บทที่ 3

การออกแบบและประกอบสร้าง

เครื่องวิเคราะห์ดีสชาร์จบางส่วนในอุปกรณ์ไฟฟ้าแรงสูงที่ทำการออกแบบสร้างเป็นแบบระบบดิจิตอลทำงานร่วมกับคอมพิวเตอร์ เพื่อใช้ความสามารถของคอมพิวเตอร์ในการประมวลผลสัญญาณดิจิตอล การแสดงผล การบันทึกข้อมูล และการวิเคราะห์ PD นำเทคโนโลยี FPGA (Field Programmable Gates Array) มาใช้ โดยการโปรแกรมวงจรภาคดิจิตอลไว้ในไอซี FPGA เพียงตัวเดียว ทำให้สามารถออกแบบวงจรทั้งภาคแอนะลอกและภาคดิจิตอลไว้บนแผงวงจรขยาย (Expansion card) แผงเดียวกันได้ ระบบวัดทั้งหมดประกอบด้วยอุปกรณ์รับสัญญาณ แผงวงจรที่ติดตั้งอยู่บนแผงวงจรหลัก (Mother board) ของคอมพิวเตอร์ที่ใช้ Intel 200 MHz Pentium pro processor เป็นตัวประมวลผล ใช้งานร่วมกับซอฟต์แวร์และอุปกรณ์ที่เกี่ยวข้องอีกเล็กน้อย

3.1 ส่วนประกอบของเครื่องวิเคราะห์และการทำงาน

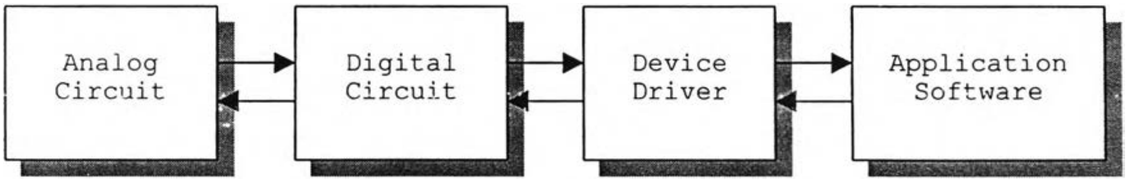
เครื่องวิเคราะห์ดีสชาร์จบางส่วนประกอบด้วย

- วงจรภาคแอนะลอก สำหรับการรับและประมวลผลสัญญาณ PD ที่เกิดขึ้นในวงจรทดสอบอุปกรณ์ไฟฟ้าแรงสูงและสัญญาณแรงดันทดสอบ

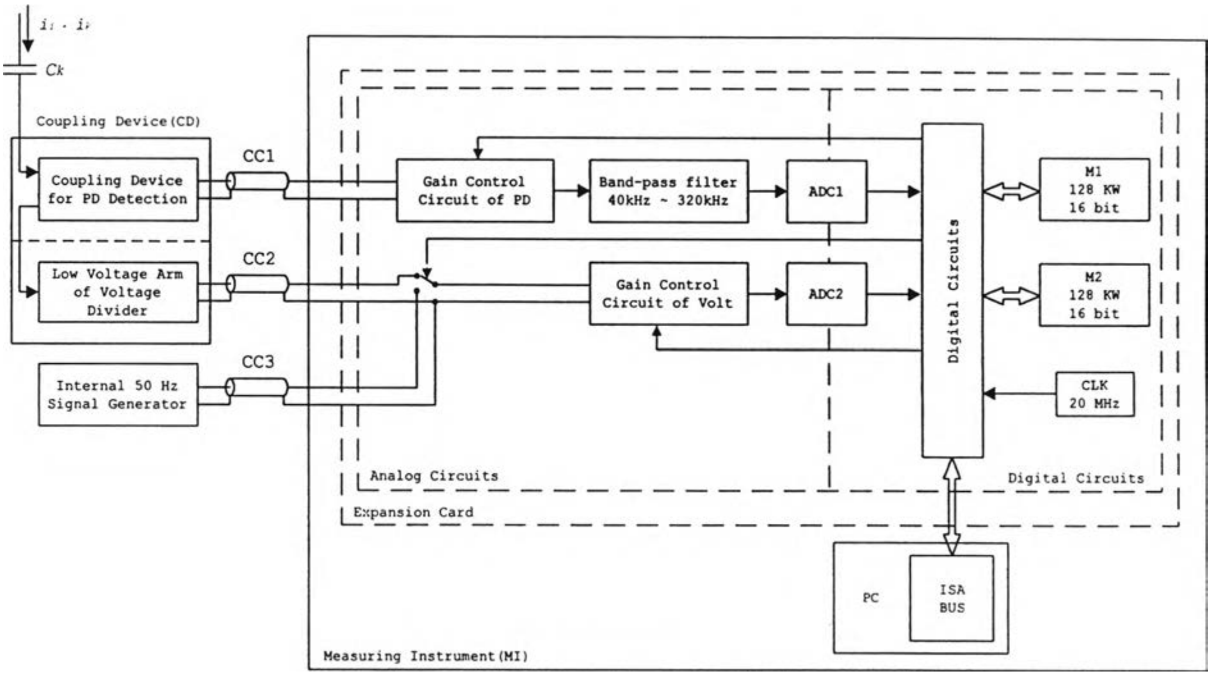
- วงจรภาคดิจิตอล สำหรับการประมวลผลสัญญาณรวมทั้งควบคุมการทำงานของวงจร

- ดีไวซ์ไดรเวอร์ (Device driver) สำหรับการอินเตอร์เฟสระหว่างวงจรอิเล็กทรอนิกส์กับคอมพิวเตอร์

- โปรแกรมประยุกต์ (Application software) สำหรับแสดงผลการตรวจวัด และวิเคราะห์สาเหตุของ PD ดังไดอะแกรมของระบบวัดแสดงในรูปที่ 3.1 และ 3.2 ตามลำดับ



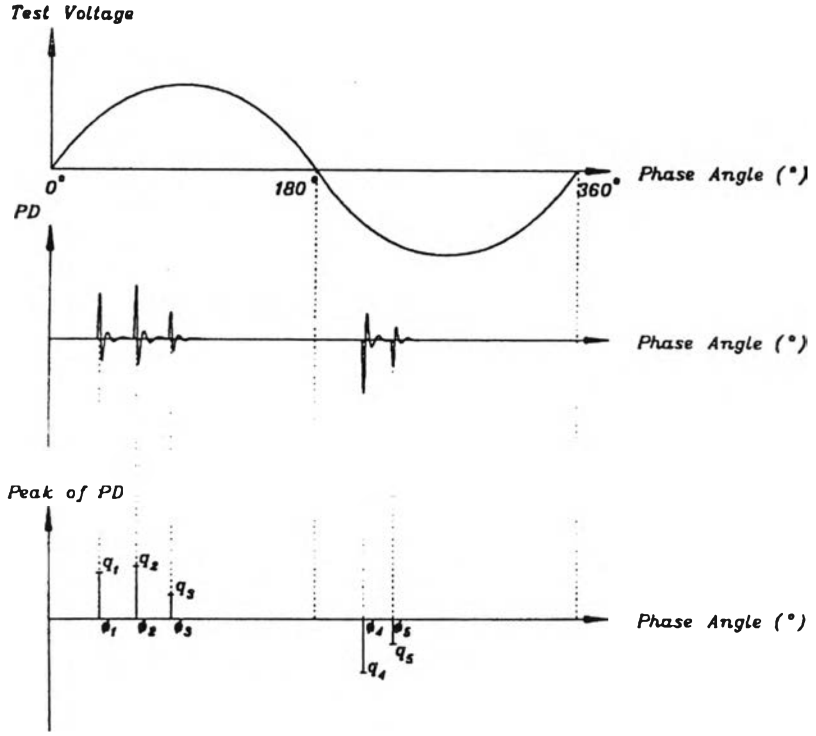
รูปที่ 3.1 ส่วนประกอบของเครื่องวิเคราะห์ PD



รูปที่ 3.2 ไดอะแกรมของระบบวัด

การทำงานของเครื่องวิเคราะห์ PD ที่ออกแบบสร้างมีดังต่อไปนี้

1) การทำงานของวงจรตรวจจับ PD จะเป็นการทำงานแบบเวลาจริง (real time) คือ คอมพิวเตอร์จะรับข้อมูลค่ายอดของพัลส์ PD และมุมเฟส (Phase Angle, ϕ) ที่เกิด PD ทุกพัลส์ที่เกิดในแต่ละไซเคิล (หรือคาบเวลาของความถี่จากแหล่งจ่ายแรงดันทดสอบ) ดังรูปที่ 3.3 เพื่อนำไปคำนวณขนาด PD และแสดงผลให้ทันทุกไซเคิล การทำงานเริ่มจากอุปกรณ์รับสัญญาณรับกระแสพัลส์ i_i ที่เกิดเนื่องจาก PD เข้ามา สัญญาณที่ได้จะส่งไปที่วงจรควบคุมอัตราขยาย PD และตัวกรองผ่านแถบช่วงความถี่ 40 กิโลเฮิรตซ์ ถึง 320 กิโลเฮิรตซ์ ตามลำดับ และได้เอาต์พุตจากการแปลงสัญญาณจากแอนะลอกเป็นดิจิตอลด้วยอัตราชักตัวอย่าง 20 ล้านตัวอย่างต่อวินาที ข้อมูลที่ได้จะส่งไปที่วงจรในภาคดิจิตอลเพื่อวิเคราะห์ค่ายอดของพัลส์และมุมเฟสที่เกิด PD แล้วจึงนำข้อมูลที่ได้ทั้งหมดในแต่ละไซเคิลไปเก็บไว้ชั่วคราวที่หน่วยความจำ M1 และ M2 โดยที่หน่วยความจำทั้งสองจะรับและส่งข้อมูลสลับกันไปทุกๆไซเคิล เช่น ถ้าในไซเคิลแรกหน่วยความจำ M1 รับข้อมูลที่มาจาก ADC1 หน่วยความจำ M2 จะโหลดข้อมูลออกเพื่อส่งให้คอมพิวเตอร์ ในทำนองเดียวกัน เมื่อถึงไซเคิลถัดไปหน่วยความจำ M2 จะรับข้อมูลที่มาจาก ADC1 หน่วยความจำ M1 จะโหลดข้อมูลออกเพื่อส่งให้คอมพิวเตอร์สลับกันไปทุกๆไซเคิลโดยใช้สัญญาณนาฬิกาที่มาจากวงจรสร้างสัญญาณซิงโครไนซ์ (synchronize) เป็นสัญญาณอ้างอิงในการให้จังหวะการทำงานของหน่วยความจำทั้งสอง



รูปที่ 3.3 การวิเคราะห์หาขนาดค่ายอดของ PD และมุมเฟสที่เกิด PD

อย่างไรก็ตาม เนื่องจากการวัด PD ตามวงจรที่กล่าวมาแล้วให้ข้อมูลเฉพาะค่ายอดและมุมเฟสที่เกิด PD เท่านั้น แต่มิได้ให้ข้อมูลเกี่ยวกับลักษณะรูปร่างที่แท้จริงของสัญญาณเอาต์พุตที่เกิดจาก PD และสัญญาณรบกวนที่ปะปนเข้ามาในขณะทำการวัด จึงได้ออกแบบวงจรเพิ่มเติมเพื่อให้สามารถรับสัญญาณดังกล่าวมาวิเคราะห์ได้โดยการแปลงสัญญาณแอนะลอกเป็นดิจิตอลด้วยอัตราชักตัวอย่าง 5 ล้านตัวอย่างต่อวินาที เก็บไว้ในหน่ายความจำ M1 จำนวน 128 KWord ก่อนส่งต่อให้คอมพิวเตอร์นำสัญญาณไปวิเคราะห์ต่อไป

2) วงจรสร้างสัญญาณซิงโครไนซ์จากแรงดันทดสอบและแรงดันภายใน เป็นวงจรส่วนที่อยู่ใน FPGA ใช้ในการสร้างสัญญาณนาฬิกาเพื่อให้จังหวะในการถ่ายโอนข้อมูลไปยังคอมพิวเตอร์และกำหนดตำแหน่งมุมอ้างอิงให้กับวงจรตรวจจับ PD สาเหตุที่ต้องมีวงจรทั้งสองเนื่องจากในระหว่างการปรับเทียบแรงดันทดสอบเป็นศูนย์ การสร้างสัญญาณซิงโครไนซ์จากแรงดันทดสอบจะไม่สามารถทำได้ ดังนั้นในระหว่างการปรับเทียบจึงต้องสร้างสัญญาณซิงโครไนซ์จากแรงดันภายใน แต่แรงดันภายในอาจมีมุมเฟสไม่ตรงกับของแรงดันทดสอบหากเลือกเฟสแรงดันที่จ่ายให้วงจรทดสอบและระบบวัดไม่ตรงกัน ดังนั้นในการจ่ายแรงดันเพื่อทดสอบจึงต้องใช้สัญญาณจากแรงดันทดสอบเป็นมุมอ้างอิง

3) วงจรวัดแรงดันทดสอบ ใช้ในการวัดระดับแรงดันโดยรับแรงดันจากภาคแรงดันต่ำของโวลเตจดิไวเดอร์ซึ่งเป็นไฟฟ้ากระแสสลับ มาแปลงเป็นค่าทางดิจิตอลโดย ADC2 เพื่อนำ

ค่าสัญญาณที่ได้ส่งให้คอมพิวเตอร์นำไปแสดงผลต่อไป แต่เนื่องจากโวลเตจดิไวเดอร์มีราคาแพง ดังนั้นโครงการวิจัยจึงได้ออกแบบให้ตัวเก็บประจุคับปลิงเป็นส่วนหนึ่งของโวลเตจดิไวเดอร์เพื่อใช้วัดแรงดันทดสอบโดยทางอ้อม

3.2 อุปกรณ์รับสัญญาณและวงจรภาคแอนะลอก

อุปกรณ์รับสัญญาณและวงจรภาคแอนะลอกประกอบด้วยวงจรย่อยคือ วงจรตรวจจับ PD วงจรสร้างสัญญาณซิงโครไนซ์และวัดแรงดันทดสอบ และวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล

3.2.1 วงจรตรวจจับดีสชาร์จบางส่วน

วงจรตรวจจับ PD จะประกอบด้วยอุปกรณ์รับสัญญาณ วงจรขยายและตัวกรองผ่านแถบ (Band–pass Filter) เพื่อใช้ในการอินทิเกรตกระแสพัลส์ i_i ที่เกิดในวงจรทดสอบ การออกแบบอุปกรณ์รับสัญญาณปกติจะใช้เป็นวงจร RLC ที่มีรูปแบบการต่อเป็นตัวกรองผ่านแถบอันดับ 2[2] ซึ่งถือว่าเป็นตัวกรองที่มีประสิทธิภาพต่ำในการลดทอนสัญญาณรบกวนนอกแถบผ่าน จะทำให้วงจรขยายภาคแรกๆมีโอกาสรับแรงดันอินพุตสูงเกินไป การแก้ปัญหาวิธีหนึ่งทำได้โดยการออกแบบอุปกรณ์รับสัญญาณให้มีคุณสมบัติเป็นตัวกรองผ่านแถบหรือผ่านสูงที่มีอันดับสูงขึ้น เพื่อเพิ่มประสิทธิภาพในการกรองสัญญาณรบกวน

หลักเกณฑ์ในการออกแบบอุปกรณ์รับสัญญาณและตัวกรองผ่านแถบเพื่อใช้ในการอินทิเกรตกระแสพัลส์มีรายละเอียดดังนี้

1) อิมพีแดนซ์ถ่ายโอน (Transfer Impedance, Z(f)) มาตรฐาน IEC [5] กำหนดให้ Z(f) มีความถี่จำกัดต่ำ (Lower limit frequency, –6 dB) f_1 ความถี่จำกัดสูง (Higher limit frequency, –6 dB) f_2 และแบนด์วิดต์(-6 dB) Δf ของเครื่องตรวจจับ PD แบบแถบกว้างคือ

$$30kHz \le f_1 \le 100kHz$$
$$f_2 \le 500kHz$$
$$100kHz \le \Delta f \le 400kHz$$

มาตรฐานได้กำหนดเฉพาะช่วงความถี่ของ Z(f) ซึ่งมีคุณสมบัติเป็นตัวกรองผ่านแถบ แต่มิได้กำหนดอันดับของตัวกรองไว้ ตัวกรองอันดับสูงจะมีประสิทธิภาพในการลดทอนสัญญาณรบกวนที่อยู่นอกช่วงความถี่แถบผ่านได้ดีกว่า แต่ถ้าอันดับของตัวกรองสูงเกินไปจะทำให้สัญญาณเอาต์พุตไม่เป็นแบบ “α-response” ดังนั้นการออกแบบจึงต้องคำนึงถึงข้อนี้ด้วย

2) การเลือกอินพุตอิมพีแดนซ์ของอุปกรณ์รับสัญญาณ เนื่องจากมาตรฐาน IEC มิได้กำหนดค่าอินพุตอิมพีแดนซ์ไว้ แต่ในการตรวจจับ PD ในเคเบิลแรงสูงนั้นต้องเลือก $R_{mi} > Z_0$ เพื่อทำให้เกิด mismatch ทำให้ได้สัญญาณสะท้อนมีขั้วเดียวกัน และ R_{mi} ต้องไม่สูงจนเกินไปจนทำให้สัญญาณเอาต์พุตเป็นแบบ " β-response" ในทางปฏิบัติเลือก

$$Z_0 < R_{mi} < \frac{1}{2\pi f_2 C_m} \tag{3.1}$$

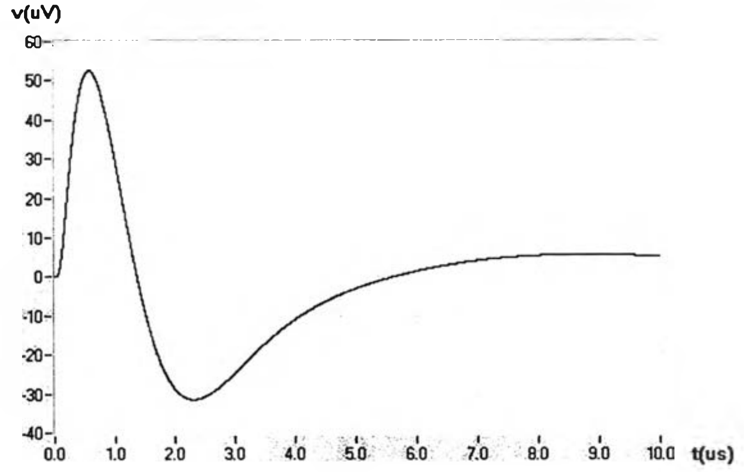
โดยที่ C_m คือ ค่าความจุไฟฟ้ารวมประสิทธิผลที่ขนานกับ R_{mi}

R_{mi} คือ อินพุตอิมพีแดนซ์ ในช่วงแถบผ่านจะเสมือนเป็นตัวต้านทาน

f_2 คือ ความถี่จำกัดบน

จากเงื่อนไขข้างต้นจึงได้ออกแบบโดยประนีประนอมเงื่อนไขต่างๆ โดยใช้ซอฟต์แวร์[46] ช่วยในการคำนวณ พบเงื่อนไขที่เหมาะสมคือเลือก R_{mi} = 75 โอห์ม เลือกอุปกรณ์รับสัญญาณเป็นวงจรกรองไม่ไวงานบัตเตอร์เวิร์ท (Butterworth filter) แบบผ่านแถบอันดับที่ 4 ที่มีช่วงแถบผ่าน (-3dB) 40 กิโลเฮิรตซ์ ถึง 2 เมกะเฮิรตซ์ และเลือกตัวกรองของเครื่องมือวัดเป็นวงจรกรองไวงานบัตเตอร์เวิร์ทแบบผ่านแถบอันดับ 4 เช่นกัน ที่มีช่วงความถี่ (-3dB) 40 กิโลเฮิรตซ์ ถึง 320 กิโลเฮิรตซ์ ตามลำดับ [47] จะทำให้ได้ Z_f ที่มี f_1 = 40 กิโลเฮิรตซ์ และ f_2 = 420 กิโลเฮิรตซ์ โดยรายละเอียดของการคำนวณในการออกแบบวงจรกรอง และแผนภาพผังงาน(Schematic diagram) ของวงจรกรองจะแสดงในภาคผนวก

