บทที่ 3

การประมวลผลสัญญาณเชิงเลข

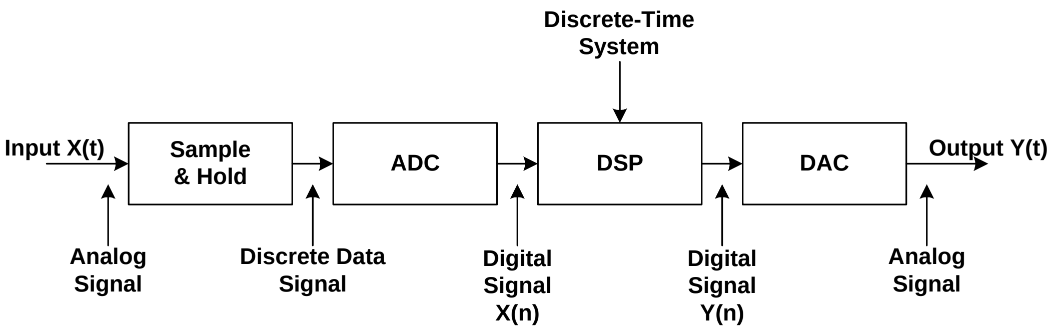
3.1 บทนำ

สัญญาณที่พบและใช้อยู่ในชีวิตประจำวันส่วนใหญ่อยู่ในรูปสัญญาณเชิงอุปมานซึ่งหมายถึงสัญญาณที่มีรูปคลื่น (Waveform) แปรไปอย่างต่อเนื่องกับพิสัยเวลาโดยที่แอมปลิจูด (Amplitude) หรือค่าขนาดของสัญญาณก็มีการแปรไปอย่างต่อเนื่องด้วย ยกตัวอย่างเช่น สัญญาณไซน์ สัญญาณเสียง ฯลฯ สัญญาณเชิงอุปมานดังกล่าวนี้ในอดีตระบบที่ใช้ในการประมวลผลส่วนใหญ่ถูกสร้างจากอุปกรณ์อิเล็กทรอนิกส์โดยอาศัยคุณสมบัติเฉพาะตัวของอุปกรณ์อิเล็กทรอนิกส์ เช่น ตัวต้านทาน ตัวเก็บประจุ ทรานซิสเตอร์ ฯลฯ ที่มีพฤติกรรมต่อปริมาณทางไฟฟ้าเช่น แรงดัน กระแส ความถี่และเฟส ผลที่ได้นี้เปรียบได้ว่ามันก็เกิดจากการคำนวณของอุปกรณ์อิเล็กทรอนิกส์นั่นเอง ยกตัวอย่างเช่น การใช้ออปแอมป์ (Operational Amplifier: Op Amp) ต่อเป็นวงจรขยายสองเท่าแบบกลับเฟส ผลที่ได้ก็คือแรงดันขาออกมีขนาดเท่ากับแรงดันขาเข้าที่ถูกคูณด้วยลบสอง ดังนั้นจึงสามารถใช้การคำนวณทางซอร์ฟแวร์แทนพฤติกรรมการคำนวณของอุปกรณ์อิเล็กทรอนิกส์ดังกล่าวได้ ด้วยเหตุนี้จึงได้มีการคิดทฤษฏีการประมวลผลสัญญาณเชิงเลขขึ้น ซึ่งในความเป็นจริงได้มีการวิจัยออกมานานแล้วแต่ไม่สามารถสร้างได้จริงในทางปฏิบัติ (Realize) เนื่องจากเทคโนโลยีการผลิตชิพไอซีไม่สามารถสร้างตัวประมวลผลที่มีความซับซ้อนสำหรับรองรับอัตราความเร็วของการคำนวณที่ต้องการได้ แต่ในปัจจุบันเทคโนโลยีวีแอลเอสไอ (VLSI: Very Large Scale Integrated Circuit) ที่พัฒนาไปอย่างรวดเร็วจึงทำให้สามารถสร้างชิพตัวประมวลผลที่มีความเร็วในการประมวลผลสูงเพียงพอกับความต้องการได้

โดยทั่วไปสัญญาณขาออกของระบบการประมวลผลสัญญาณเชิงเลขที่เราต้องการคือสัญญาณเชิงอุปมานแต่สัญญาณขาเข้าของระบบสามารถเป็นได้ทั้งสัญญาณเชิงอุปมานและสัญญาณเชิงเลข การประมวลผลสัญญาณเชิงเลขจึงสามารถแบ่งออกได้เป็นสองแบบคือ การประมวลผลสัญญาณเชิงเลขของสัญญาณใดๆที่อยู่ในรูปแบบเชิงเลขหรือการประมวลผลสัญญาณเชิงเลขของสัญญาณใดๆที่ไม่ได้อยู่ในรูปแบบเชิงเลข

3.2 สัญญาณและระบบแบบไม่ต่อเนื่อง

ในระบบการประมวลผลสัญญาณเชิงเลข สัญญาณเชิงอุปมานขาเข้าที่เป็น "สัญญาณเชิงเวลาต่อเนื่อง" (Continuous-time Signal) จะถูกสุ่มและคงค่า (Sampling and Hold) เรียกสัญญาณที่จุดนี้ว่า "สัญญาณเชิงข้อมูลเต็มหน่วย" (Discrete Data Signal) จากนั้นจะถูกแปลงเป็น "สัญญาณเชิงเลข" (Digital Signal) ทั้งสัญญาณเชิงข้อมูลเต็มหน่วยและสัญญาณเชิงเลข ถือว่าเป็น "สัญญาณเชิงเวลาเต็มหน่วย" (Discrete-time Signal) ตำราบางเล่มเรียกว่า "สัญญาณไม่ต่อเนื่อง" [3] คำว่าต่อเนื่องหรือไม่ต่อเนื่องนี้หมายถึงสัญญาณนั้นๆมีค่าต่อเนื่องทางเวลาหรือไม่ สัญญาณไม่ต่อเนื่องหมายถึงสัญญาณที่มีค่าเพียงบางจุดของเวลาเท่านั้นและใช้สัญลักษณ์ n แทนเวลาแบบไม่ต่อเนื่อง โดยที่ n เป็นตัวแปรที่มีค่าเป็นจำนวนเต็มคือ n = …, -2, -1, 0, 1, 2,… สัญญาณไม่ต่อเนื่องจะเป็นฟังก์ชั่นของ n ดังนั้นจะเขียนแทนสัญญาณนี้ว่า x[n] ส่วนระบบไม่ต่อเนื่องก็คือ ระบบที่ทำการประมวลผลสัญญาณไม่ต่อเนื่องซึ่งก็คือโปรเซสเซอร์ (Digital Signal Processor) นั่นเอง ด้วยเหตุนี้การประมวลผลสัญญาณเชิงเลขบางครั้งถูกเรียกว่า การประมวลผลสัญญาณไม่ต่อเนื่อง (Discrete-Time Signal Processing)



รูปที่ 3.1 แสดงระบบพื้นฐานของการประมวลผลสัญญาณเชิงเลข

3.2.1 ทฤษฎีการสุ่ม (Sampling Theorem)

การแปลงข้อมูลระหว่างสัญญาณเชิงอุปมานและสัญญาณเชิงเลขสิ่งที่ต้องคำนึงถึงและให้ความสำคัญเป็นอย่างมากก็คือความสมบูรณ์ครบถ้วนของข้อมูลหลังการแปลง ทฤษฎีการสุ่มกล่าวว่าสัญญาณเชิงอุปมานที่มีความหนาแน่นสเปกตรัมอยู่ในแบนด์วิดท์ที่จำกัด เราจะพบว่าในโดเมนเวลาของสัญญาณจะมีข้อมูลที่มีความซ้ำซ้อนดังนั้นถ้าเราเลือกสุ่มค่าตัวอย่างของสัญญาณด้วยคาบเวลาที่เหมาะสมและมีระยะห่างอย่างสม่ำเสมอจะทำให้เราได้ข้อมูลที่เป็นตัวแทนของสัญญาณเชิงอุปมานที่ถูกสุ่มได้อย่างสมบูรณ์ โดยที่เวลาของการสุ่มสามารถพิจารณาได้ดังนี้คือ สมมติว่าสัญญาณใดๆที่มีค่าฟังก์ชั่นมีความหนาแน่นสเปกตรัมอยู่ในช่วงความถี่ที่มีแบนด์วิดท์ไม่เกิน B เฮิรตซ์ การสุ่มค่าตัวอย่าง

