

รายงานการวิจัยฉบับสมบูรณ์

โครงการวิจัย

การอินเตอร์เฟซเอนโค้ดเดอร์แบบเพิ่มค่าได้ด้วยเอฟพีจีเอ

Incremental Encoder interface using FPGA

คณะผู้วิจัย

ดร. ปัญญาศ ไชยกาพ

นาย กรกต สุวรรณรัตน์

รศ. ดร. มนตรี กาญจนะเดชะ

ผศ. ดร. ชเนศ เการพาทวงศ์

กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่ได้ให้
ทุนสนับสนุนโครงการวิจัยนี้ และขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะ
วิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่ได้ให้การสนับสนุนในเรื่องพื้นที่ทำงาน
และเครื่องมือต่างๆ ที่ใช้ในงานวิจัยนี้

คณะผู้วิจัย

บทคัดย่อ

ในการควบคุมอัตราเร็วการหมุนนั้นสามารถควบคุมได้ทั้งตำแหน่งและอัตราเร็วเชิงมุม ซึ่งการที่จะควบคุมตำแหน่งหรืออัตราเร็วเชิงมุมได้จะต้องมีการป้อนกลับค่าตำแหน่งหรืออัตราเร็วของระบบไปที่วงจรถวลุมเพื่อคำนวณค่าที่จะชดเชยให้กับระบบ การวัดตำแหน่งและอัตราเร็วการหมุนเชิงมุมนิยมใช้เอนโค้ดเดอร์แบบชนิดเพิ่มค่า ในงานควบคุมทั่วไปนิยมต่อเอนโค้ดเดอร์กับไมโครโพรเซสเซอร์โดยตรง ซึ่งทำให้ภาระงานตกอยู่กับไมโครโพรเซสเซอร์ ทำให้อัตราเร็วในการควบคุมระบบงานลดลง งานวิจัยนี้จึงมีความสนใจในการออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์แบบความเร็วสูงหลายช่องสัญญาณด้วยพีแอลดีชนิดเอพพีจีเอ เพื่อลดภาระงานของไมโครโพรเซสเซอร์

วงจรรวมที่ออกแบบนั้นจะออกแบบด้วยภาษา VHDL และทำการโปรแกรมการทำงานบนเอพพีจีเอ ซึ่งเป็นวงจรรวมที่สามารถรองรับการเชื่อมต่อเอนโค้ดเดอร์ได้ 2 ตัว ซึ่งจะทำให้ไมโครโพรเซสเซอร์หนึ่งตัวสามารถควบคุมการหมุนได้จำนวนมากขึ้น งานวิจัยนี้ประกอบด้วยวงจรร้อยในการวัดตำแหน่ง วงจรวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ และวงจรวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ โดยแต่ละช่องสัญญาณจะประกอบไปด้วยวงจรร้อยที่แยกจากกัน แต่ใช้การควบคุม หน่วยความจำและช่องทางนำเข้าและส่งออกข้อมูลระหว่างไมโครโพรเซสเซอร์กับวงจรรวมขนาด 32 บิตที่สามารถใช้งานแบบ 8 บิต และ 16 บิตได้ชุดเดียวกัน ซึ่งวงจรรวมนี้สามารถรองรับอัตราเร็วการหมุนของเอนโค้ดเดอร์ที่มีความถี่สูงถึง 150 kHz ได้

จากผลการทดสอบการวัดตำแหน่ง วงจรสามารถวัดค่าตำแหน่งโดยมีความผิดพลาดของจำนวนจังหวะเคลื่อนที่ไม่เกิน $\frac{1}{4}$ ของคาบของสัญญาณเอนโค้ดเดอร์ ผลการทดสอบการวัดความเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์จะเกิดค่าความผิดพลาดต่ำเมื่อสัญญาณเอนโค้ดเดอร์มีความถี่ต่ำ ในขณะที่การวัดความเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์จะเกิดค่าความผิดพลาดต่ำเมื่อสัญญาณเอนโค้ดเดอร์มีความถี่สูง จากการทดสอบพบว่าจุดตัดความถี่ที่ให้ค่าความผิดพลาดที่น้อยที่สุดคือ 1550 Hz ดังนั้นจึงเลือกใช้วิธีวัดอัตราเร็วด้วยการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์เมื่อความถี่สัญญาณเอนโค้ดเดอร์ต่ำกว่า 1550 Hz แต่เมื่อความถี่สูงกว่า 1550 Hz จะใช้วิธีวัดความถี่สัญญาณเอนโค้ดเดอร์แทน โดยวงจรที่ได้มีความผิดพลาดในการคำนวณหาอัตราเร็วไม่เกิน 0.0044 %

คำสำคัญ Incremental Encoder, Position meter, Period meter, Frequency meter, VHDL, FPGA, PLD

Abstract

Position and velocity estimations are important for the feedback control of rotary systems. Incremental encoders are employed in many applications due to their reliability, availability, and insensitivity to external disturbance. In general, the encoder is directly connected to a microprocessor of the control system. This leverages the workload of the microprocessor and results in slowing down the execution speed of the microprocessor. With these limitations, the work in this report focuses on designing a high speed and multi-channel incremental encoder interfacing using PLD in order to reduce the workload of microprocessor.

The circuit is designed with VHDL language, and programmed to FPGA. It supports two encoders. The circuit has three sub-circuits, the position meter, the period meter, and the frequency meter. Each channel has its own circuit to estimate the position and velocity. The control unit, the register memory and the 32 bit I/O port are shared resources for both channels. The circuit is capable of estimating the position and velocity when the encoder frequency is not greater than 150 kHz.

With respect to the test results, the circuit gives a maximum position error of a quarter of the encoder signal's period. For velocity estimation, the period meter method yields low error at low encoder frequency. While the frequency meter yields low error at high encoder frequency. For lowest estimation error across the encoder frequency range, the period meter method is used when the encoder signal frequency is less than 1550 Hz and the frequency meter method is used when encoder signal frequency is greater than 1550 Hz. The circuit gives a velocity estimation error of no greater than 0.0044%.

Keywords: Incremental Encoder, Position meter, Period meter, Frequency meter, VHDL, FPGA, PLD

สารบัญ

กิตติกรรมประกาศ	i
บทคัดย่อ	ii
Abstract	iii
สารบัญ	iv
บทที่ 1 บทนำ	
1.1 ความสำคัญและที่มาของหัวข้อวิจัย	1
1.2 การตรวจเอกสาร	2
1.3 วัตถุประสงค์	3
1.4 ขอบเขตของการวิจัย	3
1.5 ขั้นตอนและวิธีการวิจัย	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 หลักการทำงานและแนวความคิด	5
2.1 แนวคิดในการออกแบบวงจร	5
2.2 ตัวตรวจรู้ที่ใช้ในการวัดตำแหน่งและอัตราเร็วการหมุน	5
2.3 การคำนวณหาตำแหน่ง	7
2.4 การวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์	9
2.5 การออกแบบด้วยเอฟพีจีเอ	11
บทที่ 3 โครงสร้างของวงจร	15
3.1 โครงสร้างหลักของวงจร	15
3.2 ส่วนปรับแต่งสัญญาณนำเข้าและวัดค่า	16
3.3 ส่วนคำนวณ	29
3.4 ส่วนเก็บข้อมูล	32
3.5 ส่วนติดต่อกับอุปกรณ์ภายนอก	32
บทที่ 4 การทดสอบการทำงานของเอฟพีจีเอ	35
4.1 ทรัพยากรที่ใช้ในการทดสอบ	35
4.2 การทดสอบการวัดตำแหน่ง	36
4.3 การทดสอบการวัดอัตราเร็วการหมุน	38
บทที่ 5 ผลและการอภิปรายผลการทดลอง	40
5.1 ผลการทดสอบการวัดตำแหน่ง	40
5.2 ผลการทดสอบวัดอัตราเร็วการหมุน	44

บทที่ 6 สรุปผลและข้อเสนอแนะ	51
6.1 บทสรุป	51
6.2 ข้อเสนอแนะ	52
เอกสารอ้างอิง	54
ภาคผนวก ก ผลงานทางวิชาการที่ได้รับการตีพิมพ์	56

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของหัวข้อวิจัย

ในการควบคุมอัตราเร็วการหมุนนั้นสามารถควบคุมได้ทั้งตำแหน่งและอัตราเร็วเชิงมุม (Angular position and angular velocity) ซึ่งการที่จะควบคุมตำแหน่งหรืออัตราเร็วเชิงมุมได้นั้นต้องมีการป้อนกลับค่าตำแหน่งหรืออัตราเร็วเชิงมุมของระบบไปที่วงจรควบคุมเพื่อคำนวณค่าที่จะชดเชยให้กับระบบ เพื่อให้ระบบหมุนไปตำแหน่งหรือได้อัตราเร็วเชิงมุมที่ต้องการ

ตัวตรวจรู้ (Sensor) ที่ใช้กันมากในงานเกี่ยวกับการควบคุมตำแหน่งและอัตราเร็วเชิงมุมคือเอนโค้ดเดอร์แบบหมุน (Rotary Encoder) ซึ่งมีอยู่ 3 แบบ คือ เอนโค้ดเดอร์แบบหมุนชนิดให้ผลสมบูรณ์ (Absolute Rotary Encoder) ที่ให้ค่าตำแหน่งการหมุนของแกน ข้อมูลที่ได้จะเป็นค่าตำแหน่งที่จุดนั้นๆ ซึ่งข้อมูลอาจอยู่ในรูปพีซีดี (Binary Code Decimal: BCD) หรือโค้ดเกรย์ (Gray Code), เอนโค้ดเดอร์แบบหมุนชนิดหลายรอบ (Multi-turn Rotary Encoder) ซึ่งจะคล้ายกับ เอนโค้ดเดอร์แบบหมุนชนิดให้ผลสมบูรณ์แต่สามารถอ่านค่าจากการหมุนในหลายๆ รอบได้ และเอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่า (Incremental Rotary Encoder) ซึ่งเป็นแบบที่นิยมใช้กันมากที่สุด โดยเอาต์พุตของเอนโค้ดเดอร์จะให้สัญญาณพัลส์ (Pulse) ออกมาทุกครั้งที่มีการหมุนของแกนเอนโค้ดเดอร์ ทำให้ทราบได้ว่าแกนของเอนโค้ดเดอร์ หมุนไปถึงจังหวะ เอาต์พุตของเอนโค้ดเดอร์แบบนี้จะให้สัญญาณออกมา 1-3 บิต (Bit) ถ้าเป็นแบบ 2 บิตสัญญาณเอาต์พุตทั้งสองจะอยู่ในรูปของเฟสที่ต่างกัน 90 องศา ทำให้สามารถทราบทิศทางการหมุนของแกนเอนโค้ดเดอร์ได้ ส่วนสัญญาณเอาต์พุตบิตที่ 3 จะเป็นสัญญาณอินเด็กซ์ (Index) ซึ่งจะให้พัลส์ เมื่อเอนโค้ดเดอร์หมุนครบ 360 องศาแล้ว

โดยทั่วไป เอาต์พุตของเอนโค้ดเดอร์จะต่อกับไมโครโพรเซสเซอร์ (Microprocessor) หรือไมโครคอนโทรลเลอร์ (Microcontroller) เพื่อให้ไมโครโพรเซสเซอร์หรือไมโครคอนโทรลเลอร์ ทำการสุ่มสัญญาณจากเอนโค้ดเดอร์เพื่อประมวลผลตามโปรแกรมคำนวณหาตำแหน่งหรืออัตราเร็ว เพื่อใช้เป็นพารามิเตอร์หนึ่งในการคำนวณค่าชดเชยในการควบคุมต่อไป โดยในขั้นตอนการคำนวณจะเสียเวลาการทำงานของไมโครโพรเซสเซอร์มากและส่งผลเสียต่อระบบงานควบคุมที่ต้องการอัตราเร็วสูง อันทำให้ต้องใช้ไมโครโพรเซสเซอร์ที่มีสมรรถนะสูงและมีราคาแพงรวมทั้งเสียเวลาในการเขียนอัลกอริทึม (Algorithm) สำหรับการสุ่ม

สัญญาณไปคำนวณ การคำนวณหาค่าตำแหน่ง ทิศทาง และอัตราเร็วของมอเตอร์ ดังนั้นจึงเกิดแนวคิดที่จะสร้างวงจรรวม (Integrated Circuit: IC) สำหรับเชื่อมต่อตัวเอนโค้ดเดอร์กับไมโครโพรเซสเซอร์ เพื่อลดภาระการคำนวณค่าตำแหน่งหรืออัตราเร็วเชิงมุมของหน่วยประมวลผลกลาง (Central Processing Unit: CPU)

โดยวงจรรวมที่จะสร้างขึ้นนั้นจะเป็นวงจรรวมที่สามารถรองรับการเชื่อมต่อระหว่างเอนโค้ดเดอร์แบบสัญญาณอินพุต 3 บิต กับไมโครโพรเซสเซอร์ โดยที่ยังสามารถคำนวณค่าตำแหน่งและอัตราเร็วเชิงมุมของเอนโค้ดเดอร์ด้วยอัตราเร็วสูง

1.2 การตรวจเอกสาร

1.2.1 A PLD Based Interface Velocity with Accurate Position and Velocity Estimation [1] บทความนี้ได้แสดงถึง วิธีการจัดการกับสัญญาณพัลส์ที่มาจากเอนโค้ดเดอร์ ซึ่งจะเป็นที่มาของข้อมูลที่จะนำไปคำนวณหาตำแหน่งและอัตราเร็วการหมุนเชิงมุม โดยพัฒนางจรด้วยภาษา VHDL (VHSIC Hardware Description Language: VHDL) เพื่อเชื่อมต่อกับไมโครคอนโทรลเลอร์ราคาถูก โดยนำเสนอวิธีการวัด 3 วิธีคือ Pulse Counting (PC) ซึ่งนับสัญญาณพัลส์จากเอนโค้ดเดอร์ในช่วงเวลาที่กำหนดไว้ Single Pulse Time Measurement (SPTM) เป็นการหาค่าเวลาความกว้างคาบของสัญญาณจากเอนโค้ดเดอร์ และ Constant Elapsed Time (CET) เป็นการรวมวิธีการนับพัลส์จากวิธี PC และการวัดความกว้างคาบจากวิธี SPTM เพื่อกำจัดผลกระทบจาก Quantization ความไม่สมบูรณ์ของพัลส์

1.2.2 Single-Chip Velocity Measurement System for Incremental Optical Encoders [2] บทความนี้ได้แสดงถึง การออกแบบชิปเดี่ยว (Single Chip) ด้วยเอฟพีจีเอ (Field Programmable Gate Arrays: FPGA) โดยนำเสนอวิธีการวัดอัตราเร็ว 2 วิธี คือ Period Counting เป็นการวัดความกว้างคาบ ที่เป็นฟังก์ชันของความถี่สัญญาณจากเอนโค้ดเดอร์ และความถี่สัญญาณนาฬิกาของวงจรมัน และ Frequency Counting เป็นการวัดความถี่สัญญาณจากเอนโค้ดเดอร์ ในฟังก์ชันของความถี่เอนโค้ดเดอร์ และระยะเวลาการวัด (Time Interval)

1.2.3 Speed Measurement Using Rotary Encoders for High Performance ac Drivers [3] บทความนี้ได้แสดงถึง การวัดอัตราเร็วการหมุนเชิงมุม 2 วิธี คือ Period Meter เป็นการวัดความกว้างคาบ ที่เป็นฟังก์ชันความถี่มุมละเอียดยที่สุดของเอนโค้ดเดอร์ กับเวลาช่วงคาบที่วัดได้ ซึ่งให้ความถูกต้องสูงเมื่ออัตราเร็วการหมุนของเอนโค้ดเดอร์ต่ำ และ Frequency Meter เป็นการวัดมุม

ที่เอนโค้ดเดอร์เคลื่อนที่ได้ในช่วงเวลาที่กำหนด ซึ่งให้ความถูกต้องสูงเมื่ออัตราเร็วการหมุนของเอนโค้ดเดอร์สูง

1.3 วัตถุประสงค์

งานวิจัยนี้เป็นส่วนหนึ่งในการพัฒนางจรต้นแบบ ซึ่งงานวิจัยนี้ประกอบด้วย 2 วัตถุประสงค์หลักดังนี้

1.3.1 เพื่อศึกษาและออกแบบอัลกอริทึมในการสุ่มสัญญาณจากเอนโค้ดเดอร์ เพื่อทำการคำนวณค่าตำแหน่งและอัตราเร็วเชิงมุมได้อย่างรวดเร็ว

1.3.2 เพื่อพัฒนางจรรวมต้นแบบสำหรับอ่านค่าจากเอนโค้ดเดอร์ที่มีอัตราเร็วการหมุนสูงเพื่อใช้งานควบคุมการหมุน

1.4 ขอบเขตของการวิจัย

1.4.1 ออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่า ที่ให้สัญญาณเอาต์พุต 3 สัญญาณคือ สัญญาณเฟส A และเฟส B ที่ต่างกัน 90 องศา และสัญญาณอินดิเคซ์ ด้วยภาษา VHDL และบันทึกการทำงานของวงจรบนเอฟพีจีเอ

1.4.2 วงจรเชื่อมต่อเอนโค้ดเดอร์สามารถเชื่อมต่อเอนโค้ดเดอร์ที่สามารถรองรับพัลส์สัญญาณสูงสุด 150 kHz และสามารถเชื่อมต่อกับไมโครโพรเซสเซอร์ด้วยบัสขนาด 8, 16 และ 32 บิต ได้

1.4.3 วงจรที่ออกแบบจะประมวลผลแบบ 32 บิต แต่การติดต่อกับภายนอกจะเป็นแบบ 8 บิต

1.5 ขั้นตอนและวิธีการวิจัย

1.5.1 ศึกษามาตรฐานและหลักการทำงานของเอนโค้ดเดอร์แบบหมุน

1.5.2 ค้นหาบทความทางวิชาการที่เกี่ยวกับการเชื่อมต่อกับเอนโค้ดเดอร์ และข้อมูลของอุปกรณ์เชื่อมต่อเอนโค้ดเดอร์สำเร็จรูป

1.5.3 ออกแบบวงจรในระดับการส่งข้อมูลของรีจิสเตอร์ (Register Transfer Level: RTL) ด้วยภาษา VHDL และทวนสอบการทำงานของวงจรด้วยวิธีการจำลองการทำงาน (Functional Verification) และวิธีการจำลองเชิงเวลา (Timing Simulation)

1.5.4 สังเคราะห์วงจรที่ได้จากการออกแบบเป็นวงจรในระดับเกตและทวนสอบการทำงานของวงจรในระดับเกตด้วยวิธีจำลองการทำงานในระดับโพสต์เลย์เอาต์ (Post-Layout Simulation)

1.5.5 บันทึกวงจรที่ทดสอบจนแน่ใจถึงการทำงานที่ถูกต้องแล้วลงบนเอฟพีจีเอของบอร์ดพัฒนาเอฟพีจีเอ (FPGA Development Board)

1.5.6 ออกแบบและสร้างวงจรเชื่อมต่อระหว่างบอร์ดพัฒนาเอพพีจีเอกับเอนไอคเคอร์และทดสอบการทำงานของวงจรที่ได้กับตัวเอนไอคเคอร์

1.5.7 รวบรวมผลการทดสอบ สรุปผล จัดทำรายงานฉบับสมบูรณ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้วงจรต้นแบบสำหรับเชื่อมต่อกับเอนไอคเคอร์และหมุนด้วยอัตราเร็วสูง ซึ่งผู้ใช้สามารถโปรแกรมกำหนดรูปแบบการทำงานได้

1.6.2 ลดภาระการทำงานของโปรแกรมเมอร์ในการเขียนโปรแกรมควบคุมการทำงานไมโครคอนโทรลเลอร์

บทที่ 2

หลักการทํางานและแนวความคิด

งานวิจัยนี้มีจุดประสงค์เพื่อออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์ที่สามารถคำนวณค่าอัตราเร็วการหมุน และค่าตำแหน่งการเคลื่อนที่ได้ภายในตัววงจรเอง เพื่อลดภาระงานของไมโครโพรเซสเซอร์

2.1 แนวความคิดในการออกแบบวงจร

งานวิจัยนี้เสนอแนวความคิดในการออกแบบวงจรสำหรับเชื่อมต่อเอนโค้ดเดอร์กับไมโครโพรเซสเซอร์ เพื่อลดภาระงานของไมโครโพรเซสเซอร์ในการคำนวณหาตำแหน่งหรืออัตราเร็วการหมุนของเอนโค้ดเดอร์ วงจรที่ออกแบบจะทำหน้าที่คำนวณหาตำแหน่งการหมุนและอัตราเร็วการหมุนจากข้อมูลการหมุนของเอนโค้ดเดอร์ ซึ่งต่อกับเอพฟิเจีโดยตรง เมื่อเอพฟิเจีคำนวณค่าตำแหน่งและอัตราเร็วการหมุนได้แล้วจึงจะส่งค่าขึ้นไปยังไมโครโพรเซสเซอร์ดังภาพประกอบ 2-1



ภาพประกอบ 2-1 แนวคิดของการพัฒนาไอซีเอพฟิเจีต้นแบบ

2.2 ตัวตรวจรู้ที่ใช้ในการวัดตำแหน่งและอัตราเร็วการหมุน

ตัวตรวจรู้ที่นิยมนำมาใช้ในการวัดตำแหน่งและอัตราเร็วการหมุนคือเอนโค้ดเดอร์ ซึ่งแบบเอนโค้ดเดอร์นิยมนำมาประยุกต์เพื่อวัดตำแหน่งและอัตราเร็วการหมุนคือเอนโค้ดเดอร์แบบหมุนซึ่งมีอยู่ 3 ชนิด คือ

2.2.1 เอนโค้ดเดอร์แบบหมุนชนิดให้ผลสมบูรณ์

เอนโค้ดเดอร์แบบหมุนชนิดให้ผลสมบูรณ์ จะให้ค่าตำแหน่งการหมุนของแกน ค่าตำแหน่งที่อ่านได้จะเป็นค่าตำแหน่งที่จุดนั้นๆ ซึ่งอาจเป็นข้อมูลแบบบิตซีดี หรือไค้ดเกรย์

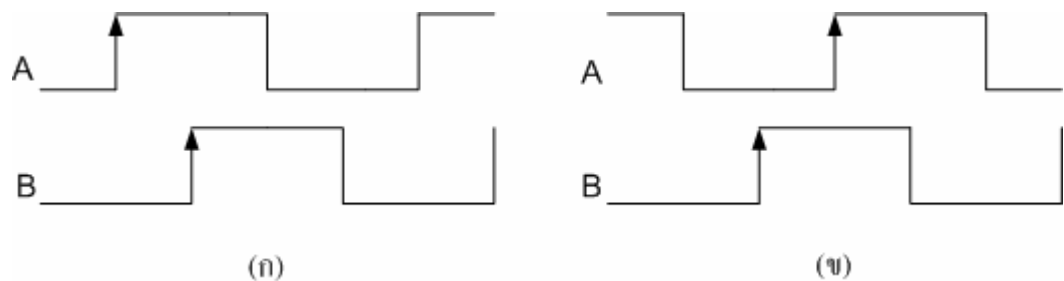
2.2.2 เอนโค้ดเดอร์แบบหมุนชนิดหลายรอบ

เอนโค้ดเดอร์แบบหมุนชนิดหลายรอบ จะมีการทำงานคล้ายกับเอนโค้ดเดอร์แบบหมุนชนิดให้ผลสมบูรณ์ คือค่าที่อ่านได้จากเอนโค้ดเดอร์จะเป็นค่าที่ตำแหน่งนั้นๆของการหมุน แต่เอนโค้ดเดอร์แบบหมุนชนิดหลายรอบสามารถอ่านค่าจากการหมุนในหลายๆ รอบได้

2.2.3 เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่า

เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่า เป็นตัวตรวจรู้ที่นิยมนำมาประยุกต์ใช้งานเพื่อวัดตำแหน่งและอัตราเร็วการหมุนมากที่สุด เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าจะให้สัญญาณเอาต์พุตเป็นสัญญาณพัลส์ที่มีความกว้างพัลส์ต่างกันขึ้นอยู่กับอัตราเร็วการหมุนของแกนเอนโค้ดเดอร์ ถ้าแกนของเอนโค้ดเดอร์หมุนเข้าความกว้างพัลส์จะมาก แต่ถ้าหมุนเร็วขึ้นจะให้ความกว้างพัลส์แคบลง

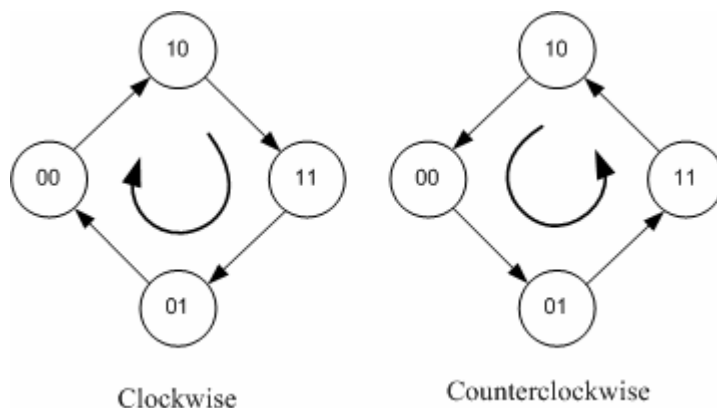
เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าเป็นอุปกรณ์สำหรับตรวจสอบทิศทางหมุนและจำนวนรอบการหมุน ซึ่งนิยมประยุกต์ใช้วัดตำแหน่งและอัตราเร็วการหมุน โดยทั่วไป เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าจะให้สัญญาณเอาต์พุตอย่างน้อย 2 สัญญาณ ซึ่งทั้งสองสัญญาณจะมีเฟสต่างกัน 90 องศา ดังภาพประกอบ 2-2



ภาพประกอบ 2-2 ก เอาต์พุต 2 สัญญาณที่ต่างกัน 90 องศา แบบสัญญาณ A นำสัญญาณ B

ภาพประกอบ 2-2 ข เอาต์พุต 2 สัญญาณที่ต่างกัน 90 องศา แบบสัญญาณ B นำสัญญาณ A

จากภาพประกอบ 2-2 ความแตกต่างของการนำระหว่างสัญญาณ A และสัญญาณ B สามารถแยกได้ว่าเป็นการหมุนตามเข็มนาฬิกาหรือเป็นการหมุนทวนเข็มนาฬิกา ซึ่งสามารถเขียนสถานะการหมุนของเอนโค้ดเดอร์ได้ดังภาพประกอบ 2-3 ตัวอย่าง สมมุติให้ค่าสัญญาณ B และสัญญาณ A เริ่มต้นเป็น “00” ตามลำดับ ถ้าสถานะต่อไปเป็น “10” และ “11” แล้วจะได้ว่าเอนโค้ดเดอร์หมุนแบบตามเข็มนาฬิกา ถ้าสถานะต่อไปเป็น “01” แสดงว่าเอนโค้ดเดอร์ยังหมุนตามเข็มนาฬิกาต่อไป แต่ถ้าสถานะที่ต่อจาก “11” เป็น “10” แสดงว่าเอนโค้ดเดอร์กำลังเปลี่ยนทิศทางเป็นหมุนทวนเข็มนาฬิกา



ภาพประกอบ 2-3 สถานะการหมุนตามและทวนเข็มนาฬิกาของเอนโค้ดเดอร์

สำหรับเอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าที่ให้สัญญาณเอาต์พุต 3 สัญญาณ นั้นจะเหมือนกับแบบให้สัญญาณเอาต์พุต 2 สัญญาณ แต่เพิ่มสัญญาณอินเด็กซ์ ซึ่งเอนโค้ดเดอร์จะสร้างสัญญาณพัลส์อินเด็กซ์นี้เมื่อเอนโค้ดเดอร์หมุนครบรอบ

ความละเอียดของเอนโค้ดเดอร์แต่ละตัวไม่เท่ากัน ซึ่งขึ้นอยู่กับจำนวนพัลส์ที่เอนโค้ดเดอร์สร้างขึ้นในแต่ละรอบ (Pulses Per Round: PPR) ของเอนโค้ดเดอร์ตัวนั้นๆ

งานวิจัยนี้ได้เลือกใช้เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าในการวัดตำแหน่งและอัตราเร็วการหมุน ซึ่งจำเป็นจะต้องมีการประมวลสัญญาณพัลส์จากเอนโค้ดเดอร์เพื่อคำนวณหาตำแหน่งและอัตราเร็วการหมุนของเอนโค้ดเดอร์

