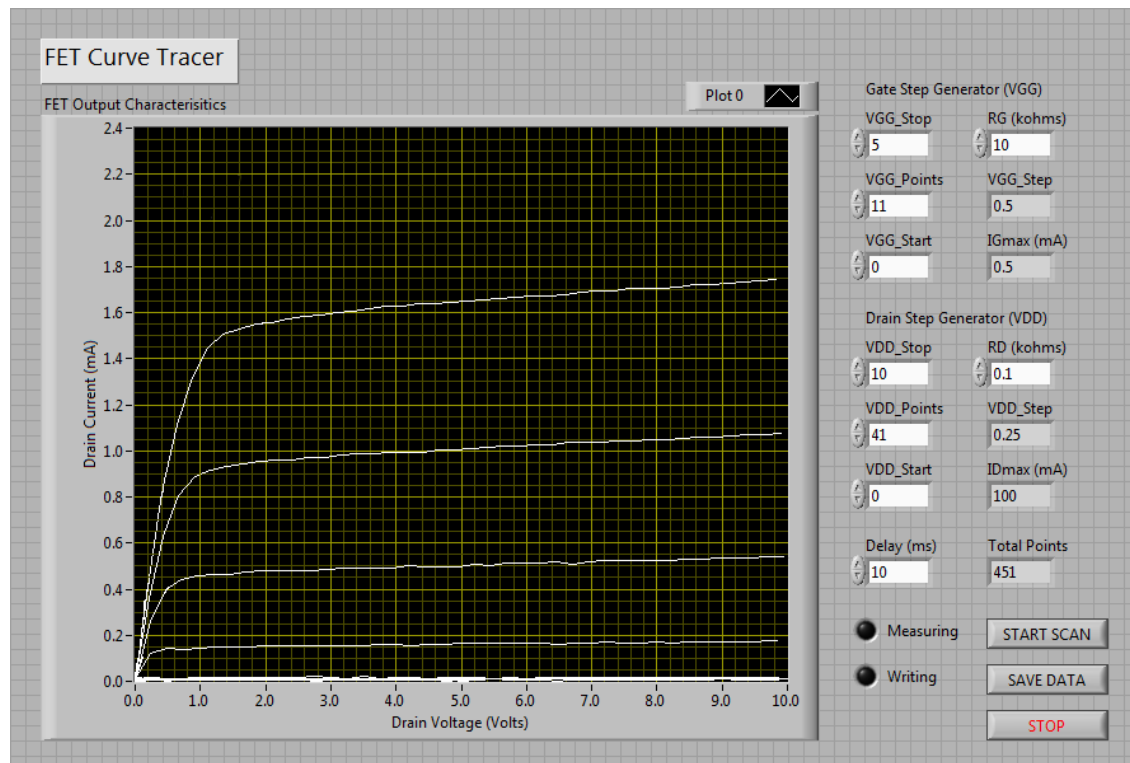
รูปที่ 5.1.2 Front Panel ของชุดคำสั่งวงจรทรานซิสเตอร์

จากการทดลองการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำประเภททรานซิสเตอร์ เมื่อทำการจ่ายค่า VBB จาก 0 ถึง 5 กำหนดให้ค่า BB Point เท่ากับ 11 กำหนดให้ตัวต้านทานมีค่าเท่ากับ 100 kohms. จะได้ค่า IBmax มีค่าเท่ากับ 0.05 mA. อีกส่วนหนึ่งจะทำการจ่ายค่า VCC จาก 0 ถึง 10 กำหนดให้ค่า VCC Point เท่ากับ 81 กำหนดให้ตัวต้านทานมีค่าเท่ากับ 1 kohms จะได้ค่า ICmax มีค่าเท่ากับ 10 ซึ่งผลการทดลองก็จะได้ผลตามรูปที่ 5.1.2



รูปที่ 5.1.3 Front Panel ของ I_C (mA) vs V_{CE} (V)

จากการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำประเภททรานซิสเตอร์ เมื่อทำการไบอัสที่ขาเบส V_{BB} 0 - 5.5 voltage จะได้กราฟของกระแส I_C กับโวลท์ V_{CE} โดยที่ได้ค่าของ R_B 470 K.ohm R_C 9.8 K ohm โดยค่า V_{CEmax} 4 Volt VAF 207 Volt กระแสอิ่มตัว I_S $2.8 E-10$ mA โดยที่ค่าอัตราขยายไฟฟ้าตรงหรือค่าเบต้า 47.5 ซึ่งผลตามรูปที่ 5.1.3