ของสัญญาณด้วยคาบเวลาไม่เกิน $\frac{1}{2B}$ วินาที หรือความถี่ของการสุ่มจะต้องมากกว่าหรือเท่ากับ 2B เฮิรตซ์ ก็จะสามารถเก็บข้อมูลที่เป็นตัวแทนของสัญญาณเชิงอุปมานที่ถูกสุ่มได้อย่างสมบูรณ์และสามารถสร้างสัญญาณเชิงอุปมานกลับคืนอย่างสมบูรณ์ด้วยข้อมูลที่ได้จากการสุ่มนี้ด้วยเช่นกัน จากคำกล่าวในข้างต้นสามารถอธิบายในเชิงคณิตศาสตร์ได้ดังนี้คือ สมมติให้สัญญาณเชิงอุปมาน $f(t)$ มีค่าความหนาแน่นเชิงสเปกตรัมอยู่ไม่เกินความถี่ B เฮิรตซ์ สัญญาณ $f(t)$ นี้ถูกสุ่มด้วยเวลาคงที่ T วินาที และสมมติให้สัญญาณที่ถูกสุ่มแล้วนี้ว่า $f_S(t)$ จะได้ว่า

$$f_S(t) = f(t).\delta_T(t) \tag{3.1}$$

โดยในที่นี้ $\delta_T(t)$ คือขบวนอิมพัลส์ (Impulse Train) ที่มีการนิยามว่า

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.2}$$

จากคุณสมบัติการคูณฟังก์ชั่นอิมพัลส์กับฟังก์ชั่นอื่นจะได้

$$f(t)\delta(t - t0) = f(t0)\delta(t - t0) \tag{3.3}$$

จาก (3.1) และ (3.2) จะได้

$$f_S(t) = \sum_{n=-\infty}^{\infty} f(nT)\delta(t - nT) \tag{3.4}$$

จาก (3.4) แสดงให้เห็นว่า $f_S(t)$ คือสัญญาณ $f(t)$ ที่มีค่าเพียงช่วงเวลา $t = nT(n = 1,2,..)$ เท่านั้นและผลการแปลงฟูริเยร์ของ $\delta_T(t)$ จะได้

$$\delta_T(t) \leftrightarrow \delta_{\omega_S}(\omega)\omega_S \tag{3.5}$$

โดยที่ $\omega_S = \frac{2\pi}{T}$ และจากทฤษฎีการทำคอนโวลูชัน

$$f_1(t)f_2(t) \leftrightarrow \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(\mu)F_2(\omega-\mu)d\mu \qquad (3.6)$$

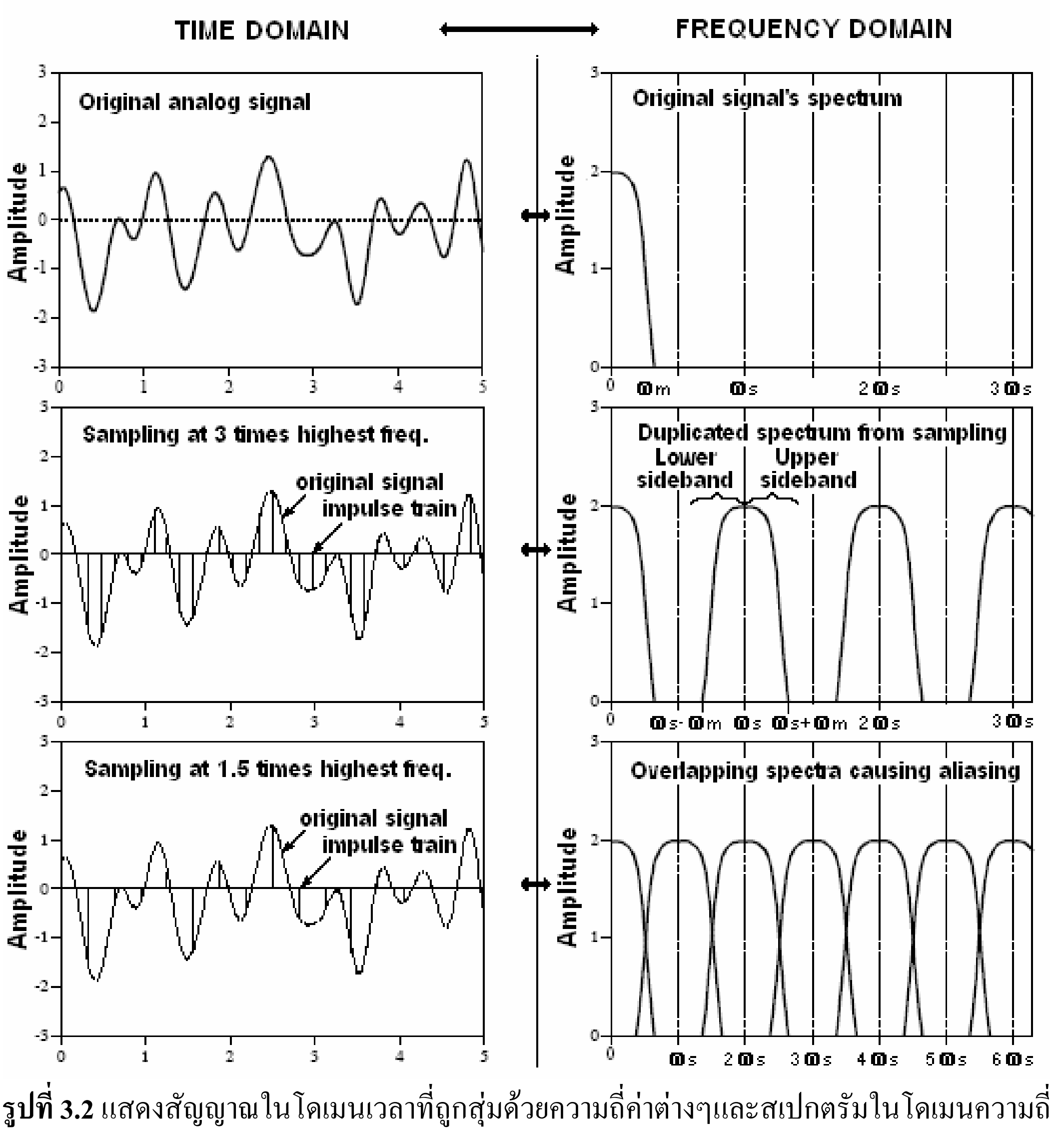
จาก (3.6) ถ้ากำหนดให้ $f_S(t) \leftrightarrow F_S(\omega)$ และ $f(t) \leftrightarrow F(\omega)$ จะได้

$$\begin{aligned} F_S(\omega) &= F(\omega)\delta_{\omega_S}(\omega)\omega_S \\ &= \frac{\omega_S}{2\pi}\int_{-\infty}^{\infty}\sum_{n=-\infty}^{\infty} F(z)\delta(\omega - n\omega_S - z)dz \\ &= \frac{1}{T}\sum_{n=-\infty}^{\infty} F(\omega - n\omega_S) \end{aligned} \qquad (3.7)$$

จาก 3.7 ทำให้ทราบว่าสัญญาณ $f_S(t)$ หรือสัญญาณที่เกิดจากการสุ่มของสัญญาณ $f(t)$ จะมีฟังก์ชั่นความหนาแน่นเชิงสเปกตรัมที่มีรูปร่างเหมือนฟังก์ชั่นความหนาแน่นเชิงสเปกตรัมของสัญญาณ $f(t)$ แต่จะเกิดอยู่ซ้ำกันทุกคาบความถี่ ω_S แสดงดังรูปที่ 3.2 เมื่อสัญญาณ $f(t)$ มีองค์ประกอบของความถี่สูงสุดคือ B เฮิรตซ์ ค่าความถี่เชิงมุมที่ตรงกันกับค่าความถี่สูงสุดนี้จะมีค่า $\omega_M = 2\pi B$ เรเดียน/วินาที จะพบว่าถ้า $\omega_S > 2\omega_M$ หรือ $T < \frac{1}{2B}$ เช่นในกรณีที่ใช้ค่าความถี่ในการสุ่มเท่ากับสามเท่าของความถี่สูงสุดของสัญญาณ $f(t)$ ($\omega_S = 3\omega_M$) จะได้ $F_S(\omega)$ มีลักษณะดังรูปที่ 3.2 (กลาง-ขวา) แต่ถ้า $\omega_S < 2\omega_M$ หรือ $T > \frac{1}{2B}$ เช่นในกรณีที่ใช้ค่าความถี่ในการสุ่มหนึ่งจุดห้าเท่าของความถี่สูงสุดของสัญญาณ $f(t)$ ($\omega_S = 1.5\omega_M$) จะได้ $F_S(\omega)$ มีลักษณะดังรูปที่ 3.2 (ล่าง-ขวา) จะสังเกตเห็นว่าเกิดการเหลื่อมทับกันของ $F(\omega)$ ทำให้เกิดผลเทียบเท่ากับการพับกลับ (Fold Over) องค์ประกอบของสัญญาณที่มีความถี่สูงเกิน $\frac{\omega_S}{2}$ ซึ่งกลับมาทับกับองค์ประกอบของสัญญาณที่มีความถี่ต่ำและรวมตัวกันจึงทำให้มีสเปกตรัมผิดเพี้ยนไปจากเดิม ปรากฏการณ์ที่มีการเหลื่อมหรือการพับกลับของส่วนประกอบของสเปกตรัมดังกล่าวนี้มีคำศัพท์เรียกเฉพาะว่าการเกิด เอเลียสซิง (Aliasing)