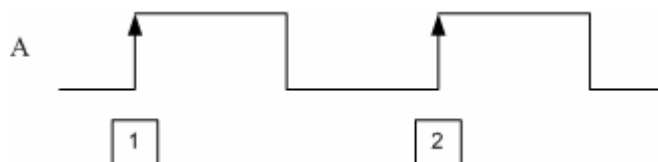
2.3 การคำนวณหาตำแหน่ง

การหาตำแหน่งการหมุนโดยใช้เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่า เป็นการนับจำนวนพัลส์ที่เกิดจากการหมุนของเอนโค้ดเดอร์ ซึ่งผู้ใช้งานจะต้องคำนวณความสัมพันธ์ระหว่างจำนวนพัลส์ต่อรอบของเอนโค้ดเดอร์ ต่อระยะทางที่เกิดขึ้นจริงจากการหมุนของระบบงาน เพื่อให้การวัดตำแหน่งได้รายละเอียดหรือช่วงระยะที่วัดได้ละเอียดมากขึ้น สามารถทำได้จากความสัมพันธ์ของสัญญาณจากเอนโค้ดเดอร์ทั้งสองสัญญาณ

2.3.1 ความละเอียด 1 เท่า

การวัดตำแหน่งโดยใช้เอนโค้ดเดอร์ด้วยความละเอียด 1 เท่า เป็นการวัดโดยใช้สัญญาณจากเอนโค้ดเดอร์สัญญาณเดียว ซึ่งจะเหมาะสมกับระบบงานที่ไม่ต้องการความละเอียดมากหรือเมื่อเอนโค้ดเดอร์ที่ใช้เป็นแบบให้สัญญาณเอาต์พุตสัญญาณเดียว

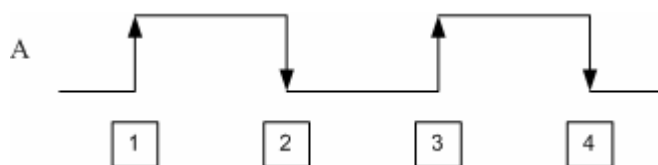
การวัดตำแหน่งด้วยความละเอียด 1 เท่า เป็นการวัดที่ทั้งขอบขาขึ้นหรือขอบขาลงอย่างใดอย่างหนึ่งของสัญญาณจากเอนโค้ดเดอร์ ดังภาพประกอบ 2-4



ภาพประกอบ 2-4 การวัดตำแหน่งด้วยความละเอียด 1 เท่า

2.3.2 ความละเอียด 2 เท่า

การวัดตำแหน่งโดยใช้เอนโค้ดเดอร์ด้วยความละเอียด 2 เท่า เป็นการวัดโดยใช้สัญญาณจากเอนโค้ดเดอร์สัญญาณเดียว เช่นเดียวกับการวัดด้วยความละเอียด 1 เท่า ซึ่งให้ความละเอียดหรือความแม่นยำมากขึ้น โดยเป็นการวัดที่ทั้งขอบขาขึ้นและขอบขาลง ดังภาพประกอบ 2-5

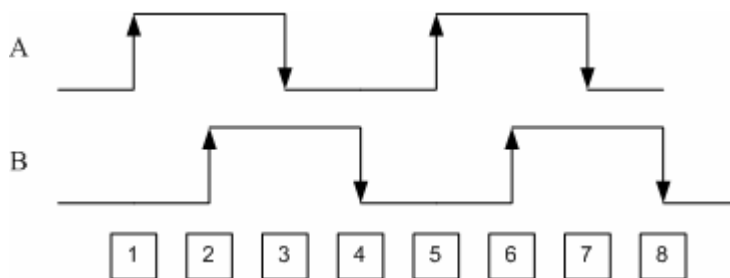


ภาพประกอบ 2-5 การวัดตำแหน่งด้วยความละเอียด 2 เท่า

2.3.3 ความละเอียด 4 เท่า

การวัดตำแหน่งโดยใช้สัญญาณเอนโค้ดเดอร์ด้วยความละเอียด 4 เท่า เป็นการวัดโดยใช้สัญญาณจากเอนโค้ดเดอร์ 2 สัญญาณ เอนโค้ดเดอร์แบบสัญญาณเดี่ยวไม่สามารถทำได้ ซึ่งจะเหมาะสมกับระบบงานที่ต้องการความละเอียดค่อนข้างสูงและใช้กับหน่วยประมวลผลที่มีความสามารถสูงกว่าการวัดด้วยความละเอียด 1 เท่า หรือ 2 เท่า

การวัดตำแหน่งด้วยความละเอียด 4 เท่า เป็นการวัดที่ทั้งขอบขาขึ้นและขอบขาลงของสัญญาณจากเอนโค้ดเดอร์ทั้ง 2 สัญญาณ ดังภาพประกอบ 2-6



ภาพประกอบ 2-6 การวัดตำแหน่งด้วยความละเอียด 4 เท่า

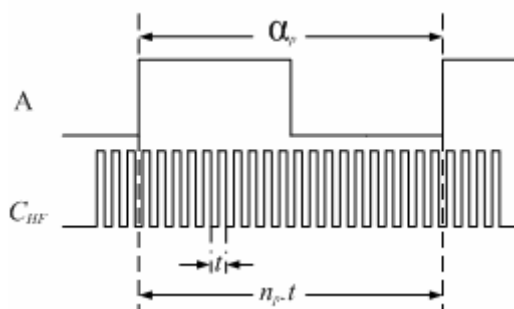
งานวิจัยนี้เน้นการนำเอนโค้ดเดอร์มาใช้ในระบบงานที่มีประสิทธิภาพสูงจึงเลือกออกแบบวิธีการวัดตำแหน่งด้วยความละเอียด 4 เท่า ซึ่งจะต้องออกแบบวิธีการจัดการสัญญาณจากเอนโค้ดเดอร์ทั้ง 2 สัญญาณ ให้ได้สัญญาณที่มีความละเอียด 4 เท่าต่อไป

2.4 การวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์

การวัดอัตราเร็วการหมุนของระบบ โดยใช้เอนโค้ดเดอร์แบบหมุนชนิดเพิ่มค่าเป็นตัวตรวจรู้ วิธีการวัดที่นิยมใช้มี 2 วิธี คือการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ (Period meter) และการวัดความถี่สัญญาณเอนโค้ดเดอร์ (Frequency meter) [1] [2] [3]

2.4.1 วิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์

การวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ เป็นการวัดความกว้างของช่วงหนึ่งคาบของสัญญาณเอนโค้ดเดอร์สัญญาณใดสัญญาณหนึ่งว่าใช้เวลาเท่าใด ดังภาพประกอบ 2-7



ภาพประกอบ 2-7 การวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์

ความกว้างคาบสัญญาณเอนโค้ดเดอร์เท่ากับ α_p สามารถประมาณค่าได้โดยใช้สัญญาณนาฬิกาความถี่สูง (C_{HF}) ที่สูงกว่าความถี่สัญญาณเอนโค้ดเดอร์มากๆ ซึ่งรู้ความถี่ที่แน่นอน มาใช้ในการนับจำนวนพัลส์สัญญาณนาฬิกาความถี่สูงในช่วง α_p จากภาพประกอบ 2-7

ช่วงเวลา 1 คาบสัญญาณนาฬิกาความถี่สูงเท่ากับ t และให้จำนวนพัลส์สัญญาณนาฬิกาความถี่สูงที่นับได้ในช่วง 1 คาบสัญญาณเอนโค้ดเดอร์เท่ากับ n_p

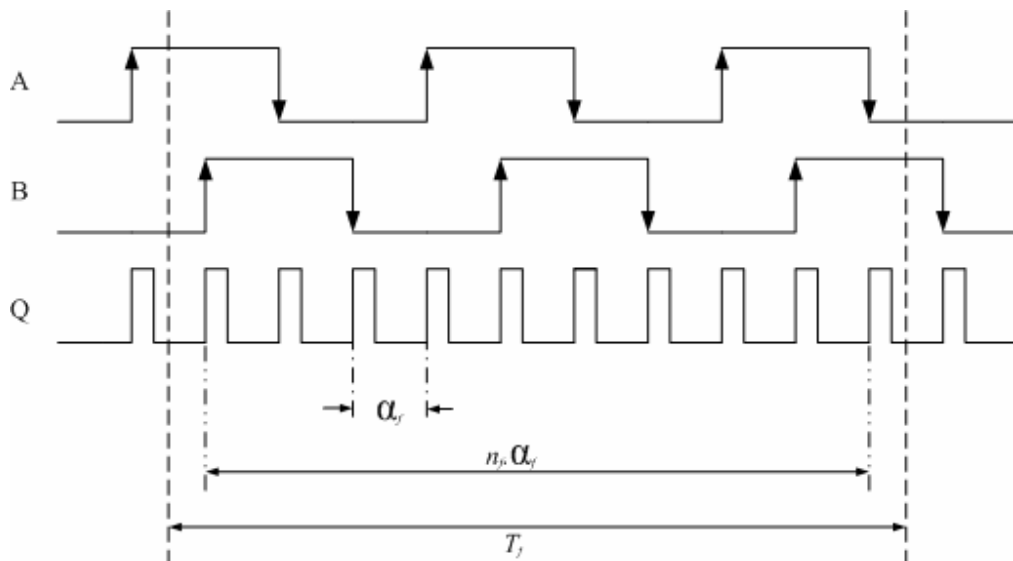
จะได้ว่า อัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ ω_p เท่ากับ

$$\omega_p = \frac{\alpha_p}{n_p \cdot t} \quad (2-1)$$

การวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ เหมาะสมกับการวัดเมื่อสัญญาณจากเอนโค้ดเดอร์มีความถี่ต่ำๆ [3]

2.4.2 วิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

การวัดความถี่สัญญาณเอนโค้ดเดอร์ เป็นการวัดจำนวนพัลส์สัญญาณที่เกิดจากเอนโค้ดเดอร์ในช่วงเวลาที่กำหนด ดังภาพประกอบ 2-8



ภาพประกอบ 2-8 การวัดความถี่สัญญาณเอนโค้ดเดอร์

การวัดอัตราเร็วด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ สามารถวัดได้โดยกำหนดช่วงเวลา (T_f) ที่ต้องการวัดสัญญาณเอนโค้ดเดอร์ แล้วนับจำนวนพัลส์สัญญาณเอนโค้ดเดอร์ที่เกิดขึ้นในช่วงเวลานั้น ดังภาพประกอบ 2-8 เป็นการวัดสัญญาณจากเอนโค้ดเดอร์แบบ 2 สัญญาณ จึงใช้การนับทุกช่วงขอบขาลงและขอบขาขึ้นของสัญญาณเอนโค้ดเดอร์ทั้ง 2 สัญญาณ ซึ่งจะได้ความละเอียดมากขึ้น

ความกว้างของสัญญาณ Q หรือสัญญาณที่สร้างจากสัญญาณเอนโค้ดเดอร์ทั้ง 2 สัญญาณกว้างเท่ากับ α_f ถ้าในช่วงเวลาที่นับ T_f สามารถนับจำนวนพัลส์สัญญาณ Q ได้ n_f ค่าจะได้อัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ ω_f เท่ากับ

$$\omega_f = \frac{n_f \cdot \alpha_f}{T_f} \quad (2-2)$$

การวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ เหมาะสมกับการวัดเมื่อสัญญาณจากเอนโค้ดเดอร์มีความถี่สูง [3]

2.5 การออกแบบวงจรด้วยเอพพีจีเอ

2.5.1 เอพพีจีเอ

การออกแบบวงจรดิจิทัลถูกพัฒนาอย่างต่อเนื่อง ในปี 1970 ได้มีการสร้างชิป (Chip) ที่มีวงจรลอจิกพื้นฐานจำนวนมากบรรจุอยู่ในและไม่มีฟังก์ชันการทำงานที่ตายตัว ซึ่งเป็นชิปดิจิทัลเอนกประสงค์ที่สามารถโปรแกรมให้เป็นฟังก์ชันการทำงานตามที่ต้องการได้ คือชิปพีแอลดี (Programmable Logic Device: PLD) ซึ่งเหมาะสมในการนำมาใช้ในการออกแบบวงจรรุ่นแบบ [4] [5]

เอพพีจีเอเป็นอุปกรณ์ประเภทพีแอลดี ที่สามารถโปรแกรมให้เป็นวงจรดิจิทัลตามที่ต้องการได้ เอพพีจีเอส่วนใหญ่จะมีความจุของเกตค่อนข้างสูงประมาณ 10,000 ถึง 6,000,000 เกต การโปรแกรมเอพพีจีเอให้เป็นวงจรตามที่ต้องการสามารถทำได้โดยการโหลดข้อมูลวงจร (Configuration Data) ลงไปที่เซลล์หน่วยความจำแบบแรมที่อยู่ภายในเอพพีจีเอ ดังนั้น เอพพีจีเอจะมีข้อเสียตรงที่ข้อมูลวงจรจะสูญหายหากมีการตัดไฟเลี้ยงออก จึงจำเป็นต้องมีหน่วยความจำภายนอกชิป เช่น Serial PROM เพื่อเก็บข้อมูลวงจรเพื่อทำการโปรแกรมลงเอพพีจีเอโดยอัตโนมัติ ทุกครั้งที่มีการจ่ายไฟเลี้ยง และเอพพีจีเอจะไม่มีข้อจำกัดในการโปรแกรมซ้ำหลายๆ ครั้ง

แม้ว่าไมโครคอนโทรลเลอร์จะเป็นที่นิยมอย่างมากในการนำมาใช้ในงานควบคุม ไม่ว่าจะเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 หรือตระกูล PIC ไมโครคอนโทรลเลอร์เป็นชิปที่ถูกออกแบบที่มีสถาปัตยกรรมและชุดคำสั่งไว้แล้วผู้ออกแบบเพียงนำชุดคำสั่งต่างๆ มาโปรแกรมให้ไมโครคอนโทรลเลอร์ทำงานตามโปรแกรมที่วางไว้ ซึ่งจะเห็นได้ว่าไมโครคอนโทรลเลอร์มีความยืดหยุ่นในการใช้งานค่อนข้างสูง แต่สำหรับงานบางประเภทที่ต้องการอัตราเร็วในการทำงานสูง ไมโครคอนโทรลเลอร์จะไม่สามารถตอบสนองในส่วนนี้ได้ เช่นวงจรเข้าและถอดรหัส วงจรของอุปกรณ์ทางการสื่อสารโทรคมนาคม หรือวงจรในเครื่องมือวัดบางประเภท เป็นต้น ทำให้ต้องอาศัยวงจรฮาร์ดแวร์มาทำงานแทน ซึ่งวงจรฮาร์ดแวร์นั้นสามารถออกแบบได้โดยใช้เอพพีจีเอ

ข้อดีของการเลือกใช้เอ็พพีจีเอ

- 1) สามารถออกแบบวงจรดิจิทัลได้โดยมีจำนวนอุปกรณ์และขนาดของแผงวงจรลดลงอย่างมาก เนื่องจากวงจรเดิมหลายๆ ส่วนได้ถูกออกแบบรวมไว้ในเอ็พพีจีเอแล้ว
- 2) วงจรที่ออกแบบสามารถทำงานที่ความถี่สูงกว่าไมโครคอนโทรลเลอร์
- 3) สามารถทำการแก้ไขวงจรได้ง่าย เพราะเป็นการโปรแกรมโดยใช้ซอฟต์แวร์และสามารถโปรแกรมซ้ำได้หลายๆ ครั้ง
- 4) ทำให้สามารถลดระยะเวลาและค่าใช้จ่ายในการออกแบบและพัฒนาผลิตภัณฑ์ ในการพัฒนาด้วยภาษา VHDL จะมีความยืดหยุ่นมาก เนื่องจากภาษา VHDL เป็นมาตรฐานไม่ขึ้นอยู่กับชนิดหรือว่ายี่ห้อของเอ็พพีจีเอที่นำมาใช้งาน

ข้อเสียของการเลือกใช้เอ็พพีจีเอ

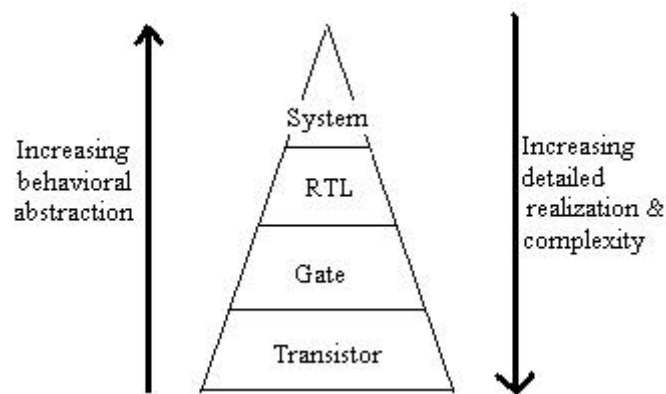
- 1) จำนวนทรัพยากรของเอ็พพีจีเอแต่ละตัวมีจำนวนจำกัด ตามแต่ละรุ่นของเอ็พพีจีเอ ทำให้การออกแบบวงจรต้องระมัดระวังไม่ให้วงจรใช้จำนวนทรัพยากรของเอ็พพีจีเอเกินจำนวนที่เอ็พพีจีเอรุ่นที่กำลังใช้งานมีอยู่
- 2) บางครั้งวงจรที่ออกแบบใช้งานทรัพยากรของเอ็พพีจีเอไม่หมด ทำให้ทรัพยากรส่วนที่เหลือต้องถูกปล่อยทิ้งไว้ โดยไม่ได้ใช้ประโยชน์
- 3) เอ็พพีจีเอมีราคาค่อนข้างแพง ไม่เหมาะสมที่ใช้ในการออกแบบเพื่อการตลาด แต่เหมาะสมในการใช้เพื่อสร้างวงจรต้นแบบมากกว่า

2.5.2 ภาษาบรรยายวงจรฮาร์ดแวร์

ภาษาที่สามารถนำมาใช้ในการออกแบบวงจรดิจิทัลได้นั้นมีหลายภาษา เช่น ภาษา VHDL, Verilog หรือ Handel-C ซึ่งแต่ละภาษาก็มีข้อดีและข้อเสียแตกต่างกันไป ภาษา VHDL เป็นภาษาที่ได้รับความนิยมมานาน ถูกคิดค้นในปี ค.ศ. 1980 โดยกระทรวงกลาโหมของสหรัฐอเมริกา ต่อมาในปี ค.ศ. 1987 ถูกพัฒนาจนสามารถประกาศเป็นมาตรฐานของสถาบัน IEEE (Institute of Electrical and Electronic Engineers) [6] [7] [8] ในหมวด IEEE 1076-1987 และได้มีการปรับปรุงอีกครั้งในปี ค.ศ. 1993 เรียกว่า IEEE 1076-1993 ภาษา VHDL มีความสามารถในการออกแบบวงจรที่ซับซ้อนได้ดี ส่วนภาษา Verilog เป็นภาษาที่ใช้ในการออกแบบวงจรดิจิทัลซึ่งอยู่ในกลุ่มภาษาบรรยายวงจรฮาร์ดแวร์ (Hardware Description Language: HDL) เช่นเดียวกับภาษา VHDL ซึ่งภาษา Verilog เป็นภาษาที่ถูกพัฒนาขึ้นในช่วงปี ค.ศ. 1984-1985 และถูกประกาศเป็นมาตรฐานของ IEEE ในปี ค.ศ. 1995 ในหมวด IEEE Standard 1365 เป็นภาษาที่ออกแบบให้มีการเขียนแล้วทำความเข้าใจที่ง่ายกว่าภาษา VHDL เนื่องจากโครงสร้างไวยากรณ์เหมือนภาษาซี แต่ความสามารถในการ

ออกแบบวงจรที่มีความซับซ้อนสูงจะดีกว่าการออกแบบด้วยภาษา VHDL [9] ภาษา Handel-C ก็เป็นภาษาที่ใช้ในการออกแบบวงจรเช่นเดียวกับภาษา VHDL และภาษา Verilog แต่ไม่ได้จัดอยู่ในกลุ่มภาษาบรรยายวงจรฮาร์ดแวร์ [10] มีการใช้ไวยากรณ์เหมือนภาษาซี ภาษา Handel-C มีข้อเสียคือรองรับสัญญาณนาฬิกาจากภายนอกเพียงสัญญาณเดียวขณะที่ภาษา VHDL สามารถรับได้หลายสัญญาณ อีกทั้งยังใช้งานเฉพาะที่ขอบสัญญาณนาฬิกาที่ขอบขาขึ้นได้อย่างเดียวขณะที่ภาษา VHDL สามารถใช้ได้ทั้งขอบขาขึ้นและขอบขาลง และภาษา Handel-C ยังมีค่าสัญญาณให้ใช้แค่ลอจิกสูง “1” และลอจิกต่ำ “0” เท่านั้น ขณะที่ภาษา VHDL มีสัญญาณให้เลือกใช้ได้หลายประเภท เช่น สัญญาณไม่ทราบค่า “X”, สัญญาณความต้านทานสูง (High Impedance) เป็นต้น [10]

ภาษา VHDL เป็นภาษาที่สามารถใช้ได้หลายระบบ การออกแบบวงจรโดยใช้ภาษา VHDL สามารถนำไปจำลองการทำงานหรือสังเคราะห์ด้วยซอฟต์แวร์ตัวใดก็ได้ที่รองรับภาษา VHDL จึงทำให้การออกแบบด้วยภาษา VHDL เป็นการออกแบบที่ไม่ยึดติดอยู่ซอฟต์แวร์ที่ใช้ในการออกแบบ เป็นภาษาในรูปแบบที่บรรยายพฤติกรรมทำให้สามารถอธิบายการทำงานของวงจรภายในได้ทันที ภาษา VHDL สามารถบรรยายวงจรดิจิทัลได้หลายระดับตั้งแต่ระดับพฤติกรรมการทำงานของวงจร (Behavioral Level) จนถึงระดับลอจิกเกต (Logic Gate Level) ระดับการออกแบบวงจรดิจิทัลในภาษา VHDL เป็นดังภาพประกอบ 2-9 [5]



ภาพประกอบ 2-9 ระดับของการออกแบบวงจรดิจิทัลในภาษา VHDL [5]

- ระดับระบบ (System Level) เป็นระดับบนสุดของการเขียนภาษา VHDL โดยจะบรรยายพฤติกรรมการทำงานของระบบ หรือแบบอัลกอริทึมของระบบ โดยคำนึงถึงเรื่องข้อมูลของเวลา
- ระดับการส่งข้อมูลของรีจิสเตอร์ เป็นการบรรยายลงลึกในรายละเอียดของวงจรระดับโครงสร้าง (Architecture) เช่นเป็นวงจรแบบอะซิงโครนัส (Asynchronous), แบบ

ซิงโครนัส (Synchronous), สถานะของเครื่อง (State machines), วิธีของข้อมูล (Data path), รีจิสเตอร์ ต่างๆ ตลอดจนการไหลเข้าออกของข้อมูลในแต่ละบล็อกภายในวงจร

- ระดับลอจิกเกต (Gate Level) เป็นการบรรยายระดับต่ำกว่าระดับการส่งข้อมูลของรีจิสเตอร์ โดยจะมีรายละเอียดวงจรระดับลอจิกเกต ฟลิปฟล็อป ต่างๆ ที่เชื่อมต่อกันเป็นวงจร
- ระดับทรานซิสเตอร์ (Transistor Level) เป็นระดับต่ำสุดของการออกแบบวงจร ในระดับนี้ภาษา VHDL ไม่สามารถใช้เขียนบรรยายวงจรได้

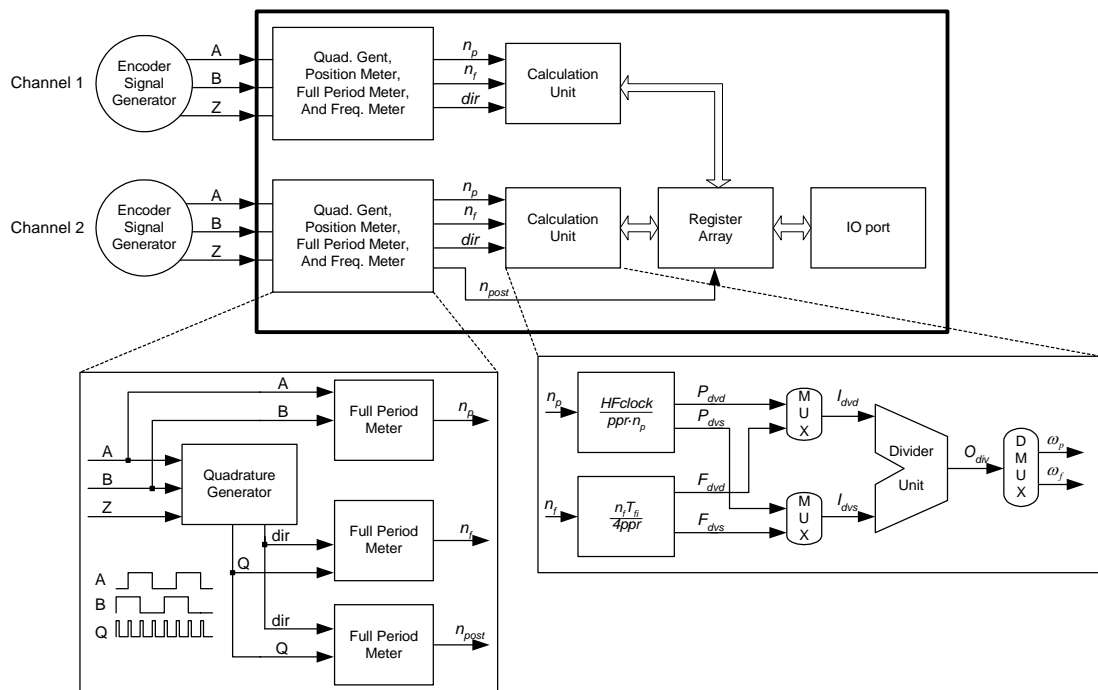
ภาษา VHDL มีคุณสมบัติการทำงานแบบขนาน (Concurrent) หมายถึงในการเขียนโค้ดในแต่ละบรรทัดจะไม่ลำดับความสำคัญของตำแหน่งแต่ละบรรทัด (Order independent) ชุดคำสั่งสามารถทำงานได้ในเวลาเดียวกันและเป็นอิสระ แต่ภาษา VHDL ก็ยังสามารถเขียนรูปแบบคำสั่งแบบตามลำดับ (Sequential) มีลำดับความสำคัญก่อนและหลังของการเขียนคำสั่งแต่จะต้องเขียนรูปแบบคำสั่งไว้ในคำสั่งที่เป็นแบบตามลำดับ

บทที่ 3

โครงสร้างของวงจร

ในบทนี้อธิบายถึงโครงสร้างโดยรวมของวงจรเชื่อมต่อเอนโคเดอร์ที่สามารถคำนวณค่าอัตราเร็วการหมุนและค่าตำแหน่งการเคลื่อนที่แบบหลายช่องสัญญาณ ซึ่งประกอบด้วยวงจรรย่อยต่างๆ ที่สัมพันธ์กันและแบบวิธีต่างๆ ที่ถูกนำมาใช้ในการออกแบบวงจร

3.1 โครงสร้างหลักของวงจร



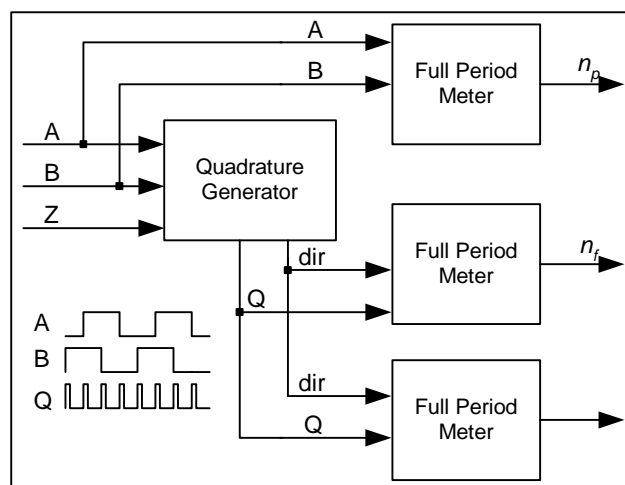
ภาพประกอบ 3-1 โครงสร้างหลักของวงจร

โครงสร้างหลักของวงจรถูกออกแบบตามที่แสดงในภาพประกอบ 3-1 ประกอบด้วย 4 ส่วนหลักดังต่อไปนี้

1. ส่วนปรับแต่งสัญญาณนำเข้าและวัดค่าประกอบด้วย 4 วงจรย่อย คือ วงจรย่อย Quadrature Generator, วงจรย่อย Position Meter, วงจรย่อย Full period Meter และวงจรย่อย Frequency Meter ซึ่งจะกล่าวอย่างละเอียดในหัวข้อ 3.2
2. ส่วนคำนวณ ประกอบด้วย 2 วงจรย่อย คือ วงจรส่วนเตรียมค่าสำหรับคำนวณ และ วงจรส่วนคำนวณตามสมการหาค่าแต่ละวิธี ซึ่งจะกล่าวอย่างละเอียดในหัวข้อ 3.3
3. ส่วนเก็บข้อมูล ซึ่งจะเก็บข้อมูลค่าเริ่มต้นต่างๆ ของระบบที่จำเป็นในการทำงานและการคำนวณ และผลลัพธ์จากการคำนวณโดยข้อมูลของทั้ง 2 ช่องสัญญาณจะเก็บในส่วนเก็บข้อมูลเดียวกัน ซึ่งจะกล่าวอย่างละเอียดในหัวข้อ 3.4
4. ส่วนติดต่อกับอุปกรณ์ภายนอก จะเป็นส่วนติดต่อนำเข้าและนำออกทั้งกับเอนโค้ดเดอร์ 2 ช่องสัญญาณ และส่วนติดต่อกับไมโครคอนโทรลเลอร์ได้สูงถึงขนาด 32 บิต ซึ่งจะกล่าวอย่างละเอียดในหัวข้อ 3.5