การวิเคราะห์ลักษณะเอาต์พุตทำได้โดยป้อนกระแสอิมพัลส์ที่มีค่าประจุ 1 พิโคคูลอมบ์ที่อินพุตของอุปกรณ์รับสัญญาณดังรูปที่ 3.4 ก) จะได้สัญญาณเอาต์พุตดังรูปที่ 3.4 ข)



รูปที่ 3.4 ลักษณะเอาต์พุตของเครื่องตรวจจับ PD

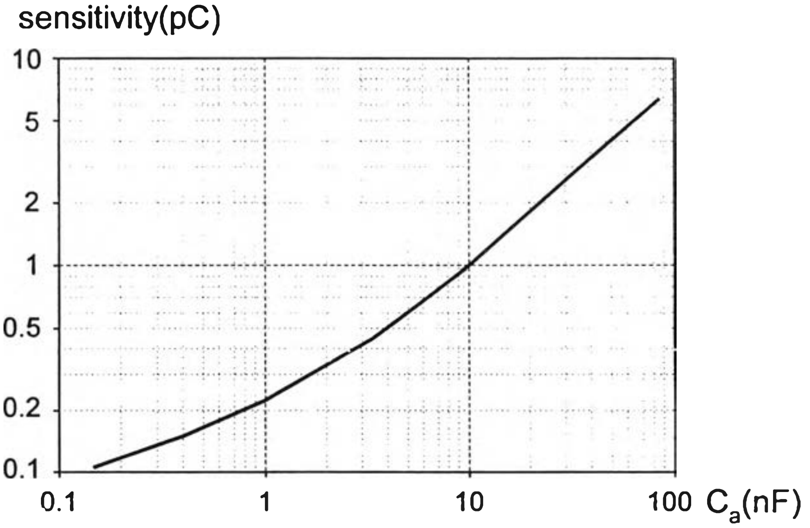
ในการคำนวณด้วยซอฟต์แวร์ได้กำหนดให้อัตราขยายเท่ากับ 1 ดังนั้นสามารถคำนวณแฟกเตอร์สำหรับการตอบสนองแบบอิมพัลส์ได้

$$\xi = \text{ค่ายอด } v_2 / \text{ อัตราขยาย}$$
$$= \frac{52.5x10^{-6}}{1}$$
$$= 52.5 \text{ ไมโครโวลต์ต่อพิโคคูลอมป์}$$

ถ้าสัญญาณรบกวนรวมที่อินพุตเท่ากับ 0.987 ไมโครโวลต์ (ดูรายละเอียดการคำนวณในภาคผนวก ก) และกำหนดให้ $C_a = 100$ พิโคฟารัด $C_k = 1$ นาโนฟารัด จากสมการ (2.20) สามารถคำนวณความไวได้

$$q_{\min} = 5\left(\frac{C_k + C_a}{\xi C_k}\right)v_{n(rms)}$$
$$= 5\left(\frac{1x10^{-9} + 0.1x10^{-9}}{52.5x10^{-6}x1x10^{-9}}\right)x0.987x10^{-6}$$
$$= 0.103 \qquad \text{พิโคคูลอมป์}$$

ความไวของเครื่องวิเคราะห์ PD สำหรับวัสดุทดสอบทั่วไปเมื่อวัสดุทดสอบมีค่าความจุตั้งแต่ 100 พิโคฟารัด ถึง 100 นาโนฟารัด โดยใช้ $C_k = 1$ นาโนฟารัด แสดงดังรูปที่ 3.5



รูปที่ 3.5 ความไวเมื่อตรวจจับวัสดุทดสอบทั่วไปเมื่อใช้ตัวเก็บประจุคับปลิง 1 นาโนฟารัด

3.2.2 วงจรวัดแรงดันทดสอบ

วงจรวัดแรงดันทดสอบจะรับสัญญาณไฟฟ้ากระแสสลับ 220 โวลต์ 50 เฮิรตซ์ ผ่านวงจรลดทอนแรงดันเพื่อลดแรงดันลงให้เหลือ 20-100 มิลลิโวลต์ และแปลงค่าเป็นดิจิตอลด้วยอัตราชักตัวอย่าง 50,000 ตัวอย่างต่อวินาทีเพื่อให้ในแต่ละรูปคลื่นแรงดันมีข้อมูลประมาณ 1,000 ข้อมูล โดยมีความละเอียด 12 บิต เพื่อให้ FPGA นำไปใช้สร้างสัญญาณนาฬิกาความถี่ 50 เฮิรตซ์ที่ซิงโครไนซ์กับแหล่งจ่ายภายใน (internal synchronizing signal) ส่วนวงจรที่ใช้สร้างสัญญาณซิงโครไนซ์กับแรงดันภายนอก (external synchronizing signal) จะมีลักษณะเดียวกันกับวงจรสร้างสัญญาณซิงโครไนซ์กับแรงดันภายใน ต่างกันตรงที่วงจรจะรับสัญญาณแรงดันทดสอบจากภาคแรงดันต่ำของโวลเตจดิไวเดอร์ที่ต่ออยู่กับวงจรทดสอบ และขนาดของสัญญาณจะนำไปคำนวณและแสดงผลเป็นค่าแรงดันแบบรากของกำลังสองเฉลี่ยของแหล่งจ่ายไฟฟ้าแรงสูง การวัดแรงดันทดสอบทำได้โดยการรับแรงดันจากเอาต์พุตของวงจรขยายสัญญาณแรงดัน และส่งให้ ADC2 แปลงค่าสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลให้คอมพิวเตอร์นำไปแสดงผลและเก็บข้อมูลต่อไป การแสดงผลบนจอภาพจะแสดงรูปสัญญาณแรงดันทดสอบแบบเวลาจริงด้วยความเร็ว 50 เฟรมต่อวินาที และแสดงผลเป็นตัวเลข 3 ครั้งต่อวินาที

เครื่องวิเคราะห์ PD นี้ใช้ตัวเก็บประจุคับปลิงขนาด 1 นาโนฟารัดเป็นตัวเก็บประจุภาคแรงสูงของโวลเตจดิไวเดอร์ร่วมกับตัวเก็บประจุภาคแรงต่ำภายในอุปกรณ์รับสัญญาณที่มีขนาด 40 ไมโครฟารัดเพื่อลดทอนแรงดันจากแหล่งจ่ายแรงดันทดสอบลง 40,000 เท่า เนื่องจาก ADC2 ที่ใช้มีพิกัดแรงดันขาเข้า ± 3 โวลต์ ทำให้สามารถวัดแรงดันค่ายอดจากแหล่งจ่ายได้ 40,000 x 3 =120 กิโลโวลต์ หรือคิดเป็นค่ารากของกำลังสองเฉลี่ยของแรงดันทดสอบได้ 85 กิโลโวลต์ ซึ่งเมื่อเผื่อค่าความคลาดเคลื่อนของอุปกรณ์ที่เกี่ยวข้องอีกประมาณ 20 เปอร์เซ็นต์เพื่อความปลอดภัยของวงจรอิเล็กทรอนิกส์ จะได้แรงดันทดสอบสูงสุดของเครื่องเท่ากับ 70 กิโลโวลต์

3.2.3 วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล

การแปลงสัญญาณแอนะลอกจากวงจรตรวจจับ PD และวงจรวัดแรงดันทดสอบเป็นสัญญาณดิจิตอล มีรายละเอียดการออกแบบดังนี้

1) วงจรตรวจจับ PD ADC1 ที่เลือกใช้คือเบอร์ AD9225AR มีคุณสมบัติโดยสรุปคือ พิสัยเต็มสเกล ± 2 โวลต์, ความละเอียด 12 บิต, อัตราชักตัวอย่างไม่เกิน 25 ล้านตัวอย่างต่อวินาที ถ้าต้องการไม่ให้ Aliasing Error มีผลกับสัญญาณ PD เกินกว่า 0.5 บิต จาก 11 บิต (เนื่องจากสัญญาณมีขั้ว) หรือ 0.488 มิลลิโวลต์ จะต้องมี Anti-aliasing prefilter ในการจำกัดแบนด์วิดต์ของสเปกตรัมของสัญญาณ PD ซึ่งคำนวณได้คือ

$$A = -20\log\left|\frac{H(f_{stop})}{H(f_{pass})}\right| \quad (3.2)$$

โดยที่ A คือ อัตราการลดทอนความถี่นอกช่วง Anti-aliasing prefilter

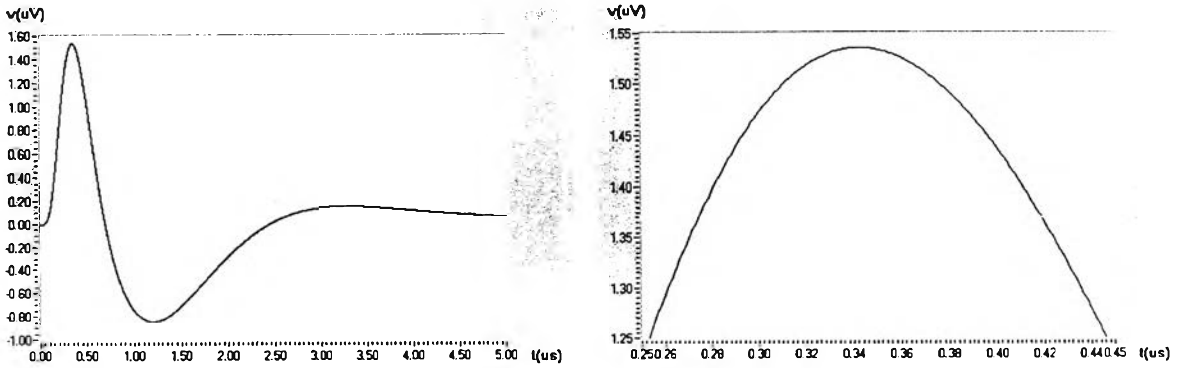
$H(f_{stop})$ คือค่าฟังก์ชันถ่ายโอนที่ความถี่หยุด (stop band)

$H(f_{pass})$ คือค่าฟังก์ชันถ่ายโอนที่ความถี่ตัดสูง (pass band)

จากสมการ (3.2) จะได้ A = -20log(0.488x10^{-3}/2) =-72.25 dB และจากคุณสมบัติทางความถี่ของระบบวัดที่ได้กล่าวไว้ในหัวข้อ 3.2.1 จะสามารถคำนวณได้ว่า f_{stop} (-72.25 dB) เท่ากับ 6.39 เมกะเฮิรตซ์ ดังนั้นอัตราชักตัวอย่าง f_s คำนวณได้จาก [48]

$$f_s = f_{stop} + f_{pass} \quad (3.3)$$

แทนค่า f_{stop} = 6390 กิโลเฮิรตซ์ และ f_{pass} = 320 กิโลเฮิรตซ์ ลงในสมการ (3.2) จะได้ f_s = 6390 + 320 = 6710 กิโลเฮิรตซ์ ดังนั้นต้องใช้อัตราชักตัวอย่างไม่ต่ำกว่า 6.71 ล้านตัวอย่างต่อวินาที แต่อย่างไรก็ตามเนื่องจากการตรวจจับค่ายอดของ PD จากสัญญาณที่แปลงค่าเป็นดิจิตอลแล้วจะให้ค่าไม่ละเอียดเพียงพอในเชิงเวลา ทำให้ค่ายอดที่ได้ผิดพลาดมากเกินไป จากการให้ซอฟต์แวร์ช่วยคำนวณพบว่าเมื่อวัสดุทดสอบเป็นเคเบิลแรงสูงจะให้ผลตอบที่มีการเปลี่ยนแปลงของค่ายอดเร็วที่สุด ดังแสดงรูปคลื่นสัญญาณและส่วนขยายบริเวณส่วนยอดของรูปคลื่นในรูปที่ 3.6ก) และ 3.6ข) ตามลำดับ และความผิดพลาดสูงสุดของค่ายอดที่อัตราชักตัวอย่างต่างๆ แสดงในตารางที่ 3.1 เนื่องจากมาตรฐาน IEC 60270(1998) ได้กำหนดไว้ว่าระบบวัด PD จะมีความผิดพลาดในการวัดได้ไม่เกิน ± 10 เปอร์เซ็นต์หรือ 1 พิโคคูลอมบ์โดยถือตามค่าที่มากกว่า จึงเลือกใช้อัตราชักตัวอย่าง 20 ล้านตัวอย่างต่อวินาที ซึ่งจะเกิดความผิดพลาดสูงสุดน้อยกว่า 2 เปอร์เซ็นต์ เพื่อให้มีผลน้อยต่อความผิดพลาดโดยรวมที่มาตรฐานยอมรับ



ก) รูปคลื่นทั้งหมด ข) ส่วนขยายบริเวณยอด

รูปที่ 3.6 ลักษณะสัญญาณ PD เมื่อวัสดุทดสอบเป็นเคเบิลแรงสูง

ตารางที่ 3.1 ความผิดพลาดสูงสุดของค่ายอดที่อัตราชักตัวอย่างต่างๆ

ความผิดพลาดสูงสุด(%)	1	2	3	5	10	15
อัตราชักตัวอย่าง(MHz)	22.7	16.7	13.2	10.0	7.1	5.8

สัญญาณ PD ที่ตรวจจับได้จะมีสัญญาณรบกวนปะปนเข้ามาด้วยในขณะทำการทดสอบในห้องปฏิบัติการ การพิจารณาข้อมูลที่ตรวจจับโดยใช้อัตราชักตัวอย่าง 5 ล้านตัวอย่างต่อวินาทีจะได้รูปคลื่นของสัญญาณเอาต์พุตของภาคแอนะลอก ทำให้สามารถวิเคราะห์ปัญหาต่างๆ ที่เกิดขึ้นได้ สาเหตุที่เลือกชักตัวอย่างด้วยอัตราดังกล่าวเนื่องจากหน่วยความจำมีขนาดจำกัด ผลการลดอัตราชักตัวอย่างนี้จะทำให้ความละเอียดลดลงเหลือประมาณ 10 บิต ซึ่งถือว่ายังเพียงพออยู่ และเมื่อนำข้อมูลไปแสดงผลจะอินเตอร์โพเลต (Interpolate) สัญญาณด้วยแฟกเตอร์ 1:8 (8 เท่า) โดยใช้เทคนิคการอินเตอร์โพเลต ซึ่งเป็นวิธีการหนึ่งในการประมวลผลสัญญาณดิจิตอลที่ใช้สร้างสัญญาณส่วนที่ขาดไปเนื่องจากการควอนไทซ์ทางเวลา (Time quantization) ให้กลับคืนมาโดยที่สเปกตรัมของสัญญาณในโดเมนความถี่ยังคงเดิม เพื่อให้เห็นรายละเอียดของรูปคลื่นสัญญาณได้ชัดเจนเสมือนว่าใช้อัตราชักตัวอย่าง 5 x 8 = 40 ล้านตัวอย่างต่อวินาที คำนวณได้จากสมการ[49]

$$Y_i(n) = \sum_{k=-M}^{M-1} d_i(k) \cdot X(n-k) \qquad (3.4)$$

$$d_i(k) = sinc(\pi(k + \frac{i}{L})) \qquad (3.5)$$

โดยที่ X(.) คือ ข้อมูลที่อัตราชักตัวอย่างเดิม

$Y_i(.)$ คือ ข้อมูลลำดับที่ i ที่นำมาแทรกเพิ่มอัตราชักตัวอย่างขึ้นเป็น L เท่า

n คือ ตำแหน่งข้อมูลที่อัตราชักตัวอย่างเดิม

L คือ จำนวนเท่าของอัตราชักตัวอย่างที่เพิ่มขึ้น

i คือ ลำดับที่แทรกระหว่างข้อมูลของอัตราเร็วเดิมมีค่าตั้งแต่ 1..L-1

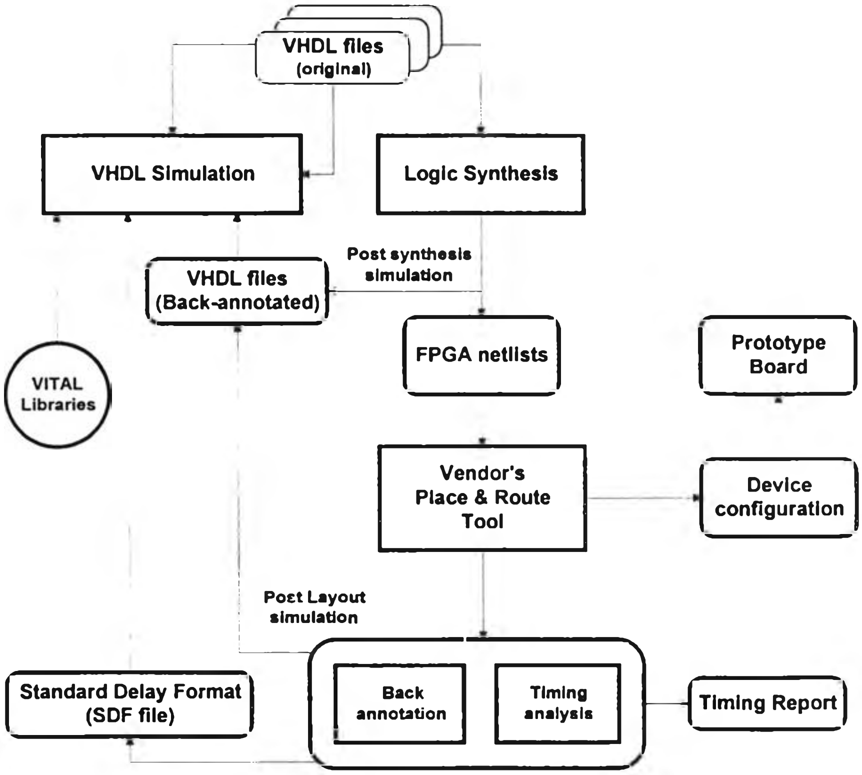
M คือ จำนวนพจน์ของข้อมูลที่นำมาคำนวณค่าของข้อมูลแทรก ณ. ตำแหน่ง i

3.3 วงจรภาคดิจิตอล

วงจรภาคดิจิตอลทั้งหมดยกเว้นหน่วยความจำจะโปรแกรมไว้ใน FPGA โดยใช้ VHDL ซึ่งเป็นภาษาที่ใช้อธิบายการทำงานของฮาร์ดแวร์แทนการวาด Schematic สำหรับการพัฒนาวงจรดิจิตอล การพัฒนาวงจรโดยใช้ VHDL สามารถทำได้รวดเร็ว สามารถสื่อความหมายได้ดี และง่ายในการแก้ไขจุดผิดพลาด (Debug)[50]

3.3.1 การออกแบบวงจรดิจิตอลโดยใช้ภาษา VHDL

การออกแบบวงจรดิจิตอลโดยใช้ภาษา VHDL มีขั้นตอนดังรูปที่ 3.7 คือ เริ่มจากการศึกษาและกำหนดรูปแบบการทำงานของวงจร แล้วจึงนำมาอธิบายการทำงานด้วยภาษา VHDL โดยใช้รูปแบบการเขียนในระดับ RTL (Register transfer level) ซึ่งจะแบ่งแผนการทำงานออกเป็น 2 ส่วนหลัก คือ ส่วนถ่ายโอนข้อมูล และส่วนรีจิสเตอร์ (Register) ซึ่งเป็นส่วนที่ทำงานตามสัญญาณนาฬิกา หลังจากเขียนเสร็จแล้วจะใช้โปรแกรมจำลองการทำงานของ VHDL (VHDL simulator) จำลองการทำงานของวงจร เมื่อวงจรทำงานได้ตามต้องการจึงใช้โปรแกรมสังเคราะห์วงจร (Logic synthesis) สังเคราะห์วงจรออกมาเป็น Netlist เช่นเดียวกับการวาด Schematic แล้วนำ Netlist ที่ได้ไปจำลองการทำงานในระดับเกตเพื่อเปรียบเทียบกับระดับ RTL ว่ามีการทำงานตรงกันหรือไม่