รูปที่ 5.1.4 Front Panel ของชุดคำสั่งวงจรมอสเฟต

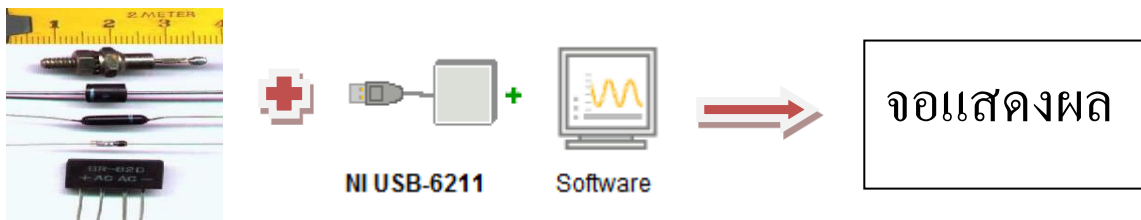
จากการทดลองการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำประเภทมอสเฟต เมื่อทำการจ่ายค่า VGG จาก 0 ถึง 5 กำหนดให้ค่า VGG Point มีค่าเท่ากับ 11 กำหนดให้ตัวต้านทานมีค่าเท่ากับ 10 kohms ซึ่งจะได้ค่า IGmax เท่ากับ 0.5 mA. อีกส่วนหนึ่งจะทำการจ่ายค่า VDD จาก 0 ถึง 10 กำหนดให้ค่า VDD Point มีค่าเท่ากับ 41 และกำหนดตัวต้านทานมีค่าเท่ากับ 0.1 kohms ซึ่งจะได้ค่า IDmax มีค่าเท่ากับ 100 mA. ซึ่งผลการทดลองก็จะได้ตามรูปที่ 5.1.4

บทที่ 6

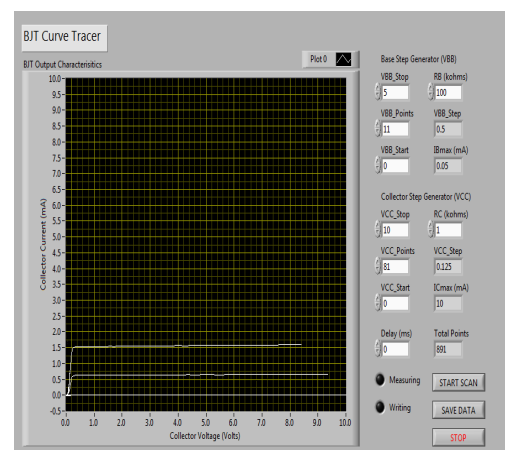
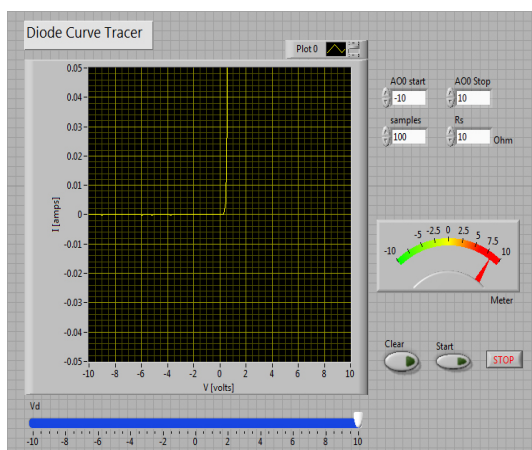
สรุปและวิจารณ์ผลการทดลอง

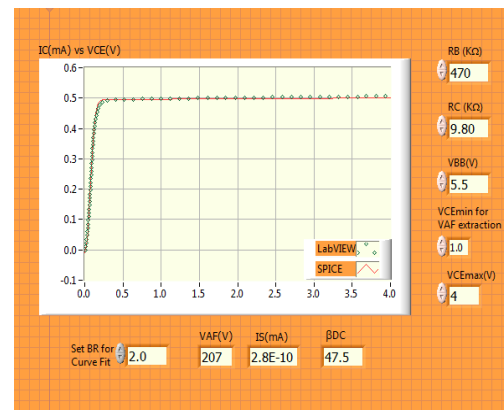
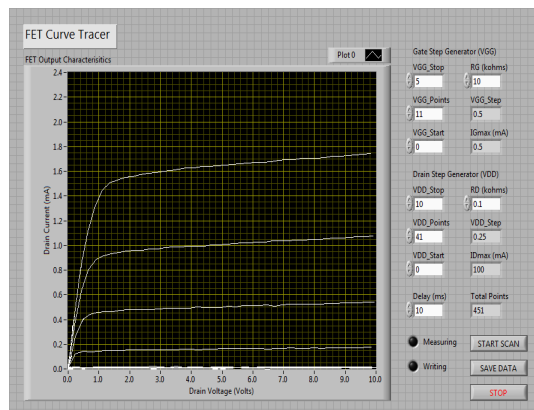
สรุปผลการทดลอง