3.2 ส่วนปรับแต่งสัญญาณนำเข้าและวัดค่า

ส่วนนี้เป็นส่วนที่รับสัญญาณนำเข้าจากภายนอกเอฟพีจีเอ คือ สัญญาณจากเอนโค้ดเดอร์ ซึ่งประกอบด้วย 3 สัญญาณ คือ สัญญาณ A, สัญญาณ B และสัญญาณ Z ตามภาพประกอบ 3-2



ภาพประกอบ 3-2 ส่วนปรับแต่งสัญญาณนำเข้าและวัดค่า

ส่วนนี้ประกอบด้วย 4 วงจรย่อยคือ วงจรย่อย Quadrature Generator วงจรย่อย Full Period Meter วงจรย่อย Frequency Meter และวงจรย่อย Position Meter ซึ่งแต่ละวงจรย่อยมีรายละเอียดดังต่อไปนี้

3.2.1 วงจรย่อย Quadrature Generator

ส่วนนี้ทำหน้าที่สร้างสัญญาณ Quadrature ตามภาพประกอบ 3-2 คือสัญญาณ Q ซึ่งถูกสร้างโดยใช้สัญญาณ A และ B ซึ่งเป็นอินพุตที่วงจรรับเข้ามา สัญญาณนำเข้าและนำออกของวงจรนี้แสดงให้เห็นในภาพประกอบ 3-3 และภาพประกอบ 3-4

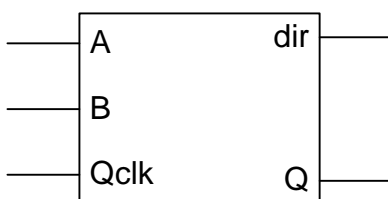
วิธีการสร้างสัญญาณ Q เป็นการสร้างพัลส์สัญญาณใหม่ โดยใช้สัญญาณ A และ สัญญาณ B โดยสร้างพัลส์ใหม่ทุกๆ ครั้งที่มีการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B

```

Entity Quadrature is
Port (   Qclk       : in std_logic;
        A          : in std_logic;
        B          : in std_logic;
        dir        : out std_logic;
        Q          : out std_logic);
End Quadrature;

```

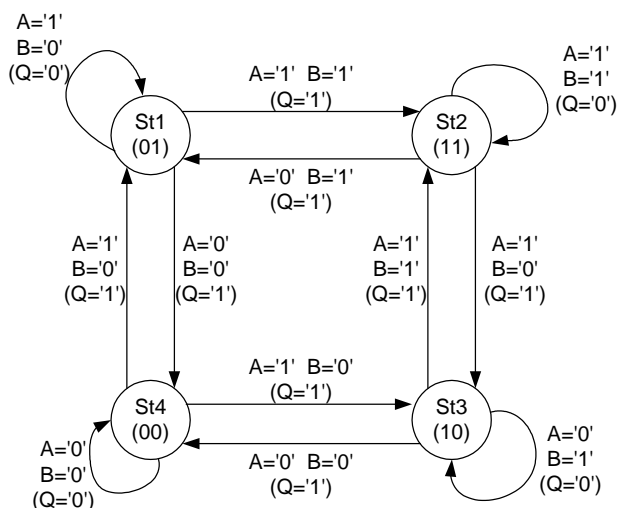
ภาพประกอบ 3-3 การประกาศสร้างสัญญาณนำเข้าและนำออก
ของวงจรย่อย Quadrature Generator



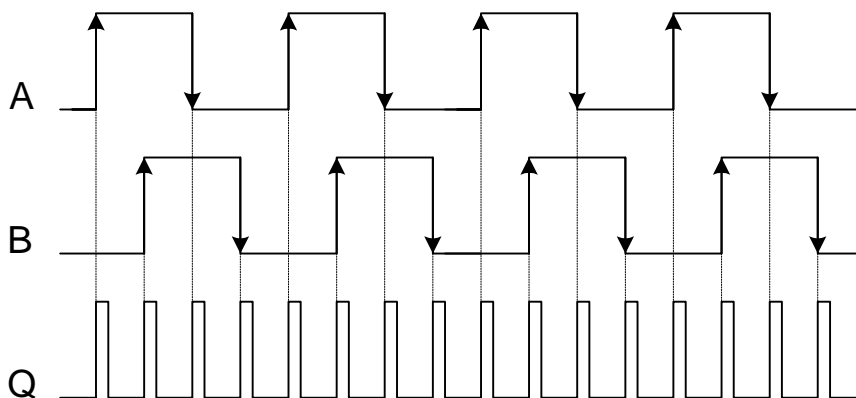
ภาพประกอบ 3-4 วงจรระดับบนจากการสังเคราะห์วงจรย่อย Quadrature Generator

สัญญาณนำเข้าและนำออกของวงจรย่อย Quadrature Generator มีดังนี้

- **Qclk** เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งสัญญาณ Qclk เป็นสัญญาณนาฬิกาที่มีความถี่สูงกว่าความถี่สัญญาณ A และสัญญาณ B มากๆ ซึ่งในที่นี้จะใช้สัญญาณนาฬิกาหลักของระบบ เพื่อให้สามารถตรวจจับการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B ได้
- **A** เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณจากเอนโค้ดเดอร์เฟส A
- **B** เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณจากเอนโค้ดเดอร์เฟส B
- **Dir** เป็นสัญญาณนำออกจากย่อยขนาด 1 บิต ซึ่งคือสัญญาณที่บอกถึงทิศทางการหมุนของเอนโค้ดเดอร์ว่าหมุนทวนเข็มนาฬิกาหรือตามเข็มนาฬิกา
- **Q** เป็นสัญญาณนำออกจากวงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณที่มีการเปลี่ยนแปลงสูงกว่าสัญญาณเอนโค้ดเดอร์ทั้งสัญญาณ A และสัญญาณ B ถึง 4 เท่า



ภาพประกอบ 3-5 การตรวจสอบสัญญาณ A และ B เพื่อสร้างสัญญาณ Q



ภาพประกอบ 3-6 การสร้างสัญญาณ Q

ภาพประกอบ 3-5 และภาพประกอบ 3-6 เป็นการสร้างสัญญาณ Q โดยการตรวจสอบการเปลี่ยนแปลงของสัญญาณจากเอนโค้ดเดอร์ทั้งสัญญาณ A และสัญญาณ B ซึ่งมีหลักการทำงานดังนี้

การสร้างสัญญาณ Q ซึ่งเป็นสัญญาณที่ให้ความละเอียดถึง 4 เท่าของสัญญาณเอนโค้ดเดอร์ โดยใช้สัญญาณนำเข้าคือสัญญาณ A และสัญญาณ B ทำให้กำหนดสถานะของสัญญาณ Q เป็น 4 สถานะตามภาพประกอบ 3-5 โดยลักษณะการเปลี่ยนแปลงสถานะของสัญญาณ A และสัญญาณ B ทำให้สามารถทราบทิศทางการหมุนของเอนโค้ดเดอร์และทราบว่าสถานะของสัญญาณ A และสัญญาณ B แต่ละจังหวะการหมุนว่าจะเป็นอย่างไร

การทำงานของวงจรย่อยนี้ จะตรวจสอบการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B ตามภาพประกอบ 2-3 เพื่อตรวจสอบการหมุนของเอนโค้ดเดอร์ว่าหมุนตามหรือทวนเข็มนาฬิกา จากรูปแบบของสัญญาณ A และสัญญาณ B เมื่อเอนโค้ดเดอร์หมุนตามหรือทวนเข็มนาฬิกา สามารถนำมาออกแบบรูปแบบการสร้างสัญญาณ Q ได้ตามภาพประกอบ 3-5 คือถ้าสัญญาณ A

และสัญญาณ B มีค่าลอจิกเป็นสูง “1” และต่ำ “0” ตามลำดับที่กำหนดให้อยู่ในสถานะ St1 เมื่อตรวจสอบได้สถานะ st1 แล้วก็ให้สัญญาณ Q เท่ากับลอจิกสูง “1” ช่วงหนึ่งตามที่ออกแบบไว้ นั้นให้มีค่าลอจิกสูง “1” ค้างไว้ 1 ช่วงคาบสัญญาณนาฬิกาที่ใช้ในการตรวจสอบการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B แล้วถ้าสัญญาณ A และสัญญาณ B ยังมีค่าลอจิกเหมือนเดิมก็จะยังให้อยู่ในสถานะ st1 เหมือนเดิมและให้สัญญาณ Q มีค่าเป็นลอจิกต่ำ “0” ตามรูปแบบการหมุนของเอนโค้ดเดอร์ตามภาพประกอบ 2-3 ถ้าเอนโค้ดเดอร์ หมุนตามเข็มนาฬิกาสัญญาณ A จะมีค่าลอจิกต่อไปเป็นลอจิกสูง “1” และสัญญาณ B จะมีค่าลอจิกต่อไปเป็นลอจิกสูง “1” หรือเอนโค้ดเดอร์หมุนเวียนทวนเข็มนาฬิกาสัญญาณ A จะมีค่าเป็นลอจิกต่ำ “0” และสัญญาณ B จะมีค่าเป็นลอจิกต่ำ “0” เมื่อวงจรย่อยนี้ตรวจสอบการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B ได้ก็จะเปลี่ยนสถานะไปยัง st2 ถ้าเป็นการหมุนตามเข็มนาฬิกา หรือสถานะ st4 ถ้าเป็นการหมุนทวนเข็มนาฬิกาแล้วให้ค่าสัญญาณ Q เป็นลอจิกสูง “1” ทันที และเมื่อสัญญาณนาฬิกาที่ใช้ในการตรวจสอบครั้งต่อไปพบว่าสัญญาณ A และสัญญาณ B มีค่าลอจิกเหมือนเดิมก็จะให้สัญญาณ Q มีค่าเป็นลอจิกต่ำ “0”

โดยสรุปแล้ว วงจรย่อยนี้จะมีการเปลี่ยนสถานะและให้สัญญาณ Q มีค่าเป็นลอจิกสูง “1” ทุก ๆ ครั้งที่ตรวจพบการเปลี่ยนแปลงของสัญญาณ A และสัญญาณ B แต่ถ้าตรวจสอบพบว่าสัญญาณ A และสัญญาณ A และสัญญาณ B ไม่มีการเปลี่ยนแปลงก็ให้อยู่ในสถานะเดิมและให้สัญญาณ Q มีค่าเป็นลอจิกต่ำ “0” ซึ่งจะได้สัญญาณ Q ตามภาพประกอบ 3-6

วงจรย่อย Quadrature Generator ที่กล่าวข้างต้น เมื่อออกแบบด้วยภาษา VHDL และทำการสังเคราะห์วงจรด้วยโปรแกรม Xilinx 7.1i แล้วได้ผลด้าน การใช้ทรัพยากรและเวลาดังภาพประกอบ 3-7

Device utilization summary:	

Selected Device : 3s200tq144-5	
Number of Slices:	3 out of 1920 0%
Number of Slices Flip Flops :	4 out of 3840 0%
Number of 4 input LUTs:	5 out of 3840 0%
Number of bonded IOBs:	5 out of 97 0%
Number of GCLKs:	1 out of 8 0%
Timing Summary	

Speed Grade: -5	
Minimum period : 2.824 ns (Maximum Frequency : 354.114MHz)	
Minimum input arrival time before clock : 3.046ns	
Maximum output required time after clock : 6.280 ns	
Maximum combinational path delay : No path found	

ภาพประกอบ 3-7 จำนวนทรัพยากรของเอฟพีจีเอที่ถูกใช้ในวงจรร้อย Quadrature Generator

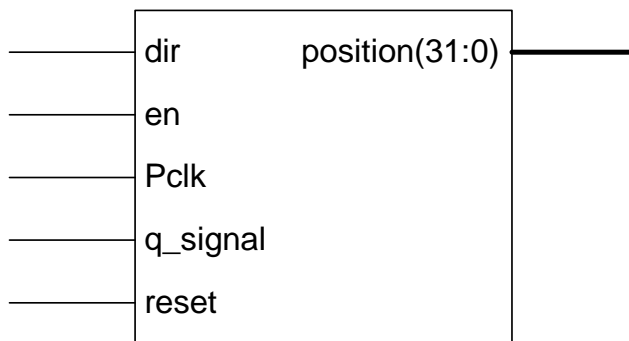
สัญญาณ Q จะถูกนำไปใช้ในอีก 2 วงจรร้อยคือ วงจรร้อย Frequency Meter เพื่อเพิ่มความถี่สัญญาณของเอนโค้ดเดอร์อีก 4 เท่า และวงจรร้อย Position Meter เพื่อเพิ่มความละเอียดของการเปลี่ยนตำแหน่งอีก 4 เท่าเช่นกัน

3.2.2 วงจรร้อย Position Meter

วงจรร้อย Position Meter เป็นวงจรร้อยที่ทำการวัดตำแหน่งการเคลื่อนที่ของเอนโค้ดเดอร์ โดยจะเป็นวงจรร้อย ซึ่งการนับจังหวะการหมุนของเอนโค้ดเดอร์ที่ให้สัญญาณนำออก 2 เฟส คือ สัญญาณ A และ สัญญาณ B ที่ต่างกัน 90 องศา นั้น สามารถแบ่งย่อยให้ละเอียดได้ถึง 4 เท่าของจำนวนพัลส์ของเอนโค้ดเดอร์ต่อรอบการหมุน ดังนั้น ในการออกแบบวงจรร้อยวัดตำแหน่งจึงเป็นการนับจังหวะการเปลี่ยนแปลงของสัญญาณ Q ซึ่งวงจรร้อยส่วนนี้มีสัญญาณนำเข้าและนำออก ดังภาพประกอบ 3-8 และภาพประกอบ 3-9

Entity Position_Est is	
Port (Pclk	: in std_logic;
reset	: in std_logic;
en	: in std_logic;
q_signal	: in std_logic;
dir	: in std_logic;
Position	: out std_logic_vector (31 downto 0));
End Position_Est;	

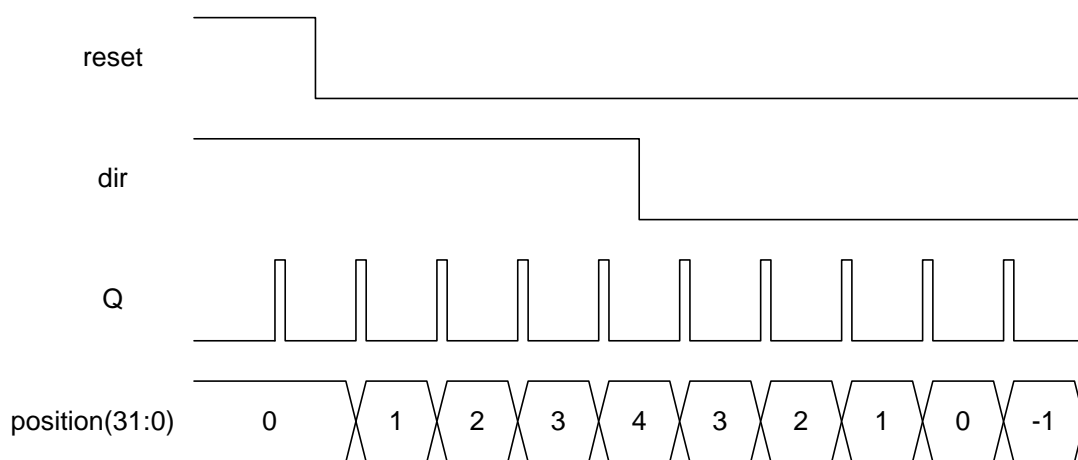
ภาพประกอบ 3-8 การประกาศสร้างสัญญาณนำเข้าและนำออกของวงจรร้อย Position Meter



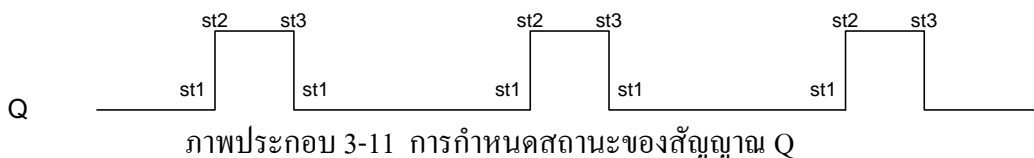
ภาพประกอบ 3-9 วงจรระดับบนจากการสังเคราะห์วงจรย่อย Position Meter

สัญญาณนำเข้าและนำออกของวงจรวัดตำแหน่งมีดังนี้

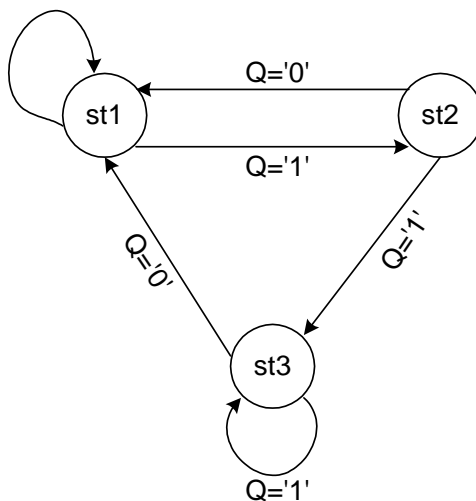
- Pclk เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณนาฬิกาที่มีความถี่สูงกว่าความถี่สัญญาณ Q เพื่อให้สามารถตรวจจับการเปลี่ยนแปลงของสัญญาณ Q
- reset เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณเพื่อทำการเคลียร์ค่าตำแหน่งเพื่อเริ่มทำการนับใหม่
- En เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณเพื่ออนุญาตวงจรวัดตำแหน่งให้หยุดหรือทำงาน
- q_signal เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณ Q ตามภาพประกอบ 3-6 ซึ่งเป็นสัญญาณ Q ที่รับมาจากวงจรย่อย Quadrature Generator
- dir เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 2 บิต ซึ่งคือสัญญาณบอกทิศทางการหมุนของเอนโค้ดเดอร์
- position เป็นสัญญาณนำออกจากวงจรย่อยขนาด 32 บิต ซึ่งคือจะเห็นค่าการหมุนของเอนโค้ดเดอร์ว่าหมุนไปได้กี่จังหวะ



ภาพประกอบ 3-10 การนับสัญญาณ Q ในวงจรวัดตำแหน่ง



ภาพประกอบ 3-11 การกำหนดสถานะของสัญญาณ Q



ภาพประกอบ 3-12 การตรวจสอบสถานะสัญญาณ Q

ภาพประกอบ 3-10 เป็นวิธีการนับสัญญาณ Q ในวงจรวัดตำแหน่ง โดยมีการกำหนดสถานะของสัญญาณ Q ตามภาพประกอบ 3-11 ซึ่งมีหลักการทำงานดังนี้

การทำงานของวงจรวัดตำแหน่งที่ออกแบบนี้ จะใช้สัญญาณนาฬิกา Pclk ซึ่งมีความถี่สูงกว่าความถี่ของสัญญาณ Q เพื่อตรวจสอบการเปลี่ยนแปลงของสัญญาณ Q ได้ โดยวงจรจะทำงานได้ก็ต่อเมื่อสัญญาณ en มีค่าลอจิกสูง “1” เมื่อวงจรถูกอนุญาตให้ทำงานได้แล้ว ถ้าสัญญาณ reset มีค่าเป็นลอจิก “1” จะเป็นการเคลียร์ค่าตำแหน่ง (position) ให้เป็นค่าเริ่มต้นและจะเริ่มนับได้เมื่อ reset มีค่าเป็นลอจิก “0”

ถ้าสัญญาณ en มีค่าเป็นลอจิกสูง “1” และ reset มีค่าเป็นลอจิกต่ำ “0” วงจรก็จะเริ่มทำการนับ โดยจะนับสัญญาณ Q ทุกครั้งที่มีการเปลี่ยนแปลงจากค่าเป็นลอจิกต่ำ “0” เป็นค่าลอจิกสูง “1” ซึ่งการตรวจสอบการเปลี่ยนแปลงของสัญญาณ Q สำหรับการนับสัญญาณ Q จะนับขึ้น (เพิ่มค่า Position) เมื่อ dir มีค่าเป็นลอจิกสูง “1” และจะนับลง (ลดค่า position) เมื่อ dir มีค่าเป็นลอจิกต่ำ “0” ดังนั้นจะสามารถนับจังหวะการหมุนได้ $\pm 2,147,483,647$ จังหวะ

สำหรับการนับสัญญาณ Q นั้นจะทำการกำหนดสถานะของสัญญาณ Q ดังภาพประกอบ 3-12 ถ้าสัญญาณ Q ค่าเป็นลอจิกต่ำ “0” ก็กำหนดให้อยู่ในสถานะที่ 1 (st1) เมื่อตรวจสอบพบว่า Q มีค่าเป็นลอจิกสูง “1” ก็จะเปลี่ยนเป็นสถานะที่ 2 (st2) ก็จะทำการเพิ่มค่า position ขึ้นหนึ่งค่า ถ้าสัญญาณ dir มีค่าเป็นลอจิกสูง “1” หรือลดค่า position ลงหนึ่งค่า ถ้าสัญญาณ dir มีค่าเป็น

ลอจิกค่า “0” แต่ถ้ายังตรวจสอบได้ว่า Q ยังมีค่าเป็นลอจิกสูง “1” ก็จะเป็นสถานะที่ 3 (st3) และจะเปลี่ยนแปลงเป็นสถานะที่ 1 เมื่อ Q มีค่าเป็นลอจิกต่ำ “0”

วงจรร้อย Position Meter ดังกล่าวข้างต้น เมื่อกแบบด้วยภาษา VHDL และทำการสังเคราะห์วงจรด้วยโปรแกรม Xilinx 7.1i แล้วได้ผลด้านการใช้ทรัพยากรและเวลา ดังภาพประกอบ 3-13

Device utilization summary:	

Selected Device : 3s200tq144-5	
Number of Slices:	19 out of 1920 0%
Number of Slices Flip Flops :	34 out of 3840 0%
Number of 4 input LUTs:	37 out of 3840 0%
Number of bonded IOBs:	37 out of 97 38%
Number of GCLKs:	1 out of 8 12%
Timing Summary	

Speed Grade: -5	
Minimum period : 5.082 ns (Maximum Frequency : 196.755MHz)	
Minimum input arrival time before clock : 6.137ns	
Maximum output required time after clock : 6.280 ns	
Maximum combinational path delay : No path found	

ภาพประกอบ 3-13 จำนวนทรัพยากรของเอฟพีไอที่ถูกใช้ในวงจรร้อย Position Meter

3.2.3 วงจรร้อย Full Period Meter

วงจรร้อย Full Period Meter หรือวงจรวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ ส่วนนี้เป็นส่วนหนึ่งของการวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์โดยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ ซึ่งการวัดที่เลือกใช้เป็นหัวข้อ 2.4.1 แต่เนื่องจากใน [1] กล่าวว่าสัญญาณเอนโค้ดเดอร์มีค่ารอบการทำงาน (Duty Cycle) ไม่เท่ากับ 50% จึงเลือกใช้การวัดความกว้างแบบเต็มคาบแทนการวัดความกว้างแบบครึ่งคาบเหมือนกับ [1] [2] [3]

โดยในวงจรร้อยนี้ จะเป็นวงจรรนับพัลส์สัญญาณนาฬิกา โดยใช้สัญญาณนาฬิกาความถี่สูงตรวจสอบหาขอบขาขึ้นของสัญญาณ A หรือสัญญาณ B ขึ้นอยู่กับว่าจะเลือกใช้สัญญาณใดในการวัด ซึ่งช่วงห่างระหว่างขอบขาขึ้นทั้งสองคือหนึ่งช่วงสำหรับวงจรรนับ โดยค่าที่นับ n_p จะมีการเปลี่ยนแปลงทุก ๆ ครั้งเมื่อครบ 1 คาบสัญญาณเอนโค้ดเดอร์ ค่าที่ได้จากการนับ n_p เป็นสัญญาณขนาด 32 บิต ซึ่งจะนับได้สูงถึง 2^{32} ค่า ซึ่งวงจรร้อย Full Period Meter มีสัญญาณนำเข้าและนำออก ดังภาพประกอบ 3-14 และภาพประกอบ 3-15

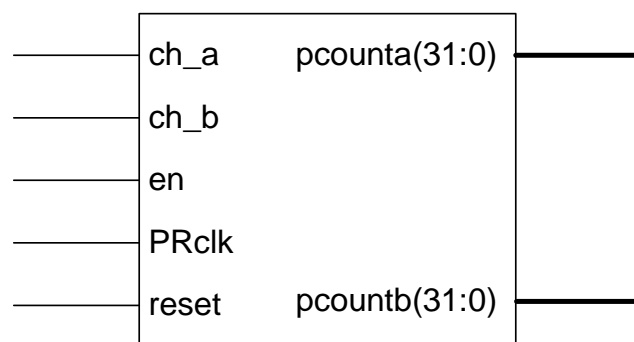
```

Entity Period_Est is
Port (  PRclk      : in std_logic;
        reset      : in std_logic;
        en         : in std_logic;
        ch_a       : in std_logic;
        ch_b       : in std_logic;
        pcounta    : out std_logic_vector (31 downto 0) ;
        pcountb    : out std_logic_vector (31 downto 0) );

End Period_Est;

```

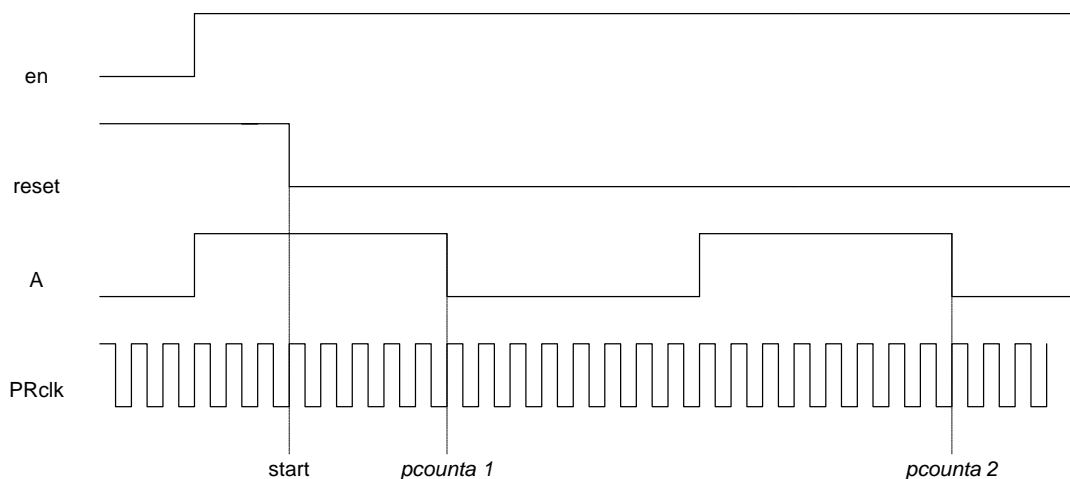
ภาพประกอบ 3-14 การประกาศสร้างสัญญาณนำเข้าและนำออกของวงจรร้อย Full Period Meter



ภาพประกอบ 3-15 วงจรระดับบนจากการสังเคราะห์วงจรร้อย Full Period Meter

สัญญาณนำเข้าและนำออกของวงจรวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์มีดังนี้

- PRclk เป็นสัญญาณนำเข้าสู่วงจรร้อยขนาด 1 บิต ซึ่งคือสัญญาณนาฬิกาที่มีความถี่สูงที่ทราบค่าคาบเวลาแน่นอน ซึ่งจะใช้ในการตรวจสอบการเปลี่ยนแปลงสัญญาณจากเอนโค้ดเดอร์ และใช้ในวงจรรันเพื่อวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์
- reset เป็นสัญญาณเข้าสู่วงจรร้อยขนาด 1 บิต ซึ่งสัญญาณเพื่อทำการเคลียร์ค่าสัญญาณภายในวงจรรันเพื่อเริ่มการทำงานใหม่
- en เป็นสัญญาณนำเข้าสู่วงจรร้อยขนาด 1 บิต ซึ่งคือสัญญาณเพื่ออนุญาตวงจรรันให้หยุดหรือทำงาน
- ch_a และ ch_b เป็นสัญญาณนำเข้าสู่วงจรร้อยขนาด 1 บิต ซึ่งคือสัญญาณจากเอนโค้ดเดอร์ทั้งสัญญาณ A และสัญญาณ B ซึ่งเป็นสัญญาณที่ต้องการวัดความกว้างคาบ
- pcounta และ pcountb เป็นสัญญาณนำเข้าสู่วงจรร้อยขนาด 32 บิต ซึ่งเป็นค่าการนับสัญญาณ PRclk ซึ่งคือค่า n_p ในสมการ 2-1