รูปที่ 3.7 วิธีการออกแบบด้วยภาษา VHDL

ถ้าตรงกันจึงนำ Netlist นี้ไปทำการ Place & Route เพื่อโปรแกรมลงใน FPGA และทำการวิเคราะห์ทางเวลา (Timing analysis) เพื่อวิเคราะห์ delay ที่เกิดในส่วนต่างๆ FPGA คือในส่วนของเกตและ interconnect ซึ่งได้ผลออกมาในรูปของไฟล์ SDF (Standard delay format) อันเป็นไฟล์มาตรฐานของ delay และไฟล์ VHDL ในระดับเกตที่ตรงกับวงจรที่ได้ภายใน FPGA จากนั้นนำไฟล์ไปจำลองการทำงานอีกครั้งหนึ่ง เพื่อทดสอบการทำงานที่สภาวะใกล้เคียงกับเมื่อโปรแกรมลง FPGA แล้วมากที่สุด ก่อนจะโปรแกรมลงใน FPGA เพื่อทดสอบการทำงานกับฮาร์ดแวร์จริงเป็นขั้นตอนสุดท้าย

3.3.2 โหมดการทำงานของวงจรภาคดิจิตอล

การทำงานของวงจรภาคดิจิตอล เริ่มจาก นำค่าสัญญาณ PD และแรงดันทดสอบที่ได้จาก ADC1 และ ADC2 ตามลำดับมาประมวลผลและบันทึกค่าลงหน่วยความจำซึ่งมี 2 ชุดสลับกันทำงาน โดยขณะที่ชุดหนึ่งกำลังเขียนข้อมูลโดยคอมพิวเตอร์ อีกชุดหนึ่งก็จะอ่านข้อมูลและส่งข้อมูลนั้นไปยังบัส ISA (the Industrial standard architecture) เพื่อนำไปคำนวณและแสดงผลที่คอมพิวเตอร์ทุกคาบแรงดันทดสอบแบบเวลาจริง วงจรดิจิตอลจะเป็นตัวให้จังหวะการถ่ายโอนข้อมูล โดยการสร้างฐานเวลาจากการตรวจจับจุดผ่านศูนย์ (zero crossing) ของสัญญาณแรงดันกระแสสลับ แบ่งการทำงานทั้งหมดออกเป็น 3 โหมดดังนี้

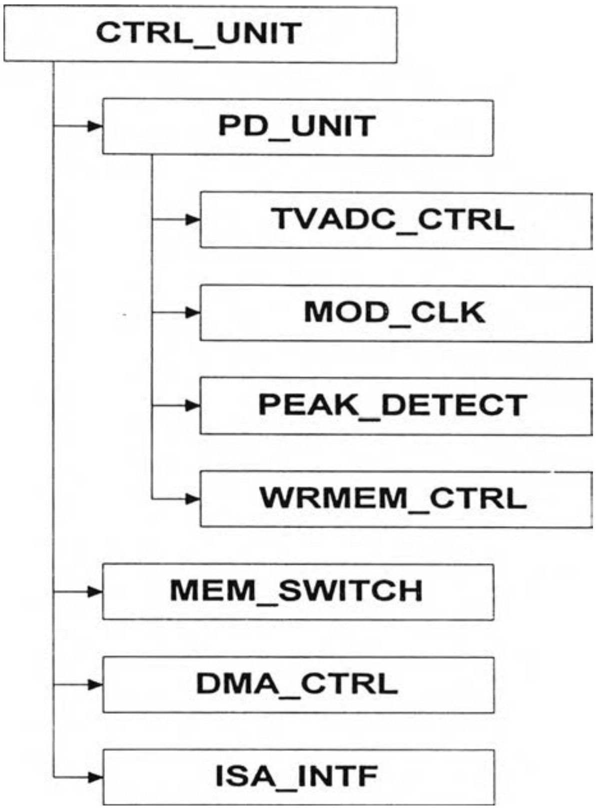
1) โหมดตรวจจับ PD (PD Detection) เป็นการรับข้อมูลจาก ADC1 ด้วยความถี่ 20MHz เพื่อหาค่ายอดใน lock-out time ที่กำหนด แล้วเก็บค่าที่ได้ลงหน่วยความจำไซเคิลละ 1,000 ค่า ในช่วงเวลา 20 มิลลิวินาที(คิดจากคาบเวลาของแรงดันทดสอบที่ความถี่ 50Hz) ซึ่งเก็บพร้อมกับค่าของแรงดันทดสอบที่ได้จาก ADC2 โดยเก็บที่แอดเดรส (address) ถัดจากค่ายอดของ PD ในหน่วยความจำเดียวกัน เมื่อครบหนึ่งไซเคิลแล้วจะทำการส่งข้อมูลที่เก็บไว้ผ่านทางบัส ISA โดยสามารถถ่ายโอนได้ 2 แบบคือ แบบ DMA transfer (Direct memory access transfer) หรือ I/O tranfer (Input/output transfer) ใช้หน่วยความจำจำนวน 2 ชุดในการเก็บและส่งข้อมูล เพื่อสลับกันรับและส่งข้อมูลเพื่อให้ถ่ายโอนข้อมูลได้ต่อเนื่องแบบเวลาจริง

2) โหมดแสดงรูปสัญญาณ (Capture) โดยการรับข้อมูลจาก ADC1 ที่ชักตัวอย่างด้วยความถี่ 5 เมกะเฮิรตซ์ แล้วนำค่าที่ได้มาเก็บที่หน่วยความจำจนครบ 128KWord จากนั้นจึงทำการอินเตอร์รัพต์ (Interrupt) และรอให้คอมพิวเตอร์มาอ่านข้อมูลทางบัส ISA โดยวิธี I/O transfer

3) โหมดวิเคราะห์ PD (PD Analysis) มีวิธีการประมวลผลและเก็บข้อมูลเช่นเดียวกับโหมด PD realtime แต่จะเก็บข้อมูล 250 ไซเคิลต่อเนื่องกันไปในหน่วยความจำทั้ง 2 ชุด แล้วจึงอินเตอร์รัพต์เพื่อส่งข้อมูลเช่นเดียวกับโหมดตรวจจับรูปสัญญาณ

3.3.3 ส่วนประกอบของวงจรภาคดิจิตอล

โครงสร้างของวงจรมีลักษณะเป็นลำดับชั้น ดังรูปที่ 3.8 โดยชั้นบนสุดคือ CTRL_UNIT ซึ่งอธิบายการทำงานของส่วนต่างๆจากชั้นล่างไปยังชั้นบนสุดได้ดังนี้



รูปที่ 3.8 โครงสร้างของวงจรภาคดิจิตอล

1) TVADC_CTRL เป็นวงจรควบคุมการทำงานของ ADC1 โดยเมื่อได้รับการเอ็นเอเบิล (enable) จะส่งสัญญาณ CS_N เป็น 0 ไปที่ ADC1 เพื่อให้เริ่มทำการแปลงแรงดันทดสอบ เมื่อแปลงเสร็จแล้วสัญญาณ BUSY_N จะเป็นลอจิก 1 วงจรนี้จะนำค่าที่ได้ไปรวมกับค่าออฟเซต (offset) ที่ตั้งไว้ แล้วนำไปเก็บที่ ADC_DOUT เพื่อรอการนำไปใช้งาน นอกจากนี้ยังสร้างสัญญาณผ่านศูนย์ส่งออกมาทาง AC_ZCROSS เพื่อให้วงจรทั้งหมดทำงานเข้าจังหวะกับแรงดันทดสอบด้วย

2) MOD_CLK เป็นวงจรสร้างสัญญาณนาฬิกาภายในตามโหมดการทำงาน สำหรับโหมดตรวจจับ PD และโหมดวิเคราะห์ PD สัญญาณ PD_CLK มีความถี่ 20 เมกะเฮิรตซ์และสัญญาณ LOT_CLK มีความถี่ตาม lock-out time ที่กำหนด ส่วนโหมดแสดงรูปสัญญาณ สัญญาณ PD_CLK และ LOT_CLK มีความถี่ 5MHz เท่ากัน

3) PEAK_DETECT เป็นวงจรหาค่ายอดของ PD โดยนำค่าที่ได้จากการชักตัวอย่างมาปรับค่าด้วยค่าออฟเซตและค่าเทรชโฮลด์ (threshold) แล้วนำมาเก็บในบัฟเฟอร์แบบ FIFO (first-in, first-out) เพื่อเก็บค่า PD จำนวน 3 ตำแหน่ง คือ Y1, Y2 และ Y3 โดยมีเงื่อนไขคือ Y2 จะเป็นค่ายอดได้ก็ต่อเมื่อ เครื่องหมาย (sign) ของผลต่างระหว่าง Y2 กับ Y1 ไม่เท่ากับเครื่องหมายของผลต่างระหว่าง Y3 กับ Y2 จากนั้นนำค่ายอดในแต่ละ lock-out time ไปเก็บไว้ที่ PEAK_DOUT ถ้าไม่มีค่ายอดในช่วงเวลานั้นเลยจะให้ค่าของ PEAK_DOUT เป็น 0

4) WRMEM_CTRL เป็นวงจรควบคุมการเขียนข้อมูล PD และ แรงดันทดสอบในหน่วยความจำทั้งสองชุด โดยในโหมดตรวจจับ PD และโหมดวิเคราะห์ PD จะเขียนค่ายอดของ PD ที่ได้จากวงจร PEAK_DETECT ตามด้วยค่าแรงดันทดสอบจากวงจร TVADC_CTRL เมื่อจบช่วง lock-out time ส่วนในโหมดแสดงรูปสัญญาณจะเขียนค่า PD จนเต็มหน่วยความจำ 1 ชุดด้วยความถี่ 5 เมกะเฮิรตซ์ วงจรนี้จะให้สัญญาณครบรอบการทำงานในแต่โหมด เพื่อการส่งข้อมูลกับบัส ISA

5) PD_UNIT เป็นวงจรตรวจจับค่า PD และแรงดันทดสอบ ทำหน้าที่ควบคุมการทำงานร่วมกันของ 4 วงจรข้างต้น

6) MEM_SWITCH เป็นวงจรเลือกหน่วยความจำที่ต้องใช้การอ่านและเขียนข้อมูลในแต่ละโหมด โดยในโหมดตรวจจับ PD จะเลือกหน่วยความจำที่ต้องใช้การเขียนและอ่านค่า PD และแรงดันทดสอบในแต่ละไซเคิลเพื่อถ่ายโอนข้อมูลกับคอมพิวเตอร์สลับชุดกันอย่างต่อเนื่อง ในโหมดแสดงรูปสัญญาณ จะเลือกหน่วยความจำชุดที่ 1 เพื่อเขียนและอ่านค่า PD ส่วนในโหมด วิเคราะห์ PD จะเลือกหน่วยความจำชุดที่ 1 เพื่อเขียนค่า PD และแรงดันทดสอบจำนวน 125 ไซเคิลแล้วสลับไปใช้ชุดที่ 2 อีก 125 cycle แล้วจึงรอการถ่ายโอนข้อมูลไปยังคอมพิวเตอร์

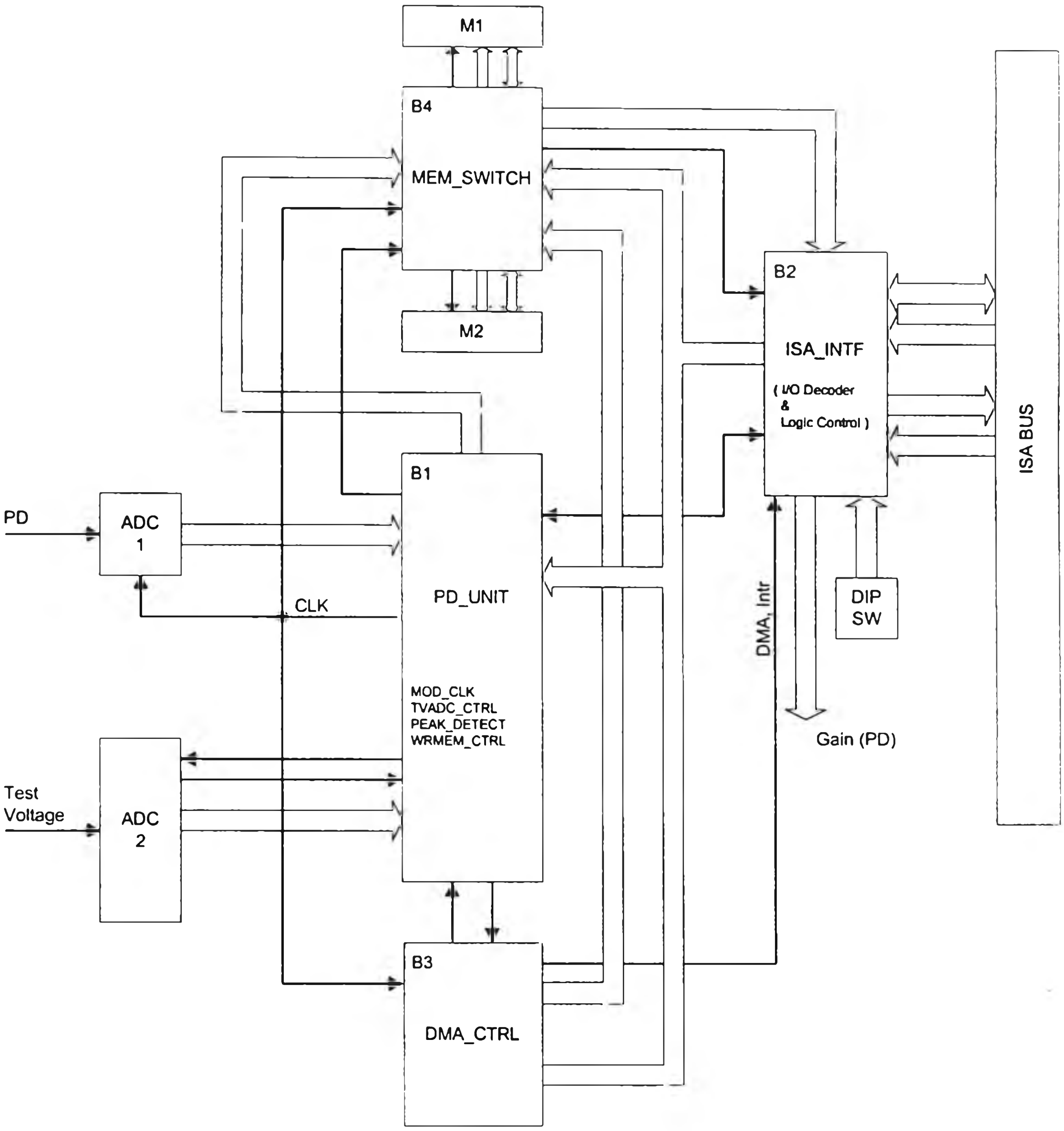
7) DMA_CTRL เป็นวงจรควบคุมการถ่ายโอนข้อมูลจากหน่วยความจำผ่านบัส ISA ทั้งในแบบ DMA transfer หรือแบบ 16-bit I/O transfer ตามโหมดการทำงานและรูปแบบการส่งข้อมูลที่กำหนด โดยในในโหมดตรวจจับ PD สามารถส่งได้ทั้ง 2 แบบ ส่วนในโหมดแสดงรูปสัญญาณ และโหมดวิเคราะห์ PD จะส่งแบบ 16 บิต I/O transfer เท่านั้น

8) ISA_INTF เป็นวงจรควบคุมการติดต่อกับบัส ISA เช่นการถอดรหัส (decode) ตำแหน่งของพอร์ต (port), การอ่านและเขียนค่าที่รีจิสเตอร์ควบคุมการทำงานของเครื่องวิเคราะห์ PD, การสร้างสัญญาณอินเตอร์รัพต์, การสร้างสัญญาณเพื่อส่งข้อมูลแบบ DMA เป็นต้น

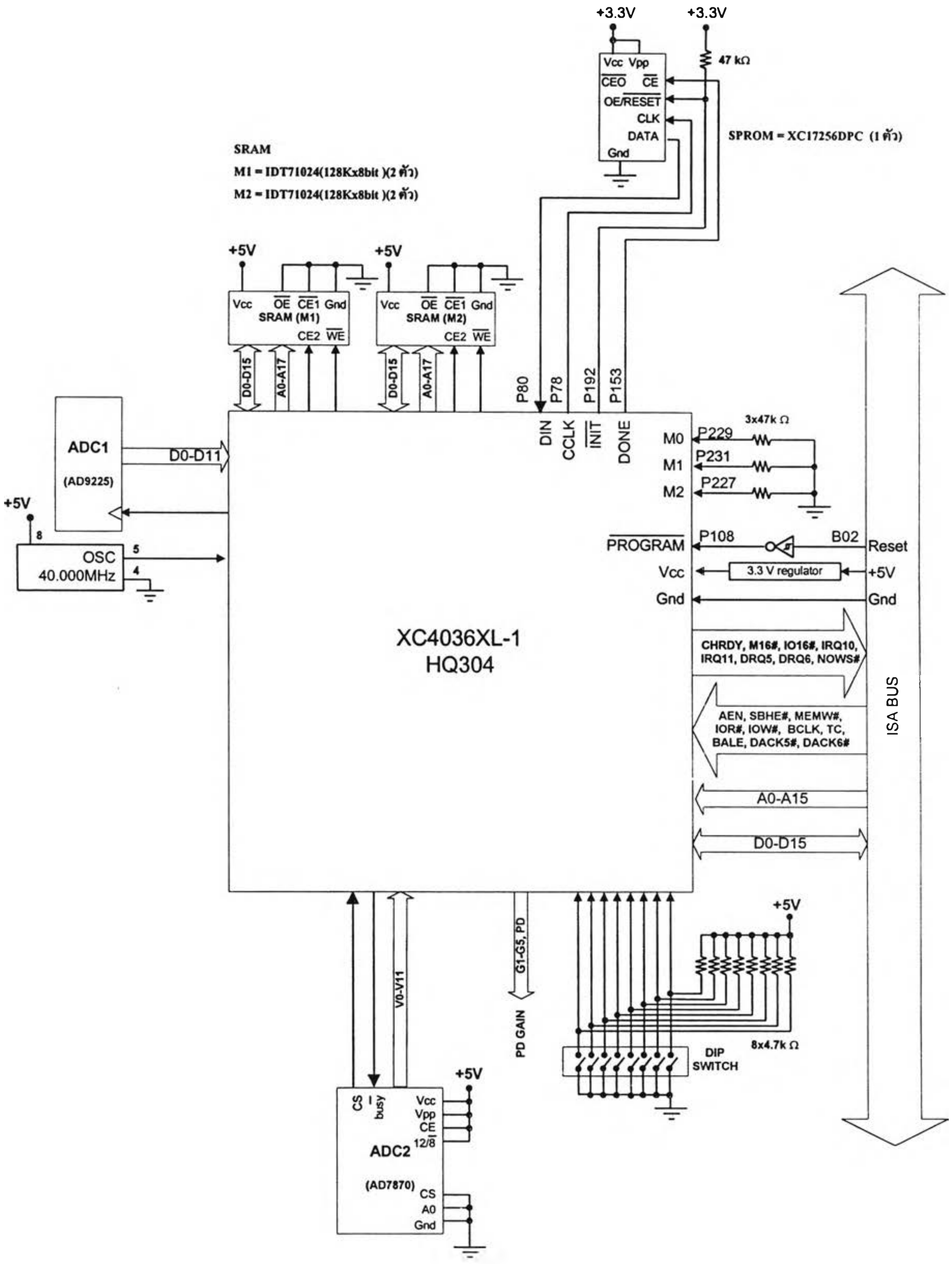
9) CTRL_UNIT เป็นวงจรภาคดิจิตอลอยู่ในลำดับชั้นสูงสุด ทำหน้าที่ควบคุมการทำงานของวงจรทั้งหมด

วงจรทั้งหมดเขียนด้วยภาษา VHDL ซึ่งสามารถทำความเข้าใจได้จากการอ่านโคดต้นฉบับ (Source code) และการพิจารณาไดอะแกรมเวลาประกอบกัน เนื่องจาก VHDL เป็นภาษาที่

สามารถอ่านทำความเข้าใจได้เช่นเดียวกันภาษาทางคอมพิวเตอร์ทั่วไปและจำลองการทำงานได้เช่นเดียวกับการวาด Schematic โดยวงจรภาคดิจิตอลอย่างง่ายแสดงในรูปที่ 3.9 และวงจรที่ออกแบบสมบูรณ์แสดงดังรูปที่ 3.10