จากการทดลอง การวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำ ได้มีการนำเอาอุปกรณ์สารกึ่งตัวนำมาทดลอง กับกล่องเครื่องมือวัด ซึ่งต่อกับ software (โปรแกรม LabView) แล้วแสดงผลออกทางหน้าจอคอมพิวเตอร์



อุปกรณ์สารกึ่งตัวนำ ที่ใช้วัดและทำการวิเคราะห์มี 3 ประเภท คือ ไดโอด, ทรานซิสเตอร์ และมอสเฟส ซึ่งมีอุปกรณ์ที่ช่วยอำนวยความสะดวกในการวัด คือกล่องเครื่องมือวัด National Instruments รุ่น NI USB-6211 ซึ่งผลของการทดลองที่ได้จากการทดลองมีค่าที่ออกมาเป็นค่าจริง ดังรูป ต่อไปนี้





ในการทดลองครั้งนี้ได้มีการเรียนรู้เกี่ยวกับการเขียนโปรแกรมLabview ซึ่งวงจรที่เขียนเสร็จแล้ว จะนำไปวัดค่ากับเครื่องวัด NI USB-6211 ซึ่งทำงานควบคู่ไปกับคอมพิวเตอร์ ค่าต่างๆ ที่ออกมาจากการวัดของเครื่องวัด NI USB-6211 จะมีค่าที่เป็นจริงและมีกราฟแสดงออกมาจากหน้าจอคอมพิวเตอร์ ซึ่งค่าต่างๆที่ได้และกราฟที่ได้นั้นได้ตามที่ต้องการ จึงทำให้ในการวัดอุปกรณ์สารกึ่งตัวนำนั้น สะดวก มีประสิทธิภาพมากขึ้น และยังสามารถวัดสารกึ่งตัวนำได้ทีละหลายๆตัว การทดลองนี้สามารถนำไปใช้ในการทำงาน, การเรียนการสอน, นำไปใช้ในห้องปฏิบัติการต่างๆได้ และการทดลองนี้สามารถที่จะพัฒนาให้ดีขึ้นไป

ในการทำการทดลองที่ผ่านมา จะเห็นได้ว่า ถ้าผู้ทดลอง มีความเชี่ยวชาญ ในการใช้โปรแกรม Labview จะทำให้เป็นผลดี ในการทำการทดลองและสามารถ นำความรู้นี้ไปประยุกต์ใช้กับการวัดสารกึ่งตัวนำชนิดอื่นได้ ถึงแม้ว่า การศึกษาการเขียนโปรแกรม Labview จะค่อนข้างใช้เวลาในการฝึกฝน แต่ถ้ามีความเชี่ยวชาญแล้ว ก็จะนำไปประยุกต์ใช้ประโยชน์ได้ ในอีกหลายๆด้าน

วิจารณ์ผลการทดลอง

จากโครงการการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำ ซึ่งเป็นเพียงการเริ่มต้นการพัฒนาของการศึกษาและการเรียนรู้การเขียนโปรแกรมเพื่อที่จะควบคุมอุปกรณ์สารกึ่งตัวนำได้ และสามารถที่จะพัฒนาไปได้อีกไกล ซึ่งมีประโยชน์มากทางการศึกษาและการทดลองต่างๆ

บรรณานุกรม

- [1] <http://th.wikipedia.org/wiki/ไอโอด>
- [2] http://www.cpe.ku.ac.th/~yuen/204471/device/diode_transistor/diode.htm
- [3] รศ.ดร.สมเกียรติ ศุกเดช “เซมิคอนดักเตอร์ ดีไวซ์” คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [4] รศ.ดร.เต็มพงษ์ เพ็ชรกุล “อุปกรณ์สารกึ่งตัวนำ (Semiconductor Devuce)” คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [5] <http://www.ni.com>
- [6] เจริญ เพชรมุณี “เรียนลัด Labview Software for Measurement and Automation”
- [7] กิจไพบูลย์ ชิวพันธุ์ศรี “การออกแบบแอปพลิเคชันในระบบกราฟิกด้วย Labview ซอฟต์แวร์อุตสาหกรรมเพื่อการวัด วิเคราะห์ ทดสอบ และควบคุมแบบอัตโนมัติ”
- [8] <http://th.wikipedia.org/wiki/มอสเฟส>
- [9] <http://th.wikipedia.org/wiki/ทรานซิสเตอร์>
- [10] http://www.sptc.ac.th/prapruet/devicesweb/books/book_6.htm