ภาพประกอบ 3-16 การวัดความกว้างสัญญาณเอนโค้ดเดอร์

การทำงานของวงจรวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ ตามภาพประกอบ 3-16 จะทำงานเมื่อสัญญาณ *en* ซึ่งเป็นสัญญาณเพื่ออนุญาตให้วงจรส่วนนี้ทำงานซึ่งวงจรส่วนนี้จะทำงานเมื่อ *en* มีค่าเป็นลอจิกสูง “1” เท่านั้น ส่วนสัญญาณ *reset* จะเป็นการรีเซ็ตค่า *pcounta* และ *pcountb* ให้เท่ากับศูนย์ เพื่อเริ่มต้นการวัดเมื่อสัญญาณ *reset* มีค่าเป็นลอจิกสูง ๆ “1” และ *reset* มีค่าเป็นลอจิกต่ำ “0” และจะได้ค่า *pcount* ทุก ๆ คาบ ซึ่งก็คือค่า n_p ในสมการ 2-1 ซึ่งเป็นตัวแปรสำคัญในการนำไปคำนวณหาค่าอัตราเร็วการหมุน

วงจรวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์นี้ เมื่อออกแบบด้วยภาษา VHDL และทำการสังเคราะห์วงจรด้วยโปรแกรม Xilinx 7.1i แล้ว ได้ผลด้านการใช้ทรัพยากรและเวลาดังภาพประกอบ 3-17

Device utilization summary:	

Selected Device : 3s200tq144-5	
Number of Slices:	147 out of 1920 7%
Number of Slices Flip Flops :	198 out of 3840 5%
Number of 4 input LUTs:	131 out of 3840 3%
Number of bonded IOBs:	69 out of 97 71%
Number of GCLKs:	3 out of 8 37%
Timing Summary	

Speed Grade: -5	
Minimum period : 6.335 ns (Maximum Frequency : 157.858MHz)	
Minimum input arrival time before clock : 5.228ns	
Maximum output required time after clock : 6.216 ns	
Maximum combinational path delay : No path found	

ภาพประกอบ 3-17 จำนวนทรัพยากรของเอฟพีจีเอที่ถูกใช้ในวงจรย่อย Full Period Meter

3.2.4 วงจรย่อย Frequency Meter

ส่วนนี้เป็นส่วนหนึ่งของการวัดอัตราการหมุนของเอนโค้ดเดอร์โดยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ ซึ่งวิธีการวัดจะเป็นตามหัวข้อ 2.4.2

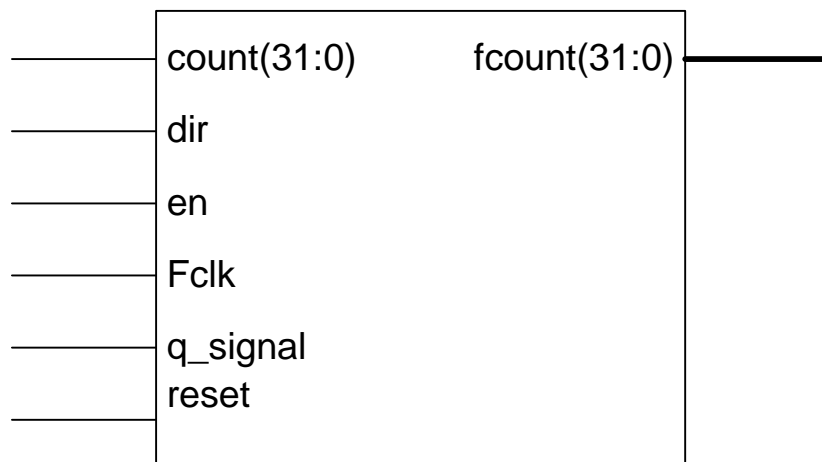
โดยในส่วนนี้ จะเป็นวงจรมับพัลส์สัญญาณ Q ที่เกิดจากวงจรย่อย Quadrature Generator ส่วนช่วงเวลาในการนับ T_f จะใช้การนับจำนวนพัลส์ของสัญญาณนาฬิกาที่มีความถี่สูงที่ทราบค่าคาบเวลาที่แน่นอน

ค่าที่นับได้จากวงจรส่วนนี้ คือค่าที่นับจำนวนพัลส์สัญญาณ Q ซึ่งคือตัวแปร n_f มีขนาด 32 บิต ซึ่งจะนับได้ถึง 2^{32} ค่า

ซึ่งวงจรส่วนนี้มีสัญญาณนำเข้าและนำออกดังภาพประกอบ 3-18 และภาพประกอบ 3-19

Entity Frequency_Est is	
Port (Fclk	: in std_logic;
reset	: in std_logic;
en	: in std_logic;
count	: in std_logic_vector(31 downto 0);
q_signal	: in std_logic;
dir	: in std_logic
fcount	: out std_logic_vector (31 downto 0) ;
End Frequency_Est;	

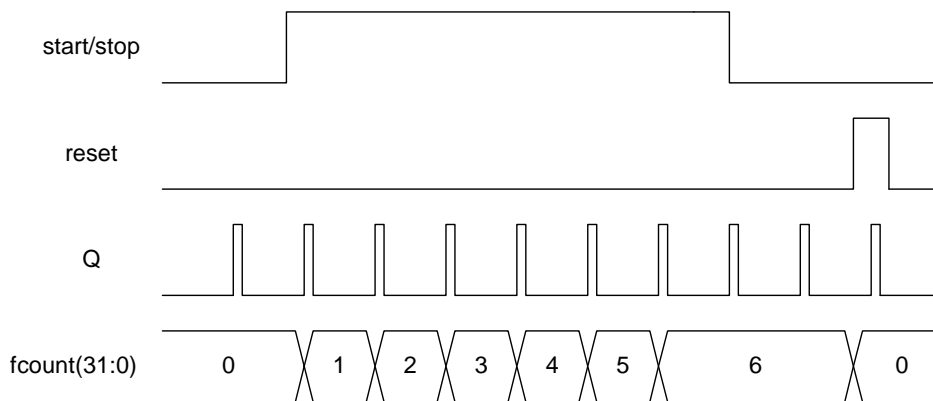
ภาพประกอบ 3-18 การประกาศสร้างสัญญาณนำเข้าและนำออกของวงจรย่อย Frequency Meter



ภาพประกอบ 3-19 วงจรระดับบนจากการสังเคราะห์วงจรย่อย Frequency Meter

สัญญาณนำเข้าและนำออกของวงจรวัดความถี่สัญญาณเอนโค้ดเดอร์ มีดังนี้

- Fclk เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งเป็นสัญญาณนาฬิกาที่มีความถี่สูงกว่าความถี่สัญญาณ Q เพื่อให้สามารถตรวจจับการเปลี่ยนแปลงของสัญญาณ Q ได้
- reset เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณเพื่อทำการเคลียร์ค่าสัญญาณภายในวงจรนี้เพื่อเริ่มต้นการทำงานใหม่
- count เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณขนาด 32 บิต ที่หมายถึงค่าช่วงเวลา T_f ตามสมการ 2-2 เพื่อทำการนับพัลส์สัญญาณ Q
- q_signal เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือสัญญาณ Q ซึ่งรับมาจากวงจรย่อย Quadrature Generator
- dir เป็นสัญญาณนำเข้าสู่วงจรย่อยขนาด 1 บิต ซึ่งคือ สัญญาณบอกทิศทางการหมุนเอนโค้ดเดอร์ ใช้เพื่อกำหนดการหยุดวัดความถี่ เมื่อตรวจสอบพบว่าเอนโค้ดเดอร์มีการเปลี่ยนทิศทาง
- fcount เป็นสัญญาณนำออกจากวงจรย่อยขนาด 32 บิต ซึ่งคือค่าการนับสัญญาณ Q คือค่า n_f ในสมการ 2-2



ภาพประกอบ 3-20 การนับสัญญาณ Q ในวงจรย่อย Frequency Meter

การทำงานของวงจรวัดความถี่สัญญาณเอนโคเดอร์ที่ออกแบบนี้ ตามภาพประกอบ 3-20 ในการนับสัญญาณ Q จะเริ่มทำการนับเมื่อ start/stop มีค่าเป็นลอจิกสูง “1” และวงจรไม่ถูกรีเซ็ต (reset มีค่าเป็นลอจิกสูง “1”) สำหรับช่วงเวลาที่ มีค่าเป็นลอจิกสูง “1” ของสัญญาณ start/stop นั้นจะขึ้นอยู่กับค่า count ซึ่งเป็นค่าช่วงเวลาที่ต้องการนับสัญญาณ Q และจะหยุดนับสัญญาณ Q เมื่อ start/top มีค่าเป็นลอจิกต่ำ “0” และจะคงค่านั้นไว้เพื่อเก็บค่า fcount ไปทำการคำนวณหาอัตราเร็วต่อไป และจะเคลียร์ค่าเมื่อ reset มีค่าเป็นลอจิกสูง “1” สำหรับการตรวจสอบสัญญาณ Q เพื่อนับในวงจรนี้นั้นจะเหมือนกับวิธีการนับสัญญาณ Q ของวงจรย่อยวัดตำแหน่งดังที่กล่าวไปแล้วในหัวข้อ 3.2.2

วงจรย่อย Frequency Meter นี้ เมื่อออกแบบด้วยภาษา VHDL และทำการสังเคราะห์วงจรด้วยโปรแกรม Xilinx 7.1i แล้วได้ผลด้านการใช้ทรัพยากรและเวลาดังภาพประกอบ 3-21

Device utilization summary:	

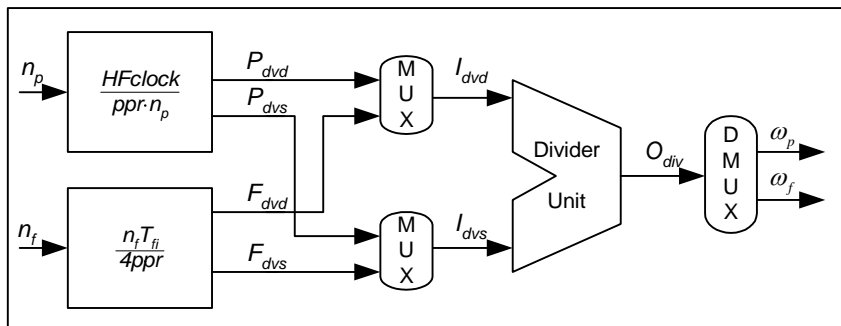
Selected Device : 3s200tq144-5	
Number of Slices:	103 out of 1920 5%
Number of Slices Flip Flops :	131 out of 3840 3%
Number of 4 input LUTs:	106 out of 3840 2%
Number of bonded IOBs:	69 out of 97 71%
Number of GCLKs:	2 out of 8 25%
Timing Summary	

Speed Grade: -5	
Minimum period : 7.839 ns (Maximum Frequency : 127.571MHz)	
Minimum input arrival time before clock : 7.988ns	
Maximum output required time after clock : 6.216 ns	
Maximum combinational path delay : No path found	

ภาพประกอบ 3-21 จำนวนทรัพยากรของเอฟพีจีเอที่ถูกใช้ในวงจรย่อย Frequency Meter

3.3 ส่วนคำนวณ

ส่วนการคำนวณนี้เป็นการคำนวณหาค่าอัตราเร็วการหมุนของเอนโค้ดเดอร์ ด้วยวิธีการวัดอัตราเร็วการหมุนตามที่อธิบายในหัวข้อ 2.4 ทั้ง 2 วิธีคือวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ และวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์



ภาพประกอบ 3-22 ส่วนคำนวณหาอัตราเร็ว

เมื่อวงจรย่อย Full Period Meter และวงจรย่อย Frequency Meter วัดค่า n_p และ n_f ได้ตามลำดับแล้วก็จะส่งค่าทั้งสองมายังส่วนของคำนวณนี้ ซึ่งมีรายละเอียดดังนี้

3.3.1 การคำนวณหาอัตราโดยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์

จากสมการคำนวณหาอัตราเร็วโดยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ในสมการ 2-1 ซึ่ง $\omega_p = \frac{\alpha_p}{n_p \cdot t}$ โดยจะเห็นได้ว่าค่า α_p คือค่ามุมที่เอนโค้ดเดอร์เคลื่อนที่ไป 1 จังหวะ ซึ่งเอนโค้ดเดอร์แต่ละตัวมีค่ามุมนี้ต่างกัน ขึ้นอยู่กับจำนวนพัลส์ต่อรอบ (ppr) ของเอนโค้ดเดอร์แต่ละตัว

ดังนั้น ค่ามุมที่เอนโค้ดเดอร์เคลื่อนที่ไป 1 จังหวะ เท่ากับ

$$\alpha_p = \frac{1}{ppr} \tag{3-1}$$

และจากค่า t ในสมการ 2-1 เป็นค่าคาบเวลาของสัญญาณนาฬิกาความถี่สูง ค่า t จึงค่าที่มีค่าต่ำมาก ซึ่งถ้าเขียนในหน่วยวินาทีจะเป็นค่าทศนิยมที่ต่ำมาก จึงทำการปรับค่าจากค่าคาบเป็นค่าความถี่ของสัญญาณนาฬิกาแทน

ดังนั้น จากค่า $\frac{1}{t}$ ในสมการ 2-1 เปลี่ยนเป็นค่าความถี่ได้ดังนี้

$$C_{HF} = \frac{1}{t} \tag{3-2}$$

ดังนั้น จากสมการการคำนวณหาค่าอัตราเร็วโดยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ 2-1 สามารถแทนด้วยค่าตัวแปรด้วยสมการ 3-1 และสมการ 3-2 ได้ดังนี้

$$\omega_p = \frac{C_{HF}}{n_p \cdot ppr} \quad (3-3)$$

3.3.2 การคำนวณหาค่าอัตราเร็วโดยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

การคำนวณหาค่าอัตราเร็วการหมุนของเอนโค้ดเดอร์ด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ โดยทั่วไปสามารถหาได้โดยใช้สมการ 2-2 $\omega_f = \frac{n_f \cdot \alpha_f}{T_f}$

จากภาพประกอบ 2-8 และสมการ 2-2 จะเห็นได้ว่าค่า α_f คือค่ามุมเล็กที่สุดที่เอนโค้ดเดอร์เคลื่อนที่ได้ซึ่งเกิดจากการทำให้สัญญาณจากเอนโค้ดเดอร์สองสัญญาณมีความละเอียดเพิ่มขึ้น 4 เท่า ดังนั้นค่า α_f นี้จะต่างกันเมื่อจำนวนพัลส์ต่อรอบของเอนโค้ดเดอร์ต่างกัน

ดังนั้น ค่ามุมเล็กที่สุดของเอนโค้ดเดอร์เท่ากับ

$$\alpha_f = \frac{1}{4 \cdot ppr} \quad (3-4)$$

และค่า T_f ในสมการ 2-2 เป็นค่าช่วงเวลาที่ใช้ในการนับ ซึ่งถ้าน้อยกว่า 1 วินาที จะทำให้ค่า T_f เป็นค่าทศนิยม ซึ่งไม่เหมาะสมกับการออกแบบวงจรส่วนอื่น เนื่องจากได้ออกแบบวงจรให้คำนวณการคูณเป็นแบบเลขจำนวนเต็ม จึงต้องปรับค่าให้เป็นเลขจำนวนเต็ม ดังนี้

$$T_{fi} = \frac{1}{T_f} \quad (3-5)$$

จากสมการ 2-2 สามารถแทนค่าตัวแปรด้วยสมการ 3-4 และสมการ 3-5 ซึ่งจะได้สมการการคำนวณหาค่าอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ใหม่ดังนี้

$$\omega_f = \frac{n_f \cdot T_{fi}}{4 \cdot ppr} \quad (3-6)$$

จากภาพประกอบ 3-4 จะเห็นได้ว่าส่วนการคำนวณนี้ จะรับค่า n_p และ n_f จากวงจรย่อย Full Period Meter และวงจรย่อย Frequency Meter ตามลำดับ ซึ่งเมื่อจะคำนวณตามสมการ 3-3 และ 3-6 ซึ่งเป็นการหารที่ตัวตั้งและตัวหารเป็นเลขจำนวนเต็ม ดังนี้

จากสมการ 3-3 แยกได้ตัวตั้ง P_{dvd} และตัวหาร P_{dvs} ดังนี้

$$P_{dvd} = C_{HF} \quad (3-7)$$

$$P_{dvs} = n_p \cdot ppr \quad (3-8)$$

และจากสมการ 3-6 แยกได้ ตัวตั้ง F_{dvd} และตัวหาร F_{dvs} ดังนี้

$$F_{dvd} = n_f \cdot T_{fi} \quad (3-9)$$

$$F_{dvs} = 4 \cdot ppr \quad (3-10)$$

ซึ่งส่วนที่ทำการคำนวณเพื่อให้ได้ตัวตั้งและตัวหาร นี้คือส่วนเตรียมข้อมูลการคำนวณซึ่งจะแยกออกเป็น 2 ส่วน ส่วนหนึ่งจากการคำนวณด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์อีกส่วนหนึ่งมาจากการคำนวณด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

จากสมการ 3-3 และสมการ 3-6 จะเห็นได้ว่าผลลัพธ์เกิดจากการหาร ดังนั้น ส่วนการคำนวณนี้ จึงต้องมีวงจรหารเป็นส่วนประกอบสำคัญ และเนื่องจากวงจรหารเป็นวงจรที่ใช้ทรัพยากรของเอพฟิอีค่อนข้างมาก จึงจำเป็นต้องใช้วงจรหารเพียงวงเดียวสำหรับการต่อเอนโค้ดเดอร์ 1 ชุด เพื่อประหยัดทรัพยากรของเอพฟิอีไว้ใช้สร้างวงจรส่วนอื่นๆ ต่อไป

จากที่กล่าวข้างต้นว่ามีวงจรหารเพียงชุดเดียว แต่ต้องการหารค่าจาก 2 สมการ จึงออกแบบให้ต้องมีวงจรมัลติเพล็กซ์ (Multiplexer : MUX) แบบ 2 นำเข้า 1 นำออก ดังภาพประกอบ 3-1 เพื่อผลัดกันนำข้อมูลจากสมการใดสมการหนึ่งเข้าไปคำนวณหาผลลัพธ์อัตราเร็วขณะนั้น ๆ วงจรมัลติเพล็กซ์ตัวหนึ่งจะเลือกส่งข้อมูลส่วนตัวตั้ง (I_{dvd}) เข้าสู่วงจรหารและวงจรมัลติเพล็กซ์อีกตัวจะเลือกส่งข้อมูลส่วนตัวหาร (I_{dvs}) เข้าสู่วงจรหาร

จากนั้นค่าตัวตั้งและตัวหารจากสมการหนึ่งๆ จะถูกคำนวณด้วยวงจรหารจากภาพประกอบ 3-1 วงจรหารเป็นการหารแบบจำนวนเลขทศนิยม ดังนั้นข้อมูลตัวตั้งและข้อมูลตัวหารจะต้องถูกเปลี่ยนรูปแบบข้อมูลจากแบบเลขจำนวนเต็มเป็นข้อมูลแบบทศนิยมก่อน เนื่องจากนำเข้าของวงจรหารที่ใช้เป็นข้อมูลแบบเลขทศนิยม [12] ซึ่งผลลัพธ์ที่ได้ก็จะอยู่ในรูปเลขทศนิยมมีขนาด 32 บิต

หลังจากวงจรหารคำนวณหาอัตราเร็วการหมุนของเอนโค้ดเดอร์ได้แล้ว ก็จะถูกส่งไปยังส่วนเก็บข้อมูล แต่ผลลัพธ์จากวงจรหารอาจเป็นอัตราการหมุนจากวิธีการวัดความกว้างสัญญาณเอนโค้ดเดอร์หรืออาจเป็นอัตราเร็วการหมุนจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ จึงต้องมีวงจรมัลติเพล็กซ์ (De-Multiplexer : DMUX) เพื่อแยกว่าข้อมูลที่ออกมาจากวงจรหารนั้นเป็นข้อมูลจากวิธีการใดเพื่อเก็บในตำแหน่งของส่วนเก็บข้อมูลได้ถูกต้อง

การออกแบบวงจรในหัวข้อ 3.2 และ 3.3 จะถูกสร้างขึ้นมาสองชุด สำหรับเอนโค้ดเดอร์สองตัว นั่นคือ วงจรส่วนปรับแต่งสัญญาณนำเข้าและวัดค่าและวงจรส่วนการคำนวณหนึ่งชุดออกมาให้ใช้งานกับเอนโค้ดเดอร์เพียงหนึ่งตัวเท่านั้น

3.4 ส่วนเก็บข้อมูล

ส่วนเก็บข้อมูล (Register Array) จะเก็บข้อมูลค่าเริ่มต้นต่างๆ ของระบบที่จำเป็นในการทำงานและการคำนวณ และผลลัพธ์จากการคำนวณทั้งค่าตำแหน่งและค่าอัตราเร็วการหมุนจากทั้ง 2 วิธี

ในการออกแบบวงจรส่วนนี้ ออกแบบให้มีส่วนเก็บข้อมูลวงจรเดียวซึ่งเก็บค่าที่เกี่ยวข้องกับเอนโค้ดเดอร์ทั้ง 2 ตัวตั้งที่กล่าวข้างต้นไว้ในชุดเก็บข้อมูลเดียวกัน เพื่อความสะดวกในการใช้งานของผู้ใช้เพื่อต้องการจัดการข้อมูลกับส่วนเก็บข้อมูลนี้

วงจรเก็บข้อมูลนี้ออกแบบในลักษณะของชุดรีจิสเตอร์โดยขนาดของรีจิสเตอร์แต่ละตัวมีขนาด 32 บิต และกำหนดข้อจำกัดการเข้าถึงให้กับรีจิสเตอร์แต่ละตัวต่างกัน ขึ้นกับลักษณะการใช้งานของรีจิสเตอร์ตัวนั้น ๆ เช่น รีจิสเตอร์เก็บค่าอัตราเร็วการหมุน ผู้ใช้จะไม่สามารถเขียนค่าลงไปได้อ่านค่าได้อย่างเดียว แต่รีจิสเตอร์เก็บค่าจำนวนพัลส์ต่อรอบของเอนโค้ดเดอร์ ผู้ใช้สามารถอ่านและเขียนค่ายังรีจิสเตอร์ตัวนั้นได้ เป็นต้น

3.5 ส่วนติดต่อกับอุปกรณ์ภายนอก

ส่วนการติดต่อกับอุปกรณ์ภายนอก แบ่งเป็นการติดต่อกับอุปกรณ์ภายนอก 2 ชนิด คือ เอนโค้ดเดอร์และไมโครคอนโทรลเลอร์

3.5.1 การติดต่อกับเอนโค้ดเดอร์

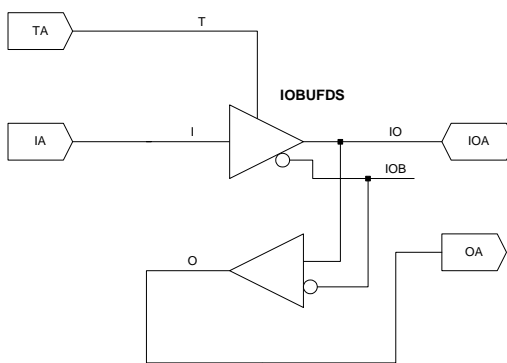
เป็นการสร้างช่องสัญญาณสำหรับนำเข้าสัญญาณจากเอนโค้ดเดอร์ คือ สัญญาณ A, สัญญาณ B และสัญญาณอินดิเคซ์ ช่องสัญญาณนี้กำหนดให้เป็นแบบช่องสัญญาณนำเข้าเพียงอย่างเดียว (IBUF) [13] สำหรับการติดต่อกับเอนโค้ดเดอร์ 2 ตัวนั้น ได้ออกแบบการต่อกับเอนโค้ดเดอร์แต่ละตัวแยกจากกันคือ เอนโค้ดเดอร์แต่ละตัวจะมีช่องชุดนำเข้าสัญญาณนำเข้าทั้ง 3 ดังที่กล่าวข้างต้นแยกจากกัน

3.5.2 ส่วนการติดต่อกับไมโครคอนโทรลเลอร์

ส่วนนี้จะมีสัญญาณ 3 ชนิด ที่ติดต่อกับไมโครคอนโทรลเลอร์ คือสัญญาณนำเข้า-ส่งออก สัญญาณข้อมูลระหว่างเอพฟิอีกับไมโครคอนโทรลเลอร์ขนาด 32 บิต, สัญญาณอ้างอิงตำแหน่งของรีจิสเตอร์ภายในเอพฟิอีที่ไมโครคอนโทรลเลอร์ติดต่อด้วย และสัญญาณควบคุมการทำงานของเอพฟิอี

1) ส่วนนำเข้า – ส่งออกสัญญาณข้อมูลระหว่างเอพฟิอีกับไมโครคอนโทรลเลอร์

วงจรส่วนนี้ออกแบบให้เป็นช่องทางแบบ 2 ทิศทาง (Bi-directional IO Port) ขนาด 32 บิต เพื่อเป็นช่องทางนำเข้าข้อมูลและส่งออกข้อมูลระหว่างเอพฟิอีและไมโครคอนโทรลเลอร์โดยเลือกใช้ส่วนประกอบ (Component) ที่เป็นทรัพยากรของเอพฟิอีคือ IOBUFDS ซึ่งเป็นบัฟเฟอร์ (Buffer) แบบ 2 ทิศทาง ซึ่งอยู่ในไลบรารี UNISIM [14]



ภาพประกอบ 3-23 วงจรสังเคราะห์ของ IOBUFDS ขนาด 1 บิต

จากภาพประกอบ 3-23 จะเห็นได้ว่าส่วนประกอบ IOBUFDS มีขนาด 1 บิต ซึ่งจะรองรับสัญญาณแบบ 3 สถานะ (Tri-state) ถ้าต้องการให้ช่องทางมีสถานะเป็น 3 สถานะ จะต้องให้สัญญาณ TA มีค่าเป็นลอจิกสูง “1” ซึ่งใช้เมื่อต้องการใช้ช่องทางนั้นเป็นสำหรับนำข้อมูลจากภายนอกเข้าเอพฟิอี และถ้าสัญญาณ TA มีค่าเป็นลอจิกต่ำ “0” จะเป็นการส่งออกข้อมูลจากภายในเอพฟิอีสู่ภายนอก แต่ส่วนการติดต่อที่ต้องการเป็น 32 บิต จึงต้องทำการเรียกใช้ส่วนประกอบดังกล่าว 32 ชุด เพื่อให้ได้ช่องทางนำเข้า-ส่งออกข้อมูลแบบ 2 ทิศทางขนาด 32 บิต แต่การใช้งานสามารถเลือกใช้งานแบบ 8 บิต หรือ 16 บิตได้เพื่อความสะดวกต่อการใช้กับไมโครคอนโทรลเลอร์ขนาดเล็ก โดยผู้วิจัยได้ตัดโหมดความละเอียดขนาด 24 บิตออกไปเนื่องจากพิจารณาแล้วเห็นว่าการคำนวณของโปรเซสเซอร์ส่วนใหญ่เป็นแบบ 8 บิต 16 บิต และ 32 บิตเท่านั้น โดยการกำหนดค่าสัญญาณเลือกขนาดช่องทางการรับส่งข้อมูล ถ้าค่าสัญญาณเลือกขนาดช่องทางการรับส่งข้อมูลมีค่าลอจิกเป็น “00” จะเป็นการเลือกใช้ช่องสัญญาณขนาด 32 บิต , ลอจิกเป็น “01” จะเป็นการเลือกใช้ช่องสัญญาณขนาด 16 บิต และลอจิกเป็น “10” จะเป็นการเลือกใช้ช่องสัญญาณขนาด 8 บิต

2) ส่วนสัญญาณอ้างอิงตำแหน่ง

ช่องสัญญาณนี้เป็นแบบช่องนำเข้าเพียงอย่างเดียว เนื่องจากใช้สำหรับให้ผู้ใช้ป้อนค่าตำแหน่งของรีจิสเตอร์ที่ต้องการติดต่อด้วย ออกแบบไว้มีขนาด 5 บิต ซึ่งสามารถอ้างอิงตำแหน่งของรีจิสเตอร์ได้ 32 ตำแหน่ง ซึ่งเพียงพอสำหรับการเก็บข้อมูลต่างๆ ที่เกี่ยวข้องกับเอน โค้ดเดอร์ ทั้ง 2 ตัว