รูปที่ 3.9 ส่วนประกอบของวงจรภาคดิจิตอล



รูปที่ 3.10 บล็อกไดอะแกรมของวงจรที่ออกแบบสมบูรณ์

3.3.4 รายละเอียดการทำงานของวงจรดิจิตอล

วงจรภาคดิจิตอลมีหลักการทำงานคือรับข้อมูลจาก ADC แล้วทำการประมวลผลข้อมูลสัญญาณที่รับมาก่อนส่งไปยังคอมพิวเตอร์ผ่านทางบัส ISA

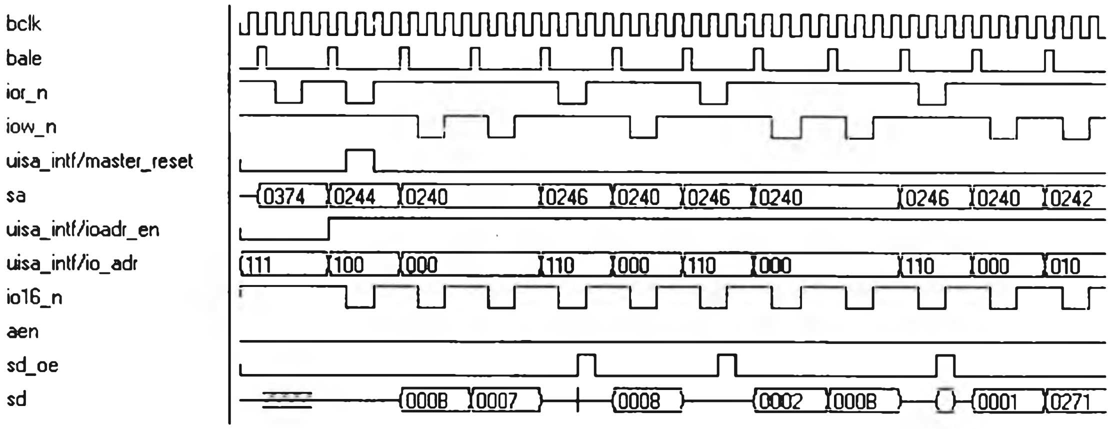
3.3.4.1 การถอดรหัสอินพุตเอาต์พุตและลอจิกควบคุม (I/O Decoder and Control Logic)

วงจรถอดรหัสอินพุตเอาต์พุตและลอจิกควบคุมอยู่ภายใน ISA_INTF ทำหน้าที่ถอดรหัสแอดเดรสอินพุตเอาต์พุตพอร์ต ควบคุมการรับส่งข้อมูลกับเครื่องคอมพิวเตอร์ และโปรแกรมลอจิกควบคุมการทำงานของวงจรภาคดิจิตอลและอัตราขยายสัญญาณของวงจรภาคแอนะลอก ภายในประกอบด้วยส่วนย่อยต่อไปนี้

1) ส่วนถอดรหัสแอดเดรสอินพุตเอาต์พุตพอร์ต สัญญาณที่ใช้ในการอ่านและเขียนพอร์ตประกอบด้วย สัญญาณแอดเดรสบัส SA[15:0], ดาต้าบัส SD[11:0], IOR, IOW, AEN, IO16 และดิปสวิตช์ SW[7:0] ภายในมีอินพุตเอาต์พุตพอร์ตขนาด 16 บิท 3 พอร์ต โดยใช้สัญญาณ SA[15:3] เป็นเบสแอดเดรส และสัญญาณ SA[2:0] เป็นช่วงแอดเดรสอินพุตเอาต์พุตพอร์ตที่ใช้งาน เบสแอดเดรสสามารถกำหนดได้โดยการตั้งดิปสวิตช์ในภาคผนวก ในการอ่านเขียนอินพุตเอาต์พุตพอร์ต สัญญาณแอดเดรสบัส SA[15:0], IOR หรือ IOW, และ AEN จะส่งออกมาทาง ISA BUS โดยจะเปรียบเทียบเบสแอดเดรส SA[15:3] กรณีเขียนหรืออ่านพอร์ตสัญญาณ จะถอดรหัสหมายเลขพอร์ตเขียนและพอร์ตอ่านจากสัญญาณ SA[2:0] ซึ่งแบ่งเป็นพอร์ตเขียน 16 บิต 3 พอร์ต และพอร์ตอ่าน 16 บิต 3 พอร์ต ขณะเดียวกันสัญญาณ IO16 จะแอกทีฟ 0 บอกให้ ISA BUS ทราบว่าการรับส่งข้อมูลมีขนาด 16 บิต โดยการถอดรหัสหมายเลขพอร์ตดูในตารางที่ 3.2 และไดอะแกรมเวลาแสดงในรูปที่ 3.11[51]

ตารางที่ 3.2 หน้าที่ของพอร์ตเขียนและพอร์ตอ่าน

เบสแอดเดรส SA[15:3]	SA[2:0]	หมายเลขพอร์ต	พอร์ตเขียน (IOW)	พอร์ตอ่าน (IOR)
ดิปสวิตช์	000	1	หมายเลขรีจิสเตอร์ควบคุมสัญญาณ (Register Index)	อ่านข้อมูลสัญญาณตรวจวัด PD
ดิปสวิตช์	010	2	รีจิสเตอร์ข้อมูล (Data Register)	อ่านรีจิสเตอร์สถานะการขออินเตอร์รัพต์
ดิปสวิตช์	100	3	เอ็นเอเบิลการขอ DMA (Enable DMA request)	รีเซตวงจร



รูปที่ 3.11 ไดอะแกรมเวลาวงจรถอดรหัสแอดเดรสอินพุตเอาต์พุต

2) ส่วนลอจิกควบคุมการทำงานของวงจร (พอร์ตเขียน 1 และพอร์ตเขียน 2) เป็นพอร์ตสำหรับโปรแกรมลอจิกควบคุมการทำงานของวงจรโดยพอร์ตเขียน 1 เป็นพอร์ตใช้ระบุหมายเลขรีจิสเตอร์ (Register index) มี 6 รีจิสเตอร์ และพอร์ตเขียน 2 เป็นพอร์ตเขียนข้อมูลไปยังรีจิสเตอร์ที่ระบุเพื่อโปรแกรมการทำงานของวงจร

การโปรแกรมลอจิกควบคุมเริ่มต้นด้วยการสั่งเขียนหมายเลขรีจิสเตอร์ที่ต้องการไปยังพอร์ตเขียน 1 หมายเลขรีจิสเตอร์จะถูกเก็บไว้ใน RegCtrl รีจิสเตอร์ จากนั้นสั่งเขียนข้อมูลหรือลอจิกควบคุมวงจรไปยังรีจิสเตอร์ RegData ผ่านทางพอร์ต 2 สัญญาณเขียนพอร์ต 2 จะถอดรหัสหมายเลขรีจิสเตอร์ที่เก็บใน RegCtrl รีจิสเตอร์ แล้วสั่งเขียนข้อมูลลงในรีจิสเตอร์ที่ระบุลอจิกควบคุมการทำงานของแต่ละรีจิสเตอร์ ดังรายละเอียดต่อไปนี้

(1) รีจิสเตอร์ 1 ควบคุมโหมดการทำงาน (RegCtrl)

D0	1 = เลือกอินเตอร์รัพต์ 10 (IRQ10)	0 = ไม่เลือก
D1	1 = เลือกอินเตอร์รัพต์ 11 (IRQ 11)	0 = ไม่เลือก
D2	1 = เลือก DMA แชนแนล 5 (DRQ 5)	0 = ไม่เลือก
D3	1 = เลือก DMA แชนแนล 6 (DRQ 6)	0 = ไม่เลือก
D4	0 = ดิสเอเบิลวงจรทั้งหมด	
	1 = เอ็นเอเบิลวงจรทั้งหมด	
D5	0 = เลือกการซิงโครไนซ์จากแหล่งจ่ายแรงดันภายใน	
	1 = เลือกการซิงโครไนซ์จากแรงดันทดสอบ	
D6	0 = เลือกการตรวจจับ PD แบบออฟไลน์ (Off-line)	
	1 = เลือกการตรวจจับ PD แบบเวลาจริง	

D8, D7	0 0	เลือกความถี่ของแรงดันทดสอบภายนอก 50 เฮิรตซ์
	1 1	เลือกความถี่ของแรงดันทดสอบภายใน 50 เฮิรตซ์
D9	1 =	เลือกให้อินเตอร์รัพต์ครั้งต่อไปได้ต่อเมื่ออ่านสถานะอินเตอร์รัพต์ครั้งก่อนแล้ว
	0 =	เลือกให้อินเตอร์รัพต์ครั้งต่อไปได้ทันที
D10	1 =	เลือกการตรวจจับ PD แบบ Realtime Analysis
	0 =	ไม่เลือก

(2) รีจิสเตอร์ 2 ควบคุมวงจรขยายสัญญาณ (RegGain)

D7 – D0	D7:D0	
	11111111	อัตราขยายสัญญาณ PDระดับ 255 (สูงสุด)
	⋮	
	00000000	อัตราขยายสัญญาณ PDระดับ 0 (ต่ำสุด)
D10 – D8	D10:D8	
	000	อัตราขยายสัญญาณแรงดันเป็น 1 เท่า
	001	อัตราขยายสัญญาณแรงดันเป็น 2 เท่า
	010	อัตราขยายสัญญาณแรงดันเป็น 4 เท่า
	011	อัตราขยายสัญญาณแรงดันเป็น 8 เท่า
	1XX	อัตราขยายสัญญาณแรงดันเป็น 16 เท่า

(3) รีจิสเตอร์ 3 กำหนดค่าระดับเทรชโฮลด์ตรวจจับค่ายอด PD (RegPDTHold)

D11 - D0 ค่าเทรชโฮลด์

(4) รีจิสเตอร์ 4 กำหนดช่วงเวลาตรวจจับค่ายอดสูงสุดของ PD ((RegPDLTime)

D11 - D0 ค่าช่วงเวลา

(5) รีจิสเตอร์ 5 ปรับค่าออฟเซต (Offset) ของ ADC1 (RegPDOffset)

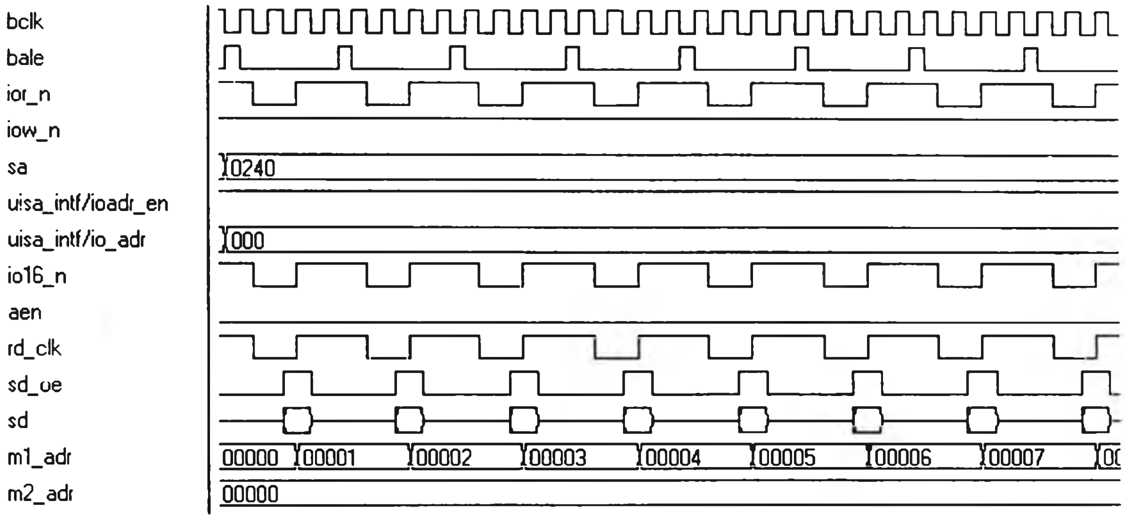
D11 – D0 ค่าออฟเซต

(6) รีจิสเตอร์ 6 ปรับค่าออฟเซต (Offset) ของ ADC2 (RegTVOffset)

D11 – D0 ค่าออฟเซต

3) พอร์ตควบคุมการปิดเปิดสัญญาณขอ DMA (พอร์ตเขียน 3) การปิดเปิดสัญญาณขอ DMA โปรแกรมได้โดยการเขียนลอจิกที่บิต SD0 ส่งไปยังพอร์ตเขียน 3 (0=ปิด, 1=เปิด) ค่าลอจิกนี้จะถูกเก็บในรีจิสเตอร์ 1 บิต เมื่อสัญญาณ DACK5(6) แอคทีฟ รีจิสเตอร์นี้จะถูกสั่งเคลียร์เพื่อปิดการขอ DMA ไม่ให้ค้างอยู่

4) พอร์ตอ่านข้อมูลตรวจจับ PD (พอร์ตอ่าน 1) พอร์ตอ่าน 1 เป็นพอร์ตใช้ในการอ่านข้อมูลตรวจจับ PD จากหน่วยความจำสัญญาณ RD_CLK เป็นสัญญาณนาฬิกาส่งไปยังวงจรรับข้อมูลตรวจหาตำแหน่ง PD เพื่ออ่านหน่วยความจำ ในขณะเดียวกันสัญญาณนี้จะไปเอ็นเอเบิล 3-state ให้บัสข้อมูลจากหน่วยความจำเชื่อมต่อกับบัส SD[15:0] เพื่อส่งข้อมูลไปยังคอมพิวเตอร์ โดยไดอะแกรมเวลาแสดงดังรูปที่ 3.12[51]

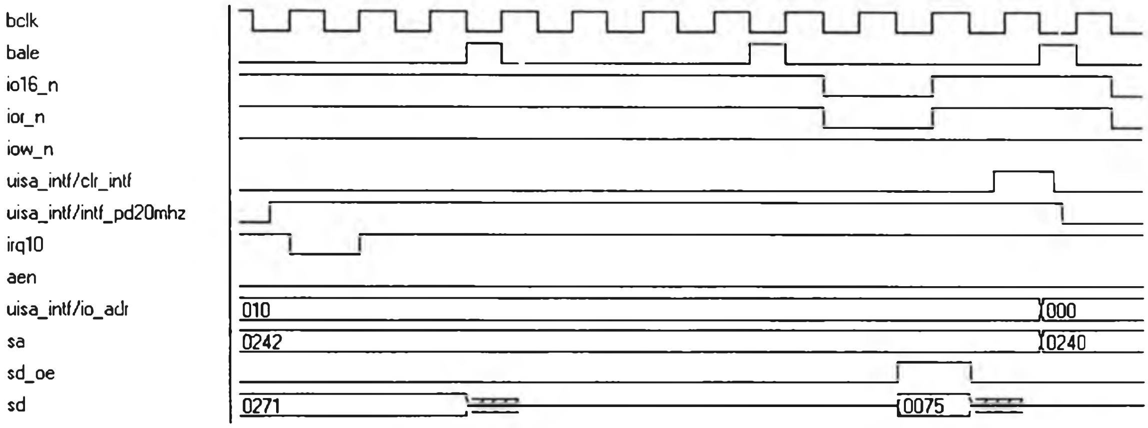


รูปที่ 3.12 ไดอะแกรมเวลาสัญญาณอ่านข้อมูลตรวจจับ PD

5) พอร์ตรีจิสเตอร์สถานะ (พอร์ตอ่าน 2) เป็นพอร์ตใช้ในการอ่านรีจิสเตอร์สถานะแสดงบิตการขออินเตอร์รัพต์และการทำ DMA โดยสัญญาณ OE_ISTATUS (Interrupt Status) เป็นสัญญาณของพอร์ตอ่าน 2 ส่งไปยังบัฟเฟอร์ให้เก็บสัญญาณที่ขออินเตอร์รัพต์หรือขอ DMA พร้อมกับเอ็นเอเบิลให้ข้อมูลจากบัฟเฟอร์ส่งไปยังบัส SD[15:0] และถูกอ่านเข้าเครื่องคอมพิวเตอร์ จากนั้นสัญญาณเคลียร์อินเตอร์รัพต์ (CLR_INTF) จะส่งไปเคลียร์ฟลิปฟลอป (Flip-flop) ที่สร้างสัญญาณขออินเตอร์รัพต์ เพื่อให้พร้อมที่จะสร้างสัญญาณอินเตอร์รัพต์ในครั้งต่อไป บิตแสดงรีจิสเตอร์สถานะมีดังนี้

บิต	ความหมาย
D0	1 = อินเตอร์รัพต์จากวงจรรับข้อมูลตรวจจับ PD แบบเวลาจริง
D1	1 = อินเตอร์รัพต์จากวงจรรับข้อมูลตรวจจับ PD แบบออฟไลน์
D2	1 = ขอทำ DMA ต่อ
D3	0 = ดิสเอเบิลการขอ DMA
	1 = เอ็นเอเบิลการขอ DMA
D8-4	1 = ค่าใน RegCtrl[8:4]

6) **พอร์ตรีเซตวงจร (พอร์ตอ่าน 3)** พอร์ตอ่าน 3 เป็นพอร์ตที่ใช้ในการรีเซตวงจร เมื่อสั่งอ่านพอร์ตนี้ สัญญาณ Master Reset จะแอคทีฟเป็นผลให้ ฟลิปฟลอป, รีจิสเตอร์ และ 3-state ภายในวงจรทั้งหมดจะเคลียร์ ดังไดอะแกรมเวลาในรูปที่ 3.13[51]



รูปที่ 3.13 ไดอะแกรมเวลาสัญญาณควบคุมการขออินเตอร์รัพต์

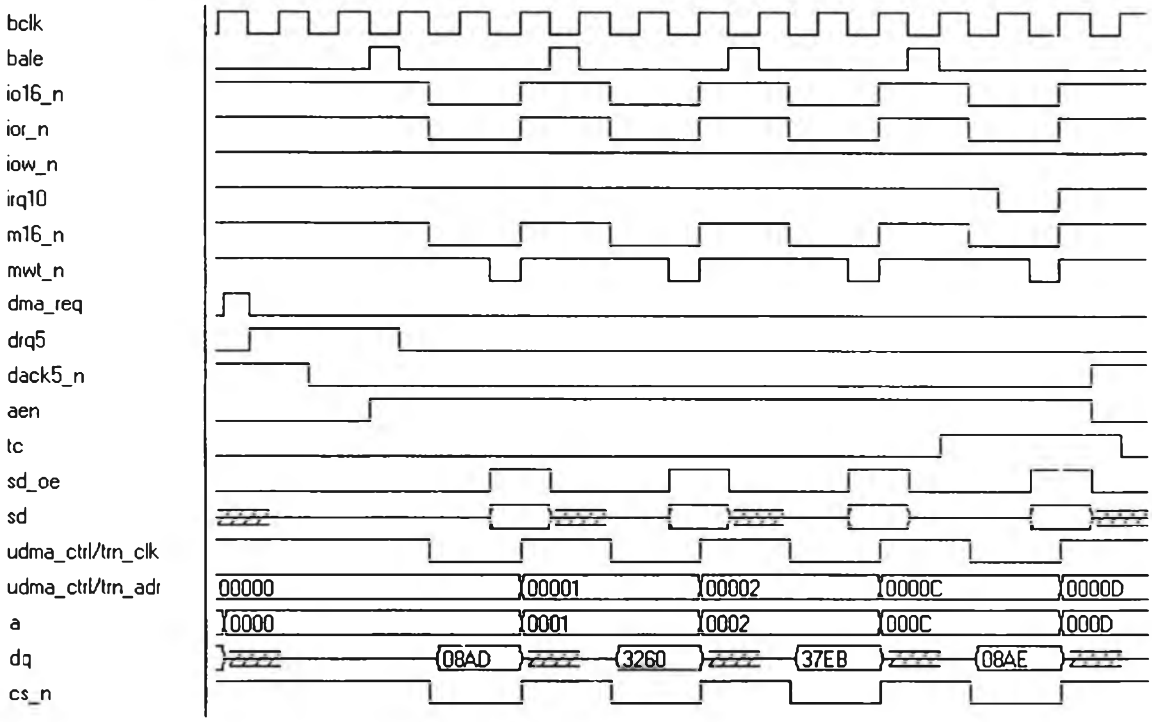
3.3.4.2 ส่วนควบคุมการขออินเตอร์รัพต์

ส่วนของการควบคุมอินเตอร์รัพต์จะมีฟลิปฟลอปควบคุมการขออินเตอร์รัพต์ โดยจะส่งสัญญาณ INT_PD20MHz ควบคุมวงจรส่วนที่สร้างพัลส์ขออินเตอร์รัพต์ไปยังขาสัญญาณ IRQ10 หรือ IRQ11 ของ ISA บัสตามที่เลือกไว้ในรีจิสเตอร์ RegCtrl เมื่อมีการสั่งอ่านพอร์ตรีจิสเตอร์สถานะ (พอร์ตอ่าน 2) วงจรควบคุมการขออินเตอร์รัพต์จะรีเซต