ภาคผนวก

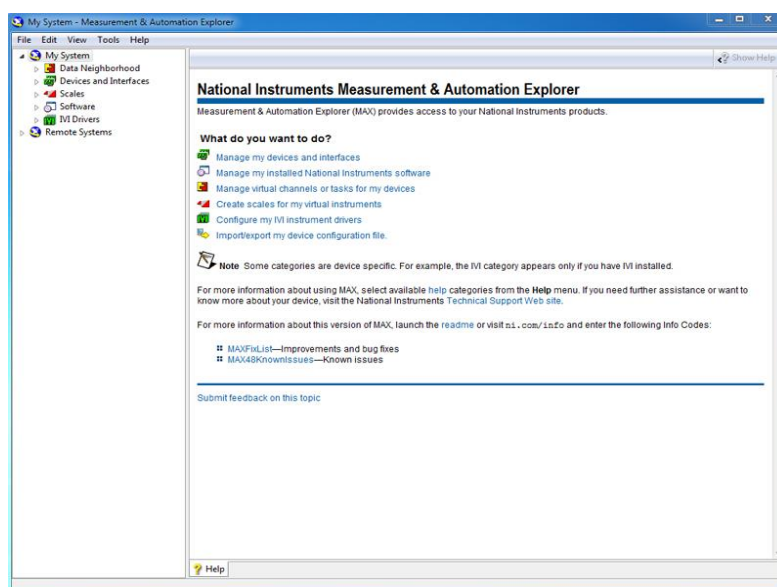


เครื่องวัด NI USB-6211

ในการทำโครงการเรื่องการวัดและวิเคราะห์อุปกรณ์สารกึ่งตัวนำ เราได้ใช้เครื่องวัดที่มีชื่อว่า “NI USB-6211” ซึ่งเป็นเครื่องมือวัดที่บริษัท NATIONAL INSTRUMENTS ได้สร้างและพัฒนาออกมา เครื่องมือวัด NI USB-6211 สามารถใช้งานร่วมกับเครื่องคอมพิวเตอร์ผ่านทางโปรแกรม Measurement & Automation เพื่อลงข้อมูลของ Data Acquisition ของเครื่องมือวัดและนำไปใช้ในโปรแกรม Labview ต่อไป



ไอคอนของโปรแกรม Measurement & Automation



หน้าหลักของโปรแกรม Measurement & Automation

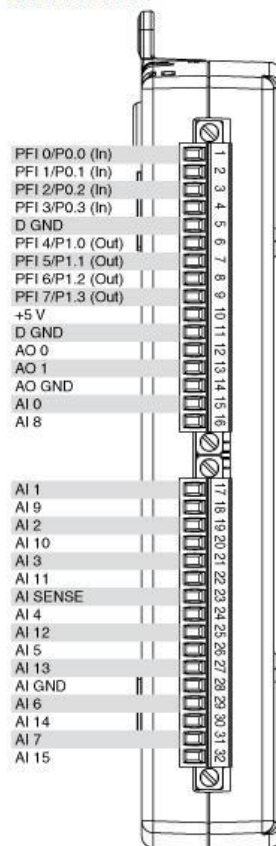


เครื่องมือวัด National Instruments รุ่น NI
USB-6211

- 16 analog inputs (16-bit, 250 kS/s)
- 2 analog outputs (16-bit, 250 kS/s); 4 digital inputs; 4 digital outputs; 32-bit counters
- Bus-powered USB for high mobility; built-in signal connectivity
- NI signal streaming for sustained high-speed data streams over USB; OEM version available
- Compatible with LabVIEW, LabWindows™/CVI

Measurement Studio

NI USB-6211



- NI-DAQmx driver software and NI LabVIEW SignalExpress LE interactive data-logging