จากที่ได้กล่าวในข้างต้น การใช้คาบเวลาในการสุ่ม $T < \frac{1}{2B}$ วินาที จะทำให้สามารถใช้วงจรกรองความถี่ในการแยกเอาเฉพาะส่วนที่มีความถี่ต่ำของ $F_S(\omega)$ ซึ่งเหมือนกับ $F(\omega)$ ออกมาได้แต่อย่างไรก็ตามในกรณีที่ $T > \frac{1}{2B}$ แล้วมีการซ้อนทับกันของส่วนประกอบความถี่ ทำให้วงจรกรองความถี่ไม่สามารถแยกเอา $F(\omega)$ จาก $F_S(\omega)$ ได้ สำหรับในกรณีที่ใช้คาบเวลาในการสุ่ม $T = \frac{1}{2B}$ พอดีหรือที่เรียกว่าช่วงเวลาไนควิสต์ (Nyquist Interval) จะต้องใช้วงจรกรองความถี่ในอุดมคติเท่านั้น

จึงจะสามารถแยกสเปกตรัมในช่วงความถี่ที่ต้องการคือ $F(\omega)$ ออกมาได้ ดังนั้นโดยปกติในทางปฏิบัติจึงต้องใช้คาบเวลาการสุ่มให้น้อยกว่า $\frac{1}{2B}$ เพื่อแก้ปัญหาการโรลออฟ (Roll Off) หรือการค่อยๆ ลดลงของค่าฟังก์ชั่นถ่ายโอนบริเวณจุดตัดความถี่ (Cutoff) ของวงจรกรองความถี่ซึ่งไม่เป็นไปตามอุดมคติ



รูปที่ 3.2 แสดงสัญญาณในโดเมนเวลาที่ถูกสุ่มด้วยความถี่ค่าต่างๆและสเปกตรัมในโดเมนความถี่

ทฤษฎีการสุ่มนี้สามารถประยุกต์ใช้ได้ทั้งในส่วนของการแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลขและการแปลงจากสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมาน แต่จะมีเงื่อนไขอื่นๆที่จะต้องนำมาวิเคราะห์ร่วมเช่น ผลของการคงค่าข้อมูล (Hold) ฯลฯ ซึ่งจะกล่าวโดยละเอียดในหัวข้อระบบการประมวลผลสัญญาณเชิงเลข

3.2.2 การแปลงแซดและการแปลงแซดผกผัน

ในสาขาวิศวกรรมไฟฟ้าอิเล็กทรอนิกส์มีคณิตศาสตร์ที่เรียกว่าการแปลงอยู่หลายรูปแบบเพื่อใช้ในการวิเคราะห์สัญญาณ, ระบบ หรือการแก้ปัญหาโดยอ้อม ยกตัวอย่างเช่น การแปลงลาปลาซ (Laplace's Transform) ที่ช่วยให้การวิเคราะห์ระบบหรือโครงข่ายเชิงอุปมานสามารถทำได้ง่ายขึ้น ทั้งนี้โดยทั่วไปในโดเมนเวลานั้น คุณสมบัติของระบบหรือโครงข่ายสามารถเขียนอธิบายได้โดยใช้สมการเชิงอินทริกรัลอนุพันธ์ (Integro-Differential Equation) [2] ระบบที่ซับซ้อนจะมีพจน์ของสมการเชิงอินทริกรัลอนุพันธ์จำนวนมาก ซึ่งเป็นที่ทราบกันว่าการหาคำตอบจากสมการเชิงอินทริกรัล อนุพันธ์โดยตรงทำได้ค่อนข้างยาก แต่ถ้าเราทำการแก้ปัญหาโดยการประยุกต์ใช้การแปลงลาปลาซ ผลของการแปลงจะทำให้เราได้สมการพีชคณิตในโดเมนความถี่และการหาคำตอบก็สามารถใช้วิธีการของพีชคณิตซึ่งทำได้ง่ายกว่า คำตอบที่ได้จะอยู่ในรูปของตัวแปรโดเมนความถี่จะต้องใช้การแปลงผกผันเพื่อหาคำตอบที่อยู่ในโดเมนเวลาที่เราต้องการ จากหลักการดังกล่าวระบบเชิงเลขก็เช่นเดียวกัน มีการใช้การแปลงเป็นเครื่องมือที่สำคัญในการวิเคราะห์คุณสมบัติของสัญญาณและระบบแต่ต่างชนิดกับที่ใช้ในระบบเชิงอุปมาน ยกตัวอย่างเช่น การออกแบบตัวกรองความถี่เชิงอุปมานใช้การแปลงลาปลาซเป็นเครื่องมือในการวิเคราะห์ ในขณะที่การออกแบบตัวกรองเชิงเลขใช้การแปลงแซด อาจกล่าวได้ว่าการแปลงทั้งสองชนิดนี้ใช้กลยุทธเดียวกันคือการหาโพลและซีโร่ของระบบ การแปลงลาปลาซใช้กับสมการเชิงอนุพันธ์ ผลที่ได้อยู่ในโดเมน S ส่วนการแปลงแซดใช้กับสมการผลต่าง (Difference Equation) และผลที่ได้อยู่ในโดเมน Z แต่การแปลงทั้งสองนี้ยังมีความแตกต่างกันบ้างเช่นระนาบ S ใช้ระบบพิกัดฉากแต่ระนาบ Z ใช้ระบบพิกัดเชิงขั้วเป็นต้น สูตรการแปลงแซดแสดงดังสมการที่ (3.8)

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \tag{3.8}$$

หลังจากได้ผลเฉลยที่อยู่ในโดเมน z แล้วจะต้องแปลงกลับผลเฉลยนั้นให้กลับไปอยู่ในโดเมนเวลาโดยการแปลงแซดผกผัน (Inverse Z- Transform) แสดงดังสมการที่ (3.9) เพื่อใช้ในการออกแบบตัวประมวลผลหรือการเขียนโปรแกรม สำหรับในโดเมนเวลาจะใช้สมการผลต่าง (Difference Equation) ในการแสดงความสัมพันธ์ระหว่างสัญญาณขาออก y[n] และ สัญญาณขาเข้า x[n] ยกตัวอย่างเช่น y[n] = 0.5x[n]+2x[n-1] เป็นต้น

$$x[n] = \frac{1}{2\pi j} \oint_c X(z) z^{n-1} dz \tag{3.9}$$

เมื่อ C เป็นคอนทัวร์ในทิศทางทวนเข็มนาฬิกาที่ล้อมรอบจุดเอกฐาน (Singular Point) ของ $X(z)z^{n-1}$

3.2.3 การวิเคราะห์ฟูริเยร์

การวิเคราะห์ฟูริเยร์เป็นคณิตศาสตร์แขนงหนึ่งซึ่งใช้การแยกสัญญาณออกเป็นสัญญาณไซน์ความถี่ต่างๆ เนื่องจากการวิเคราะห์สัญญาณไซน์สามารถทำได้ง่ายกว่าสัญญาณต้นฉบับ [11] นอกจากนี้สัญญาณไซน์และโคไซน์ยังมีคุณสมบัติ Sinusoidal fidelity ในขณะที่สัญญาณต้นฉบับไม่มี สัญญาณที่พิจารณาอาจจะเป็นสัญญาณต่อเนื่องหรือสัญญาณไม่ต่อเนื่องและอาจจะเป็นสัญญาณคาบหรือไม่ใช่สัญญาณคาบ ด้วยเหตุนี้การวิเคราะห์ฟูริเยร์จึงสามารถแบ่งออกได้เป็นสี่ชนิดคือ