3) ส่วนสัญญาณควบคุมการทำงานของเอฟพีจีเอ

สัญญาณส่วนนี้เป็นสัญญาณสำหรับควบคุมการทำงานของเอฟพีจีเอ ซึ่งประกอบด้วย

- ก) สัญญาณเริ่มต้นการทำงานใหม่ (Reset) ขนาด 1 บิต
- ข) สัญญาณอนุญาตการทำงานของเอฟพีจีเอ (Enable) ขนาด 1 บิต
- ค) สัญญาณเลือกขนาดช่องทางการรับส่งข้อมูล (Port size) ขนาด 2 บิต
- ง) สัญญาณอ่านหรือเขียนข้อมูล (Read/Write) ขนาด 1 บิต
- จ) สัญญาณควบคุมจังหวะการเขียนข้อมูลเข้า (Strobe) ขนาด 1 บิต
- ฉ) สัญญาณกำหนดชุดข้อมูล (Data Set) ขนาด 2 บิต

ในบทนี้ได้แสดงถึงโครงสร้างทั้งหมดวงจรเอฟพีจีเอที่ออกแบบเป็นวงจรเชื่อมต่อเอน โค้ดเดอร์ที่สามารถคำนวณค่าอัตราเร็วการหมุนและค่าตำแหน่งการเคลื่อนที่แบบหลายช่องสัญญาณ ซึ่งได้แสดงให้เห็นถึงวิธีการจัดการกับสัญญาณตั้งแต่การนำเข้าสัญญาณเอน โค้ดเดอร์ไปจนถึงการคำนวณค่าตำแหน่งและค่าอัตราเร็วการหมุนจากวิธีการวัดความกว้างคาบสัญญาณเอน โค้ดเดอร์ และวิธีการวัดความถี่สัญญาณพัลส์จากเอน โค้ดเดอร์ ไปจนถึงการนำค่าจากการคำนวณไปใช้

บทที่ 4

การทดสอบการทำงานของเอฟพีจีเอ

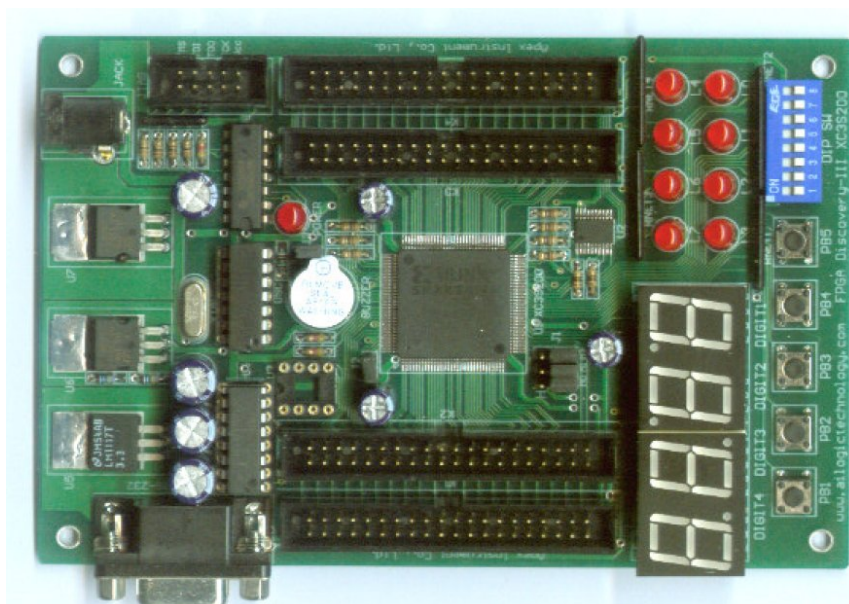
ในบทนี้อธิบายถึงการออกแบบการทดสอบการทำงานของเอฟพีจีเอ โดยการป้อนสัญญาณ เอนโค้ดเดอร์เฟส A และ เฟส B ที่ความถี่ต่างๆ และจำนวนพัลส์ต่อรอบของเอนโค้ดเดอร์แต่ละตัว แตกต่างกันไปแล้วทำการวัดความตำแหน่งการเคลื่อนและอัตราการหมุนของเอนโค้ดเดอร์

4.1 ทรัพยากรที่ใช้ในการทดสอบ

ในการวิจัยนี้เป็นการออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์แบบ 2 ช่องสัญญาณ ซึ่งวงจรนั้น ออกแบบโดยใช้เอฟพีจีเอ ซึ่งมีซอฟต์แวร์และอุปกรณ์ที่ใช้ในการออกแบบและทดสอบดังนี้

1. คอมพิวเตอร์ส่วนบุคคล (Personal Computer) ที่ใช้ในการทำงานวิจัยนี้เป็นคอมพิวเตอร์ ที่มีหน่วยประมวลผลกลางเป็น Pentium III อัตราเร็ว 866 MHz และหน่วยความจำ RAM ขนาด 256 MB

2. บอร์ดทดสอบเอฟพีจีเอ รุ่น FPGA Discovery-III XC3S200F ของบริษัท AILOGIC TECHNOLOGY จำกัด ซึ่งใช้เอฟพีจีเอของบริษัท Xilinx จำกัด รุ่น SPARTAN XC3X200-TQ144 ดังภาพประกอบ 4-1



ภาพประกอบ 4-1 บอร์ดทดลองเอฟพีจีเอ รุ่น FPGA Discovery-III XC3S200F

3. บอร์ดทดลอง MCS-51 ที่สามารถเชื่อมต่อกับเครื่องคอมพิวเตอร์ทางพอร์ตอนุกรมได้ และมีไอซี 8255 เพื่อใช้ในการขยายพอร์ตการเชื่อมต่อระหว่าง MCS-51 กับบอร์ดทดลองเอฟพีจีเอ

4. เอนโค้ดเดอร์ของบริษัท NIDEC NEMICON CORP. ซึ่งเป็นเอนโค้ดเดอร์แบบหมุนรุ่น OEW2-25-2MC ซึ่งให้ความละเอียดที่ 2500 พัลส์ต่อรอบ โดยให้สัญญาณ A และสัญญาณ B ที่ต่างกัน 90 องศา และให้สัญญาณอินดิเคซ์ ซึ่งสัญญาณเอาต์พุตเป็นแบบ Open Collector ดังภาพประกอบ 4-2 (ก) [15] และเอนโค้ดเดอร์ของบริษัท Hohner รุ่น SL 58-11112-5000 ซึ่งให้ความละเอียดที่ 5000 พัลส์ต่อรอบโดยให้สัญญาณ A และสัญญาณ B ที่ต่างกัน 90 องศาและให้สัญญาณอินดิเคซ์ ดังภาพประกอบ 4-2 (ข) [15]



(ก)

ภาพประกอบ 4-2 (ก) เอนโค้ดเดอร์ รุ่น OEW 2-25-2MC



(ข)

ภาพประกอบ 4-2 (ข) เอนโค้ดเดอร์ รุ่น SL 58-11112-5000

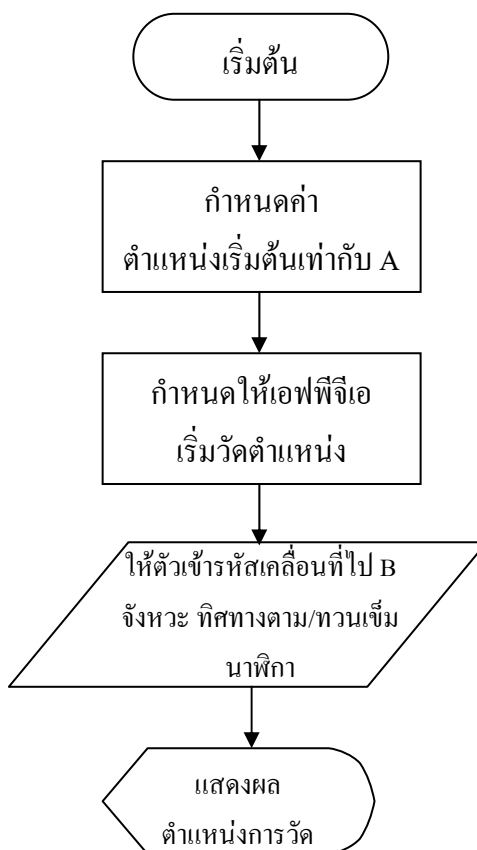
5. โปรแกรม Xilinx 7.1i สำหรับใช้ในการออกแบบวงจรด้วยภาษา VHDL ซึ่งโปรแกรมนี้อยู่มีความสามารถในการสังเคราะห์วงจรลงบนเอฟพีจีเอได้ด้วย

6. โปรแกรม uVision2 V2.40a สำหรับใช้ในการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เพื่อนำมาทดสอบกับเอฟพีจีเอ

4.2 การทดสอบการวัดตำแหน่ง

การทดสอบการวัดตำแหน่งการเคลื่อนที่นี้เป็นการวัดว่าเอนโค้ดเดอร์มีการหมุนไปแล้วกี่จังหวะ และเกิดขึ้นในทิศทางใด

ขั้นตอนการทดสอบการวัดตำแหน่งจะเป็นตามภาพประกอบ 4-3



ภาพประกอบ 4-3 ขั้นตอนการทดสอบการวัดตำแหน่ง

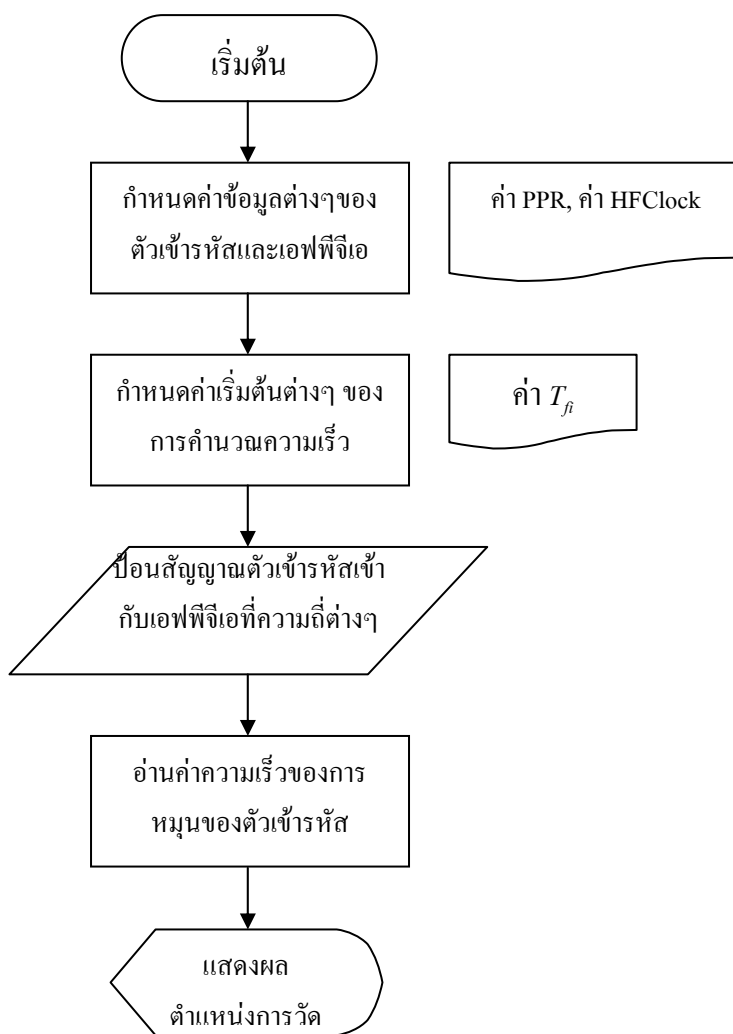
จากภาพประกอบ 4-3 เป็นขั้นตอนการทดสอบการวัดตำแหน่ง มีรายละเอียดดังต่อไปนี้

1. กำหนดค่าตำแหน่งเริ่มต้น A ให้กับเอพพีจีเอ ค่าเริ่มต้น A นี้คือค่าตำแหน่งปัจจุบันของเอนโค้ดเดอร์ โดยการเขียนค่าตำแหน่งปัจจุบันไปยังตำแหน่ง 05H หรือ 15H ของรีจิสเตอร์ สำหรับเอนโค้ดเดอร์ช่องที่ 1 และช่องที่ 2 ตามลำดับ
2. กำหนดให้เอพพีจีเอเริ่มต้นการทำงานวัดตำแหน่งโดยการส่งสัญญาณอนุญาตการทำงาน (Enable) เพื่อให้เอพพีจีเอเริ่มต้นการวัดตำแหน่ง
3. กำหนดให้เอนโค้ดเดอร์เคลื่อนที่ไปที่ B จังหวะโดยหมุนในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาทิศใดทิศหนึ่ง
4. เมื่อเอนโค้ดเดอร์เคลื่อนที่ไปครบตามที่กำหนดแล้วก็ทำการอ่านค่าตำแหน่งของเอนโค้ดเดอร์จากตำแหน่งที่เก็บค่าตำแหน่งของเอนโค้ดเดอร์ทั้ง 2 ตัว คือ ตำแหน่ง 00H และ 10H สำหรับเอนโค้ดเดอร์ที่ต่อกับช่องที่ 1 และช่องที่ 2 ตามลำดับ แล้วทำการเปรียบเทียบว่าค่าที่อ่านได้เท่ากับ $A+B$ หรือไม่เมื่อเอนโค้ดเดอร์หมุนตามเข็มนาฬิกา และเท่ากับ $A-B$ หรือไม่เมื่อเอนโค้ดเดอร์หมุนทวนเข็มนาฬิกา

โดยการทดสอบจะทดสอบกับสัญญาณเอนโค้ดเดอร์ที่ความถี่ต่างๆ ตั้งแต่ความถี่ต่ำๆ จนถึงความถี่สูงๆ และตรวจสอบความถูกต้องจากการวัด

4.3 การทดสอบการวัดอัตราเร็วการหมุน

การทดสอบการวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์ เป็นการคำนวณหาค่าการหมุนของเอนโค้ดเดอร์เป็นจำนวนรอบต่อวินาที ในการทดสอบนี้อัตราเร็วของเอนโค้ดเดอร์เป็นแบบอัตราเร็วคงที่ ซึ่งสัญญาณเอนโค้ดเดอร์เป็นสัญญาณที่สร้างขึ้นโดยเอฟพีจีเอ ซึ่งมีสัญญาณเฟส A สัญญาณเฟส B และสัญญาณอินเด็กซ์ ซึ่งขั้นตอนการทดสอบการวัดตำแหน่งจะเป็นตามภาพประกอบ 4-4



ภาพประกอบ 4-4 ขั้นตอนการทดสอบการวัดอัตราเร็วการหมุน

จากภาพประกอบ 4-4 ขั้นตอนการทดสอบการวัดอัตราเร็วการหมุน มีรายละเอียดดังนี้

1. กำหนดค่าข้อมูลของเอนโค้ดเดอร์ คือ ค่าจำนวนพัลส์ต่อรอบของเอนโค้ดเดอร์โดยการเขียนค่าจำนวนพัลส์ต่อรอบไปยังตำแหน่ง 03H และ 13H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และช่องที่ 2 ตามลำดับค่าความถี่ของสัญญาณนาฬิกาที่ป้อนให้เอฟพีจีเอทำงาน ซึ่งค่านี้จะมีเพียงค่าเดียว โดยเขียนไปยังตำแหน่ง 05H และค่าอัตราเร็วเกณฑ์ (V Threshold) โดยเขียนไปยังตำแหน่ง 08H และ 18H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และช่องที่ 2 ตามลำดับ

2. กำหนดค่าช่วงเวลาในการนับ (T_p) ของการวัดอัตราเร็วการหมุนด้วยการวัดความถี่สัญญาณเอนโค้ดเดอร์ โดยเขียนค่าช่วงเวลาไปยังตำแหน่ง 06H และ 16H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และช่องที่ 2 ตามลำดับ

3. ป้อนสัญญาณเอนโค้ดเดอร์ที่สร้างโดยเอฟพีจีเออีกตัวหนึ่ง เข้ากับเอฟพีจีเอที่ทำหน้าที่เป็นตัววัดอัตราเร็ว โดยป้อนสัญญาณเอนโค้ดเดอร์ที่ความถี่ต่างๆ

4. อ่านค่าอัตราเร็วการหมุนของเอนโค้ดเดอร์แต่ละช่องสัญญาณโดยอ่านค่าอัตราเร็วการหมุนของเอนโค้ดเดอร์จากเอฟพีจีเอได้ 6 ค่า คือค่าอัตราเร็วจากวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ โดยอ่านได้จากตำแหน่ง 01H และ 11H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และ 2 ตามลำดับค่าอัตราเร็วจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ โดยอ่านได้จากตำแหน่ง 02H และ 12H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และ 2 ตามลำดับ และค่าอัตราเร็วจากการใช้เกณฑ์ในการเลือก โดยอ่านได้จากตำแหน่ง 07H และ 17H สำหรับเอนโค้ดเดอร์ช่องที่ 1 และ 2 ตามลำดับ แล้วทำการเปรียบเทียบค่าอัตราเร็วที่อ่านได้กับการคำนวณโดยใช้ความถี่สัญญาณเอนโค้ดเดอร์ที่ป้อนเป็นค่ามาตรฐาน

โดยการทดสอบ จะทดสอบกับสัญญาณเอนโค้ดเดอร์ที่ความถี่ต่างๆ ตั้งแต่ความถี่ต่ำๆ จนถึงความถี่สูงๆ และตรวจสอบความถูกต้องของอัตราเร็วการหมุนที่วัดได้

ในบทนี้ได้แสดงถึงวิธีการทดสอบวงจรเอฟพีจีเอที่ออกแบบมาเพื่อใช้ในการวัดตำแหน่งและวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์ โดยแสดงรายละเอียดของวิธีการทดสอบและค่าที่จำเป็นต้องกำหนดให้กับเอฟพีจีเอเพื่อให้เอฟพีจีเอทำงานและคำนวณค่าอัตราเร็วได้

บทที่ 5

ผลการทดลองและอภิปรายผลการทดลอง

บทนี้อธิบายถึงผลการทดสอบการวัดตำแหน่งและผลการทดสอบการวัดอัตราเร็วการหมุนตามขั้นตอนการทดสอบที่ได้แสดงไว้ในบทที่ 4

5.1 ผลการทดสอบการวัดตำแหน่ง

วงจรวัดตำแหน่งเป็นวงจรที่จะนับจังหวะการหมุนของเอนโค้ดเดอร์ ซึ่งการนับจังหวะการหมุนของเอนโค้ดเดอร์ที่ให้นำออก 2 สัญญาณ คือ สัญญาณ A และสัญญาณ B ที่ต่างกัน 90 องศา นั้น สามารถแบ่งย่อยให้ละเอียดได้ถึง 4 เท่าของจำนวนพัลส์ของเอนโค้ดเดอร์ต่อรอบการหมุน ซึ่งก็คือผลลัพธ์ที่ได้จากวงจร Quadrature Generator

จากการออกแบบการทดสอบการวัดตำแหน่งในหัวข้อ 4.2 ได้ทำการทดสอบโดยการกำหนดให้ค่าเริ่มต้นตำแหน่งมีค่าเป็น 0, 120,000 และ -56,000 ค่าเริ่มต้นเหล่านี้เป็นค่าที่กำหนดขึ้นมาเอง โดยการสมมติระบบงานให้เป็นระบบงานหมุนของมอเตอร์ที่มีการเคลื่อนที่แนวราบบนแกนโดยมีจุดกึ่งกลางแกนเป็นจุดเริ่มต้นของตำแหน่งมอเตอร์ ซึ่งในที่นี้คือการกำหนดให้ค่าเริ่มต้นเท่ากับ 0 โดยการเขียนค่าไปยังตำแหน่ง 05H และสมมติให้มอเตอร์ย้ายจุดเริ่มต้นไปทางขวาของจุดกึ่งกลางแกนเป็นระยะ 120,000 จังหวะ (แบบความละเอียด 4 เท่า) และสมมติให้มอเตอร์มีจุดเริ่มต้นที่ทางซ้ายของจุดกึ่งกลางแกนเป็นระยะ 56,000 จังหวะ (แบบความละเอียด 4 เท่า)

จากการกำหนดจุดเริ่มต้น 3 จุด ข้างต้น แต่ละจุดจะทำการทดสอบแยกครั้งละ 1 จุด ซึ่งเมื่อกำหนดจุดเริ่มต้นแล้วก็ได้ทำการทดสอบการวัด โดยการป้อนสัญญาณเอนโค้ดเดอร์ที่มีความถี่ต่างๆ ซึ่งจะกำหนดให้เอนโค้ดเดอร์เคลื่อนที่ไปที่จังหวะ ดังนี้

สมมติให้สัญญาณเอนโค้ดเดอร์มีความถี่ 8.6 kHz และกำหนดให้มีการป้อนสัญญาณเอนโค้ดเดอร์เป็นระยะเวลา 1 วินาที จะได้ว่า

$$\begin{aligned} \text{จำนวนจังหวะการเคลื่อนที่} &= 4 \times 1 \text{ (วินาที)} \times 8600 \text{ (จังหวะต่อวินาที)} \\ &= 34400 \text{ จังหวะ} \end{aligned}$$

ถ้ากำหนดให้ความถี่ของสัญญาณเอนโค้ดเดอร์เป็น F_e ระยะเวลาที่ป้อนสัญญาณเอนโค้ดเดอร์เป็น t และวงจรวัดใช้สัญญาณที่มาจากวงจร Quadrature Generator ที่ให้ความละเอียดคูณ 4 จะได้จำนวนจังหวะการเคลื่อนที่ $step$ ดังสมการ 5-1

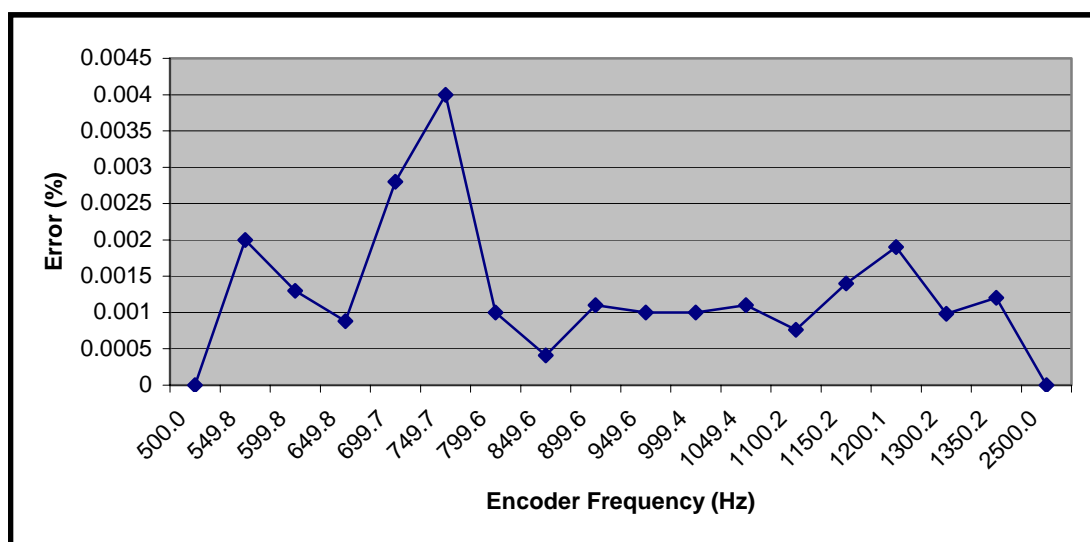
$$step = 4 \cdot t \cdot F_e \quad (5-1)$$

กำหนดให้ Step เป็นจำนวนจังหวะที่จะเกิดขึ้นในช่วงเวลาที่สัญญาณเอนโค้ดเดอร์ที่ค่าความถี่หนึ่งเกิดขึ้นในช่วงเวลาที่กำหนดไว้ ซึ่งในที่นี้กำหนดให้มีค่าระยะเวลาป้อนสัญญาณเอนโค้ดเดอร์เท่ากับ 5 วินาที ซึ่งได้ผลดังตาราง 5-1

ตาราง 5-1 ผลการทดสอบการวัดตำแหน่งเมื่อกำหนดค่าเริ่มต้นตำแหน่งเท่ากับ 0

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ค่าจังหวะที่นับได้โดย เอฟพีจีเอ	ค่าจังหวะที่ควร จะเป็น	ร้อยละของความ ผิดพลาด (%)
5.0	100	100.0	0.0×10^0
10.0	200	200.0	0.0×10^0
20.0	400	400.0	0.0×10^0
50.0	1000	100.0	0.0×10^0
100.0	2000	2000.0	0.0×10^0
500.0	10000	10000.0	0.0×10^0
549.8	10996	10995.8	2.0×10^{-3}
599.8	11996	11996.2	1.3×10^{-3}
649.8	12995	12995.1	8.8×10^{-4}
699.7	13995	13994.6	2.8×10^{-3}
749.7	14994	14993.4	4.0×10^{-3}
799.6	15993	15992.8	1.0×10^{-3}
849.6	16993	16992.9	4.1×10^{-4}
899.6	17991	17990.8	1.1×10^{-3}
949.6	18991	18991.2	1.0×10^{-3}
999.4	19987	19987.2	1.0×10^{-3}
1049.4	20987	20987.2	1.1×10^{-3}
1100.2	22003	22003.2	7.6×10^{-4}

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ค่าจิ้งหะที่นับได้โดย เอฟพีจีเอ	ค่าจิ้งหะที่ควร จะเป็น	ร้อยละของความ ผิดพลาด (%)
1150.2	23003	23003.3	1.4×10^{-3}
1200.1	24002	24001.5	1.9×10^{-3}
1300.2	26004	26003.7	9.8×10^{-4}
1350.2	27004	27003.7	1.2×10^{-3}
2.5 k	50000	50000.0	0.0×10^0
5.0 k	100000	100000.0	0.0×10^0
8.6 k	172176	172176.0	0.0×10^0
10.0 k	200000	200000.0	0.0×10^0
25.0 k	500000	500000.0	0.0×10^0
50.0 k	1000000	1000000.0	0.0×10^0
75.3 k	1506024	1506024.0	0.0×10^0
125.0 k	2500000	2500000.0	0.0×10^0
152.4 k	3048780	3048780.0	0.0×10^0



ภาพประกอบ 5-1 ร้อยละของความผิดพลาดจากการวัดตำแหน่ง

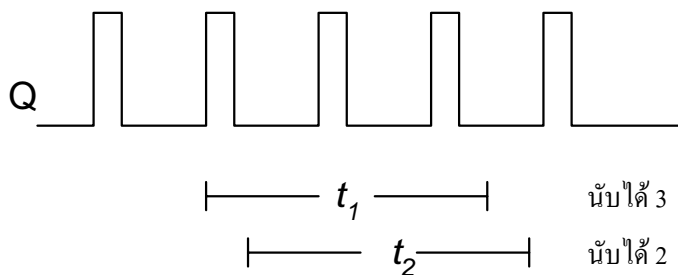
จากตาราง 5-1 และภาพประกอบ 5-1 เป็นค่าร้อยละของความผิดพลาดจากการทดสอบการวัดตำแหน่งโดยมีความถี่สัญญาณที่ใช้ในการตรวจสัญญาณ Q เท่ากับ 25 MHz และกำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ จะเห็นว่าที่ความถี่สัญญาณเอนโค้ด

เดอริ่งค่ามีค่าร้อยละของความผิดพลาดเกิดขึ้น แต่ในขณะที่บางค่ามีค่าร้อยละของความผิดพลาดเป็น 0.0 % ซึ่งในการทดสอบทั้งการย้ายจุดเริ่มต้นไปทางซ้ายของแกนหรือทางขวาของแกน ก็เกิดความผิดพลาดขึ้นเช่นกัน ค่าร้อยละของความผิดพลาดสูงสุดเท่ากับ 4.0×10^{-3}

ค่าร้อยละของความผิดพลาดนั้นไม่สามารถหลีกเลี่ยง ซึ่งเกิดขึ้นจากเมื่อทำการคำนวณหาจำนวนจังหวะที่ควรจะเป็นนั้น เป็นการคำนวณตามสมการ 5-1 ทำให้จำนวนจังหวะที่ควรจะเป็นมีค่าทศนิยมเกิดขึ้นจากราย 5-1 จะเห็นได้ว่าค่าร้อยละของความผิดพลาดที่เกิดขึ้นนั้น จะปรากฏเฉพาะกรณีที่มีมีการคำนวณค่าจังหวะที่ควรจะเป็นมีค่าเป็นเลขทศนิยมเท่านั้น

ตัวอย่าง กรณีสัญญาณเอนโค๊ดเดอร์มีความถี่ 1150.2 Hz ค่าตำแหน่งที่ได้จากการวัดด้วยเอฟพีจีเอ เท่ากับ 23003 จังหวะ แต่ค่าตำแหน่งที่เกิดจากการคำนวณตามสมการ 5-1 นั้นได้จำนวนจังหวะที่ควรจะเป็นเท่ากับ 23003.3 จังหวะ นั่นคือสัญญาณจากเอนโค๊ดเดอร์ที่มีความถี่ 1150.2 Hz นั้นมีช่วงที่เป็นลอจิก 1 เกิดขึ้น 23003 ครั้ง ในช่วง 5 วินาที ซึ่งค่าจังหวะ 0.3 นั้นเป็นช่วงที่เป็นลอจิก 0 ซึ่งยังเกิดสัญญาณเอนโค๊ดเดอร์ไม่ครบ 1 พัลส์ทำให้ค่าจังหวะที่เป็นทศนิยมนี้เป็นความผิดพลาดที่แสดงให้เห็นในตาราง 5-1 และภาพประกอบ 5-1