3.3.4.3 ส่วนการถ่ายโอนข้อมูลไปยังเครื่องคอมพิวเตอร์

สามารถถ่ายโอนข้อมูลได้สองแบบคือ แบบ DMA transfer สำหรับโหมดการทำงานแบบเวลาจริง และแบบ I/O transfer สำหรับทุกโหมดการทำงาน ดังรายละเอียดต่อไปนี้

1) การถ่ายโอนข้อมูลใช้วิธี DMA transfer เมื่อเก็บข้อมูลครบ 1 คาบ สัญญาณ DMA_REQ จะส่งไปยังส่วนควบคุมการขอ DMA เชื่อมต่อกับขาสัญญาณ DRQ5 หรือ DRQ6 ของบัส ISA ตามที่เลือกในรีจิสเตอร์ RegCtrl รอจนเริ่มมีการส่งข้อมูลสัญญาณ AEN, IOR, MWTC และ DACK5(6) แล้วจึงนำมาสร้างเป็นสัญญาณนาฬิกา TRN_CLK เพื่อควบคุมการอ่านข้อมูลจากหน่วยความจำเมื่อการทำ DMA เสร็จ 1 บล็อก สัญญาณ TC จะแอคทีฟ 1 ผ่านส่วนที่สร้างสัญญาณอินเตอร์รัพต์ไปยังคอมพิวเตอร์ เมื่อโปรแกรมตรวจสอบพบการขออินเตอร์รัพต์มาจากวงจรควบคุมการ DMA โปรแกรมจะทำการย้ายข้อมูลไปเก็บในหน่วยความจำที่ต้องการ และส่งคำสั่งไปยังพอร์ต 3 เอ็นเอเบิลให้ทำ DMA ในบล็อกต่อไปได้ ดังไดอะแกรมเวลาในรูปที่ 3.14[51]



รูปที่ 3.14 ไดอะแกรมเวลาของสัญญาณควบคุมการ DMA

2) การถ่ายโอนข้อมูลโดยใช้วิธี I/O transfer เป็นการส่งข้อมูลโดยการอ่านพอร์ตอ่าน 1 ซึ่งเป็นพอร์ตที่ใช้อ่านข้อมูลตรวจจับ PD รูปที่ 3.16 แสดงไดอะแกรมเวลาของวิธี I/O transfer

3.3.4.4 การกำจัดเบาซ์ความถี่แรงดันทดสอบ

เป็นการกำจัดเบาซ์ (bounce) ที่เกิดขึ้นเนื่องจากในขณะที่แรงดันทดสอบอยู่ในช่วงผ่านศูนย์ โดยเมื่อสัญญาณเริ่มเปลี่ยนสถานะครั้งแรกที่มุมเฟสศูนย์องศา วงจรนับจะรีเซตเริ่มนับหน่วงเวลาจนถึงมุมเฟสที่ 180 องศาจึงจะหยุดนับ หลังจากนั้นจะอ่านค่าแรงดันทดสอบเพื่อรอหาจุด

ผ่านศูนย์ครั้งต่อไป การทำงานจะเป็นเช่นนี้เหมือนกันทุกคาบ จึงไม่เกิดเบาซ์ที่ขึ้นที่บริเวณรอบๆ จุดผ่านศูนย์

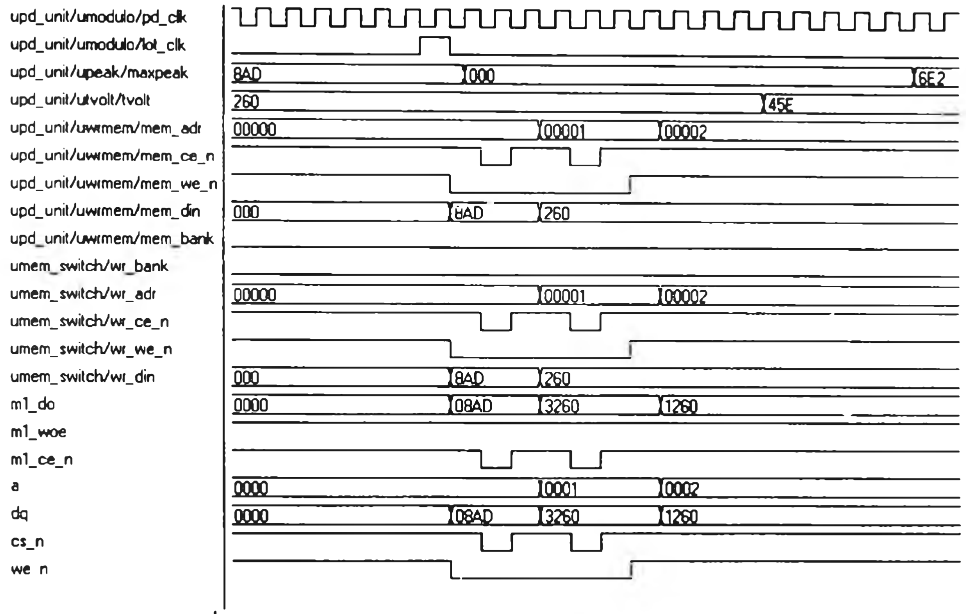
3.3.4.5 ส่วนรับข้อมูลตรวจจับ PD

ส่วนรับข้อมูลตรวจจับ PD ทำหน้าที่รับข้อมูลตรวจจับ PD จาก ADC1 ด้วยอัตราชักตัวอย่าง 20 ล้านตัวอย่างต่อวินาที ข้อมูลที่ตรวจจับได้จะมีทั้งค่ายอดของ PD และค่าแรงดันทดสอบ นอกจากนี้เรายังสามารถจัดเรียง (rectify) ค่ายอดให้แสดงค่าสัมบูรณ์ของสัญญาณได้ โดยการส่งสัญญาณควบคุมไปยังวงจรตรวจจับค่ายอดภายในวงจรรับข้อมูลตรวจจับ PD

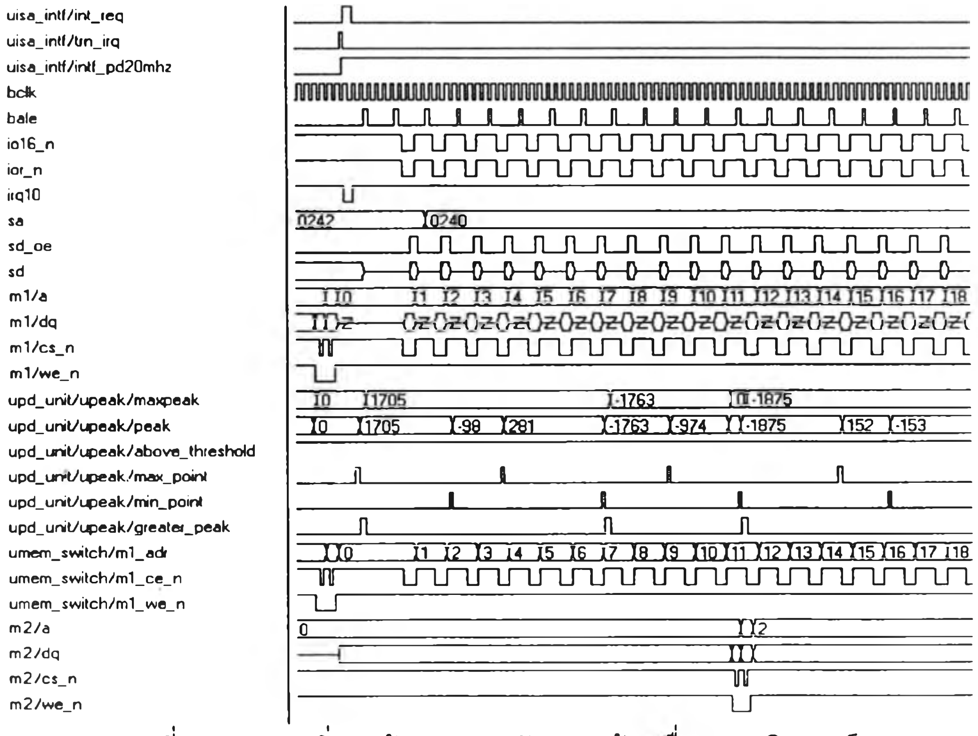
1) การทำงานของวงจรส่วนรับข้อมูลตรวจจับ PD ในการรับข้อมูลตรวจจับ PD จะมีการกำหนดค่าเริ่มต้นให้กับวงจรก่อนคือ RegPDTHold[11:0] เป็นค่าเทรชโฮลด์ใช้เปรียบเทียบระดับสัญญาณที่มีค่าสูงกว่าเทรชโฮลด์ให้ผ่านไปได้ และค่า PDLTime[11:0] เป็นค่าช่วง lock-out time ในการตรวจจับหาค่ายอดสูงสุด เมื่อวงจรเริ่มทำงานจะมีสัญญาณ CYCLE_signal เป็นสัญญาณซิงโครไนซ์สำหรับรับข้อมูลตรวจจับ PD ในแต่ละไซเคิล และจะมีการสร้างสัญญาณไปควบคุมการเขียนหน่วยความจำผ่านวงจร MEM_SWITCH เมื่อสัญญาณ M1_WE_N เป็น 0 สัญญาณ M1_CE_N จะสั่งให้บัสข้อมูล M1_DO[15:0] เชื่อมกับหน่วยความจำ ปิดบัสข้อมูลของหน่วยความจำไม่ให้เชื่อมต่อกับบัส SD[15:0] และสั่งให้หน่วยความจำทำงานในโหมดเขียน เริ่มที่วงจรตรวจจับค่ายอด PD จะรับข้อมูลจาก ADC1[11:0] เพื่อตรวจจับหาค่ายอดของสัญญาณ เมื่อตรวจจับได้ค่ายอด วงจรจะส่งสัญญาณจะสั่งให้บัฟเฟอร์ PEAK (Peak Buffer) เก็บค่ายอดในแต่ละ lock-out time ค่ายอดสูงสุดจะเก็บในบัฟเฟอร์ MAXPEAK และส่งสัญญาณไปยังวงจรสัญญาณควบคุมการเขียนหน่วยความจำ WRMEM_CTRL ให้เขียนข้อมูลในบัฟเฟอร์ส่งไปยังหน่วยความจำ 2 ค่า โดยแอดเดรสแรกจะเป็นแอดเดรสคู่ส่งออกไปที่ M1_ADR[17:0] พร้อมกับส่งค่ายอดใน MAXPEAK ออกไปที่ M1_DO[15:0] ไซเคิลที่สองจะเป็นแอดเดรสคี่ส่งออกไปที่ M1_ADR[17:0] พร้อมกับส่งค่าแรงดันทดสอบที่เก็บในบัฟเฟอร์ TVOLT ออกไปที่ M1_DO[15:0] โดยมีสัญญาณ M1_CE_N จำนวนสองไซเคิลไปสั่งให้หน่วยความจำ M1 เขียนค่ายอดและแรงดันทดสอบลงในหน่วยความจำ ดังไดอะแกรมเวลาในรูปที่ 3.15[51]

ในขณะมีการเขียนค่ายอดลงหน่วยความจำชุดหนึ่ง ก็จะมีการอ่านข้อมูลจากหน่วยความจำอีกชุดหนึ่งเพื่อส่งไปยังคอมพิวเตอร์ โดยวงจร DMA_CTRL จะมีการสร้างสัญญาณไปควบคุมการอ่านหน่วยความจำผ่านวงจร MEM_SWITCH เพื่อเปิดบัสข้อมูลจากหน่วยความจำให้เชื่อมต่อกับบัส SD[15:0] โดยรอให้สัญญาณ M2_CE_N แอกทีฟ และสัญญาณ M2_WE_N จะไปควบคุมให้หน่วยความจำทำงานในโหมดอ่าน เริ่มจากสัญญาณ DMA_REQ จะแอกทีฟเป็น 1 ส่งไปยังส่วนควบคุมการขอ I/O หรือ DMA เมื่อวงจรได้รับสัญญาณตอบสนองจากคอมพิวเตอร์แล้วจะทำ

หน้าที่สร้างสัญญาณอ่านข้อมูลส่งไปยัง วงจร MEM_SWITCH นับแอดเดรสออกมาที่ M2_ADR [17:0] (แอดเดรสจะนับขึ้นเริ่มจากแอดเดรสศูนย์จนถึงแอดเดรสสุดท้ายของข้อมูลที่เก็บอยู่ในหน่วยความจำ) พร้อมกับส่งสัญญาณ M2_CE_N อ่านข้อมูลจากหน่วยความจำ M2 สัญญาณเปิดให้บัสข้อมูลจากหน่วยความจำเชื่อมต่อกับบัส SD[15:0] ข้อมูลจากหน่วยความจำจะอ่านเข้าสู่คอมพิวเตอร์ ดังไดอะแกรมเวลาในรูปที่ 3.16[51]



รูปที่ 3.15 เวลารับข้อมูลตรวจจับ PD เข้าหน่วยความจำ

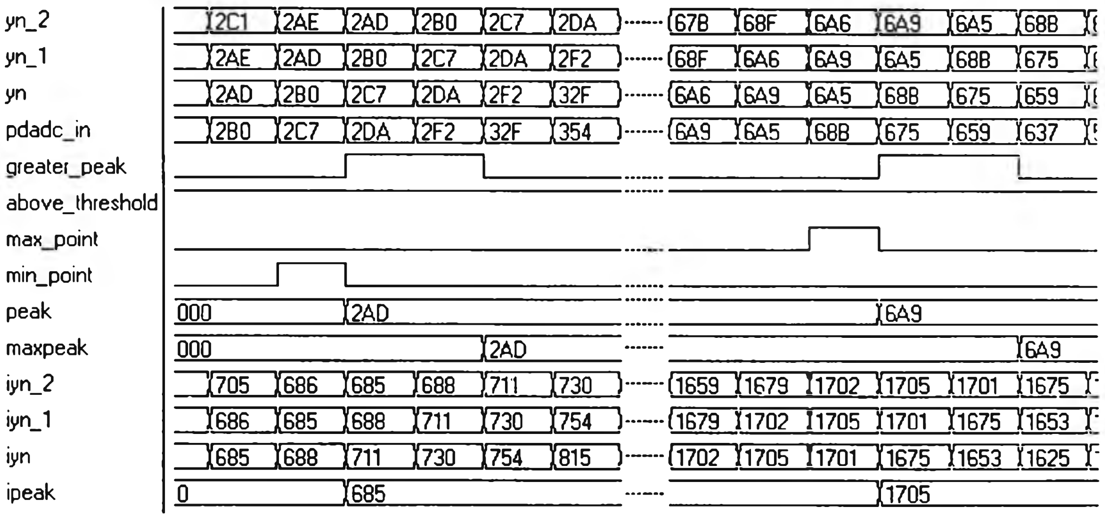


รูปที่ 3.16 เวลาเริ่มส่งข้อมูลตรวจจับ PD เข้าเครื่องคอมพิวเตอร์

2) ส่วนตรวจจับค่ายอด PD ทำหน้าที่ตรวจจับค่ายอด (Peak) PD แล้วส่งข้อมูลไปเก็บพักไว้ในบัฟเฟอร์รอส่งไปยังหน่วยความจำ หลังจากปรับข้อมูล PD ด้วยค่าออฟเซตแล้ว ข้อมูลที่ได้จะเก็บลงในบัฟเฟอร์แบบ FIFO เพื่อเก็บค่า PD จำนวน 3 ตำแหน่ง การเปรียบเทียบค่ายอดจะนำค่าทั้ง 3 ไปพิจารณา

```
greater_peak      <= '1' when (ABS_PEAK > ABS_MAXPEAK) else '0';
above_threshold   <= '0' when (ABS_PEAK < PDADC_THOLD) else '1';
max_point         <= '1' when (YN_2 < YN_1) and (YN_1 > YN) else '0';
min_point         <= '1' when (YN_2 > YN_1) and (YN_1 < YN) else '0';
........
if ((max_point = '1') or (min_point = '1')) then
      PEAK   <= YN_1;
else
........
```

โดยจะถือว่าตำแหน่ง Y_{n-1} เป็นค่ายอดก็ต่อเมื่อ Y_{n-1} เป็นจุดสูงสุดหรือจุดต่ำสุดเมื่อเทียบกับ Y_n และ Y_{n-2} ส่วนค่ายอดสูงสุดพิจารณาจากสัญญาณ greater_peak และ above_threshold โดยเมื่อค่าสัมบูรณ์ของบัฟเฟอร์ PEAK มีค่าสูงกว่าค่าสัมบูรณ์ของบัฟเฟอร์ MAXPEAK ที่มีอยู่เดิม และมีค่ามากกว่าค่าเทรชโฮลด์ที่กำหนด ในช่วงเวลาแต่ละ lock-out time วงจรนำค่ายอดที่สูงสุดเก็บในบัฟเฟอร์ MAXPEK ลงในหน่วยความจำ จากนั้นวงจรจะตรวจหาค่ายอดถัดไปและเริ่มจับเวลาอีกครั้งเพื่อตรวจหาค่ายอดสูงสุดใน lock-out time ใหม่ ดังไดอะแกรมเวลาในรูปที่ 3.17



รูปที่ 3.17 ไดอะแกรมเวลาวงจรตรวจจับค่ายอด PD

3.4 ดีไวซ์ไดรเวอร์

ดีไวซ์ไดรเวอร์คือซอฟต์แวร์ที่ทำหน้าที่เป็น device controller ของฮาร์ดแวร์ และสามารถเชื่อมต่อกับโปรแกรมประยุกต์ได้ ในงานวิจัยนี้พัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการ (Operating system) วินโดวส์98

3.4.1 ส่วนประกอบของระบบปฏิบัติการวินโดวส์98 ที่เกี่ยวข้อง

ระบบปฏิบัติการวินโดวส์98 ประกอบไปด้วยส่วนของโปรแกรมที่ทำหน้าที่ต่างๆ สำหรับส่วนที่มีความเกี่ยวข้องกับการพัฒนาดีไวซ์ไดรเวอร์สำหรับงานวิจัยโดยตรง มีรายละเอียดดังนี้

1) VMM (Virtual machine manager) จะทำงานอยู่ในระดับล่างสุดของวินโดวส์98 ซึ่งเป็นส่วนที่ทำงานใน protected mode ของ CPU ตระกูล Intel ตั้งแต่รุ่น 386 ขึ้นไป[52] ทำหน้าที่ดูแลจัดการ Virtual machine (VM) ทั้งหมดที่ทำงานอยู่ภายใต้วินโดวส์ 98