Default NI-DAQmx Counter Terminals

Counter/Timer Signal	Default Pin Number	Signal Name
CTR 0 SRC	1	PFI 0
CTR 0 GATE	2	PFI 1
CTR 0 AUX	1	PFI 0
CTR 0 OUT	6	PFI 4
CTR 0 A	1	PFI 0
CTR 0 Z	3	PFI 2
CTR 0 B	2	PFI 1
CTR 1 SRC	4	PFI 3
CTR 1 GATE	3	PFI 2
CTR 1 AUX	4	PFI 3
CTR 1 OUT	7	PFI 5
CTR 1 A	4	PFI 3
CTR 1 Z	2	PFI 1
CTR 1 B	3	PFI 2
FREQ OUT	8	PFI 6

Pin out เครื่องมือวัด National Instruments รุ่น NI USB-6211

NATIONAL INSTRUMENTS USB-6211

NI USB-6211



- 16 analog inputs (16-bit, 250 kS/s)
- 2 analog outputs (16-bit, 250 kS/s); 4 digital inputs; 4 digital outputs; 2 32-bit counters
- Bus-powered USB for high mobility; built-in signal connectivity
- NI signal streaming for sustained high-speed data streams over USB; OEM version available
- Compatible with LabVIEW, LabWindows™/CVI, and Measurement Studio for Visual Studio .NET
- NI-DAQmx driver software and NI LabVIEW SignalExpress LE interactive data-logging software

Data Acquisition ทำหน้าที่ประมวลผลความหมายหรือเปลี่ยนสัญญาณในลักษณะ Analog ให้มาอยู่ในรูปของ digital signal เพื่อประโยชน์ในการตีความหมายและใช้ในการควบคุม หน้าที่ของ DAQ boards อาจจะเป็นการอ่านสัญญาณ analog (A/D Conversion) การสร้างสัญญาณ อันalog (D/A conversion) เขียนและอ่านสัญญาณ เพื่อเชื่อมต่อกับ Transducer

การติดต่อระหว่างคอมพิวเตอร์และ transducer จึงเป็นเรื่องสำคัญสำหรับคอมพิวเตอร์โดยปกติแล้ว สามารถที่จะติดต่อสื่อสารกับอุปกรณ์ภายนอกได้โดยการผ่าน Input/Output Board (I/O Board) ซึ่ง I/O board นี้จะมีหลายแบบแต่แบบที่สำคัญและสามารถเชื่อมต่อโดยผ่านคำสั่งของ LabVIEW ได้ทันที

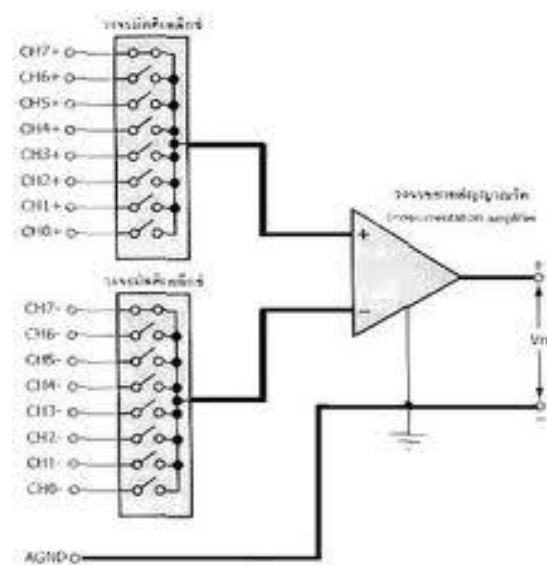
แหล่งสัญญาณทางไฟฟ้า แหล่งกำเนิดสัญญาณประเภทแรงดันไฟฟ้าเรียกว่า Voltage Source เป็นสัญญาณที่จะถูกวัดด้วยตัวอุปกรณ์ DAQ โดยสัญญาณใน NI USB-6211 จะแบ่งเป็นสัญญาณสองชนิด

แหล่งสัญญาณแบบต่อกราวด์ (Grounded Source) กับแหล่งสัญญาณที่ไม่ต่อกราวด์ หรือแบบลอย (Floating Source)

การออกแบบในส่วนของ USB-DAQ ทางกลุ่มโครงการได้เลือกใช้ NI USB-6/ ซึ่งจะเป็น DAQ โดยมีจำนวนช่อง Analog Input สำหรับวัดแรงดันที่อยู่ในช่วง ± 10 โวลต์ จำนวน 16 ช่องสัญญาณ ที่ความเร็ว Sampling Rate 250kS/s, Analog Output 2 ช่องสัญญาณ, Digital I/O 12 ช่องสัญญาณ และ 32-bit Counter สำหรับการนับ 1 ตัว โดยใช้ไฟเลี้ยงจากสาย USB จึงไม่ต้องต่อสายแพเวอร์ให้ยุ่งยาก พร้อมช่องต่อสายสัญญาณแบบขั้วสกรูในตัว ขนาดเล็กสามารถนำไปใช้ได้ในงานทั่วไป