- สัญญาณที่สนใจไม่เป็นสัญญาณคาบและเป็นสัญญาณต่อเนื่องมีความยาวตามแกนเวลาจากลบอนันต์จนถึงบวกอนันต์ สัญญาณชนิดนี้ต้องใช้ “การแปลงฟูริเยร์” (Fourier Transform) ในการวิเคราะห์
- สัญญาณที่สนใจเป็นสัญญาณคาบและเป็นสัญญาณต่อเนื่องมีความยาวตามแกนเวลาจากลบอนันต์จนถึงบวกอนันต์ สัญญาณชนิดนี้ต้องใช้ “อนุกรมฟูริเยร์” (Fourier Series) ในการวิเคราะห์
- สัญญาณที่สนใจไม่เป็นสัญญาณคาบและเป็นสัญญาณไม่ต่อเนื่องมีความยาวจุดของข้อมูลที่ได้จากการสุ่มตามแกนเวลาจากลบอนันต์จนถึงบวกอนันต์ สัญญาณชนิดนี้ต้องใช้ “ดีสครีตไทม์ฟูริเยร์ทรานสฟอร์ม” (Discrete Time Fourier Transform: DTFT) ในการวิเคราะห์
- สัญญาณที่สนใจเป็นสัญญาณคาบและเป็นสัญญาณไม่ต่อเนื่องมีความยาวจุดของข้อมูลที่ได้จากการสุ่มตามแกนเวลาจากลบอนันต์จนถึงบวกอนันต์ สัญญาณชนิดนี้ต้องใช้ “ดีสครีตฟูริเยร์ทรานสฟอร์ม” (Discrete Fourier Transform: DFT) ในการวิเคราะห์ ซึ่งการแปลงชนิดนี้บางครั้งเรียกว่า “ดีสครีตฟูริเยร์ซีรีส์” (Discrete Fourier Series: DFS)

การวิเคราะห์ฟูริเยร์ทั้งสี่ชนิดดังกล่าวสามารถแบ่งออกเป็นชนิดย่อยได้อีกสองชนิดคือ คอมเพล็ก (Complex) และ เรียล (Real) ทั้งสองชนิดนี้แตกต่างกันที่รูปแบบของคณิตศาสตร์ที่ใช้ในการวิเคราะห์แสดงรายละเอียดดังรูปที่ 3.3-3.6 ถึงแม้ว่าการวิเคราะห์ฟูริเยร์จะมีอยู่หลายชนิดด้วยกันแต่สำหรับงานด้านการประมวลผลสัญญาณเชิงเลขจะใช้เพียงชนิดเดียวคือ DFT เนื่องจากตัวประมวลผลสามารถทำงานได้กับเฉพาะข้อมูลแบบไม่ต่อเนื่องและมีความยาวข้อมูลจำกัด

Fourier Transform

complex transform	real transform
synthesis $$x(t) = \int_{-\infty}^{+\infty} X(\omega)\, e^{j\omega t}\, d\omega$$	*synthesis* $$x(t) = \int_{0}^{+\infty} ReX(\omega)\cos(\omega t) - ImX(\omega)\sin(\omega t)\, dt$$
analysis $$X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} x(t)\, e^{-j\omega t}\, dt$$	*analysis* $$ReX(\omega) = \frac{1}{\pi}\int_{-\infty}^{+\infty} x(t)\cos(\omega t)\, dt$$ $$ImX(\omega) = \frac{-1}{\pi}\int_{-\infty}^{+\infty} x(t)\sin(\omega t)\, dt$$

รูปที่ 3.3 แสดงสมการที่ใช้ในการแปลงฟูริเยร์

Fourier Series

complex transform	real transform
synthesis $$x(t) = \sum_{k=-\infty}^{+\infty} X[k]\, e^{j2\pi kt/T}$$	*synthesis* $$x(t) = \sum_{k=0}^{+\infty} ReX[k]\cos(2\pi kt/T) - ImX[k]\sin(2\pi kt/T)$$
analysis $$X[k] = \frac{1}{T}\int_{0}^{T} x(t)\, e^{-j2\pi kt/T}\, dt$$	*analysis* $$ReX[k] = \frac{2}{T}\int_{0}^{T} x(t)\cos(2\pi kt/T)\, dt$$ $$ImX[k] = \frac{-2}{T}\int_{0}^{T} x(t)\sin(2\pi kt/T)\, dt$$

รูปที่ 3.4 แสดงสมการที่ใช้ในการคำนวณอนุกรมฟูริเยร์

Discrete Time Fourier Transform (DTFT)

complex transform	real transform
synthesis $x[n] = \int_{0}^{2\pi} X(\omega)\, e^{j\omega n}\, d\omega$	*synthesis* $x[n] = \int_{0}^{\pi} ReX(\omega)\cos(\omega n) - ImX(\omega)\sin(\omega n)\, d\omega$
analysis $X(\omega) = \frac{1}{2\pi} \sum_{n=-\infty}^{+\infty} x[n]\, e^{-j\omega n}$	*analysis* $ReX(\omega) = \frac{1}{\pi} \sum_{n=-\infty}^{+\infty} x[n]\cos(\omega n)$ $ImX(\omega) = \frac{-1}{\pi} \sum_{n=-\infty}^{+\infty} x[n]\sin(\omega n)$

รูปที่ 3.5 แสดงสมการที่ใช้ในการคำนวณดีสครีตไทม์ฟูริเยร์ทรานสฟอร์ม

Discrete Fourier Transform (DFT)

complex transform	real transform
synthesis $x[n] = \sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$	*synthesis* $x[n] = \sum_{k=0}^{N/2} ReX[k]\cos(2\pi kn/N) - ImX[k]\sin(2\pi kn/N)$
analysis $X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$	*analysis* $ReX[k] = \frac{2}{N} \sum_{n=0}^{N-1} x[n]\cos(2\pi kn/N)$ $ImX[k] = \frac{-2}{N} \sum_{n=0}^{N-1} x[n]\sin(2\pi kn/N)$

รูปที่ 3.6 แสดงสมการที่ใช้ในการคำนวณดีสครีตฟูริเยร์ทรานสฟอร์ม

3.2.4 ตัวกรองเชิงเลข

โดยทั่วไปคำว่าตัวกรองสัญญาณอาจหมายถึง อุปกรณ์ซึ่งอาจเป็นอุปกรณ์ทางเครื่องมือกลวงจรอิเล็กทรอนิกส์ ซอร์ฟแวร์ หรือ คอมพิวเตอร์ฮาร์ดแวร์ ที่สามารถเปลี่ยนแปลงสเปกตรัมซึ่งเป็นองค์ประกอบทางความถี่ของสัญญาณให้มีคุณสมบัติตามที่ต้องการได้ ดังนั้นในที่นี้คำว่าตัวกรองเชิงเลข

จะหมายถึงระบบการประมวลผลสัญญาณเชิงเลขที่ทำให้สเปกตรัมความถี่เปลี่ยนแปลงไป [2] วัตถุประสงค์เพื่อแยกสัญญาณที่ถูกรวมแล้วหรือสร้างคืน (Restoration) สัญญาณเมื่อมีความเพี้ยนซึ่งเกิดจากการรบกวนจากสัญญาณที่ไม่ต้องการ [11] ในการจำแนกชนิดของตัวกรองเชิงเลขสามารถแบ่งได้ตามกลุ่มของการใช้งาน และวิธีการสร้างแสดงดังรูปที่ 3.7

FILTER USED FOR:	FILTER IMPLEMENTED BY: Convolution *Finite Impulse Response (FIR)*	FILTER IMPLEMENTED BY: Recursion *Infinite Impulse Response (IIR)*
Time Domain *(smoothing, DC removal)*	Moving average	Single pole
Frequency Domain *(separating frequencies)*	Windowed-sinc	Chebyshev
Custom *(Deconvolution)*	FIR custom	Iterative design