2) VxDs (Virtual devices) เป็นโปรแกรมแบบ 32 บิต ที่ทำงานจัดการทรัพยากรของระบบ ได้แก่ อุปกรณ์ฮาร์ดแวร์ หรือโปรแกรมประยุกต์ต่างๆ เพื่อทำให้โปรแกรมทั้งหมดสามารถใช้งานทรัพยากรของระบบร่วมกันได้ VxD ทำงานประสานกับส่วนของ VMM และส่วนอื่นในการจัดการสัญญาณอินเตอร์รัพท์และการกระทำอินพุตเอาต์พุตสำหรับโปรแกรมประยุกต์หนึ่งๆ โดยไม่ให้มีผลกระทบต่อโปรแกรมประยุกต์อื่นๆ VxD เป็นส่วนหนึ่งของ VMM ซึ่งมีสิทธิพิเศษในการทำงานที่เหนือกว่าโปรแกรมประยุกต์ที่ทำงานอยู่ในระบบ VxD code และ data นั้นทำงานใน Ring 0, 32 bit flat memory model ในการทำงานนี้ VxD จะสามารถอ้างถึงแอดเดรส ณ ตำแหน่งใด ๆ ก็ได้ภายใน 4 จิกะไบต์ โดยไม่ต้องเปลี่ยนค่าของ segment register เหมือนอย่างโปรแกรมแบบ 16 บิต (ใน protected mode นั้น segment register จะมีหน้าที่อื่นและเรียกว่า selector) เนื่องจาก VxD ทำงานอยู่ในระดับของ Kernel ในระบบปฏิบัติการ ซึ่งมีสิทธิในการใช้งาน privileged instruction ของ CPU ได้ ดังนั้น VxD จึงสามารถเข้าถึงส่วนต่าง ๆ ของ CPU และระบบได้ วินโดวส์ใช้ VxD สำหรับทำหน้าที่ virtualize ทรัพยากรของระบบ เช่น อุปกรณ์ฮาร์ดแวร์ หรือ การขออินเตอร์รัพต์ เป็นต้น (การทำ virtualize คือการทำให้โปรแกรมประยุกต์เสมือนกับเป็นเจ้าของทรัพยากรนั้นๆแต่เพียงผู้เดียว เช่น โปรแกรมประยุกต์ที่พัฒนาขึ้นนี้สามารถทำงานกับ IRQ 5, I/O Port 280h ได้โดยตรงและไม่มีผลกระทบต่อโปรแกรมอื่นที่ใช้งานอุปกรณ์เดียวกันอยู่นี้ และในทางกลับกันก็เป็นจริงด้วย) โดยสรุป VxD นั้นมีคุณสมบัติหลายอย่างที่เหมาะกับการพัฒนาส่วนของซอฟต์แวร์ที่ทำหน้าที่เป็น device controller ของการ์ด PD ดังนี้

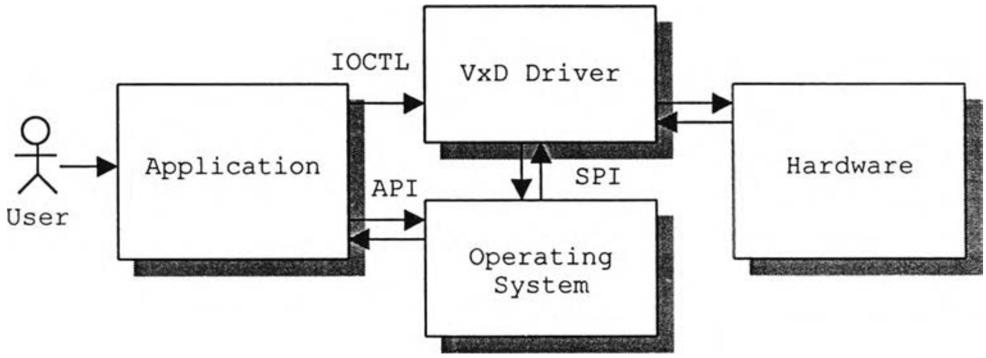
(1) ความเร็วและการเข้าถึงส่วนต่างๆของระบบ จะเหนือกว่าและไม่มีข้อจำกัดเหมือนกับโปรแกรมประยุกต์ ความเร็วเหนือกว่าการใช้งานไดรเวอร์ที่พัฒนาบน MS-DOS เนื่องจากไม่ต้องมีการทำ Context switching ซึ่งเป็นลักษณะที่เกิดขึ้นในการทำงานแบบ Multitasking

(2) สามารถเชื่อมต่อกับโปรแกรมประยุกต์ 32 บิตบนวินโดวส์ได้ โดยผ่านอินเตอร์เฟส แบบ device I/O control

นอกจากทั้งสองส่วนที่ได้กล่าวมาแล้ว ระบบปฏิบัติการวินโดวส์98 ยังมีส่วนประกอบที่สำคัญอื่นๆ เช่น Read-only memory (ROM) basic input and output system (BIOS), Installable device drivers and terminate-and-stay-resident (TSR) programs, 16- and 32-bit Windows dynamic-link libraries (DLLs), Microsoft MS-DOS–based applications, 16- and 32-bit Windows-based applications เป็นต้น

3.4.2 การสร้าง Device Controller

ในการสร้าง (implement) PD.VxD ซึ่งเป็นโปรแกรมที่ทำหน้าที่เป็น device controller นั้น ได้ใช้ซอฟต์แวร์ VtoolsD ของบริษัท Vireo Software, Inc. (ปัจจุบันเป็นของ Compuware Co., Ltd.) เป็นเครื่องมือ โดยที่ PD.VxD นั้นสามารถทำงานได้ทั้งระบบวินโดวส์95 และวินโดวส์ 98 แม้จะพัฒนาบนวินโดวส์95 ทั้งนี้เนื่องจากวินโดวส์98 ได้ทำ backward compatability ในส่วนของ VxD ไว้ด้วย สำหรับ VxD ที่พัฒนาขึ้นนี้โปรแกรมประยุกต์จะติดต่อ VxD โดยใช้ IOCTL ดังในรูปที่ 3.18 โดยที่ API (Application programming interface) เป็นฟังก์ชันในการเรียกใช้บริการต่าง ๆ ของระบบปฏิบัติการ ส่วน SPI (System programming interface) เป็นฟังก์ชันในการเรียกใช้ VxD ที่สามารถใช้ได้โดย Kernel เท่านั้น ซึ่งเป็นจุดที่แตกต่างระหว่างการที่โปรแกรมประยุกต์ติดต่อกับระบบปฏิบัติการ กับการที่ดีไวซ์ไดรเวอร์แบบ VxD ติดต่อกับระบบปฏิบัติการ



รูปที่ 3.18 โมเดลของ VxD ที่พัฒนาขึ้น

โปรแกรม VxD มีลักษณะเป็น Event driven โดยตัวโปรแกรม VxD จะได้รับเมสเสจ (message) ต่าง ๆ จากตัวระบบ เสมือนว่ารับรู้เหตุการต่างๆ ที่เกิดขึ้นในระบบ ฟังก์ชันการทำงานใน PD.VxD สำหรับเมสเสจต่างๆ จะเรียกใช้ตามที่ได้กำหนดไว้ เมื่อ VxD ได้รับเมสเสจเช่น Device Create ขณะที่สร้าง device ขึ้นมา หรือเมื่อเกิดฮาร์ดแวร์อินเตอร์รัพต์ VMM ก็จะบันทึก

ไว้ว่ามีอินเตอร์รัพต์เกิดขึ้นแล้ว โดยเก็บสถานะทุกอย่างที่จำเป็นไว้ ในขั้นนี้เรียกว่า first-level interrupt handler รอจนกระทั่งสามารถส่งเมสเสจเหล่านี้ไปยังส่วนที่จะทำการทำงานของอินเตอร์รัพต์ ซึ่งเรียกว่า second-level interrupt handler จาก VMM ผ่านเข้ามายัง VxDs ในระบบทุกตัวเรียงตามลำดับในลิสต์ (list) แม้ VMM จะไม่สามารถตอบสนองเมสเสจได้ทันทีก็ตาม แต่ VxD ก็เป็นวิธีที่สามารถควบคุมและจัดการฮาร์ดแวร์ได้เร็วที่สุดสำหรับระบบปฏิบัติการวินโดวส์98[53]

3.4.3 การออกแบบดีไวซ์ไดรเวอร์

ในการออกแบบโปรแกรม PD.VxD สำหรับใช้ในงานวิจัยนี้ จะประกอบด้วยส่วนของ PD Detection, Capture และ PD Analysis ซึ่ง PD.VxD จะทำหน้าที่สำคัญ 3 ส่วน คือ

1) การจัดการ virtualize อินเตอร์รัพต์ และ อินพุตเอาต์พุตพอร์ต จะเรียกใช้บริการของ VPICD (Virtual programmable interrupt controller device) ซึ่งเป็น VxD ตัวหนึ่งของวินโดวส์ ด้วยการเรียกใช้ฟังก์ชัน VPICD_Virtualize_IRQ ร่วมกับฟังก์ชัน VPICD_Physically_Mask ,VPICD_Physically_Unmask และ VPICD_Force_Default_Behavior ในการจัดการเกี่ยวกับการ virtualize , mask & unmask IRQ และ ยกเลิกการ virtualize

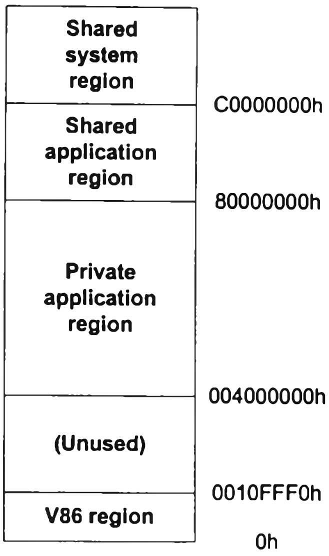
2) การจัดการ memory buffer เนื่องจากเราใช้ส่วนของพื้นที่หน่วยความจำส่วนที่ใช้ร่วมกันใน Win32 application ทุกตัว ร่วมกับ System VM ของระบบซึ่งเป็นที่ที่ VxD ของเราทำงานอยู่ ทำให้ไม่จำเป็นต้องทำการสำเนาข้อมูลจากส่วนของ VxD ไปสู่ส่วนของโปรแกรมประยุกต์ พื้นที่หน่วยความจำส่วนนี้เป็นส่วนของ Virtual address space ซึ่งจำเป็นจะต้องมีการแมปจาก buffer ที่จองพื้นที่ไว้ไปสู่พื้นที่ที่ใช้งานร่วมกัน เราสามารถทำการ lock และ commit ลงสู่ Physical memory โดยใช้ฟังก์ชัน LinPageLock รูปที่ 3.19 แสดงพื้นที่ของ memory ส่วนที่ใช้งานสำหรับ buffer ซึ่งจะแมปเข้าไปในส่วนของ shared system region ในการสั่งงานถ่ายโอนข้อมูลจากฮาร์ดแวร์จะใช้ลักษณะของ Program I/O

3) การจัดการติดต่อกับโปรแกรมประยุกต์แบ่งเป็น 2 แบบคือ

(1) จากโปรแกรมประยุกต์ไปยัง VxD โดยผ่าน Device I/O Control Interface ซึ่ง Win32 application สามารถเรียกใช้ Win32 API ที่มีชื่อว่า DeviceIoControl ได้โดยตรง หลังจากที่ได้สร้าง Handle สำหรับการติดต่อกับ VxD มาแล้วจาก API ที่มีชื่อว่า CreateFile

(2) จาก VxD ไปยังโปรแกรมประยุกต์โดยใช้ undocumented service ของวินโดวส์ ลักษณะที่ PD.VxD ทำโดยให้โปรแกรมประยุกต์สร้าง Event ขึ้นมาโดยใช้ Win32 API ที่มีชื่อว่า CreateEvent แล้วทำการแปลง Handle ที่ได้ให้อยู่ในรูป Ring 0 Event Handle ซึ่ง VxD สามารถเรียกใช้ได้โดยใช้ undocumented API ที่มีชื่อว่า OpenVxDHandle ซึ่งอยู่ในไฟล์ kernel32.dll จากนั้นจึงส่ง Handle นี้ไปยัง VxD โดยใช้ DeviceIoControl เมื่อ VxD ต้องการจะติดต่อกับ

โปรแกรมประยุกต์ VxD จะใช้ Ring 0 Event Handle ที่ได้ และฟังก์ชัน VWIN_SetWin32Event เพื่อทำ Event signaling ให้กับโปรแกรมประยุกต์



System memory map of Windows 95 virtual address space

รูปที่ 3.19 พื้นที่ของ memory ส่วนที่ใช้งานสำหรับ buffer

3.4.4 Device I/O Control Interface ของดีไวซ์ไดรเวอร์

ค่าต่อไปนี้คือ Device I/O Control Interface ของดีไวซ์ไดรเวอร์ PD.VxD ส่วนที่ application สามารถเรียกใช้ได้

DIOC_DISABLEPD	ใช้สำหรับปิดวงจรการทำงานของการ์ด
DIOC_RESETGSR	ใช้สำหรับรีเซต Xilinx
PD_ANALYSIS	ใช้สำหรับการทำงานในโหมด PD Analysis
PD_OFFLINE	ใช้สำหรับการทำงานในโหมด PD Offline
READ_PD_REALTIME	ใช้สำหรับการอ่านข้อมูลของ PD Analysis
READ_PD_REALTIME_DATA_TAKEN	ใช้สำหรับการแจ้ง (notify) VxD ว่าการอ่านข้อมูล PD Analysis เรียบร้อย
READ_PD_REALTIME_END	ใช้สำหรับการสั่งจบขั้นตอนการทำงานกับ PD Analysis
SETPARAM	ใช้สำหรับการเซตค่าต่าง ๆ ของการ์ด

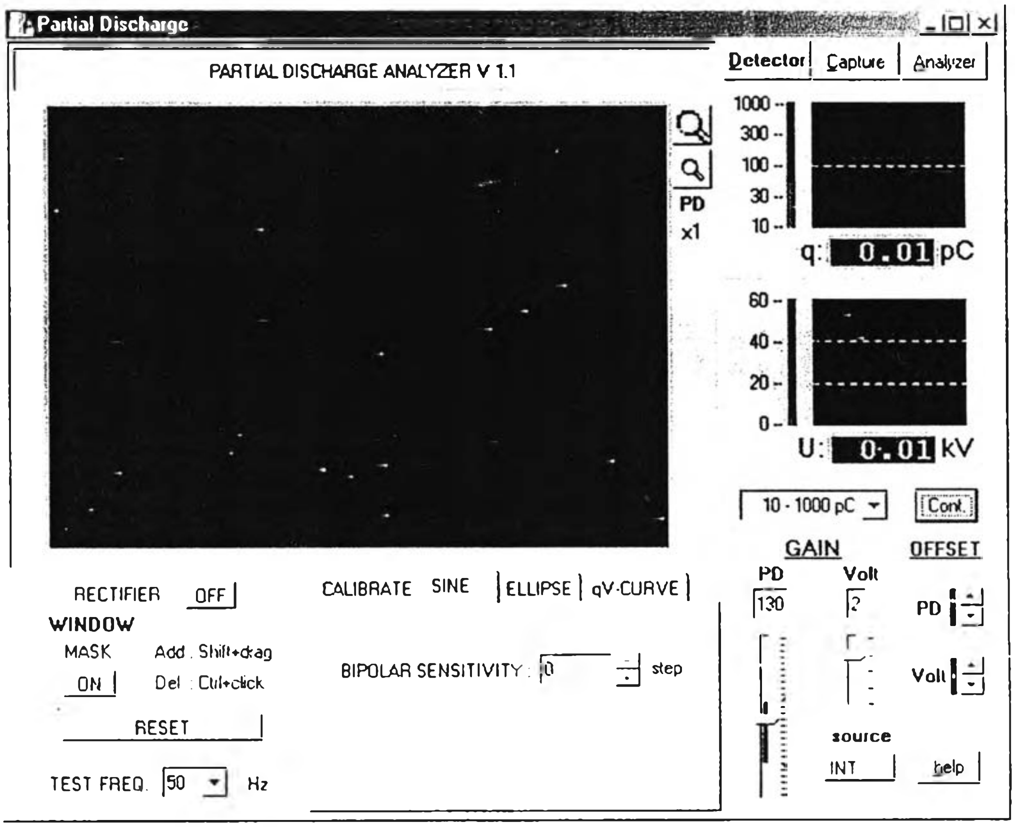
สำหรับโครงสร้างข้อมูลที่ใช้ในการติดต่อกับ PD.VxD นั้น มีรายละเอียดแสดงไว้ในภาคผนวก ค

3.5 ซอฟต์แวร์สำหรับเครื่องวิเคราะห์ดีสชาร์จบางส่วน

ซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้กับเครื่องวิเคราะห์ดีสชาร์จบางส่วนประกอบด้วย 3 ภาคหลัก คือ ภาคตรวจจับสัญญาณ PD ภาคแสดงรูปคลื่นสัญญาณ และภาควิเคราะห์ PD ซอฟต์แวร์ทั้ง 3 ภาคนี้พัฒนาขึ้นด้วยภาษา C++ โดยใช้คอมไพเลอร์ (Compiler) Borland C++ Builder เวอร์ชัน 4.0 ของบริษัท Inprise Corporation ทำงานภายใต้ระบบปฏิบัติการวินโดวส์98 (Windows98) โดยจะแสดงผลบนจอภาพสี SVGA การควบคุมการทำงานและการส่งผ่านข้อมูลกับวงจรฮาร์ดแวร์จะติดต่อผ่านทางบัส ISA โดยใช้สัญญาณอินพุตเอาต์พุตพอร์ต และสัญญาณอินเทอร์รัพต์

3.5.1 ภาคตรวจจับสัญญาณดีสชาร์จบางส่วน

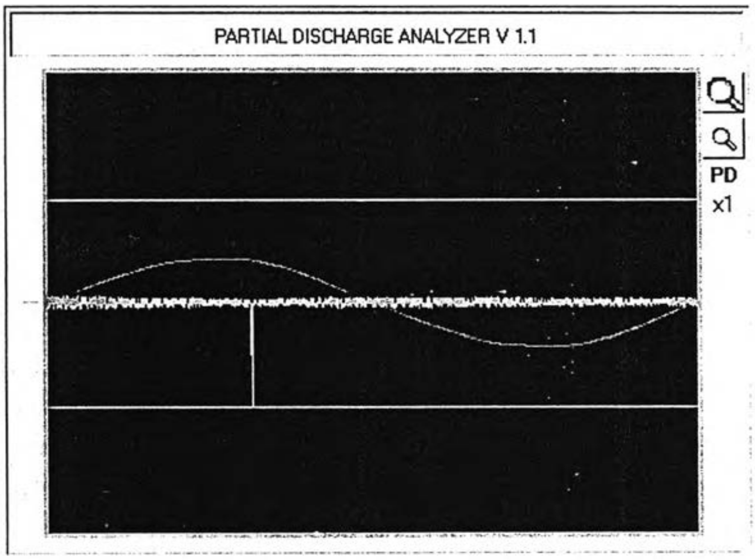
ภาคตรวจจับสัญญาณ PD ประกอบด้วยฟอร์ม (form) 1 ฟอร์ม ทำหน้าที่ตรวจจับสัญญาณ PD มีส่วนแสดงผลประกอบด้วยกรอบแสดงภาพสัญญาณ และปุ่มปรับเพื่อควบคุมการทำงานดังรูปที่ 3.20 ฟอร์มนี้ทำหน้าที่เป็น ฟอร์มหลักในการควบคุมการทำงานของเครื่องในการเรียกใช้ฟอร์มอื่นๆ ต่อไปและควบคุมการทำงานของฟังก์ชันต่างๆในการตรวจจับสัญญาณ PD



รูปที่ 3.20 ส่วนแสดงผลของภาคตรวจจับสัญญาณ PD

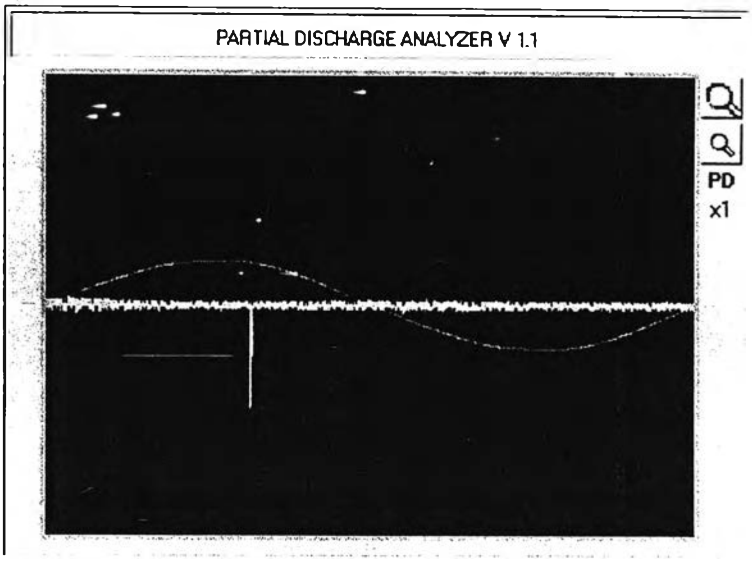
รายละเอียดของฟังก์ชันต่างๆในการตรวจจับ PD มีดังต่อไปนี้