ความผิดพลาดในการวัด สามารถเกิดขึ้นได้อีกกรณีที่จุดเริ่มต้นการนับ ไม่ได้อยู่ที่ตำแหน่งเริ่มต้นของพัลส์สัญญาณเอนโค๊ดเดอร์ ดังภาพประกอบ 5-2



ภาพประกอบ 5-2 ความผิดพลาดที่อาจเกิดขึ้นในการวัดตำแหน่ง

จากภาพประกอบ 5-2 จะให้ได้ว่า ถ้าค่าเวลา t_1 และ t_2 เท่ากัน แต่จุดเริ่มต้นทำการนับต่างกัน เวลา t_1 เริ่มนับที่ขอบขาขึ้นของสัญญาณ Q พอดี ทำให้นับจังหวะได้ 3 จังหวะ แต่เวลา t_2 เริ่มนับในช่วงที่สัญญาณ Q มีค่าลอจิกเป็น 0 แม้จะมีค่าเวลาเท่ากัน แต่จำนวนจังหวะที่นับได้กลับไม่เท่ากันซึ่งนับได้เพียง 2 จังหวะ

5.2 ผลการทดสอบการวัดอัตราเร็วการหมุน

การทดสอบการวัดอัตราเร็วการหมุน จะมีการทดสอบเพื่ออ่านค่าจากการวัดทั้ง 3 วิธีคือค่าจากการวัดด้วยวิธีการวัดความกว้างคาบสัญญาณตัวเข้า วิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ และวิธีการกำหนดค่าเกณฑ์

5.2.1 วิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์

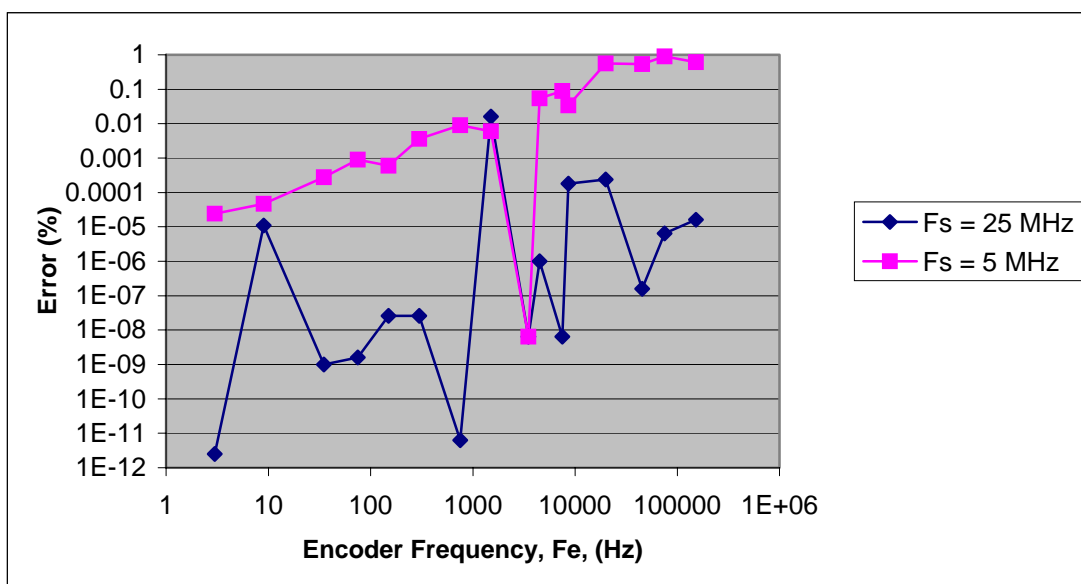
การทดสอบการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ได้ทำการทดสอบใช้ความถี่สุมหาช่วงคาบสัญญาณเอนโค้ดเดอร์ที่ 25 MHz และ 5 MHz โดยทดสอบกับการป้อนสัญญาณเอนโค้ดเดอร์ที่ความถี่ต่ำ จนถึงความถี่สูงถึง 150 kHz โดยกำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ ซึ่งเมื่อนำค่าอัตราเร็วที่วัดได้มาคำนวณหาค่าร้อยละของความผิดพลาด ได้ดังตาราง 5-2

ตาราง 5-2 ค่าร้อยละของความผิดพลาดจากวิธีการวัดความกว้างคาบ

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ความผิดพลาด (%)	
	$F_s = 25.0 \text{ MHz}$	$F_s = 5.0 \text{ MHz}$
3.0	2.5×10^{-12}	2.4×10^{-5}
9.0	1.1×10^{-5}	4.7×10^{-5}
34.9	1.0×10^{-9}	2.8×10^{-4}
75.0	1.6×10^{-9}	9.0×10^{-4}
150.0	2.6×10^{-8}	6.0×10^{-4}
300.0	2.6×10^{-8}	3.6×10^{-3}
750.0	6.3×10^{-12}	9.0×10^{-3}
1.5 k	1.6×10^{-2}	6.0×10^{-3}
3.5 k	6.4×10^{-9}	6.4×10^{-9}
4.5. k	9.9×10^{-7}	5.4×10^{-2}
7.5 k	6.4×10^{-9}	9.0×10^{-2}

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ความผิดพลาด (%)	
	$F_s = 25.0 \text{ MHz}$	$F_s = 5.0 \text{ MHz}$
8.6 k	1.8×10^{-4}	3.4×10^{-2}
20.0 k	2.4×10^{-4}	5.6×10^{-1}
45.3 k	1.6×10^{-7}	5.4×10^{-1}
75.3 k	6.4×10^{-6}	9.0×10^{-1}
152.4 k	1.6×10^{-5}	6.1×10^{-1}

จากข้อมูลในตาราง 5-2 สามารถนำมาเสนอในรูปของกราฟ ดังภาพประกอบ 5-3



ภาพประกอบ 5-3 ค่าร้อยละของความผิดพลาดจากวิธีการวัดความกว้างคาบ

จากตาราง 5-2 และภาพประกอบ 5-3 เป็นค่าร้อยละของความผิดพลาดจากการวัดอัตราการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ เมื่อความถี่สุ่มหาช่วงคาบสัญญาณเอนโค้ดเดอร์ที่ 25 MHz และ 5 MHz และกำหนดให้เอนโค้ดเดอร์ มีจำนวนพัลส์ต่อรอบ เท่ากับ 100 พัลส์ต่อรอบ จะเห็นว่าเมื่อค่าความถี่สัญญาณเอนโค้ดเดอร์เพิ่มขึ้น ค่าร้อยละของความผิดพลาดก็จะเพิ่มมากขึ้น ซึ่งตรงกับใน [1] [2] [3] และข้อมูลโดยทั่วไปของวิธีการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบที่กล่าวไว้ว่า วิธีการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบนั้นเหมาะสมกับงานที่มีความถี่ของสัญญาณเอนโค้ดเดอร์ต่ำและเห็นได้ชัดว่าค่าร้อยละของความผิดพลาดสามารถลดได้โดยการเพิ่มความถี่ของสัญญาณที่ใช้ในการสุ่ม

ดังนั้นจากตาราง 5-2 และภาพประกอบ 5-3 จะได้ว่าวงจรวัดอัตราเร็วการหมุนด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ให้ประสิทธิภาพดีเมื่อความถี่ของสัญญาณเอนโค้ดเดอร์มีค่าต่ำ และจะให้ประสิทธิภาพที่ดีมากขึ้นเมื่อใช้ความถี่สัญญาณที่ใช้ในการสุ่มสูงๆ เมื่อใช้ความถี่ในการสุ่มเท่ากับ 25 MHz มีค่าร้อยละของความผิดพลาดสูงสุดเพียง 1.6×10^{-2}

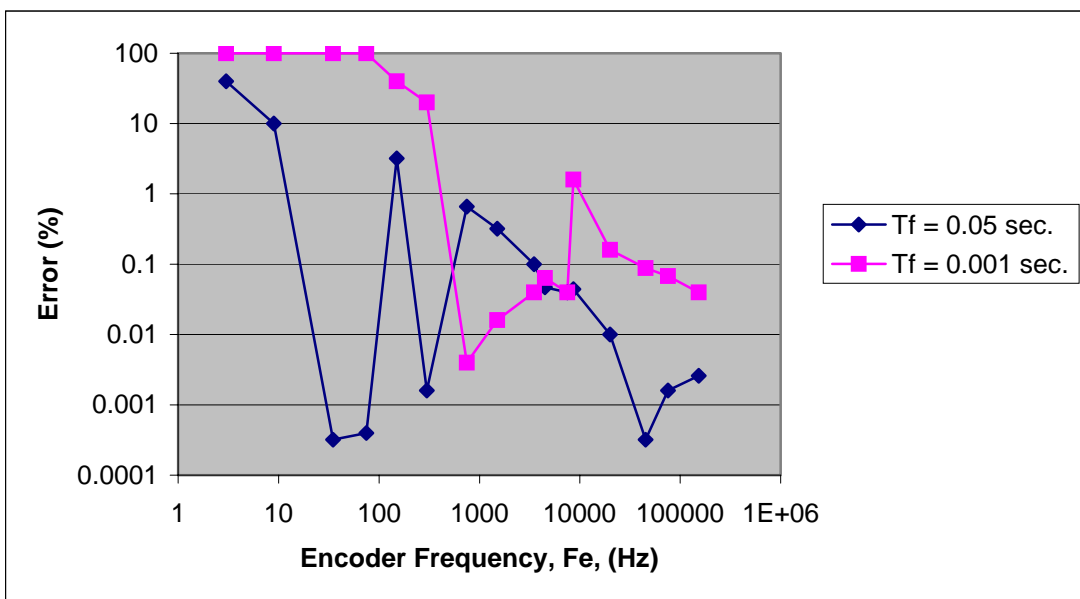
5.2.2 วิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

การทดสอบการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ ได้ทำการทดสอบโดยใช้ช่วงเวลากการวัด (T_f) เท่ากับ 0.05 วินาที และ 0.001 วินาที โดยทดสอบกับการป้อนสัญญาณเอนโค้ดเดอร์ที่ความถี่ต่ำ จนถึงความถี่สูง 150 kHz โดยกำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ ต่อรอบ และสัญญาณนาฬิกาที่ใช้ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz ซึ่งเมื่อนำค่าอัตราเร็วที่วัดได้มาคำนวณหาค่าร้อยละของความผิดพลาด ได้ดังตาราง 5-3

ตาราง 5-3 ค่าร้อยละของความผิดพลาดจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ความผิดพลาด (%)	
	$T_f = 0.05$ วินาที	$T_f = 0.001$ วินาที
3.0	4.0×10^1	1.0×10^2
9.0	1.0×10^1	1.0×10^2
34.9	3.2×10^{-4}	1.0×10^2
75.0	4.0×10^{-4}	1.0×10^2
150.0	3.2×10^0	4.0×10^1
300.0	1.6×10^{-3}	2.0×10^1
750.0	6.6×10^{-1}	4.0×10^{-3}
1.5 k	3.2×10^{-1}	1.6×10^{-2}
3.5 k	1.0×10^{-1}	4.0×10^{-2}
4.5 k	4.7×10^{-2}	6.4×10^{-2}
7.5 k	4.0×10^{-2}	4.0×10^{-2}
8.6 k	4.4×10^{-2}	1.6×10^0
20.0 k	1.0×10^{-2}	1.6×10^{-1}
45.3 k	3.2×10^{-4}	8.8×10^{-2}
75.3 k	1.6×10^{-3}	6.8×10^{-2}
152.4 k	2.6×10^{-3}	4.0×10^{-2}

จากข้อมูลในตาราง 5-3 สามารถนำมาเสนอในรูปของกราฟ ดังภาพประกอบ 5-4



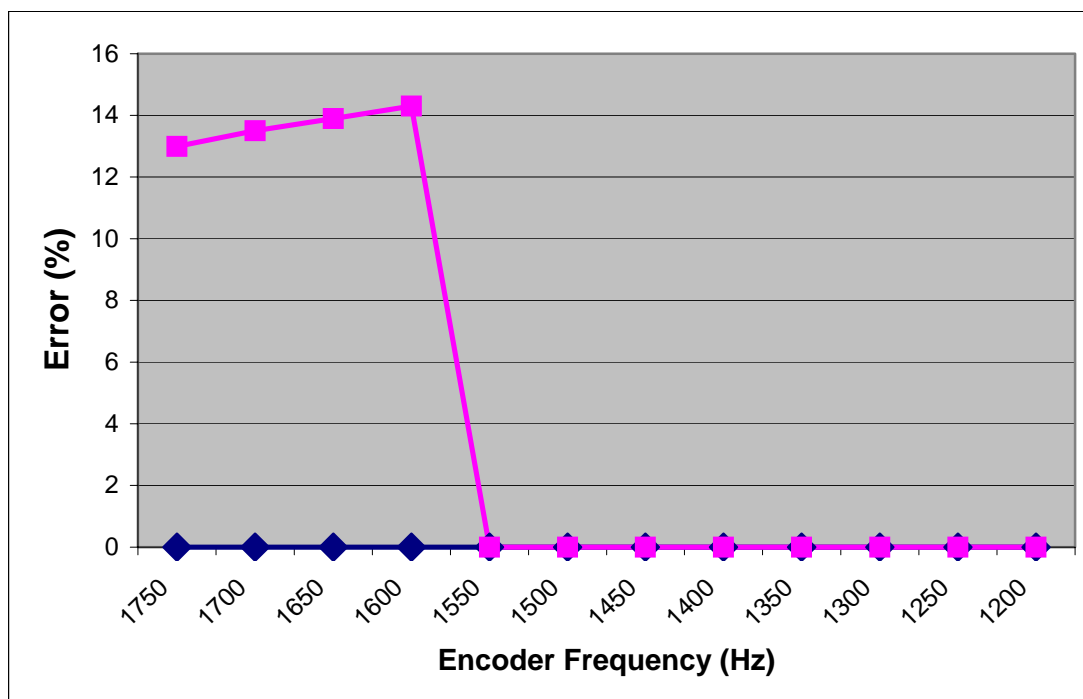
ภาพประกอบ 5-4 ค่าร้อยละของความผิดพลาดจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

จากตาราง 5-3 และภาพประกอบ 5-4 เป็นค่าร้อยละของความผิดพลาดจากการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ เมื่อสัญญาณนาฬิกาที่ใช้ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz ค่าช่วงเวลารับเท่ากับ 0.05 วินาที และ 0.001 วินาที และ กำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ จะเห็นได้ว่าเมื่อมีความถี่สัญญาณเอนโค้ดเดอร์เพิ่มขึ้น ค่าร้อยละของความผิดพลาดก็จะลดลง ซึ่งตรงกับใน [1] [2] [3] และข้อมูลโดยทั่วไปของวิธีการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ว่า วิธีการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์นั้นเหมาะสมกับงานที่มีความถี่ของสัญญาณเอนโค้ดเดอร์สูงและเห็นได้ว่าเมื่อกำหนดให้ช่วงเวลารับเท่ากับ 0.001 วินาที นั้นให้ผลค่าร้อยละของความผิดพลาดสูง และบางค่าก็สูงถึง 100 % เนื่องจากช่วงเวลาในการวัดนั้นสั้นกว่าความกว้างของสัญญาณ Quadrature แต่ก็ยังให้ประสิทธิภาพที่ดีที่ความถี่สัญญาณเอนโค้ดเดอร์สูง และเมื่อกำหนดช่วงเวลารับเท่ากับ 0.05 วินาที นั้นจะได้ค่าร้อยละของความผิดพลาดลดต่ำกว่าการกำหนดช่วงเวลารับเท่ากับ 0.001 วินาที

ดังนั้นจากตาราง 5-3 และภาพประกอบ 5-4 จะได้ว่าวงจรวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ให้ผลประสิทธิภาพดีเมื่อความถี่ของสัญญาณเอนโค้ดเดอร์มีค่าสูง และให้ประสิทธิภาพลดลงเมื่อมีความถี่สัญญาณเอนโค้ดเดอร์ต่ำ ๆ และการกำหนดค่าช่วงเวลารับให้มากขึ้นก็จะให้ผลการวัดที่ถูกต้องมากขึ้น ซึ่งค่าร้อยละของความผิดพลาดสูงสุดที่เกิดขึ้นไม่เกิน $6.4 \times 10^{-2}\%$ เมื่อกำหนดช่วงเวลารับเท่ากับ 0.001 วินาที เมื่อความถี่สัญญาณเอนโค้ดเดอร์มีค่าสูงกว่า 150 Hz และความถี่ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz

5.2.3 วิธีการกำหนดอัตราเร็วเกณฑ์

จากภาพประกอบ 5-3 และภาพประกอบ 5-4 จะได้ว่า ด้วยวิธีการวัดความกว้างให้ความถูกต้องสูงเมื่อความถี่สัญญาณเอนโค้ดเดอร์ต่ำ แต่วิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ให้ความถูกต้องสูงเมื่อความถี่สัญญาณเอนโค้ดเดอร์มีค่าสูง จึงนำกราฟทั้ง 2 มาหาจุดที่กราฟจากทั้ง 2 วิธีตัดกัน ซึ่งได้ดังภาพประกอบ 5-5



ภาพประกอบ 5-5 จุดตัดของกราฟ จากทั้ง 2 วิธี

จากภาพประกอบ 5-5 จะเห็นได้ว่าจุดตัดของกราฟอยู่ที่ความถี่สัญญาณเอนโค้ดเดอร์เท่ากับ 1550 Hz และจะใช้เกณฑ์นี้ในการเลือกแสดงผลค่าอัตราเร็วคือ ถ้าความถี่สัญญาณเอนโค้ดเดอร์มีค่าน้อยกว่า 1550 Hz จะใช้ผลค่าอัตราเร็วการหมุนจากวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ แต่ถ้าความถี่มากกว่า 1550 Hz ก็จะใช้ค่าอัตราเร็วการหมุนจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

เนื่องจาก ความถี่ของสัญญาณเอนโค้ดเดอร์มีค่าไม่คงที่ขึ้นกับอัตราเร็วการหมุนและจำนวนพัลส์ต่อรอบ เอฟพีจีเอจะไม่สามารถตรวจวัดได้ว่าความถี่สัญญาณเอนโค้ดเดอร์ขณะนั้นเป็นเท่าใด ดังนั้นจึงต้องคำนวณหาอัตราเร็วการหมุนเมื่อเอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบ เท่ากับ 100 และหมุนด้วยอัตราเร็ว 1550 Hz ซึ่งเท่ากับ 15.5 รอบต่อวินาที โดยวงจรจะทำการเปรียบเทียบค่าอัตราเร็วปัจจุบันของทั้ง 2 วิธีการวัดกับอัตราเร็วเกณฑ์ 15.5 รอบต่อวินาที ถ้าอัตราเร็วน้อยกว่า

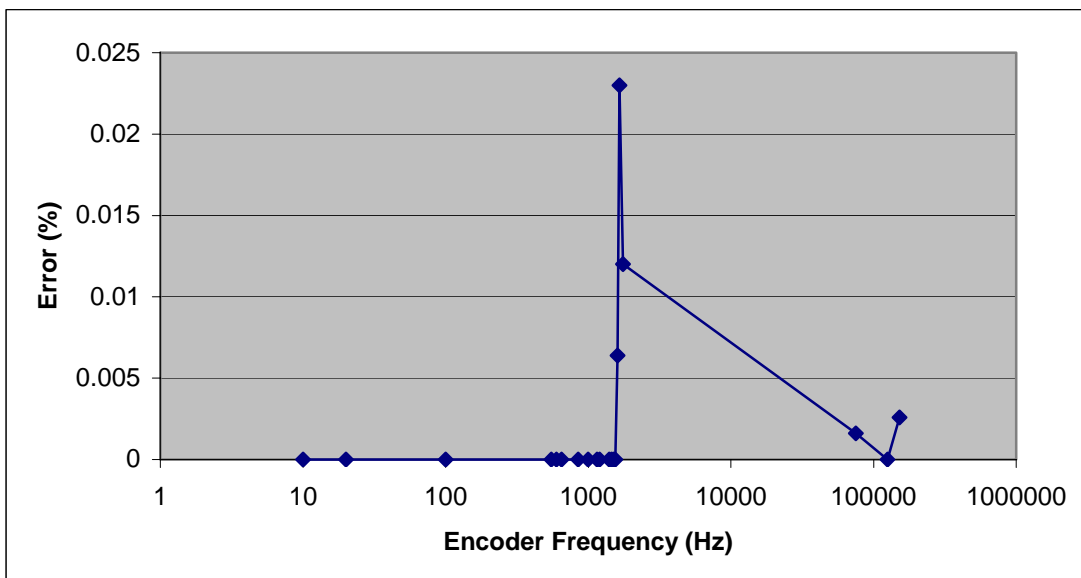
15.5 รอบต่อวินาที ก็จะแสดงค่าอัตราเร็วจากการวัดความกว้าง คาบสัญญาณเอนโค้ดเดอร์ แต่ถ้าอัตราเร็วมากกว่าอัตราเร็วเกณฑ์ ก็จะแสดงค่าอัตราเร็วจากวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์

ในการทดสอบนี้ กำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ และการวัดอัตราเร็วด้วยวิธีการวัดความกว้างสัญญาณเอนโค้ดเดอร์ มีความถี่สุ่มหาช่วงคาบสัญญาณเอนโค้ดเดอร์ที่ 25 MHz และการวัดอัตราเร็วด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ มีสัญญาณนาฬิกาที่ใช้ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz และค่าช่วงเวลาการนับเท่ากับ 0.05 วินาที ได้ผลการทดสอบดังตาราง 5-4

ตาราง 5-4 ค่าร้อยละของความผิดพลาดจากวิธีกำหนดค่าเกณฑ์

ความถี่สัญญาณจาก เอนโค้ดเดอร์ (Hz)	ความผิดพลาด (%)
10.0	0.0×10^0
20.0	0.0×10^0
100.0	0.0×10^0
549.8	1.9×10^{-7}
599.8	2.4×10^{-7}
649.8	2.9×10^{-7}
849.6	3.5×10^{-7}
999.4	6.6×10^{-8}
1150.2	2.1×10^{-6}
1200.1	4.1×10^{-7}
1400.1	4.1×10^{-7}
1450.1	6.4×10^{-7}
1500.2	2.6×10^{-6}
1550.1	4.1×10^{-7}
1600.1	6.4×10^{-3}
1650.4	2.3×10^{-2}
1750.2	1.2×10^{-2}
75.3 k	1.6×10^{-3}
125.0 k	1.8×10^{-14}
152.4 k	2.6×10^{-3}

จากข้อมูลในตาราง 5-4 สามารถนำมาเสนอในรูปแบบของกราฟ ดังภาพประกอบ 5-6



ภาพประกอบ 5-6 ผลจากการวัดเมื่อใช้ความถี่เอน โค้ดเดอร์เท่ากับ 1550 Hz เป็นค่าเกณฑ์

จากตาราง 5-4 และภาพประกอบ 5-6 เป็นผลค่าร้อยละของความผิดพลาดเมื่อค่าอัตราเร็วเกณฑ์เท่ากับ 15.5 รอบต่อวินาที หรือความถี่สัญญาณเอน โค้ดเดอร์เท่ากับ 1550 Hz โดยกำหนดให้เอน โค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ การวัดอัตราเร็วด้วยวิธีการวัดความกว้างคาบสัญญาณเอน โค้ดเดอร์ มีความถี่สุ่มหาช่วงคาบสัญญาณเอน โค้ดเดอร์ที่ 25 MHz และการวัดอัตราเร็วด้วยวิธีการวัดความถี่สัญญาณเอน โค้ดเดอร์ มีสัญญาณนาฬิกาที่ใช้ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz และค่าช่วงเวลากرنับเท่ากับ 0.05 วินาที จะได้ว่าค่าร้อยละของความผิดพลาดสูงสุดที่เกิดกับการวัดโดยวิธีนี้มีค่าไม่เกิน $4.4 \times 10^{-2}\%$

ในบทนี้ได้แสดงถึงผลการทดสอบวงจรเอฟพีจีเอที่ออกแบบมาเพื่อเชื่อมต่อกับเอน โค้ดเดอร์เพื่อใช้ในการวัดตำแหน่งและวัดอัตราเร็วการหมุน ซึ่งการทดสอบพบว่าวงจรสามารถวัดค่าต่างๆ ได้ผลลัพธ์เป็นที่น่าพอใจ มีค่าร้อยละของความผิดพลาดน้อย สามารถนำไปใช้งานได้จริง

บทที่ 6

บทสรุปและข้อเสนอแนะ

บทนี้จะกล่าวสรุปผลการวิจัยที่ได้ดำเนินการสำหรับงานวิจัยนี้ รวมทั้งข้อเสนอแนะต่างๆ ที่จะเป็นประโยชน์ต่อการทำวิจัยด้านการออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์

6.1 บทสรุป

งานวิจัยชิ้นนี้เป็นการนำเสนอการออกแบบวงจรเชื่อมต่อเอนโค้ดเดอร์แบบอัตราเร็วสูงและหลายช่องสัญญาณด้วยพีแอลดีแบบเอฟพีจีเอ โดยการออกแบบในวงจรมีความสามารถในการวัดตำแหน่งและวัดอัตราเร็วการหมุนของเอนโค้ดเดอร์ได้ 2 ช่องสัญญาณ

จากการทดสอบการวัดตำแหน่ง พบว่าวงจรสามารถวัดค่าตำแหน่งให้ความถูกต้องที่ความผิดพลาดของจำนวนจังหวะการเคลื่อนที่ไม่เกิน $\frac{1}{4}$ ของคาบสัญญาณเอนโค้ดเดอร์ เนื่องจากเหตุผลที่ก่อให้เกิดความคลาดเคลื่อนได้ดังที่อธิบายในหัวข้อ 5-1 ซึ่ง จากการทดสอบพบว่าค่าร้อยละของความผิดพลาดสูงสุดไม่เกิน 0.004%

จากการทดสอบการวัดอัตราเร็วการหมุน โดยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ พบว่าวงจรสามารถวัดอัตราเร็วการหมุนได้ประสิทธิภาพดีและเป็นที่น่าพอใจ ซึ่งจากการทดสอบพบค่าร้อยละของความผิดพลาดสูงสุดไม่เกิน $9.0 \times 10^{-1}\%$ ซึ่งเกิดขึ้นเมื่อใช้ความถี่ในการสุ่มตรวจขอบขาขึ้นของสัญญาณ Q เท่ากับ 5 MHz ซึ่งค่าร้อยละของความผิดพลาดนั้นสามารถลดลงได้โดยการใช้สัญญาณในการสุ่มตรวจขอบขาขึ้นของสัญญาณ Q จากการทดสอบพบว่าเมื่อเพิ่มความถี่สัญญาณนาฬิกาในการสุ่มเป็น 25 MHz ค่าร้อยละของความผิดพลาดลดลงมีค่าไม่เกิน $1.6 \times 10^{-2}\%$

จากการทดสอบการวัดอัตราเร็วการหมุน โดยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์พบว่าวงจรสามารถวัดอัตราเร็วการหมุนได้ประสิทธิภาพดีและเป็นที่น่าพอใจ ซึ่งจากการทดสอบแม้จะพบว่าค่าร้อยละของความผิดพลาดสูงสุดมีค่าถึง 100% เนื่องจากช่วงเวลาที่กำหนดไว้ในการนับนั้นสั้นเกินไปทำให้ไม่สามารถใช้วิธีการวัดอัตราเร็วการหมุนต่างๆ ได้ แต่ก็สามารถลดค่าร้อยละของความผิดพลาดนั้นได้โดยการเพิ่มช่วงเวลาในการนับให้มากขึ้น และพบว่าการวัดอัตราเร็วการหมุนด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์จะได้ประสิทธิภาพดี เมื่อกำหนดช่วงเวลาการนับเท่ากับ 0.001 วินาที เมื่อความถี่สัญญาณเอนโค้ดเดอร์มีค่าสูงกว่า 1550 Hz ให้ค่าร้อยละของความผิดพลาดสูงสุดไม่เกิน $6.4 \times 10^{-2}\%$