รูปที่ 3.7 แสดงการจำแนกชนิดของตัวกรองเชิงเลขตามกลุ่มของการใช้งานและวิธีการสร้าง

การใช้งานสามารถแบ่งย่อยออกเป็นสามประเภทได้แก่

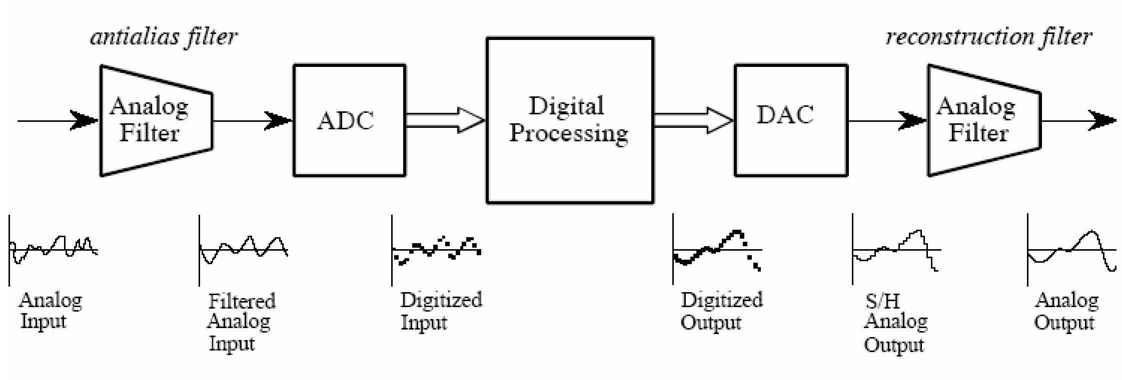
- การใช้งานในโดเมนเวลา ออกแบบเพื่อใช้กับข้อมูลที่สนใจรูปร่างของรูปคลื่นยกตัวอย่าง เช่น ตัวกรองสัญญาณไฟตรงออกจากสัญญาณที่ต้องการ หรือ ตัวกรองประเภท Waveform Shaping
- การใช้งานในโดเมนความถี่ ออกแบบเพื่อใช้กับข้อมูลที่แฝงอยู่ในขนาด, ความถี่และเฟสขององค์ประกอบสัญญาณไซน์ วัตถุประสงค์เพื่อการแยกสัญญาณช่วงความถี่ที่ต้องการออก
- การใช้งานพิเศษเฉพาะอย่าง (Custom) เป็นตัวกรองที่มีคุณสมบัติแตกต่างไปจากตัวกรองพื้นฐานทั้งสี่ชนิด (Low-pass, High-pass, Band-pass, Band reject) ซึ่งหมายถึงการตอบสนองต่อความถี่ของตัวกรองจะมีลักษณะพิเศษตามความต้องการของผู้ออกแบบ

วิธีการสร้างตัวกรองสามารถแบ่งออกได้เป็นสองชนิดได้แก่

- Convolution คือตัวกรองไม่ป้อนกลับเชิงเลขหรือที่เรียกว่าเอฟไออาร์ (Finite Impulse Response: FIR) ตัวกรองชนิดนี้มีความยาวของผลตอบสนองอิมพัลส์จำกัด
- Recursion คือตัวกรองไม่ป้อนกลับเชิงเลขหรือที่เรียกว่าเอฟไออาร์ (Infinite Impulse Response: IIR) เป็นตัวกรองที่มีความยาวของผลตอบสนองอิมพัลส์ไม่จำกัด

3.3 ระบบการประมวลผลสัญญาณเชิงเลข

ระบบพื้นฐานของการประมวลผลสัญญาณเชิงเลขประกอบด้วย ส่วนแรกคือตัวกรองเชิงอุปมานหน้าที่กรองความถี่ต่ำผ่านและสัญญาณจะถูกป้อนให้ส่วนที่สองตัวแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข (Analog to Digital Converter: ADC) ทำหน้าที่สุ่มสัญญาณขาเข้าด้วยคาบเวลาค่าหนึ่งแล้วแบ่งระดับเพื่อแปลงเป็นสัญญาณเชิงเลขส่งต่อให้ส่วนที่สามคือตัวประมวลผลทำหน้าที่คำนวณตามขั้นตอนวิธี (Algorithm) ซึ่งอาจเป็นได้ทั้งซอร์ฟแวร์หรือฮาร์ดแวร์ หลังจากได้คำตอบจากการคำนวณแล้วขั้นตอนที่สี่คือการแปลงสัญญาณเชิงเลขกลับไปเป็นสัญญาณเชิงอุปมานโดยตัวแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมาน (Digital to Analog Converter: DAC) และขั้นตอนสุดท้ายของระบบก็คือตัวกรองเชิงอุปมานชนิดความถี่ต่ำผ่านเพื่อตัดความถี่เงา (Image) เพื่อให้ได้สัญญาณเชิงอุปมานที่สมบูรณ์



รูปที่ 3.8 แสดงบล็อคไดอะแกรมของระบบการประมวลผลสัญญาณเชิงเลข

3.3.1 ตัวกรองเพื่อป้องกันเอเลียส (Anti-alias Filter)

เป็นตัวกรองเชิงอุปมานใช้เพื่อตัดสัญญาณขาเข้าที่มีความถี่สูงกว่าครึ่งหนึ่งของความถี่ของการสุ่มเพื่อป้องกันการเกิดความเพี้ยนเอเลียส (Aliasing) จากทฤษฏีการสุ่มที่กล่าวว่าความถี่ของการสุ่มต้องเท่ากับสองเท่าหรือมากกว่าสองเท่าความถี่สูงสุดของสัญญาณขาเข้าที่เราสนใจ ดังนั้นความถี่สูงกว่าครึ่งหนึ่งของความถี่ของการสุ่มถือว่าเป็นสัญญาณรบกวนของระบบ ในการเลือกใช้งานตัวกรองมีคุณสมบัติหลายอย่างที่จะต้องพิจารณารวมไปถึงงานที่จะนำไปประยุกต์ใช้ว่าสนใจสัญญาณในโดเมนความถี่หรือเวลาเป็นหลัก ข้อมูลคุณสมบัติของตัวกรองที่นิยมใช้ในการป้องกันเอเลียสแสดงดังรูปที่ 3.9

	Voltage gain at DC	Step Response			Frequency Response		
		Overshoot	Time to settle to 1%	Time to settle to 0.1%	Ripple in passband	Frequency for x100 attenuation	Frequency for x1000 attenuation
Bessel							
2 pole	1.27	0.4%	0.60	1.12	0%	12.74	40.4
4 pole	1.91	0.9%	0.66	1.20	0%	4.74	8.45
6 pole	2.87	0.7%	0.74	1.18	0%	3.65	5.43
8 pole	4.32	0.4%	0.80	1.16	0%	3.35	4.53
Butterworth							
2 pole	1.59	4.3%	1.06	1.66	0%	10.0	31.6
4 pole	2.58	10.9%	1.68	2.74	0%	3.17	5.62
6 pole	4.21	14.3%	2.74	3.92	0%	2.16	3.17
8 pole	6.84	16.4%	3.50	5.12	0%	1.78	2.38
Chebyshev							
2 pole	1.84	10.8%	1.10	1.62	6%	12.33	38.9
4 pole	4.21	18.2%	3.04	5.42	6%	2.59	4.47
6 pole	10.71	21.3%	5.86	10.4	6%	1.63	2.26
8 pole	28.58	23.0%	8.34	16.4	6%	1.34	1.66

รูปที่ 3.9 แสดงข้อมูลคุณสมบัติของตัวกรองที่นิยมใช้ในการป้องกันเอเลียส

3.3.2 ตัวแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข (Analog to Digital Converter: ADC)

ทำหน้าที่ในการแปลงสัญญาณเชิงอุปมานให้เป็นสัญญาณเชิงเลข ปัจจุบันได้รวมวงจรสุ่มและคงค่า (Sample and Hold) เอาไว้ภายใน ในการเลือกใช้มีสิ่งที่จะต้องพิจารณาหลายอย่างเช่น สถาปัตยกรรม, ความละเอียด (Resolution), ความถี่สุ่มสูงสุดหรือเวลาที่ใช้ในการแปลง (Conversion Time) ฯลฯ เป็นต้น ค่าต่างๆดังกล่าวมีผลทำให้คุณสมบัติทางสแตติค (Static) และไดนามิค (Dynamic) ของ ADC เกิดความแตกต่างกันด้วย [12]

3.3.3 ตัวประมวลผลสัญญาณเชิงเลข (Digital Signal Processor)