1) การปรับเทียบ เป็นฟังก์ชันโดยปริยาย (default) ของภาคตรวจจับสัญญาณ PD สามารถเรียกใช้โดยการเลือก page ที่ชื่อ"CALIBRATE"ในส่วน Page control การทำงานเริ่มจากเลือกแหล่งจ่ายแรงดันเป็น INT หรือ EXT ซึ่งหมายถึงแหล่งจ่ายแรงดันกระแสสลับแบบภายในหรือแบบภายนอก ตามลำดับ ปรับค่าประจุปรับเทียบ (Calibration charge) ตามค่าประจุที่ป้อนคร่อมวัสดุทดสอบขณะทำการปรับเทียบ และปรับระดับเส้นแสดงความสูงของสัญญาณให้เท่ากับความสูงของสัญญาณปรับเทียบ โปรแกรมจะทำการคำนวณแฟกเตอร์ตัวคูณที่ใช้แปลงค่าดิจิตอลของสัญญาณ PD เป็นค่าประจุที่ใช้ปรับเทียบหน่วยเป็นพิโคคูลอมบ์ การปรับขนาดของสัญญาณให้เหมาะสมกับการแสดงผลทำได้โดยปรับแถบเลื่อนแสดงอัตราขยาย โดยคอมพิวเตอร์จะสั่งการไปยังวงจรขยายในส่วนฮาร์ดแวร์ให้เพิ่มหรือลดอัตราขยายตามที่ต้องการ การกรองสัญญาณรบกวนที่มีระดับต่ำกว่าค่าเทรชโฮลด์ทำได้โดยการปรับค่า Bipolar sensitivity threshold ส่วนการกำจัดสัญญาณรบกวนที่เข้าจังหวะกับแรงดันทำได้โดยการลากเมาส์ (mouse) พร้อมกับกดคีย์พิเศษประกอบกันเพื่อสร้างแถบ"หน้าต่างเวลา" (time window) ซึ่งมีรายละเอียดในหัวข้อ *2.4.2 การลดสัญญาณรบกวน* ลักษณะของส่วนแสดงผลหลักแสดงในรูปที่ 3.21



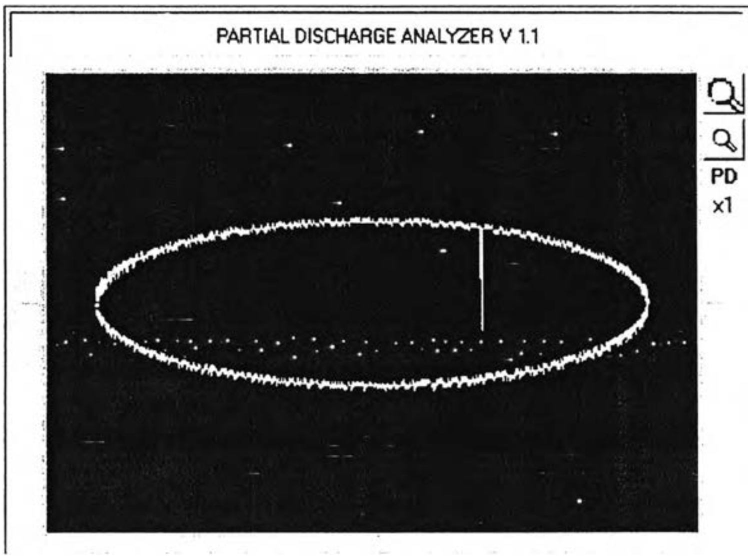
รูปที่ 3.21 ลักษณะของส่วนแสดงผลหลักของฟังก์ชันการปรับเทียบ

2) การแสดงผลสัญญาณบนฐานเวลาแบบเชิงเส้น (linear time base) เรียกใช้ฟังก์ชันนี้โดยการเลือก Page ที่ชื่อ"SINE" การทำงานของฟังก์ชันนี้คือ จะแสดงสัญญาณ PD และแรงดันทดสอบในแต่ละคาบบนฐานเวลาแบบเชิงเส้นในกรอบแสดงภาพหลัก แสดงค่าประจุที่ปรากฏและค่าแรงดันทดสอบค่าปัจจุบันเป็นตัวเลข และแสดงค่า 100 ค่าล่าสุดในแผนภูมิ (chart) ของกรอบแสดงภาพรองทั้งสองกรอบ ลักษณะของส่วนแสดงผลหลักแสดงในรูปที่ 3.22



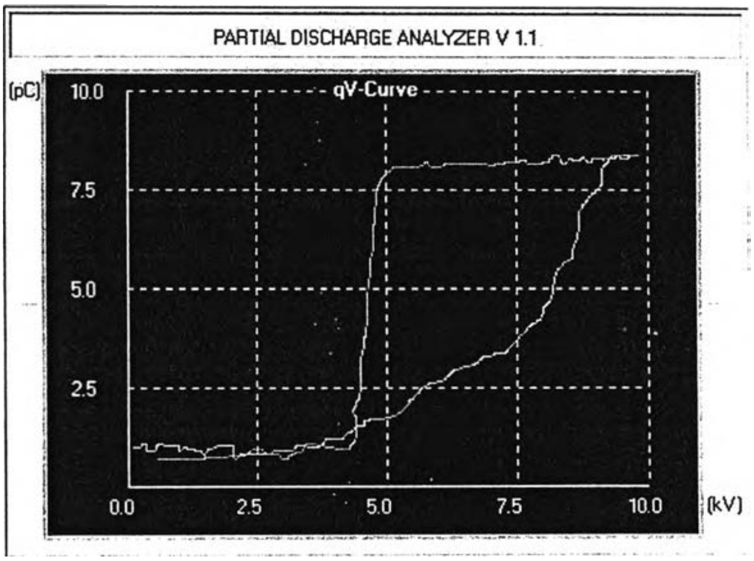
รูปที่ 3.22 ลักษณะของส่วนแสดงผลหลักของการแสดงผลบนฐานเวลาแบบเชิงเส้น

3) การแสดงผลสัญญาณบนฐานเวลารูปวงรี (elliptical time base) เรียกใช้ฟังก์ชันนี้โดยการเลือก Page ที่ชื่อ"ELLIPSE" การทำงานของฟังก์ชันนี้คือ จะแสดงสัญญาณ PD บนฐานเวลารูปวงรีในกรอบแสดงภาพหลัก แสดงค่าประจุที่ปรากฏและค่าแรงดันทดสอบค่าปัจจุบันเป็นตัวเลข และแสดงค่า 100 ค่าล่าสุดเป็นแผนภูมิ (chart) ในกรอบแสดงภาพรองทั้งสองกรอบในลักษณะเช่นเดียวกันกับฐานเวลาแบบเชิงเส้น สามารถหมุนวงรีเพื่อสังเกตสัญญาณบริเวณรอบๆ จุดผ่านศูนย์ให้ชัดเจนโดยการเปลี่ยนค่ามุมในช่อง Rotation ลักษณะของส่วนแสดงผลหลักแสดงในรูปที่ 3.23



รูปที่ 3.23 ลักษณะของส่วนแสดงผลหลักของการแสดงผลบนฐานเวลารูปวงรี

4) การแสดงผลความสัมพันธ์ระหว่างค่า PD กับแรงดัน (q-V curve) เรียกใช้ฟังก์ชันนี้โดยเลือก page ที่ชื่อ"qV-CURVE" ในการใช้งานฟังก์ชันนี้ผู้ทดสอบจะทำการปรับระดับแรงดันทดสอบขึ้นและลงเพื่อบันทึกค่าพิกัดระหว่างประจุที่ปรากฏ (pC) กับแรงดันทดสอบ (kV) ไว้ใช้ในการแสดงผลในกรอบแสดงภาพหลัก เส้นสีเขียวและสีแดงแสดงถึงการเพิ่มและลดแรงดันทดสอบตามลำดับ ผู้ทดสอบสามารถกำหนดระยะเวลาทดสอบและจำนวนจุดพิกัดได้โดยการเลือกค่าที่ต้องการใน Combo box ลักษณะของส่วนแสดงผลหลักแสดงในรูปที่ 3.24



รูปที่ 3.24 ลักษณะของส่วนแสดงผลหลักของการแสดงผล q-V curve

5) การแสดงค่าประจุ q ของ PD เนื่องจาก q เป็นค่าประจุของ PD ในแต่ละคาบของแรงดันทดสอบที่ใช้ในการแสดงผลขนาดของ PD และนำไปคำนวณพิกัดใน q-V Curve ค่าประจุนี้ได้มาจากการนำค่ายอดสูงสุดของ PD ในแต่ละคาบของแรงดันทดสอบมาคำนวณและพิจารณาตามเงื่อนไขดังนี้

$$q_n = \begin{cases} q_m & if \quad q_m > q_{n-1}e^{-1/f} \\ q_{n-1}e^{-1/f} & if \quad q_m < q_{n-1}e^{-1/f} \end{cases} \qquad (3.6)$$

โดยที่ q_m คือ ค่ายอดสูงสุดของ PD ณ คาบแรงดันทดสอบล่าสุด

q_n คือ ค่าประจุของ PD ณ คาบแรงดันทดสอบปัจจุบัน

q_{n-1} คือ ค่ายอดของ PD ณ คาบแรงดันทดสอบที่ผ่านมา

f คือ ความถี่ของแรงดันทดสอบ

6) การแสดงค่าแรงดันทดสอบ U โปรแกรมสามารถแสดงค่าแรงดันทดสอบจากข้อมูลดิจิตอลที่รับมาจาก ADC2 และแฟกเตอร์ตัวคูณ ซึ่งคำนวณได้จากอัตราการลดทอนแรงดันจากภาคแรงสูง อัตราขยายสัญญาณก่อนเข้าสู่ ADC2 ค่าแรงดันเต็มสเกลและความละเอียดของ ADC2 ดังนี้ สมมติให้อัตราขยายเป็น 1 และให้ค่าดิจิตอลที่ ADC2 แปลงได้เป็น 1 ขั้น เนื่องจากค่าเต็มสเกลคือ 2^{12} = 4096 ขั้น พิสัยของ ADC2 คือ ± 3 โวลต์ หรือ 6 โวลต์จากยอดถึงยอด และอัตราการลดทอนแรงดันทดสอบเป็น 40,000 เท่า ดังนั้น ค่าแรงดันทดสอบคือ (40000 / 1) x (6 / 4069) = 58.6 โวลต์ แสดงว่าแฟกเตอร์ตัวคูณของการแปลงค่าดิจิตอลเป็นแรงดันทดสอบเมื่ออัตราขยายเป็น 1 คือ 58.6 โวลต์ต่อขั้น สำหรับแฟกเตอร์ตัวคูณและค่าแรงดันทดสอบสูงสุดที่อัตราขยายต่างๆแสดงในตารางที่ 3.3

ตารางที่ 3.3 แฟกเตอร์ตัวคูณและแรงดันทดสอบสูงสุดที่อัตราขยายต่างๆ

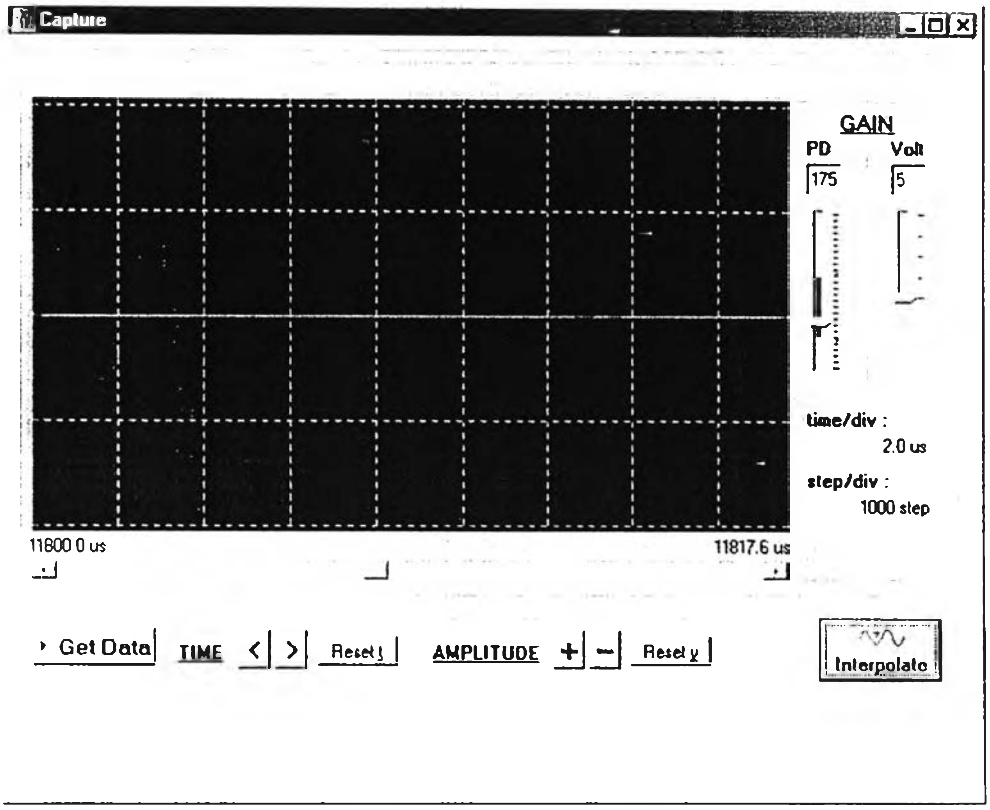
อัตราขยาย (เท่า)	1	2	4	8	16
แฟกเตอร์ตัวคูณ (V / step)	58.6	29.3	14.7	7.33	3.66
แรงดันทดสอบสูงสุด (kV_{rms})	84.9	42.4	21.2	10.6	5.30

3.5.2 ภาคแสดงรูปคลื่นสัญญาณดีสชาร์จบางส่วน

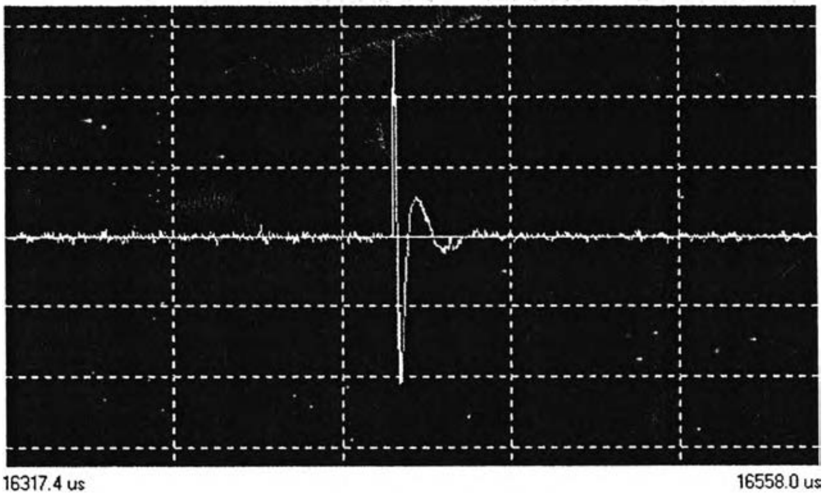
ภาคแสดงรูปคลื่นสัญญาณ PD ประกอบด้วยฟอร์ม 1 ฟอร์มทำหน้าที่แสดงรูปสัญญาณ PD มีส่วนแสดงผลประกอบด้วย กรอบแสดงภาพสัญญาณ PD ที่ได้จากการชักตัวอย่างด้วยอัตรา 5 ล้านตัวอย่างต่อวินาที และปุ่มปรับค่าเพื่อควบคุมการทำงานดังรูปที่ 3.25 ฟอร์มนี้สามารถเรียกใช้งานโดยการกดปุ่ม"Capture" ที่อยู่บนฟอร์มหลัก โดยรายละเอียดของฟังก์ชันต่างๆมีดังต่อไปนี้

1) การสั่งการตรวจจับและบันทึกรูปคลื่น โดยการกดปุ่ม"Get Data" โปรแกรมส่งคำสั่งผ่านดีไวซ์ไดรเวอร์ไปยังฮาร์ดแวร์ให้เริ่มทำการชักตัวอย่างข้อมูล หลังจากฮาร์ดแวร์บันทึกข้อมูลจนครบแล้วจะส่งสัญญาณอินเตอร์รัพต์มายังดีไวซ์ไดรเวอร์ซึ่งจะนำข้อมูลที่ได้ไปเก็บไว้ที่หน่วยความจำในคอมพิวเตอร์ เพื่อให้โปรแกรมมาอ่านค่าของข้อมูลไปแสดงผล ดังรูปที่ 3.26

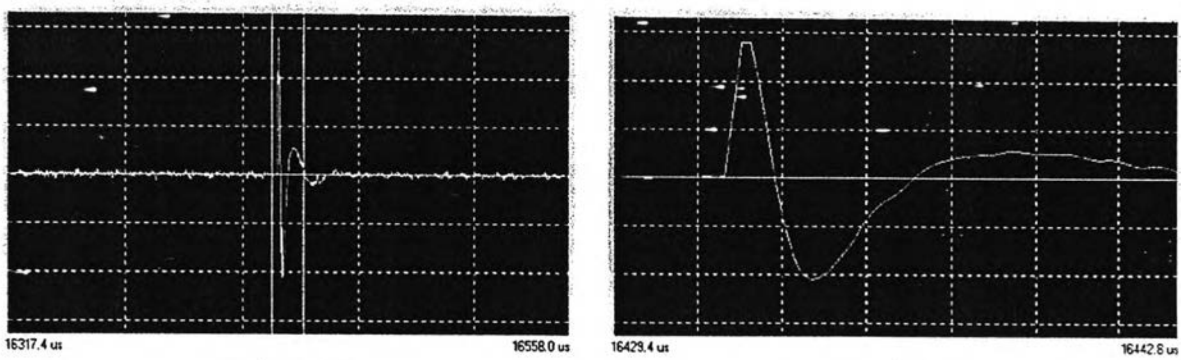
2) การขยายภาพในแกนเวลา ทำได้โดยคลิกเมาส์ในกรอบแสดงภาพที่ตำแหน่งเริ่มต้นของบริเวณที่ต้องการขยาย แล้วลากเมาส์ไปปล่อย ณ ตำแหน่งสิ้นสุด โปรแกรมจะทำการขยายภาพในบริเวณที่เลือกให้เต็มความกว้างของกรอบ ดังรูปที่ 3.27



รูปที่ 3.25 ส่วนแสดงผลของภาคแสดงรูปสัญญาณ PD



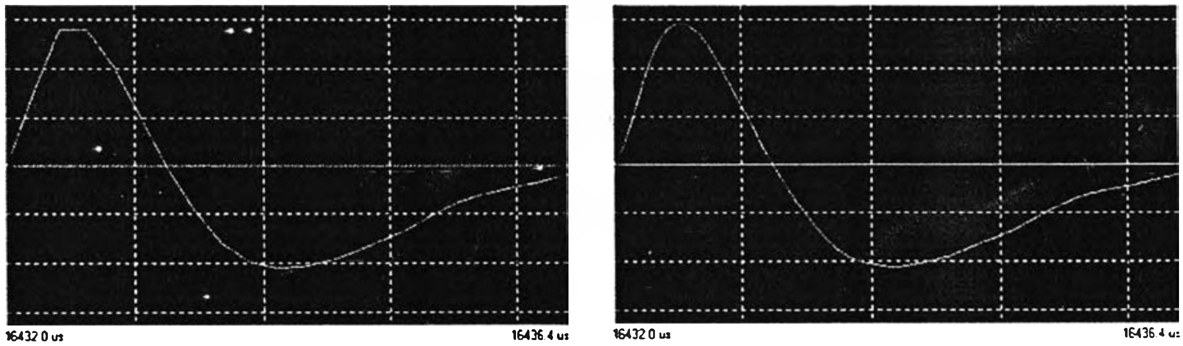
รูปที่ 3.26 การแสดงรูปสัญญาณ PD ที่บันทึกได้