จากการทดสอบ เมื่อรวมทั้งสองวิธีการวัดเข้าด้วยกัน โดยวิธีการวัดความกว้างคาบจะให้ค่าความถูกต้องสูงเมื่อเอนโค้ดเดอร์หมุนที่อัตราเร็วต่ำ และโดยวิธีการวัดความกว้างคาบจะให้ค่าความถูกต้องสูงเมื่อเอนโค้ดเดอร์หมุนที่อัตราเร็วสูง เมื่อการทดสอบกำหนดให้เอนโค้ดเดอร์มีจำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ การวัดอัตราเร็วด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ มีความถี่สุ่มหาช่วงคาบสัญญาณเอนโค้ดเดอร์ที่ 25 MHz และการวัดอัตราเร็วด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์มีสัญญาณนาฬิกาที่ใช้ในการสุ่มตรวจสัญญาณ Q เท่ากับ 25 MHz และค่าช่วงเวลาการนับเท่ากับ 0.05 วินาที ได้ความถี่เอนโค้ดเดอร์ 1550 Hz เป็นอัตราเร็วเกณฑ์ ถ้าความถี่เอนโค้ดเดอร์มีค่าต่ำกว่า 1550 Hz ก็จะแสดงผลจากอัตราเร็วด้วยวิธีการวัดความกว้างคาบสัญญาณเอนโค้ดเดอร์ แต่ถ้าความถี่สูงกว่า 1550 Hz ก็จะแสดงผลจากอัตราเร็วด้วยวิธีการวัดความถี่สัญญาณเอนโค้ดเดอร์ และได้ค่าร้อยละของความผิดพลาดสูงสุดที่เกิดกับการวัดโดยวิธีการกำหนดอัตราเร็วเกณฑ์มีค่าไม่เกิน $4.4 \times 10^{-2} \%$

โดยสรุปแล้ว งานวิจัยชิ้นนี้บรรลุวัตถุประสงค์ตามเป้าหมายที่ได้กำหนดไว้ในข้อเสนอโครงการวิจัยคือ

- ได้วงจรเชื่อมต่อเอนโค้ดเดอร์บน FPGA ซึ่งสามารถควบคุมการทำงาน โดยการใช้ Control Register
- เนื่องจากความถี่สูงสุดที่วงจร Frequency Meter สามารถทำงานได้อยู่ที่ 127.57 MHz ดังนั้นจึงสามารถรองรับการเชื่อมต่อเอนโค้ดเดอร์ที่มีความละเอียด 30,000 พัลส์ต่อรอบที่ความเร็ว 10-10,000 รอบต่อวินาทีได้อย่างไม่มีปัญหา
- มีเอาต์พุตแบบ 3 สถานะซึ่งสามารถเชื่อมต่อกับบัสข้อมูลของไมโครโพรเซสเซอร์ได้
- สามารถบอกค่าความกว้างของพัลส์ได้
- ตอบสนองการแซมปลิงแบบ Quadrature
- สามารถเลือกโหมดความละเอียดของเอาต์พุตได้ 3 รูปแบบ คือ 8 บิต 16 บิต และ 32 บิต โดยตัดโหมดความละเอียดขนาด 24 บิตออกไป

6.2 ข้อเสนอแนะ

1. งานวิจัยนี้ได้ออกแบบวงจรสำหรับเชื่อมต่อเอนโค้ดเดอร์ 2 ช่องสัญญาณ แต่วงจรที่ใช้ในการวัดของแต่ละช่องสัญญาณแยกจากกัน ไม่ได้ออกแบบให้เป็นวงจรที่มีวงจรวัดชุดเดียวกันแต่สามารถจัดการข้อมูลได้ทั้ง 2 ช่องสัญญาณ ซึ่งทั้งสองแบบจะมีข้อดีข้อเสียต่างกัน ถ้าออกแบบให้มีวงจรวัดชุดเดียวได้นั้นจะทำให้ประหยัดทรัพยากรของเอฟพีจีเอ ทำให้สามารถเลือกใช้เอฟพีจีเอรุ่นที่มีจำนวนเกตน้อยกว่าได้ แต่การออกแบบให้วงจรวัดแยกจากกันนั้น ทำได้ง่ายกว่าและสามารถจัดการกับสัญญาณเอนโค้ดเดอร์ที่เข้ามาได้รวดเร็วกว่า ดังนั้นหากจะมีการนำงานวิจัยนี้ไปพัฒนาเพื่อปรับปรุงให้มีประสิทธิภาพเพิ่มขึ้น คณะผู้วิจัยขอเสนอให้ทำการหาวิธีปรับปรุงสถาปัตยกรรม

ภายในของวงจรเชื่อมต่อเอนโคคเคอร์เพื่อให้สามารถใช้งานวัตต์ของทั้งสองช่องสัญญาณร่วมกัน อันจะส่งผลให้ลดการใช้จำนวนทรานซิสเตอร์ในเอฟพีจีเอลงแต่ยังคงไว้ซึ่งประสิทธิภาพด้านความเร็วในการจัดการกับสัญญาณได้ดีเทียบเท่ากับการออกแบบวงจรการวัดของแต่ละช่องสัญญาณแยกจากกัน

2. งานวิจัยนี้ ได้ออกแบบวงจรให้เชื่อมต่อกับเอนโคคเคอร์ได้เพียง 2 ช่องสัญญาณเท่านั้น เนื่องจากข้อจำกัดทางทรานซิสเตอร์ของเอฟพีจีเอที่ใช้ในการทดลองที่มีจำกัด หากต้องการขยายจำนวนช่องสัญญาณให้มากขึ้นก็สามารถทำได้ด้วยการเปลี่ยนไปใช้เอฟพีจีเอรุ่นที่มีจำนวนเกตมากขึ้นและใช้วิธีการทำซ้ำวงจรซึ่งได้นำเสนอในงานวิจัยนี้ให้มีมากกว่า 1 ชุดลงในเอฟพีจีเอเพื่อเพิ่มจำนวนช่องสัญญาณให้ได้จำนวนตามที่คุณใช้งานต้องการ

3. การวิจัยนี้ ได้ออกแบบส่วนของการติดต่อกับผู้ใช้เป็นแบบใช้สายสัญญาณขนาด 8 หรือ 16 หรือ 32 สายสัญญาณ ซึ่งอาจไม่สะดวกต่อการใช้งานกับไมโครคอนโทรลเลอร์ขนาดเล็ก ถ้าออกแบบวงจรให้มีส่วนติดต่อกับผู้ใช้เป็นแบบอนุกรม เช่นการเชื่อมต่ออนุกรมตามมาตรฐาน RS-232 หรือ USB จะทำให้ลดจำนวนสายสัญญาณลงได้มาก ดังนั้นหากจะมีการนำงานวิจัยนี้ไปพัฒนาเพื่อปรับปรุงให้มีประสิทธิภาพเพิ่มขึ้น คณะผู้วิจัยขอเสนอให้เพิ่มวงจรอินเตอร์เฟซแบบอนุกรมเพิ่มเติม

เอกสารอ้างอิง

- [1] E.Sisinni, A. Flammini, D. Marioli and A. Taroni, “A PLD Based Encoder Interface with Accurate Position and Velocity Estimation,” in Proc. ISIE 2002. Industrial Electronics., vol. 2, Jul. 2002, pp. 606-611.
- [2] B. Pamela and H. Blake, “Single-Chip Velocity Measurement System for Incremental Optical Encoders,” IEEE Trans. Control Systems Technology, vol.5, no 6, Nov. 1997 , pp.654-661.
- [3] F.Briz, J.A. Cancelas and A. Diez, “Speed Measurement Using Rotary Encoders for High performance ac Drives,” IECON '94.Int. Con. Control. Instrumentation, vol. 1 , Sep. 1994, pp. 538-548.
- [4] ณรงค์ ทองนิม และ เจริญ วงษ์ชุ่มเย็น, 2547, “การออกแบบวงจรรวมดิจิทัล โดยใช้ EPGA และ CPLD”, <http://www.aiogictechnoloty.com>. Apex Instrument Co.,Ltd
- [5] ชำนาญ ปัญญาใส และ วัชรกร หนูทอง, 2547, “ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล”, กรุงเทพฯ, ซีเอ็ดดูเคชั่น.
- [6] B. Stephen, V. Zvonko, 2005, “Fundamentals of Digital Logic with VHDL Design”,Singapore, McGraw-Hill.
- [7] K. William. 2004, “Digital Electronics with VHDL”, USA, Pearson-Printice Hall.
- [8] S.Stefan, L Lennart, 1997, “VHDL for Designers”, Great Britain, Printice Hall.
- [9] ชีรยศ เวียงทอง, 2548 เรียนรู้การออกแบบระบบดิจิทัลด้วยภาษา Verilog เบื้องต้น, http://www.ee.mut.ac.th/fpga/HomeFile/Verilog_Thai.pdf, 15 มิ.ย. 2550.
- [10] T. Stöcklein, J. Bäsig, Handel-C an effective method for designing FPGAs (and ASICs), <http://www.celoxica.co.jp/techlib/files/CEL-W030717HTM-16.pdf>, June. 15, 2007.
- [11] Hewlett Packard Application Brief M-109, “Incremental Encoder Errors: Causes and Methods to Reduce Them”, <http://www.hpl.hp.com>, Dec. 28 , 2004.
- [12] J. Detrey, F.D. Dinechin,” A VHDL Library of Parametrisable Floating-Point and LNS Operators for FPGA”, <http://perso.ens-lyon.fr/jeremie.detrey/FPLibrary>, May. 25, 2006

- [13] IOBUF, http://toolbox.xilinx.com/docsan/xilinx5/data/docs/lib/lib0256_240.html,
Sep. 29, 2005.
- [14] IOBUFDS, http://toolbox.xilinx.com/docsan/xilinx5/data/docs/lib/lib0257_241.html,
Sep. 29, 2005.
- [15] Incremental Encoder, <http://www.servotek.com/page6.htm>, Jan. 7, 2006.

ภาคผนวก ก

ผลงานวิชาการที่ได้รับการตีพิมพ์

1. Korakot Suwannarat, MontriKarnjanadecha, Thanate Khaorapapong and Panyayot Chaikan, "A Multi-channel Incremental Encoder Interfacing Circuit Design using FPGA," in Proc. ECTI-CON 2006, May. 2006, pp. 307-311.
2. กรกต สุวรรณรัตน์, มนต์รี กาญจนเดชะ, ธเนศ เถาว์พวงศ์ และ ปัญญยศ ไชยกาพ, "การประมาณค่าความเร็วตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่าด้วยวิธีวัดความกว้างเต็มคาบโดยใช้เอฟพีจีเอ," การประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษา มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 4, 31 พฤษภาคม 2549, หน้า 60

A Multi-Channel Incremental Encoder Interfacing Circuit Design using FPGA

Korakot Suwannarat¹, Montri Karnjanadecha²,
Thanate Khaorapapong³, Panyayot Chaikan⁴

Department of Computer Engineering, Prince of Songkla University,
Hat Yai, Songkhla 90112 THAILAND

Email: korakots@hotmail.com¹, montri@coe.psu.ac.th²,
kthanate@coe.psu.ac.th³, panyayot@fivedots.coe.psu.ac.th⁴

Abstract

The use of an incremental encoder as a speed sensor in a motion control system can consume a large amount of microprocessor time, especially when a high pulse-per-revolution encoder is needed. This paper proposes an FPGA design to interface with the encoder, which reduces the microprocessor's workload. Our implementation is based on the full period meter and the frequency meter methods for velocity estimation. Test results show that our FPGA works correctly.

Keywords: FPGA, incremental encoder interfacing.

1. Introduction

Position and velocity estimations are important for the feedback control of robotic systems. Incremental encoders are employed in many applications due to their reliability, availability, and insensitivity to external disturbance. The encoder signals are pulses with a period proportional to the encoder's rotation speed, and position data can be estimated by counting the number of pulses. An incremental encoder's output signals consist of two square waves, with a 90° electrical gap (quadrature), as shown in Figure 1.

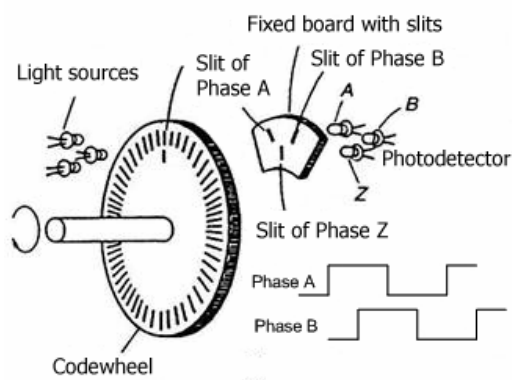


Figure 1. The basic components of an incremental encoder.

The basic components of an incremental encoder are a light source, a codewheel or codestrip, and photodetectors [4]. As the codewheel rotates, the light from the source travels through the codewheel's slits to the photodetector. An alternating pattern of light and dark is seen by the photodetector. Circuitry attached to the photodetector turns the light/dark pattern into electrical output signals. The number of codewheel slits determines the number of pulses per revolution (PPR) for the encoder.

The frequency meter and the period meter methods are two well-known approaches for estimating the velocity of an incremental encoder. The frequency meter counts the number of pulses generated by an encoder over a period of time, yielding high accuracy when the encoder pulses are high frequency. However, when the encoder is rotating slowly, the number of counted pulses may not accurately represent the rotation speed. In that case, the period meter method can be utilized. It requires a high frequency clock signal since it measures rotation velocity by counting the number of clock ticks during the duration of the encoder pulse.

A high PPR encoder is usually needed to build an accurate system, but care must be taken with the encoder's interface circuitry. For systems where simplicity and low-cost are the main concerns, the encoder is directly connected to the microprocessor. Hence, the microprocessor will have to perform the pulse counting and speed calculations. For a high PPR encoder, the microprocessor will have to spend much of its time on speed and/or position estimation, and the danger is that if the microprocessor lacks the necessary performance, then the estimated speed and position will be incorrect. One solution is to design an encoder interfacing circuit which reduces the microprocessor workload by having the encoder take care of the position and speed calculations itself.

This paper presents the design of a circuit employing a field programmable gate array (FPGA) which estimates the position and velocity of an incremental encoder using VHDL [8, 10, 13].

2. Velocity Estimation

The period meter method is more accurate than the frequency meter method when producing velocity estimates for an incremental encoder [1, 2, 3, 4]. The period meter method counts the ticks generated by a high frequency clock (F_s) on the time interval between two adjacent rising edges in the encoder's output pulses. This is equivalent to measuring the encoder's pulse period. Figure 2 illustrates how this method works and some notation employed in the speed calculation.

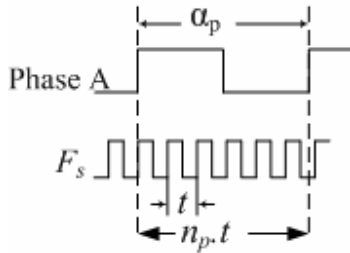


Figure 2. A full period meter.

In Figure 2, α_p is the angle that the encoder's shaft has to rotate to span one full period of the encoder's output signal. $t = \frac{1}{F_s}$ is the period of the high clock frequency, and n_p is the number of pulses occurring during the angle α_p . The period of the encoder signal, T_e , is determined by:

$$T_e = n_p \cdot t \quad (1)$$

The estimated velocity (ω_p) is:

$$\omega_p = \frac{\alpha_p}{T_e} \quad (2)$$

Note that $\alpha_p = \frac{1}{PPR}$, where PPR is the number of pulses per revolution of the encoder. It follows that

$$\omega_p = \frac{F_s}{PPR \cdot n_p} \quad (3)$$

For a particular system, F_s and PPR are known and fixed, while n_p depends on the encoder speed. High frequency clock results in a high n_p value, and accurate speed estimates. However, a compromise has to be reached between high accuracy and high clock

frequency, since it is preferable to keep the F_s value as low as possible.

Most period meter method implementations only use the "high" part or "low" parts of the encoder signal in order to reduce processing time. However, encoder signals are not always symmetry (50% duty cycle) [6, 11], and duty cycle deviations can reach as much as 25% [11, 12]. To reduce the estimation error, our period meter method implementation utilizes the full period of the encoder signal.

The frequency meter method offers higher accuracy, than the period meter method when the encoder frequency is high [1, 2, 3, 4]. The frequency approach counts the number of quadrature pulses over a fixed duration (T_f), as shown in Figure 3 [4]. The generation of the quadrature signal employs the Phase A and Phase B signals from the encoder, and the resulting signal has a four times higher frequency than the Phase A or Phase B signals. For example, a quadrature signal obtained from a 100 PPR encoder is equivalent to 400 pulses per revolution.

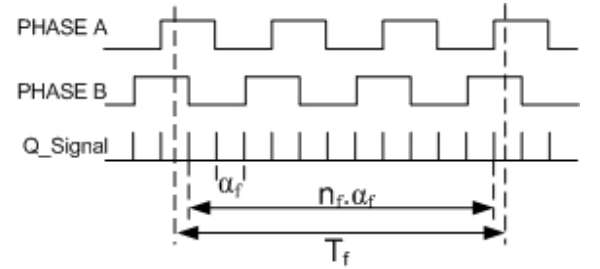


Figure 3. The output signals of an incremental encoder, and the resulting quadrature signal.

In Figure 3, α_f is a quarter of the encoder pulse angle, T_f is the duration for counting the number of pulses from the quadrature signal, and n_f is the number of counted pulses. The rotated angle, α_{ef} , over the duration T_f is:

$$\alpha_{ef} = \frac{n_f}{4 \cdot PPR} \quad (4)$$

The estimated velocity, ω_f , obtained with the frequency meter method is:

$$\omega_f = \frac{\alpha_{ef}}{T_f} \quad (5)$$

In equation (5), α_{ef} can be replaced by $\frac{n_f}{4 \cdot PPR}$, and T_{fi} by $\frac{1}{T_f}$ when $T_f < 1$ second. This gives:

$$\omega_f = \frac{n_f T_{fi}}{4.PPR} \quad (6)$$

In equation (6), T_{fi} and PPR are fixed, and n_f depends on the encoder speed. Large T_{fi} values result in large n_f 's and higher accuracy.

3. FPGA design

The encoder interfacing circuit was designed in VHDL [8, 10, 13], and Xilinx ISE 7.1i was used for its synthesis, placement and routing. The design was implemented on Placement and routing. The design was implemented on a SPARTAN-XC3S200-TQ144-4C, running at 25MHz [5].

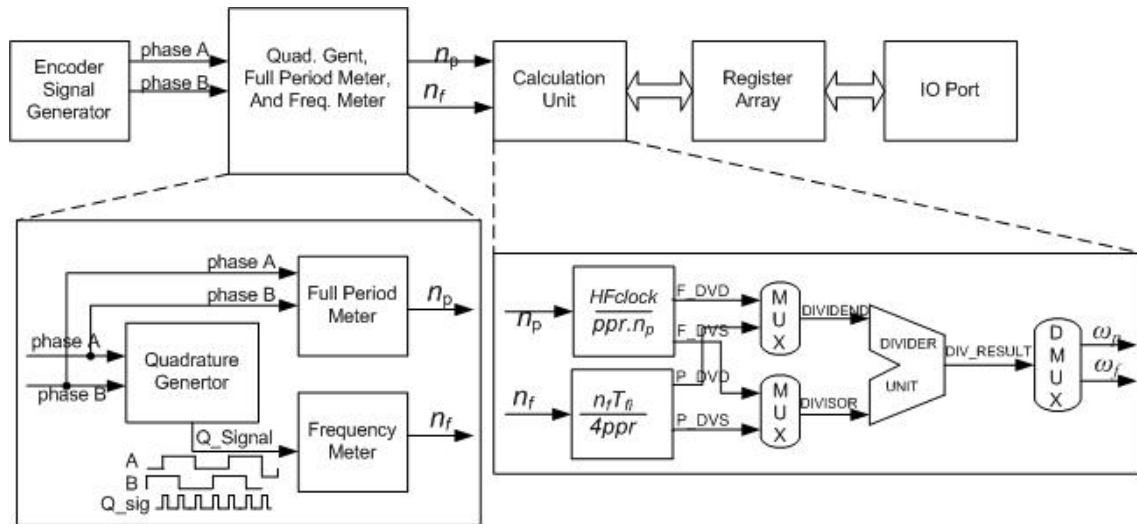


Figure 4. Block diagram of the circuit design.

The circuit has five main components, as shown in Figure 4. The first is the encoder signal generator, which produces 3-phase encoder pulses (Phase A, Phase B, and Phase Z). The second part consists of a quadrature generator, a full period meter, and a frequency meter. The third component estimates velocities using the n values output from the second part. The fourth main unit is a register array, which stores data such as the velocity, and pulses per revolution. The fifth part is a 32-bit I/O port.

The encoder signal generator produces pulses at several different frequencies, and different pulses per revolution. There are three type of signal, Phase A, Phase B and Phase Z, and the frequencies can vary between 5Hz – 152kHz. The circuit is mainly used for testing purposes.

In the second part of Figure 4, the quadrature generator produces the Q_Signal, that is shown in Figure 3, with its inputs coming from the encoder signal generator. The rising and falling edges of Phase A and Phase B are detected, triggering a Q_Signal pulse for each edge signal. The outputs of the quadrature generator become the inputs to the frequency meter module. It detects the rising edge of the Q_Signal, and counts the number of Q_Signal pulses. These are passed as the n_p and n_{pf} parameter to the third part. The full period meter module detects the rising edge of the encoder signals (Phase A or Phase B), and counts the number of F_s to n_p . When the module detects the next rising edge, it

updates the n_p value. The number of counted pulses, n_p and n_p are sent from the second part to the calculation module.

The calculation module consists of a divider unit, multiplexers and a control unit (see Figure 4). Since the divider unit requires a lot of FPGA resources, our design uses one divider unit for each encoder.

The register array (component 4 in Figure 4) stores data coming from other modules and the I/O ports.

The 32-bit I/O port is tri-state and bi-directional, and can interface with 8-, 16-, or 32-bit microprocessors. Our design employs IOBUFDS [7], which is a component in SPARTAN 3.

4. Results

During the design phase, our VHDL code was tested and verified using ModelSim software [9]. Final design was synthesized and then programmed onto the FPGA. The encoder signal generator was programmed to produce encoder signals at different frequencies. This is a very convenient way to test the FPGA, without the used of the real encoders. Outputs were directly observed from the register array, using a set of LEDs and some extra VHDL code and external circuitry.

Test results show that, the FPGA produces no velocities estimation error when the ratio between the encoder frequency, F_e , and the sampling frequency, F_s , in the period meter method, is an integer, and when the ratio between the duration T_f and the period of the Q_signal, in the frequency meter method, is an integer. This is a confirmation that the design is correct.

To further test our FPGA design in period meter mode, we programmed the encoder signal generator to produce 16 different encoder signals ranging from low to high frequencies. In each case, the sampling frequency was chosen such that the ratio between F_s and F_{en} is non-integer. Table 1 shows all the used frequencies and resulting estimation error.

Table 1. Full period meter.

Encoder Frequency, F_{en} , (Hz)	Error for period meter at 100 PPR.	
	$F_s = 25.0\text{MHz}$	$F_s = 5.0\text{MHz}$
3.0	2.5100E-12	2.4000E-05
9.0	1.1111E-05	4.7111E-05
34.9	1.0240E-09	2.8000E-04

75.0	1.5999E-09	8.9999E-04
150.0	2.5600E-08	5.9998E-04
300.0	2.5600E-08	3.5999E-03
750.0	6.3118E-12	8.9996E-03
1.5k	1.5843E-02	6.0007E-03
3.5k	6.4001E-09	6.3999E-09
4.5k	9.8560E-07	5.4004E-02
7.5k	6.3998E-09	9.0339E-02
8.6k	1.7906E-04	3.4251E-02
20.0k	2.4039E-04	5.5801E-01
45.3k	1.6000E-07	5.4054E-01
75.3k	6.4000E-06	8.9552E-01
152.4k	1.6000E-05	6.0604E-01

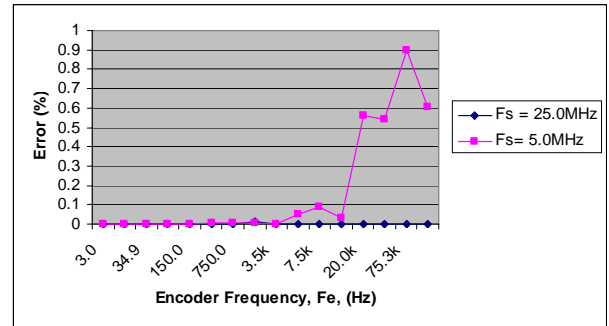


Figure 5. Error for the full period meter.

Figure 5 shows a plot of errors versus encoder frequencies. It is clearly seen from Figure 5 that the period meter method yields high error at high encoder speed for a given F_s . In addition, increasing F_s can reduce the error.

Theoretically, the frequency meter method yields high accuracy when the encoder frequency is high and the duration T_f is large. We confirm this by testing our FPGA at various encoder frequencies ($T_f = 0.05$ second and $T_f = 0.001$ second). The results are shown in Table 2 and Figure 6.

Table 2. Error for the frequency meter.

Encoder Frequency, F_{en} , (Hz)	Error for frequency meter (%).	
	$T_f = 0.05$ sec.	$T_f = 0.001$ sec.
3.0	4.0000E+01	1.0000E+02
9.0	1.0000E+01	1.0000E+02
34.9	3.2000E-04	1.0000E+02

75.0	4.0000E-04	1.0000E+02
150.0	3.2243E+00	3.9999E+01
300.0	1.6000E-03	2.0002E+01
750.0	6.5828E-01	4.0002E-03
1.5k	3.1628E-01	1.6003E-02
3.5k	1.0269E-01	4.0016E-02
4.5k	4.7019E-02	6.4040E-02
7.5k	4.0016E-02	4.0016E-02
8.6k	4.4141E-02	1.6137E+00
20.0k	1.0483E-02	1.6050E-01
45.3k	3.2016E-04	8.8077E-02
75.3k	1.5936E-03	6.8040E-02
152.4k	2.6240E-03	4.0000E-02

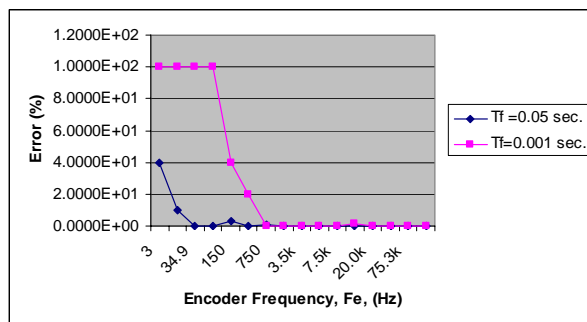


Figure 6. Error for the frequency meter at 100 PPR.

5. Conclusions

This paper presents the design and implementation of an interfacing circuit for an incremental encoder using FPGA. The designed FPGA was tested and performed correctly. The experiment results agree with the fact that the period meter method performs well at slow encoder speed and the frequency meter method performs well at fast encoder speed. Our future work will be to determine an optimum criterion for combining the two methods.

6. Acknowledgement

The authors would like to thank Dr. Andrew Davison for correcting English grammar and improving the readability of this manuscript.

7. References

- [1] B. Pamela and H. Blake, "Single-Chip Velocity Measurement System for Incremental Optical Encoders," *IEEE Trans. Control Systems Technology*, vol. 5, no 6, Nov. 1997, pp. 654-661.
- [2] D'Arco, S., Piegari, L., Rizzo, R. "Experimental validation of a velocity estimation algorithm for speed

control of electrical drives". *ISIE '03*. vol. 1, June 2003, pp. 318 – 322.

[3] E. Sisinni, A. Flammini, D. Marioli and A. Taroni, "A PLD Based Encoder Interface with Accurate Position and Velocity Estimation," in *Proc. ISIE 2002. Industrial Electronics.*, vol. 2, Jul. 2002, pp. 606 – 611.

[4] F. Briz, J. A. Cancelas and A. Diez, "Speed Measurement Using Rotary Encoders for High performance ac Drives," *IECON '94. Int. Con. Control. Instrumentation*, vol. 1, Sep. 1994, pp. 538-548.

[5] FPGA Discovery-III XC3S200 Board Manual, <http://www.aiogictechnology.com>, Date accessed: 30 March, 2006.

[6] Hewlett-Package. "Incremental Encoder Errors: Causes and Methods to Reduce Them". *Hewlett Packard Application Brief M-109*. <http://www.hpl.hp.com>, Date accessed: 28 December 2004

[7] IOBUFDS, http://toolbox.xilinx.com/docsan-xilinx7/de/libs/spartan3e_sch/iobufds.pdf, Date accessed: 30 March, 2006.

[8] K. William, *Digital electronics with VHDL*, Pearson Prentice Hall, United State, 2004.

[9] ModelSim, <http://www.model.com>, Date accessed: 30 March, 2006.

[10] S. Brown, Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, McGraw-Hill, Singapore , 2005.

[11] SERVO-TEK PRODUCTS, Incremental Encoder. <http://www.servotek.com/PAGE6.HTM>, Date accessed: 7 January 2006.

[12] Shaft Encoder, <http://www.nemicon.co.jp/en/product/category/shaft/2.html>, Date accessed: 1 July, 2003.

[13] S. Sjöholm, L. Lindth, *VHDL for Designers*, Prentice Hall, Great Britain , 1997.