การประมวลผลสัญญาณเชิงเลขเป็นการคำนวณทางคณิตศาสตร์ซึ่งมีความซับซ้อนมากถ้าจะเปรียบเทียบกับการประมวลผลทั่วๆไปเช่น การประมวลผลคำ (Word Processing) ฯลฯ เป็นต้น นั่นหมายถึงว่าตัวประมวลผลที่ถูกออกแบบมาเพื่อใช้งานโดยทั่วไปไม่สามารถทำงานได้ประสิทธิภาพสูงสุดกับขั้นตอนวิธี (Algorithm) ของการประมวลผลสัญญาณเชิงเลขเช่น ตัวกรองเชิงเลขหรือการคำนวณดีสครีตฟูริเยร์ทรานสฟอร์ม ฯลฯ เป็นต้น ตัวประมวลผลสัญญาณเชิงเลขก็คือตัวประมวลผลที่ถูกออกแบบมาเพื่องานด้านการประมวลผลสัญญาณเชิงเลขโดยเฉพาะ ดังนั้นมันจะสามารถทำงานได้ผลเร็วกว่าตัวประมวลผลที่ถูกออกแบบมาเพื่อใช้งานทั่วๆไปเมื่อใช้ความถี่สัญญาณนาฬิกาเท่ากัน รายละเอียดของความแตกต่างทางด้านสถาปัตยกรรมสามารถศึกษาเพิ่มเติมได้จาก [11]

3.3.4 ระบบเลขฐานสองที่ใช้ในการประมวลผล

เนื่องจากตัวประมวลผลถูกสร้างจากวงจรเชิงเลขการเก็บค่าต่างๆจะใช้ระบบเลขฐานสอง แต่ในความเป็นจริงตัวเลขที่ใช้ในชีวิตประจำวันเป็นเลขจำนวนจริงที่ประกอบด้วยเลขหลังจุดทศนิยมจำนวนหลายตำแหน่ง เช่น π = 3.14159265358979323846264338327952 ดังนั้นจึงได้มีการคิดระบบตัวเลขในการแทนเลขจำนวนจริงด้วยเลขฐานสองที่มีจำนวนบิตจำกัดเพื่อใช้ในการประมวลผล สำหรับตัวประมวลผลซึ่งมีขีดความสามารถจำกัดอันเนื่องมาจากสถาปัตยกรรมภายในที่ถูกออกแบบมา เช่น ขีดจำกัดในการเก็บข้อมูลของรีจีสเตอร์หรือขีดจำกัดในการประมวลผลของหน่วยคำนวณ (Arithmetic Logic Unit: ALU) ดังนั้นในการออกแบบระบบการประมวลผลสิ่งสำคัญที่ต้องคำนึงถึงก็คือการเลือกว่าจะใช้ระบบตัวเลขแบบไหนจึงจะเหมาะสมกับงานนั้นๆ ระบบตัวเลขสำหรับงานประมวลผลแบ่งออกเป็น 2 ประเภทใหญ่ๆได้แก่

3.3.4.1 เลขฐานสองแบบจำนวนโดยตรง (Fixed Point)

ตัวเลขฐานสองแบบจำนวนโดยตรงสามารถแบ่งย่อยได้อีกหลายชนิดแสดงดังรูปที่ 3.10 ซึ่งแต่ละชนิดมีหลักการคล้ายๆกันดังนั้นจะอธิบายเฉพาะส่วนที่เกี่ยวข้องกับงานวิจัยนี้เท่านั้น

- Two's Complement Sign Fractional เป็นระบบตัวเลขที่นิยมใช้มากในระบบการประมวลผลสัญญาณเชิงเลขเพราะสามารถแทนค่าได้ทั้งจำนวนเต็มและแบบมีเศษทั้งบวกและลบ [13] ค่าสูงสุดและต่ำสุดที่สามารถแทนได้แสดงดังสมการที่ (3.10) – (3.11)

$$MinValue = -2^{(N-1)} \tag{3.10}$$

$$MaxValue = 2^{(N-1)} - 2^{-M} \tag{3.11}$$

เมื่อ

$MinValue$ และ $MaxValue$ คือค่าต่ำสุดและสูงสุดที่สามารถแทนได้

N คือ จำนวนบิตของเลขฐานสองก่อนจุดทศนิยม

M คือ จำนวนบิตของเลขฐานสองหลังจุดทศนิยม

ยกตัวอย่างในกรณีที่ระบบมีขนาด 8 บิต ให้ $N = 1$ และ $M = 7$ ดังนั้นจะสามารถแทนค่าได้ตั้งแต่ -1 ถึง 0.9921875 โดยค่าละเอียดที่สุดที่สามารถแทนได้คือ 2^{-M} ซึ่งเท่ากับ 0.0078125 ตัวอย่างต่อไปแสดงการบวกเลขสองจำนวนระหว่าง -36.25 และ 13.90625 โดยกำหนด $N = 9$ และ $M = 5$ ดังนี้

36.25 = 000100100.01000 โดยที่เลขหลังจุดคำนวณได้จาก $Round(.25 * 2^{M})$

เมื่อ

Round คือฟังก์ชั่นการปัดเศษ (น้อยกว่า 0.5 ตัดทิ้งแต่ถ้า เท่ากับหรือมากกว่า 0.5 บวกเพิ่ม 1)

-36.25 = 111011011.11000 จากบรรทัดแรกทำเป็นจำนวนลบโดยทำ 2's Complement

13.90625 = 000001101.11101

111011011.11000 + 000001101.11101 = 111101001.10101 คำตอบบิต MSB เป็น 1 แสดงว่าค่าเป็นลบ

พิสูจน์โดยการแปลงกลับเพื่อดูขนาด

-36.25 + 13.90625 = -22.34375

111101001.10101 = 10110.01011

10110 = 22, 01011 = 11, $11/2^5$ = 0.34375

ดังนั้นคำตอบที่ได้คือ -22.34375 เป็นคำตอบที่ถูกต้อง

System	Number Range	Advantages	Disadvantages
Unsigned Integer	0 to $2^N - 1$	Universal numbering system. Easy to perform arithmetic operations such as addition and subtraction.	Cannot store negative numbers.
Two's Complement Integer	$-2^{(N-1)}$ to $2^{(N-1)} - 1$	Stores both positive and negative numbers. Easy to perform arithmetic with regular adders.	Requires one extra bit of storage space when only positive numbers are necessary.
Unsigned Fractional	0 to $2^N - 2^M$	Stores positive numbers greater than and less than 1. Operations are identical to unsigned integer operations.	Cannot store negative numbers.
Two's Complement Signed Fractional	$-2^{(N-1)}$ to $2^{(N-1)} - 2^{-M}$ in 2^{-M} steps	Stores positive and negative numbers both greater than and less than 1. Operations are identical to two's complement operations.	–
Gray Code	0 to $2^{(N-1)}$	Only one bit changes between adjacent numbers, which facilitates interfaces with physical systems.	Difficult to perform arithmetic operations without first converting to one of the systems listed above.
Signed-Magnitude	$-2^{(N-1)} - 1$ to $2^{(N-1)} - 1$	Useful for applications that require the magnitude to be distinct from the sign.	Difficult to perform arithmetic operations (although easier than with Gray code).
Offset Two's Complement	$-2^{(N-1)}$ to $2^{(N-1)} - 1$	Used by many analog-to-digital (A/D) and digital-to-analog (D/A) converters. Easy to perform arithmetic operations.	–
One's Complement	$-2^{(N-1)} - 1$ to $2^{(N-1)} - 1$	Easy to perform negations.	Difficult to perform arithmetic operations other than negations.