NI USB 6211 ทำหน้าที่รับข้อมูลจากกล่องชุดวัดสารกึ่งตัวนำของอุปกรณ์ทางอิเล็กทรอนิกส์ที่ทำการประมวลผลด้วยโปรแกรม LabVIEW แล้วมายังชุดควบคุมฮาร์ดแวร์ ซึ่งเป็นสัญญาณแบบดิจิทัล ในขณะที่เดียวกันการส่งข้อมูลกลับไปยังกล่องชุดวัดสารกึ่งตัวนำของอุปกรณ์ทางอิเล็กทรอนิกส์ ต้องใช้วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล (A/D) ซึ่ง NI USB 6211 นี้เป็นอุปกรณ์ที่สามารถแปลงสัญญาณแรงดัน ± 10 V เป็นสัญญาณดิจิทัลได้

ประเภทของสัญญาณของ USB-6211 เป็นสัญญาณแบบ Differential หรือ DIFF เป็นการวัดสัญญาณขาบวก ขาลบโดยไม่ได้ต่อวัดกับจุดอ้างอิง โดยจะต่อขาบวกช่องสัญญาณ CHx+ และขาลบที่ช่อง CHx- โดยที่ AIGND (Analog input ground) เป็นกราวด์ของระบบวัด



รูปที่ 5. ลักษณะของการวัดโดยไม่เทียบกับจุดอ้างอิง

I/O Connector Signal Descriptions

AI GND กราวด์อินพุตแบบแอนะล็อกซึ่งปลายเหล่านี้เป็นจุดอ้างอิงสำหรับ Single-ended

AI ในโหมด RSE และการไบอัสกระแสที่จุดอ้างอิงของการวัด จุดกราวด์ทั้งสาม— AI GND, GND

อ่าว และ D GND ซึ่งเป็น เชื่อมต่ออุปกรณ์

AI<0...31> ช่องอินพุตแบบแอนะล็อก 0-31 — สำหรับการวัดแบบ single-ended สัญญาณแต่ละเป็นช่องทางที่แรงดันอินพุตแบบแอนะล็อก ในโหมด RSE, AI GND เป็นจุดอ้างอิงของสัญญาณ ในโหมด NRSE จุดอ้างอิงอื่นๆแต่ละ AI<0...31> สัญญาณคือ AI SENSE สำหรับสิ่งที่แตกต่างของการวัด AI 0 และ AI 8 มีอินพุตบวก และลบของช่องป้อนข้อมูลส่วนที่แตกต่างแบบแอนะล็อก 0

ช่องสัญญาณอินพุต

<AI 1, AI 9>, <AI 2, AI 10>, <AI 3, AI 11>, <AI 4, AI 12>, <AI 5, AI 13>, <AI 6, AI 14>,

<AI 7, AI 15>, <AI 16, AI 24>, <AI 17, AI 25>, <AI 18, AI 26>, <AI 19, AI 27>,

<AI 20, AI 28>, <AI 21, AI 29>, <AI 22, AI 30>, <AI 23, AI 31>

AI SENSE

Analog Input Sense อยู่ในโหมด NRSE , จุดอ้างอิงอื่นๆ AI<0...31> คือสัญญาณ AI SENSE

AO <0..1>

Analog Output Channels 0 to 1 คือขาของแหล่งจ่ายโวลต์เอาต์พุต AO 0 ถึง 1

AO GND

Analog Output Ground หรือ **AO GND** คือจุดอ้างอิงสำหรับ AO 0-1 ทั้งสามจุดอ้างอิง AI

GND AO GND และ D GND

D GND

Digital Ground หรือ D GND เป็น supplies จุดอ้างอิงสำหรับ PFI<0 ถึง 15> /P0/P1

มีโวลต์ +10 โวลต์ ทั้งสามจุดอ้างอิงของกราวด์ AI GND AO GND และ D GND และจุดเชื่อมต่อของอุปกรณ์

PFI

Programmable Function Interface or Static Digital Input มีช่องสัญญาณ 0 ถึง 7 ขั้วอื่นๆ ของ

PFI สามารถต่อแหล่งจ่ายภายนอกได้สำหรับ AI, AO หรือ counter/timer inputs