การประมาณค่าความเร็วตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่าด้วยวิธีวัดความกว้างเต็มคาบโดยใช้เอฟพีจีเอ
Rotary Incremental Encoder Velocity Estimation with Full Period Meter Using FPGA

กรกต สุวรรณรัตน์¹ มนตรี กาญจนเดชะ² ธเนศ เคารพพงศ์³ ปัญญาศ ไชยกาฬ⁴
 ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

อ.หาดใหญ่ จ.สงขลา 90112

Email: korakots@hotmail.com¹, montri@coe.psu.ac.th²,

kthanate@coe.psu.ac.th³, panyayot@fivedots.coe.psu.ac.th⁴

บทคัดย่อ

บทความนี้เสนอการวิธีการวัดความเร็วการหมุนของตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่า (Rotary Incremental Encoder) ซึ่งให้สัญญาณเอาต์พุตเป็นสัญญาณพัลส์ โดยวิธีการวัดความกว้างเต็มคาบสัญญาณพัลส์จากตัวเข้ารหัส (Full Period Meter) ซึ่งเหมาะสมกับการวัดเมื่อตัวเข้ารหัสหมุนด้วยความเร็วต่ำ เพื่อให้ได้ความถูกต้องมากกว่าโดยวิธีการวัดความครึ่งคาบสัญญาณพัลส์จากตัวเข้ารหัส (Half Period Meter) ที่นิยมใช้กันมาก เนื่องจากการศึกษาพบว่าสัญญาณพัลส์จากตัวเข้ารหัสมีรอบการทำงาน (Duty cycle) ไม่เป็น 50% โดยออกแบบวงจรวัดความกว้างเต็มคาบด้วยภาษาบรรยายวงจรฮาร์ดแวร์ (VHDL) และโปรแกรมวงจรถนเอฟพีจีเอ (FPGA) การทดลองที่นำเสนอในบทความนี้เป็นการวัดความกว้างเต็มคาบ โดยมีสัญญาณจากตัวเข้ารหัส (Encoder Frequency) ที่มีความถี่ตั้งแต่ 5Hz -150kHz และใช้ความถี่ในการสุ่มวัดค่าความกว้างคาบ (Sampling Frequency) ที่ 5MHz, 6.25MHz, 8.33MHz, 12.5MHz และ 25MHz และได้ผลลัพธ์ของความผิดพลาดน้อยกว่า 1 %

คำหลัก: ตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่า, วิธีการวัดความกว้างเต็มคาบ, ภาษาบรรยายวงจรฮาร์ดแวร์, เอฟพีจีเอ

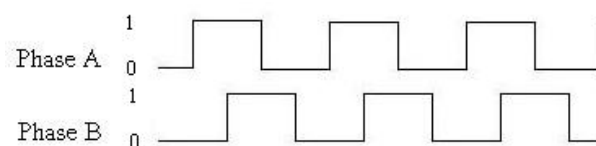
Abstract

This paper presents a full period meter method to estimate the velocity of rotary incremental encoders. This method gives a higher accuracy than the half period meter method, because encoder pulse signals are not 50% duty cycle. The circuit has been designed in VHDL and implemented on an FPGA. The system is tested with several encoder signal frequency ranging from 5Hz to 150kHz and sampling frequency ranging from 5MHz to 25MHz. The proposed method yields less than 1% of error.

Keyword: Rotary incremental encoder, Full period meter method, VHDL, FPGA

1. บทนำ

การควบคุมความเร็วการหมุน ต้องใช้ตัวรับรู้ (Sensor) มาตรวจวัดความเร็วขณะนั้นๆ เพื่อนำไปควบคุมโดยการเพิ่มหรือลดความเร็วให้ได้ตามที่ต้องการ ตัวตรวจรู้ที่นิยมใช้คือตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่า (Rotary Incremental Encoder) ซึ่งให้สัญญาณเอาต์พุต 2 เฟสที่ต่างกัน 90 องศา (กิติพงษ์ และ ปิณชกร, 2546; B. Pamela and H. Blake, 1997; E. Sisinni, A. Flammini, D. Marioli and A. Taroni, 2002) ดังรูปที่ 1 การวัดความเร็วการหมุนโดยใช้ตัวเข้ารหัสนี้ วิธีที่นิยมใช้มี 2 วิธีคือวิธีวัดความกว้างคาบสัญญาณพัลส์จากตัวเข้ารหัส ซึ่งเหมาะสมกับการวัดเมื่อความเร็วการหมุนต่ำ และวิธีวัดความถี่สัญญาณพัลส์จากตัวเข้ารหัสซึ่งเหมาะสมกับการวัดเมื่อความเร็วการหมุนสูง(F. Briz, J. A. Cancelas and A. Diez,1994; B. Pamela and H. Blake1997; E. Sisinni, A. Flammini, D. Marioli and A. Taroni, 2002)



รูปที่ 1 เอาต์พุต 2 เฟส ที่ต่างกัน 90 องศา

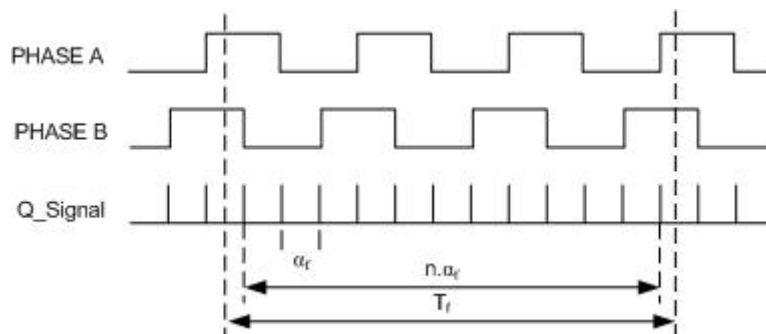
วิธีการวัดความกว้างคาบนั้น โดยทั่วไปจะเหมาะสมกับการวัดความเร็วการหมุนต่ำ เมื่อความเร็วการหมุนต่ำจะทำให้ความกว้างพัลส์สัญญาณจากตัวเข้ารหัสมีความกว้างมากขึ้น ซึ่งจะเสียเวลาในการวัดมาก จึงนิยมใช้วิธีการวัดความกว้างครึ่งคาบสัญญาณพัลส์แทน แต่จากการศึกษาค้นคว้าพบว่าพัลส์สัญญาณจากตัวเข้ารหัสแต่ละพัลส์มีรอบการทำงาน (Duty Cycle) ไม่เท่ากับ 50% (Hewlett-Package, 2004; SERVO-TEK PRODUCTS COMPANY, 2006) ทั้งเกิดจากความไม่แม่นยำของตัวเข้ารหัส หรือการจ่ายกระแสเลี้ยงวงจรของตัวเข้ารหัสไม่เพียงพอ ดังนั้นการวัดความเร็วการหมุนโดยวิธีการวัดความกว้างครึ่งคาบพัลส์สัญญาณจากตัวเข้ารหัส จะไม่ได้ค่าที่ถูกต้อง

สำหรับบทความนี้ได้นำเสนอการออกแบบวงจรวัดความเร็วการหมุนด้วยวิธีการวัดความกว้างเต็มคาบพัลส์สัญญาณจากตัวเข้ารหัสด้วยเอฟพีจีเอ โดยใช้ความถี่สุ่มวัดสัญญาณจากตัวเข้ารหัสที่ 5MHz - 25MHz

2. การวัดความกว้างเต็มคาบและการออกแบบวงจร

การวัดความเร็วการหมุนโดยใช้ตัวเข้ารหัสแบบหมุนชนิดเพิ่มค่า วิธีการวัดความกว้างเต็มคาบ (Period Meter) เป็นวิธีการวัดที่เหมาะสมกับสภาวะการหมุนที่ความเร็วต่ำของตัวเข้ารหัส ส่วนวิธีการวัดความถี่สัญญาณพัลส์จากตัวเข้ารหัส (Frequency Meter) เป็นวิธีการวัดที่เหมาะสมกับสภาวะการหมุนที่ความเร็วสูงของตัวเข้ารหัส

การวัดความเร็วการหมุนโดยวิธีการวัดความถี่สัญญาณพัลส์จากตัวเข้ารหัสเป็นนับจำนวนพัลส์ของสัญญาณจากตัวเข้ารหัสในช่วงเวลา T_f ที่กำหนด ตามรูปที่ 2 เพื่อให้วัดความเร็วของตัวเข้ารหัสได้ละเอียดมากขึ้นจึงใช้วิธีการแยกความละเอียดของสัญญาณ 2 เฟสที่ต่างกัน 90 องศา ดังนั้น 1 ช่วงพัลส์จะแบ่งได้ถึง 4 ส่วน เช่นตัวเข้ารหัสที่ให้จำนวนพัลส์ต่อรอบเท่ากับ 100 พัลส์ต่อรอบ จะสามารถแบ่งความละเอียดถึง 400 ส่วน ทำให้การวัดความเร็วได้ความละเอียดมากขึ้น

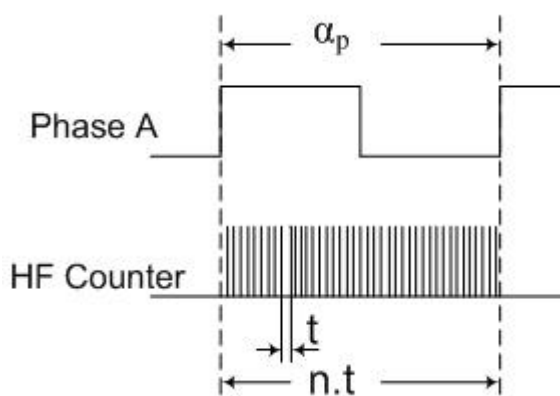


รูปที่ 2 การวัดความเร็วโดยวิธีการวัดความถี่สัญญาณจากตัวเข้ารหัส

(F. Briz, J. A. Cancelas and A. Diez, 2002)

การวัดความเร็วการหมุนด้วยวิธีวัดความถี่นี้จะต้องทำการกำหนดค่าช่วงเวลา (T_f) ที่จะนับค่า n จึงไม่เหมาะที่นำมาใช้กับการหมุนที่ความเร็วต่ำ เนื่องจากช่วงเวลาที่ใช้นับอาจจะสั้นกว่าช่วงเวลา 1 คาบสัญญาณ ดังนั้นเมื่อตัวเข้ารหัสหมุนช้าจึงใช้วิธีวัดความกว้างเต็มคาบแทน

การวัดความเร็วการหมุนโดยวิธีการวัดความกว้างเต็มคาบเป็นการนับสัญญาณนาฬิกาในช่วงคาบพัลส์สัญญาณจากตัวเข้ารหัส ดังรูปที่ 3



รูปที่ 3 การวัดความกว้างแบบเต็มคาบ

การวัดความกว้างแบบเต็มคาบ จะเป็นการนับในช่วง 1 คาบ (α_p) ของสัญญาณจากตัวเข้ารหัสเฟส A และเฟส B โดยความถี่ของวงจรถับคือ HF Counter จำนวนที่นับจะเก็บในตัวแปร n ซึ่งเป็นค่าจำนวนเต็มเท่านั้น

ถ้าคาบของความถี่ HF Counter เท่ากับ t

คาบของสัญญาณเฟส A (T_e) เท่ากับ

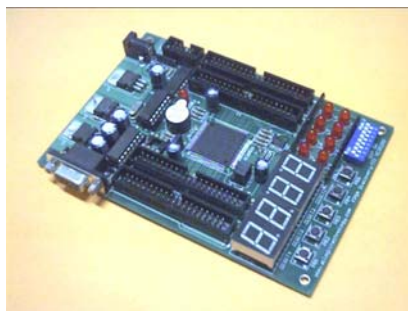
$$T_e = n.t \quad (1)$$

ดังนั้นความเร็วการหมุนเท่ากับ

$$\omega_p = \alpha_p / T_e \quad (2)$$

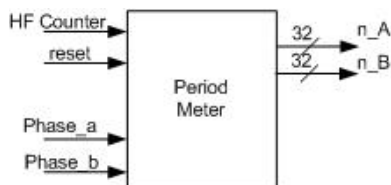
จากสมการ (1) และ (2) จะเห็นว่าตัวแปร n เป็นตัวแปรที่เปลี่ยนค่าได้ ขึ้นกับความกว้างคาบและความถี่ของ HF Counter ในขณะที่ตัวแปรอื่นเป็นตัวแปรคงที่สำหรับตัวเข้ารหัสแต่ละตัว ดังนั้นความต้องการของการคำนวณหาความเร็วการหมุนตามสมการ (1) และ (2) นั้นขึ้นอยู่กับความถูกต้องของค่าตัวแปร n

ในส่วนของการแบบวงจรเพื่อวัดความเร็วโดยวิธีการวัดความกว้างเต็มคาบนี้ ใช้ภาษาบรรยายวงจรฮาร์ดแวร์ (VHDL) (ชำนาญ และ วัชรกร, 2547) ซึ่งเป็นภาษาคอมพิวเตอร์สำหรับการออกแบบวงจรดิจิทัล โดยการสังเคราะห์วงจรที่ออกแบบแล้วโปรแกรมบนเอฟพีจีเอ (FPGA) ในการทดสอบนี้ใช้เอฟพีจีเอของบริษัท Xilinx รุ่น SPARTAN3 XC3S200-TQ144-4C ตามรูปที่ 4 (ณรงค์ และ เจริญ, 2547)



รูปที่ 4 บอร์ดทดลองเอฟพีจีเอ

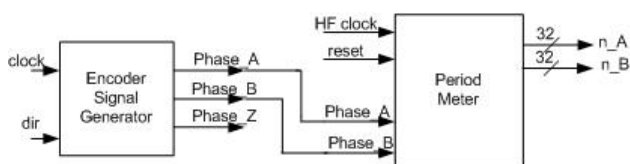
ในการออกแบบวงจรวัดความกว้างเต็มคาบ จะต้องทำการตรวจสอบสัญญาณขอบขาขึ้นของสัญญาณเฟส A และเฟส B แล้วเริ่มทำการนับค่า n จนกว่าจะถึงขอบขาขึ้นของสัญญาณเฟสนั้นๆ ผลลัพธ์ที่ได้จะเป็นค่า n จากเฟส A (n_A) และเฟส B (n_B) ตามรูปที่ 5 ถ้าตัวเข้ารหัสมีการเปลี่ยนความเร็วการหมุน ค่า n ก็จะเปลี่ยนแปลงด้วย



รูปที่ 5 บล็อกไดอะแกรมวงจรวัดความกว้างเต็มคาบ

ในบทความนี้จะนำเสนอผลการออกแบบวงจรวัดความกว้างเต็มคาบในส่วนของ การหาค่า n ตามรูปที่ 4 โดยวงจรมีส่วนนี้จะรับอินพุตสัญญาณเฟส A และเฟส B สำหรับหาค่า n_A และ n_B โดยความถี่ของวงจรมันคือ HF Counter การเก็บข้อมูลภายในจะเก็บข้อมูลขนาด 32 บิต ซึ่งจะทำให้วงจรมันสามารถนับได้สูงถึง 2^{32} ค่า ทำให้สามารถใช้วัดการหมุนที่มีความเร็วต่ำและตัวเข้ารหัสที่มีจำนวนพัลส์ต่อรอบต่ำๆ ซึ่งความกว้างคาบจะกว้างมากๆ ได้

ในการทดลองกับบอร์ดทดลองเอพพีจีเอ สัญญาณของตัวเข้ารหัสที่เป็นอินพุตของวงจรวัดความกว้างเต็มคาบเป็นสัญญาณที่สร้างขึ้นโดยเอพพีจีเอดังรูปที่ 6 ซึ่งสามารถที่จะสร้างให้มีความถี่ต่างกันหลายๆค่า เพื่อใช้ในการทดสอบการทำงานของวงจรวัด



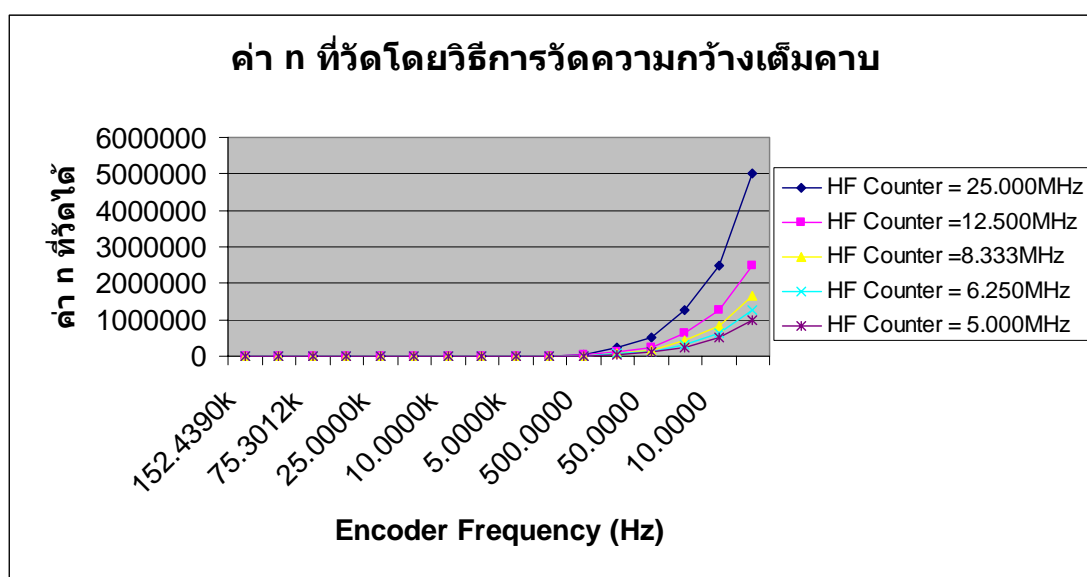
รูปที่ 6 บล็อกไดอะแกรมวงจรมองสร้างสัญญาณตัวเข้ารหัส

3.ผลการทดลอง

ในการทดลองได้ออกแบบวงจรมองสร้างสัญญาณตัวเข้ารหัสให้สร้างสัญญาณตัวเข้ารหัสความถี่ตั้งแต่ 5Hz ถึง 152kHz เพื่อให้ได้ทั้งความถี่ต่ำและสูง 16 ความถี่ตามตารางที่ 1 และความถี่การสุ่มหรือ HF Counter 5 ความถี่คือ 5.000MHz, 6.250MHz, 8.333MHz, 12.5MHz และ 25.000MHz ได้ผลดังตารางที่ 1 และรูปที่ 7

ตารางที่ 1 ค่า n จากการวัดโดยวิธีการวัดความกว้างเต็มคาบที่ความถี่การสุ่มต่างๆ

ความถี่สัญญาณจาก ตัวเข้ารหัส (Hz)	ค่า n ที่วัดได้ที่ความถี่การสุ่มต่างๆ				
	25.000MHz	12.500MHz	8.333MHz	6.250MHz	5.000MHz
152.4390k	164	82	55	41	33
125.0000k	200	100	67	50	40
75.3012k	332	166	111	83	67
50.0000k	500	250	167	125	100
25.0000k	1000	500	335	250	200
20.0321k	1248	624	416	312	251
10.0000k	2500	1250	835	625	500
8.6088k	2904	1452	968	726	581
5.0000k	5000	2500	1667	1250	1000
2.5000k	10000	5000	3335	2500	2000
500.0000	50000	25000	16667	12500	10000
100.0000	250000	125000	83335	62500	50000
50.0000	500000	250000	166667	125000	100000
20.0000	1250000	625000	416667	312500	250000
10.0000	2500000	1250000	833334	625000	500000
5.0000	5000000	2500000	1666667	1250000	1000000

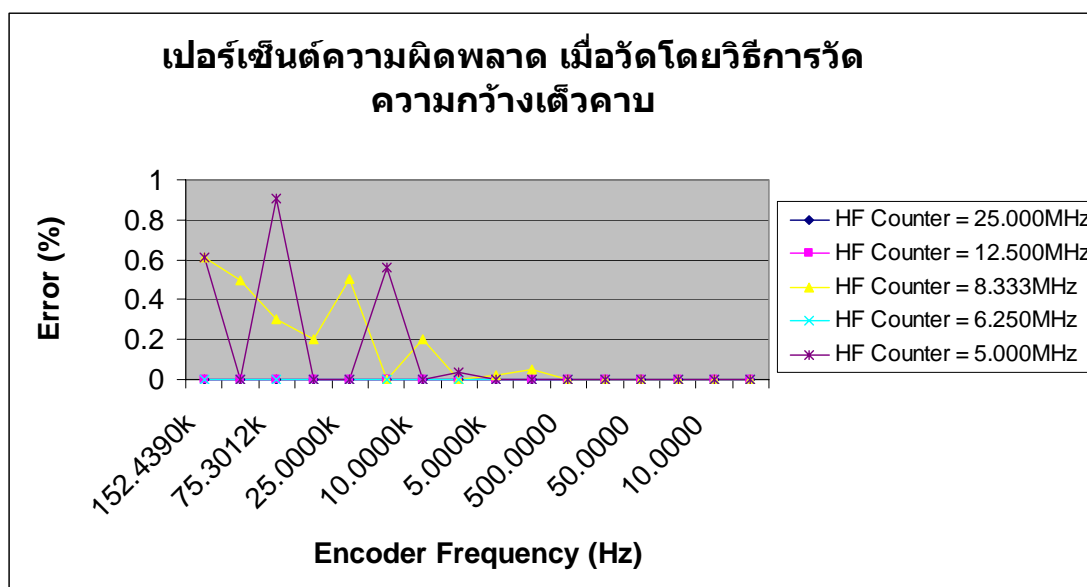


รูปที่ 7 ผลการวัดค่า n ที่ความถี่การสุ่มต่างๆ

จากตารางที่ 1 และรูปที่ 7 จะได้ว่าเมื่อความถี่สัญญาณตัวเข้ารหัสต่ำ ค่า n ที่วัดได้จะสูงขึ้น เนื่องจากที่ความถี่สัญญาณตัวเข้ารหัสต่ำความกว้างของพัลส์จะกว้างมากขึ้นด้วย ทำให้เวลาในการนับ (T_c) มากขึ้นทำให้การนับค่า n จึงนับได้มากขึ้น แต่เมื่อเปลี่ยนความถี่การสุ่มหรือความถี่การนับค่า n จะได้ว่าที่ความถี่สัญญาณตัวเข้ารหัสเท่ากัน ถ้าความถี่การสุ่มสูงก็จะได้ค่า n สูงและถ้าความถี่การสุ่มต่ำจะได้ค่า n ต่ำลง ดังนั้นจะได้ว่า ค่า n แปรผกผันกับความถี่สัญญาณตัวเข้ารหัส และแปรผันตรงกับความถี่การสุ่ม

ตารางที่ 2 เปรอ์เซ็นต์ความผิดพลาด การนับค่า n ที่ความถี่การสุ่มต่างๆ

ความถี่สัญญาณจาก ตัวเข้ารหัส (Hz)	ความผิดพลาดที่ความถี่การสุ่มต่างๆ (%)				
	25.000MHz	12.500MHz	8.333MHz	6.250MHz	5.000MHz
152.4390k	0	0	0.609695	0	0.609756
125.0000k	0	0	0.49995	0	0
75.3012k	0	0	0.301175	0	0.903614
50.0000k	0	0	0.19998	0	0
25.0000k	0	0	0.50001	0	0
20.0321k	0.00024	0.00024	0	0.000256	0.561139
10.0000k	0	0	0.200004	0	0
8.6088k	0.000179	0.000179	0.000176	0.000179	0.034263
5.0000k	0	0	0.019998	0	0
2.5000k	0	0	0.050001	0	0
500.0000	0	0	0.002	0	0
100.0000	0	0	0.002	0	0
50.0000	0	0	0.0002	0	0
20.0000	0	0	8E-05	0	0
10.0000	0	0	8E-05	0	0
5.0000	0	0	2E-05	0	0



รูปที่ 8 เปรอ์เซ็นต์ความผิดพลาด การนับค่า n ที่ความถี่การสุ่มต่างๆ

จากตารางที่ 2 และรูปที่ 8 จะได้ว่าความผิดพลาดน้อยกว่า 1% เมื่อความถี่สัญญาณตัวเข้ารหัสต่ำลงเปอร์เซ็นต์ความผิดพลาดก็ลดลงเป็นศูนย์ ไม่ว่าจะวัดโดยใช้ความถี่การสุ่มสูงหรือต่ำ ซึ่งตรงกับ

ทฤษฎีที่ว่า การวัด โดยวิธีการวัดความกว้างเต็มเมื่อวัดที่ความเร็วการหมุนต่ำ (ความถี่สัญญาณตัวเข้ารหัสต่ำ) จะให้ความถูกต้องมากกว่าที่ความเร็วการหมุนสูง

เปอร์เซ็นต์ความผิดพลาดเมื่อทำการสุ่มสัญญาณที่ 8.333MHz และ 5.000MHz ในช่วงที่สัญญาณตัวเข้ารหัสมีความถี่สูง มีเปอร์เซ็นต์ความผิดพลาดที่แกว่งขึ้น-ลง เนื่องจากการสัญญาณที่ความถี่ 8.333MHz นั้นเป็นสัญญาณที่เกิดจากการนำความถี่ 25.000MHz มาหารด้วย 3 ทำให้ความถี่ที่ได้ไม่คงที่ เมื่อนำไปสุ่มนับสัญญาณตัวเข้ารหัสจึงมีความผิดพลาดเกิดขึ้น โดยเฉพาะเมื่อความถี่สัญญาณตัวเข้ารหัสมีค่าเป็น 152.4390kHz, 75.3012kHz, 20.0321kHz และ 8.6088kHz ซึ่งเป็นความถี่ที่เกิดจากการหารที่ไม่ลงตัว ทำให้การนับในช่วงคาบนั้นๆ มีความผิดพลาดเกิดขึ้น ทั้งกับการสุ่มที่ความถี่ 8.333MHz และ 5.000MHz

4. สรุป

ในการออกแบบวงจรวัดความเต็มคาบสัญญาณตัวเข้ารหัสด้วยภาษาบรรยายวงจรฮาร์ดแวร์ และทดสอบกับการวัดสัญญาณตัวเข้ารหัสที่มีความถี่แตกต่างกันระหว่าง 5Hz ถึง 152kHz โดยให้มีการสุ่มสัญญาณที่ความถี่ต่างกันระหว่าง 5MHz ถึง 25MHz จากผลการทดสอบได้เปอร์เซ็นต์ความผิดพลาด มีทำการสุ่มที่ความถี่ต่างๆ ดังนี้ ความถี่ 25.000MHz มีความผิดพลาด 2.62×10^{-5} %, ความถี่ 12.500MHz มีความผิดพลาด 2.62×10^{-5} %, ความถี่ 8.333MHz มีความผิดพลาด 1.49×10^{-1} %, ความถี่ 6.250MHz มีความผิดพลาด 2.72×10^{-5} % และที่ความถี่ 5.000MHz มีความผิดพลาด 1.32×10^{-1} % ซึ่งถือว่า มีเปอร์เซ็นต์ความผิดพลาดที่ต่ำ เมื่อนำค่า n ที่วัดได้ซึ่งมีเปอร์เซ็นต์ความผิดพลาดต่ำไปคำนวณหาค่าความเร็วการหมุนต่อ ก็จะได้ค่าความเร็วที่มีความถูกต้องสูง

ในการเลือกความถี่ในการสุ่มนั้น ควรเลือกความถี่ที่มีค่าคงตัว เพื่อไม่ให้เกิดความผิดพลาดจากความไม่คงที่ของสัญญาณการสุ่ม และไม่ควรเลือกที่มีความถี่การสุ่มต่ำเกินไป เพื่อหลีกเลี่ยงความผิดพลาดที่เกิดจากการสุ่มซ้ำเกินไป ซึ่งส่งผลต่อการนับค่า n เมื่อความถี่สัญญาณตัวเข้ารหัสมีความถี่สูง

5. เอกสารอ้างอิง

กิติพงษ์ หงส์วรานนท์ และ ปัญชร ชัยศรี. ระบบควบคุมเครื่องจักรขับเคลื่อน 2.3 เมกะวัตต์. 2546.

ชำนาญ ปัญญาใส และ วัชรกร ทองหนู. 2547. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล, กรุงเทพฯ, ซีเอ็ดดูเคชั่น.

ณรงค์ ทองนิม และ เจริญ วงษ์ชุ่มเย็น. 2547. การออกแบบวงจรรวมดิจิทัล โดยใช้ FPGA และ CPLD.

<http://www.ailogictechnology.com>. Apex Instrument Co., Ltd.

B. Pamela and H. Blake, "Single-Chip Velocity Measurement System for Incremental Optical

Encoders," IEEE Trans. Control Systems Technology, vol. 5, no 6, Nov. 1997, pp. 654-661.

- E. Sisinni, A. Flammini, D. Marioli and A. Taroni, "A PLD Based Encoder Interface with Accurate Position and Velocity Estimation," in Proc. ISIE 2002. Industrial. Electronics., vol. 2, Jul. 2002, pp. 606 – 611
- F. Briz, J. A. Cancelas and A. Diez, "Speed Measurement Using Rotary Encoders for High performance ac Drives," IECON '94. Int. Con. Control. Instrumentation, vol. 1, Sep. 1994, pp. 538-548.
- Hewlett-Packard. Incremental Encoder Errors: Causes and Methods to Reduce Them. Hewlett Packard Application Brief M-109. <http://www.hpl.hp.com> Access date 28 December 2004
- SERVO-TEK PRODUCTS COMPANY. Incremental Encoder. <http://www.servotek.com/PAGE6.HTM> Access date 7 January 2006