รูปที่ 3.10 แสดงตัวเลขฐานสองแบบจำนวนโดยตรงชนิดต่างๆ

- Offset Two’s Complement เป็นระบบตัวเลขที่ใช้กับไอซีประเภท ADC หรือ DAC มีรูปแบบคล้ายกับ 2’Complement รูปที่ 3.11 แสดงการเปรียบค่าระว่าง Two’s Complement และ Offset Two’s Complement จะสังเกตเห็นว่าการแปลงจาก Two’s Complement เป็น Offset Two’s Complement สามารถทำได้โดยการกลับ MSB นั่นเอง

Decimal Value	Two’s Complement	Offset Two’s Complement
–4	100	000
–3	101	001
–2	110	010
–1	111	011
0	000	100
1	001	101
2	010	110
3	011	111

รูปที่ 3.11 แสดงการเปรียบค่าระหว่างตัวเลขแบบ Two’s Complement และ Offset Two’s Complement

3.3.4.2 เลขฐานสองแบบจำนวนอิงดัชนี (Floating Point)

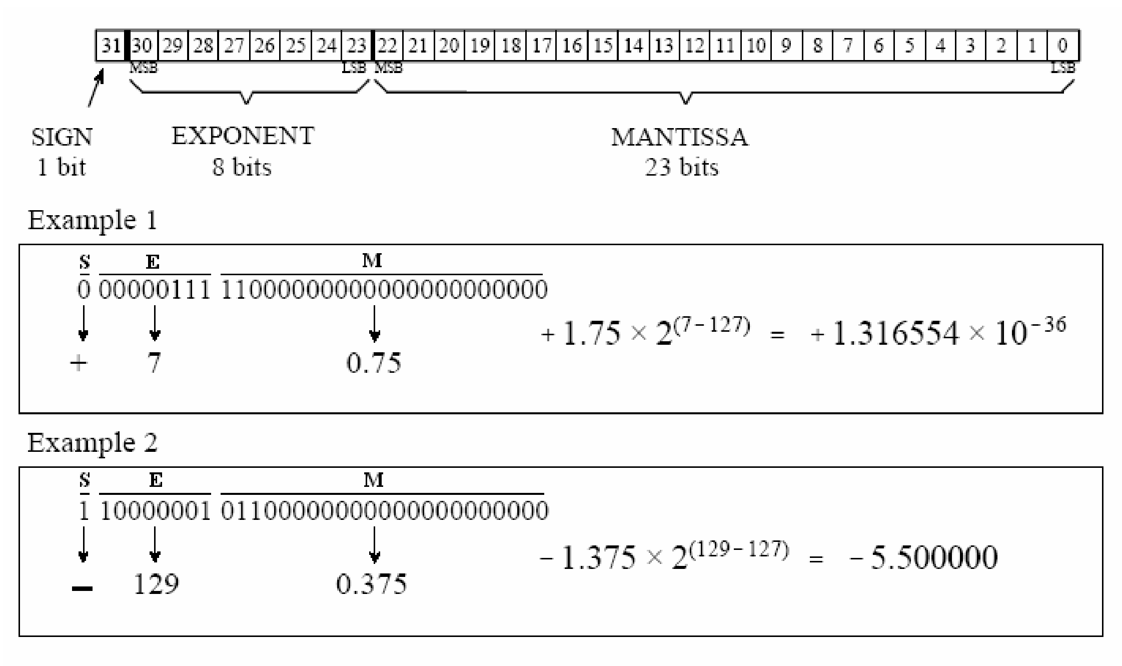
การเขียนตัวเลขแบบจำนวนอิงดัชนีตัวเลขจะถูกเขียนโดยการใช้เลขฐานสองแบบจำนวนโดยตรงสองส่วนคือ ส่วนแรกเป็นค่า แมนทีสซา (Mantissa) และส่วนที่สอง ส่วนชี้กำลัง (Exponent) รูปแบบของการเขียนตัวเลขแบบจำนวนอิงดัชนี แสดงดังสมการที่ (3.12) ตามมาตรฐาน ANSI/IEEE Std. 754-1985 จะเรียกว่า Single Precision เมื่อใช้เลขฐานสองจำนวน 32 บิตในการแสดงค่า และเรียกระบบ 64 บิตว่า Double Precision

$$Value = (-1)^{S} * M * 2^{E-127} \qquad (3.12)$$

สำหรับระบบ 32 บิต พจน์ $(-1)^{S}$ นี้ใช้เลขฐานสองจำนวน 1 บิตเป็นตัวแสดงเครื่องหมายนั่นคือ ถ้า S = 0 เป็นค่าบวกแต่ถ้า S = 1 เป็นค่าลบ ตัวแปร E มีค่าอยู่ในช่วง 0 ถึง 255 ใช้เลขฐานสองจำนวน 8 บิต จากสมการจะเห็นว่ามีการลบค่านี้ด้วย 127 เพื่อให้ส่วนชี้กำลังแสดงค่าได้ในช่วง 2^{-127} ถึง 2^{128} และค่าแมนทีสซา M ใช้เลขฐานสองจำนวน 23 บิต ด้วยวิธีการนี้ค่าสูงสุดที่สามารถแทนได้เท่ากับ

$\pm(2-2^{-23})*2^{128} = \pm 6.8*10^{38}$ และค่าต่ำสุดที่สามารถแทนได้เท่ากับ $\pm 1.0*2^{-127} = \pm 5.9*10^{-39}$ ตัวอย่างการแปลงแสดงดังรูปที่ 3.12

จากสมการที่ (3.10) จะพบว่าถ้านำตัวเลขแบบจำนวนอิงดัชนีไปเขียนโปรแกรมหรือสร้างฮาร์ดแวร์ในการประมวลผลจะทำได้ยุ่งยากมาก ยกตัวอย่างเช่น การบวก ตัวเลขที่จะบวกกันได้ส่วนชี้กำลังต้องเท่ากัน ถ้าส่วนชี้กำลังไม่เท่ากันจะต้องทำการเลื่อนจุดทวินิยม (Binary Point) ของตัวเลขที่มีค่าส่วนชี้กำลังน้อยกว่าไปทางขวาโดยเลื่อนไปเท่ากับจำนวนผลต่างของส่วนชี้กำลัง ส่วนการคูณกันของตัวเลขแบบจำนวนอิงดัชนีทำได้โดยนำเอาส่วนแมนทีสซามาคูณกันและนำส่วนชี้กำลังมาบวกกัน ข้อดีของการใช้ตัวเลขแบบจำนวนอิงดัชนีคือ สามารถเขียนแทนค่าของสัญญาณได้ละเอียดและแม่นยำกว่าแบบจำนวนโดยตรง แต่อย่างไรก็ตามฮาร์ดแวร์ที่ใช้ในการคำนวณยุ่งยากซับซ้อนและแพงกว่าแบบจำนวนโดยตรง นอกจากนั้นมันยังต้องการเวลาในการประมวลผลที่มากกว่า ดังนั้นสำหรับการประมวลผลแบบระบบเวลาจริงจึงนิยมใช้ระบบตัวเลขแบบจำนวนโดยตรงมากกว่า รวมไปถึงงานวิจัยที่กำลังนำเสนอก็ได้เลือกใช้ระบบตัวเลขแบบจำนวนโดยตรงเช่นกัน



รูปที่ 3.12 แสดงวิธีการแทนค่าด้วยตัวเลขแบบจำนวนอิงดัชนี

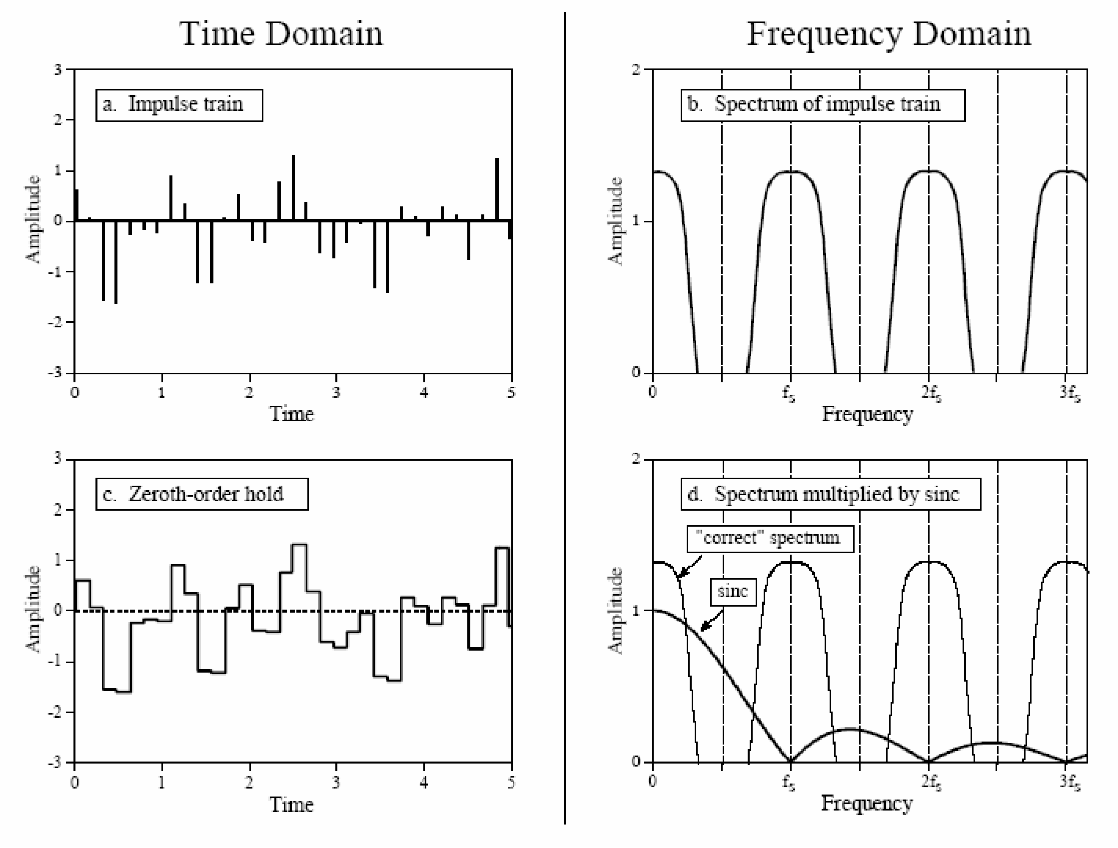
3.3.5 ตัวแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมาน (Digital to Analog Converter: DAC)