ก) ก่อนขยาย ข) หลังขยาย

รูปที่ 3.27 การขยายภาพในแกนเวลา

3) การอินเตอร์โพเลต โดยการกดปุ่ม"Interpolate" โปรแกรมจะทำการคำนวณข้อมูลที่จะนำมาแทรกเพื่อให้เห็นภาพสัญญาณเสมือนว่าใช้อัตราชักตัวอย่างสูงกว่าเดิม 8 เท่า ดังแสดงในรูปที่ 3.28 รายละเอียดการคำนวณมีในหัวข้อ *3.2.3* ในเรื่อง *เทคนิคการอินเตอร์โพเลต*



ก) ก่อนทำการอินเตอร์โพเลต ข) หลังจากทำการอินเตอร์โพเลต

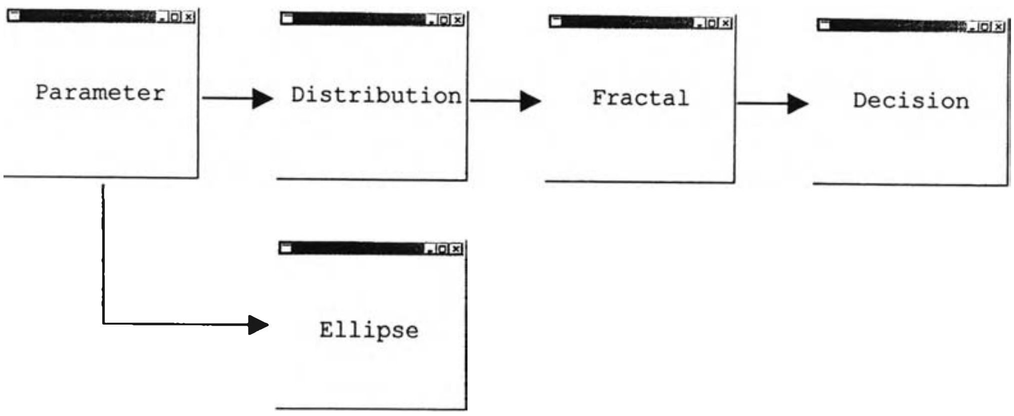
รูปที่ 3.28 การอินเตอร์โพเลต

3.4.3 **ภาควิเคราะห์ดีสชาร์จบางส่วน**

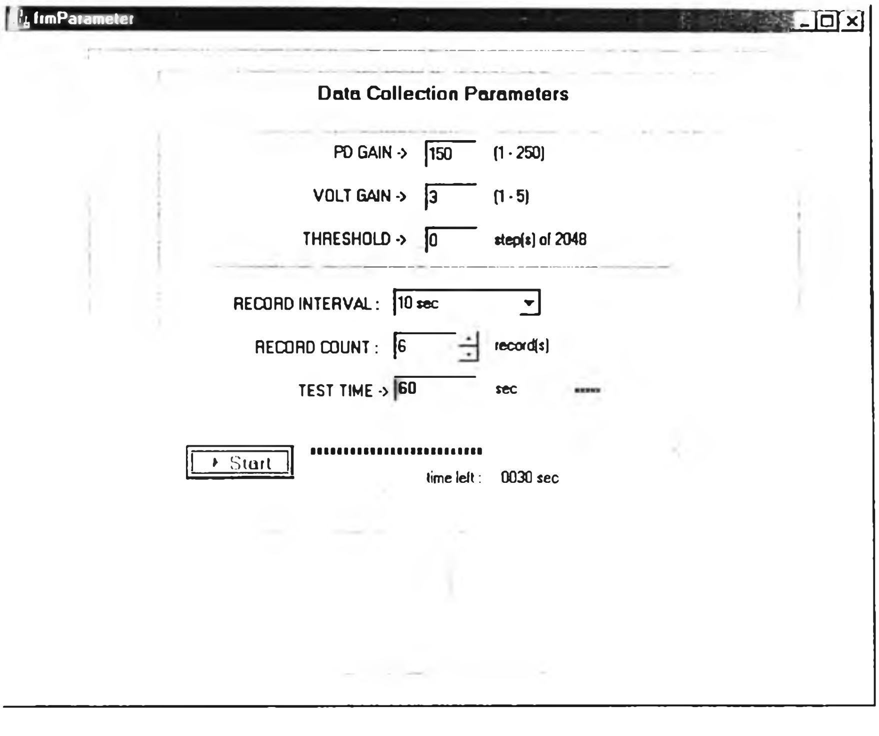
ซอฟต์แวร์ที่ทำหน้าที่ควบคุมการทำงานและวิเคราะห์สัญญาณ PD มีส่วนแสดงผลประกอบด้วย ฟอร์มทั้งหมด 5 ฟอร์ม แต่ละฟอร์มจะมีกรอบแสดงภาพและปุ่มปรับค่าสำหรับควบคุมการทำงาน โดยมีลำดับของการเรียกใช้ฟอร์มต่างๆดังรูปที่ 3.29

3.5.3.1 **ฟอร์มตั้งค่าการบันทึกสัญญาณ PD**

ฟอร์ม "Parameter" จะแสดงเป็นฟอร์มแรกเมื่อกดปุ่ม "Analyzer" ที่อยู่บนฟอร์มหลักดังรูปที่ 3.30 โดยมีรายละเอียดของฟังก์ชันต่างๆ ดังต่อไปนี้



รูปที่ 3.29 ลำดับของการเรียกใช้ฟอร์มต่างๆในการวิเคราะห์ PD

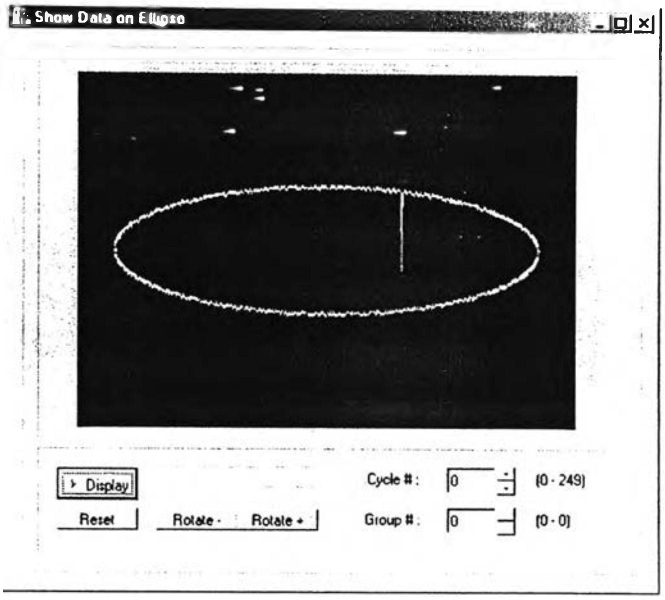


รูปที่ 3.30 ส่วนแสดงผลของฟอร์มตั้งค่าการบันทึกสัญญาณ PD

1) การตั้งค่าระยะเวลาบันทึกข้อมูลและจำนวนระเบียน (record) ของข้อมูล โดยเลือกค่าที่ต้องการจาก Combo box โปรแกรมจะคำนวณระยะเวลาทั้งหมดและแสดงค่าในหน่วยวินาที

2) การเริ่มบันทึกข้อมูล โดยการกดปุ่ม"Start" โปรแกรมจะแสดงเวลาแบบนับถอยหลังจนเสร็จสิ้นกระบวนการบันทึกข้อมูล

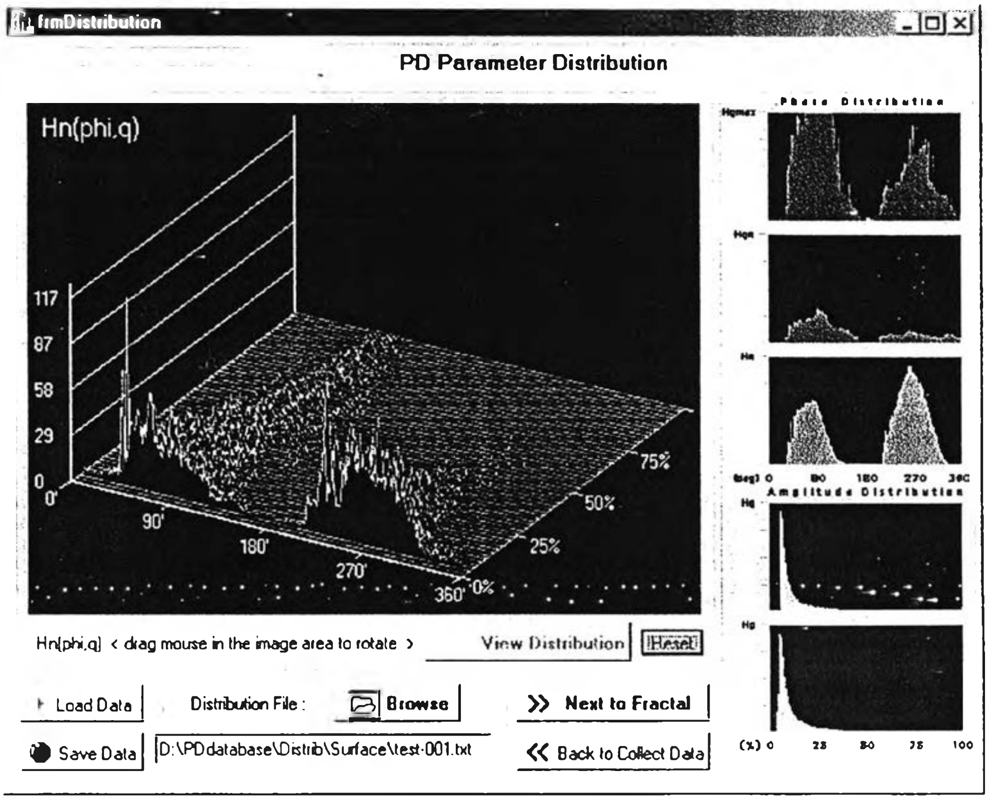
3) การแสดงผลข้อมูลที่บันทึกได้ จะแสดงบนฐานเวลารูปวงรีในฟอร์ม "Ellipse" ซึ่งเป็นฟอร์มย่อยของฟอร์ม "Parameter" การหมุนวงรีทำได้โดยการกดปุ่ม"Rotate+" และ"Rotate-" ดังแสดงในรูปที่ 3.31



รูปที่ 3.31 ฟอร์มแสดงผลข้อมูลที่บันทึกได้บนฐานเวลารูปวงรี

3.5.3.2 **ฟอร์มแสดงการกระจายของข้อมูล PD**

ฟอร์ม "Distribution" จะแสดงเป็นลำดับต่อมาเมื่อกดปุ่ม "Next to Distribution" ซึ่งอยู่บนฟอร์ม "Parameter" ดังรูปที่ 3.32 โดยมีรายละเอียดของฟังก์ชันต่างๆ ดังต่อไปนี้



รูปที่ 3.32 ส่วนแสดงผลของฟอร์มแสดงการกระจายของข้อมูล PD

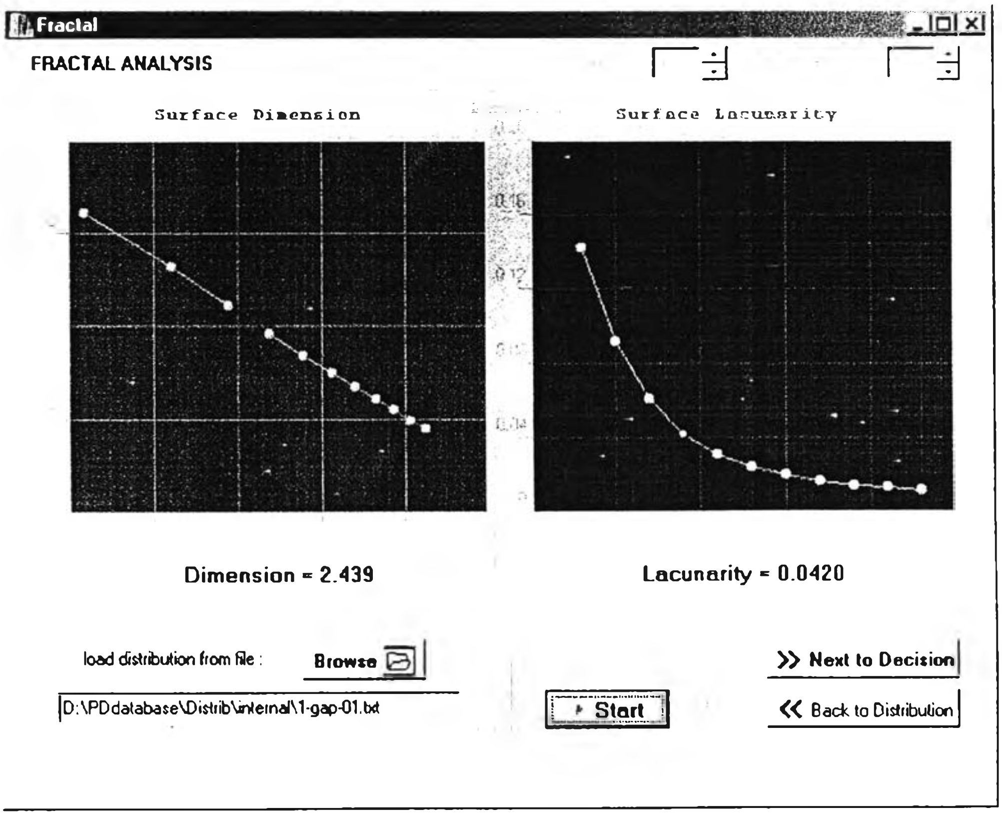
1) การโหลดข้อมูลการกระจายของ PD โดยการกดปุ่ม "Load Data" โปรแกรมจะโหลดข้อมูลสัญญาณ PD จากไฟล์ pddata.anl ที่บันทึกไว้มาแจกแจงตามขนาดประจุและมุมเฟสที่เกิด PD และเก็บข้อมูลการกระจายที่คำนวณได้ไว้ในหน่วยความจำ

2) การบันทึกข้อมูลการกระจายของ PD โดยการกดปุ่ม "Save Data" โปรแกรมจะบันทึกข้อมูลการกระจายที่อยู่ในหน่วยความจำลงสู่ไฟล์ข้อมูลการกระจายของ PD

3) การแสดงผลการกระจายของ PD โดยการกดปุ่ม "View Distribution" โปรแกรมจะนำข้อมูลไปแสดงผลในกรอบแสดงภาพหลัก และกรอบแสดงภาพรองอีก 5 กรอบ โดยในกรอบหลักจะแสดงการกระจายแบบ 3 มิติ ส่วนกรอบรองจะแสดงการกระจายแบบ 2 มิติ ซึ่งมีรายละเอียดการคำนวณในหัวข้อ *2.6.1.1* ในเรื่อง *การกระจาย* $H_n(\phi,q)$

3.5.3.3 **ฟอร์มแสดงคุณลักษณะแฟรกตัล**

ฟอร์ม "Fractal" จะแสดงต่อจากฟอร์ม "Distribution" เมื่อกดปุ่ม "Next to Fractal" ซึ่งอยู่บนฟอร์ม "Distribution" ดังรูปที่ 3.33 ฟังก์ชันของฟอร์มนี้คือคำนวณค่าคุณลักษณะแฟรกตัลจากข้อมูลการกระจายของ PD โดยมีรายละเอียดการคำนวณในหัวข้อ *2.6.2 การคำนวณคุณลักษณะจากรูปแบบที่ได้*



รูปที่ 3.33 ส่วนแสดงผลของฟอร์มแสดงคุณลักษณะแฟรกตัล

3.5.3.4 **ฟอร์มการจำแนกประเภทของ** PD

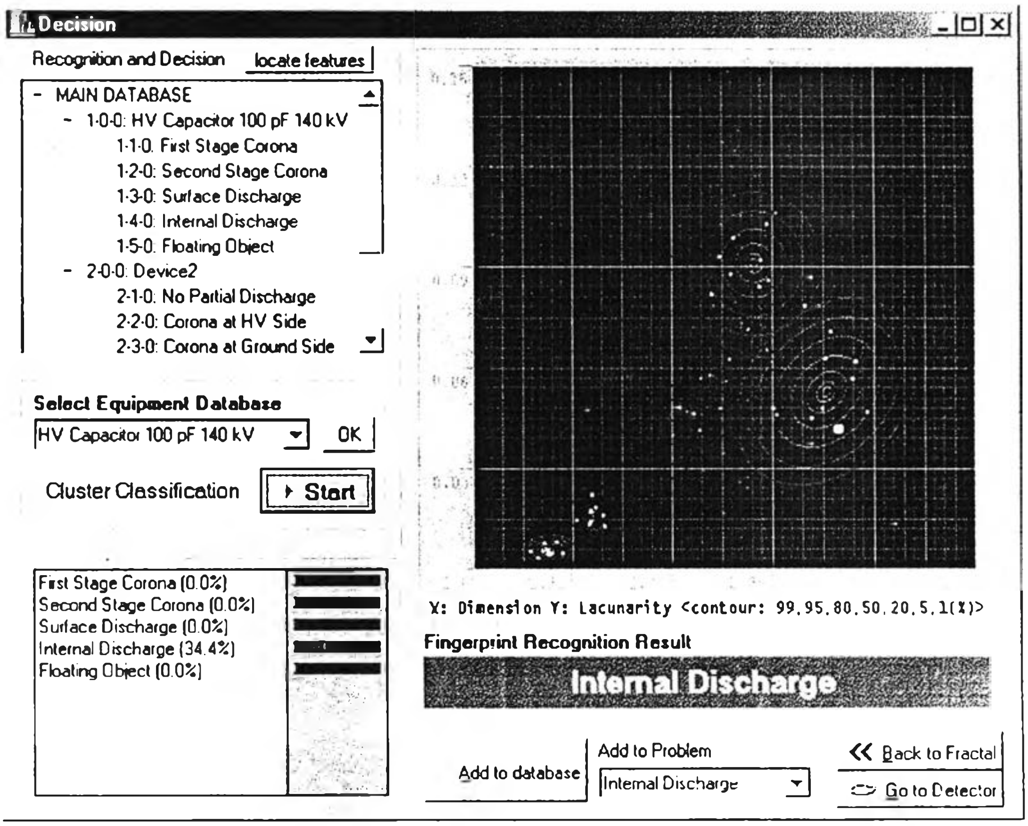
ฟอร์ม "Decision" **จะแสดงต่อจากฟอร์ม** "Fractal" **เมื่อกดปุ่ม** "Next to Decision" **ซึ่งอยู่บนฟอร์ม** "Fractal" **ดังรูปที่** 3.34 **ฟังก์ชันของฟอร์มนี้คือ**

1) การแสดงฐานข้อมูลของสาเหตุการเกิด PD ในอุปกรณ์ไฟฟ้าต่างๆใน Tree view

2) การคำนวณค่าเปอร์เซ็นต์ไทล์ของข้อมูลที่นำมาจำแนกสาเหตุ โดยมีรายละเอียดการคำนวณในหัวข้อ *2.6.4 การจำแนกประเภท*

3) การแสดงผลการจำแนกเป็นค่าเปอร์เซ็นต์ไทล์ของข้อมูล เมื่อคิดเทียบกับกลุ่มข้อมูลที่เป็นของสาเหตุต่างๆที่มีในฐานข้อมูล

4) การระบุสาเหตุ โดยดูจากค่าเปอร์เซ็นต์ไทล์ที่สูงสุดเป็นเกณฑ์ กรณีเปอร์เซ็นต์ไทล์ที่คำนวณได้สำหรับทุกกลุ่มข้อมูลต่ำกว่าค่าที่กำหนดไว้คือ 10 เปอร์เซ็นต์ หรือเปอร์เซ็นต์ไทล์ที่สูงสุดสองอันดับแรกมีค่าใกล้เคียงกันมาก (ต่างกันน้อยกว่า 10 เปอร์เซ็นต์) โปรแกรมจะแสดงข้อความเตือนเพื่อให้ผู้ทดสอบพิจารณาข้อมูลอื่นประกอบการตัดสินใจด้วย



รูปที่ 3.34 ส่วนแสดงผลของฟอร์มการจำแนกประเภทของ PD