หลังการประมวลผลสัญญาณผลลัพธ์ที่ได้จะอยู่ในรูปสัญญาณเชิงเลขเป็นข้อมูลจุดซึ่งเป็นสัญญาณชั่วขณะของสัญญาณ แต่สัญญาณขาออกที่ต้องการจากระบบคือสัญญาณเชิงอุปมาน ดังนั้นจึง

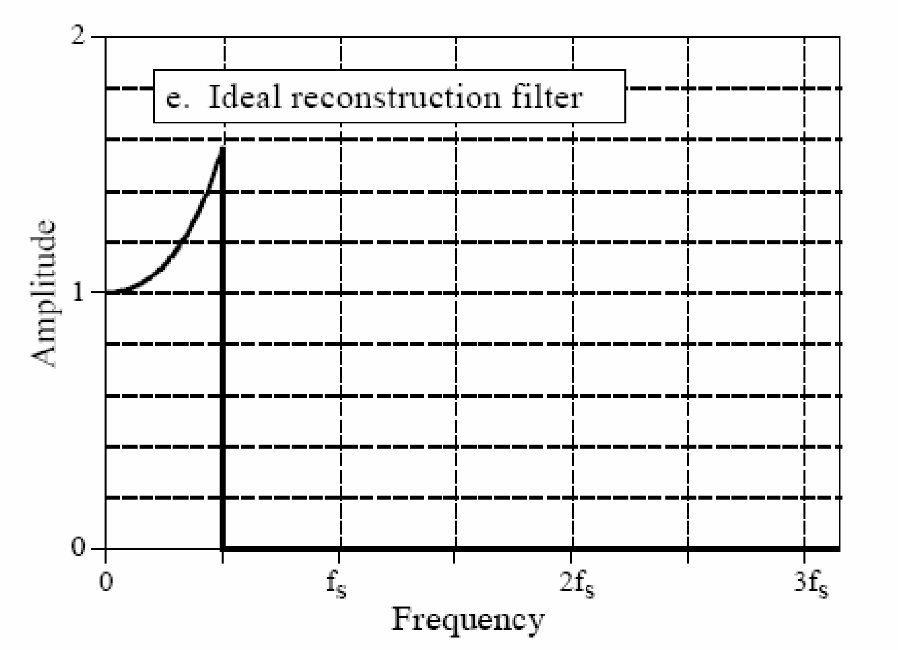
ต้องมีการแปลงสัญญาณจากเชิงเลขไปเป็นเชิงอุปมานโดย ตัวแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมาน

3.3.6 ตัวกรองสร้างสัญญาณคืน (Reconstruction Filter)

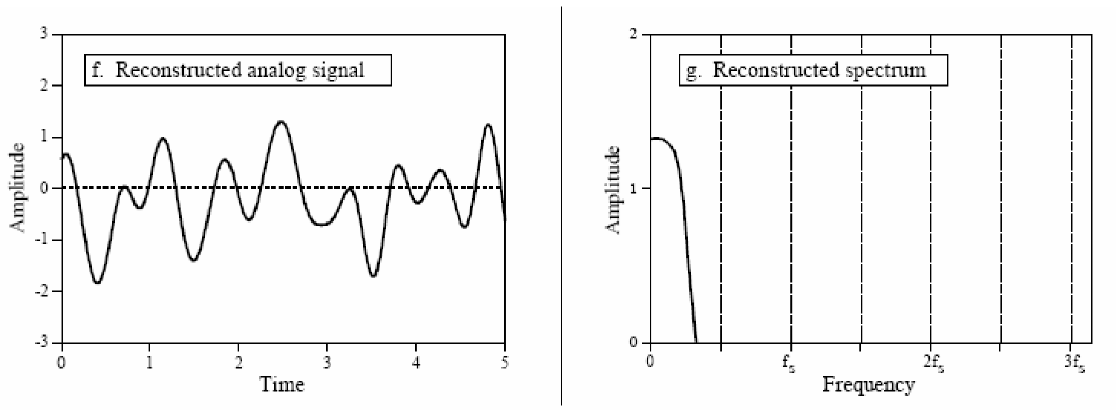
สัญญาณที่ได้จากตัวแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมานเป็นสัญญาณชั่วขณะหรือเป็นสัญญาณที่ถูกสุ่ม ดังนั้นมันจะมีองค์ประกอบทางความถี่แสดงดังรูปที่ 3.2 ในกรณีที่ไม่มีการคงค่าข้อมูล (Hold) ดังนั้นการที่จะทำให้ได้สัญญาณที่ถูกต้องจะต้องมีการกรองสัญญาณชั่วขณะดังกล่าวด้วยตัวกรองความถี่ต่ำผ่านเพื่อตัดองค์ประกอบของความถี่ที่มากกว่าครึ่งหนึ่งของความถี่สุ่มทิ้งไป ตัวกรองชนิดนี้เรียกว่า ตัวกรองสร้างสัญญาณคืน (Reconstruction Filter) แต่ในความเป็นจริงสัญญาณที่ได้จากตัวแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมานจะถูกคงค่าดังนั้นสเปกตรัมของสัญญาณจะมีความแตกต่างไปจากที่แสดงดังรูปที่ 3.2 เนื่องจากมันถูกคูณด้วยซิงค์ฟังก์ชั่น [11] ดังนั้นในการออกแบบตัวกรองจะต้องพิจารณาแยกเป็นแต่ละกรณีไป



รูปที่ 3.13 แสดงสัญญาณในโดเมนเวลาและความถี่ของสัญญาณก่อนและหลังการคงค่า



รูปที่ 3.14 แสดงผลตอบสนองความถี่ของตัวกรองสร้างสัญญาณคืนในอุดมคติ

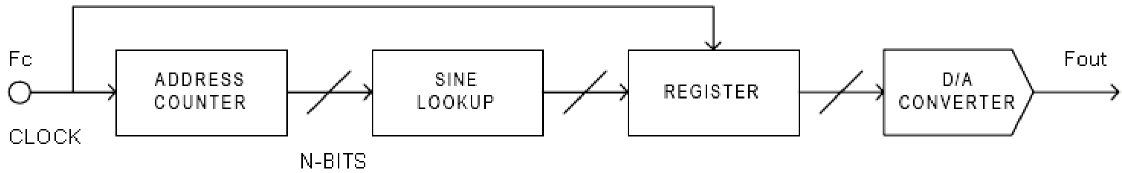


รูปที่ 3.15 แสดงสัญญาณในโดเมนเวลาและความถี่ของสัญญาณที่ได้จากตัวกรองสร้างสัญญาณคืนในอุดมคติ

3.4 Direct Digital Synthesis (DDS)

บางส่วนการทำงานของตัวเข้ารหัสฯที่กำลังนำเสนอต้องการสัญญาณรูปไซน์เพื่อการประมวลผลบางอย่าง เช่น เป็นพาหะรองของการมอดูเลตสัญญาณความแตกต่างสีหรือใช้ในการสังเคราะห์สัญญาณเบิรสต์ เทคนิคที่นำมาใช้ในการสังเคราะห์สัญญาณรูปคลื่นไซน์ได้แก่ ไดเร็คดิจิตอล

ซินทีซีส (Direct Digital Synthesis: DDS) [14], [15] ซึ่งเป็นวิธีการสร้างสัญญาณเชิงอุปมานด้วยวิธีการของระบบเชิงเลข ระบบดีดีแอสอย่างง่ายแสดงดังรูปที่ 3.16

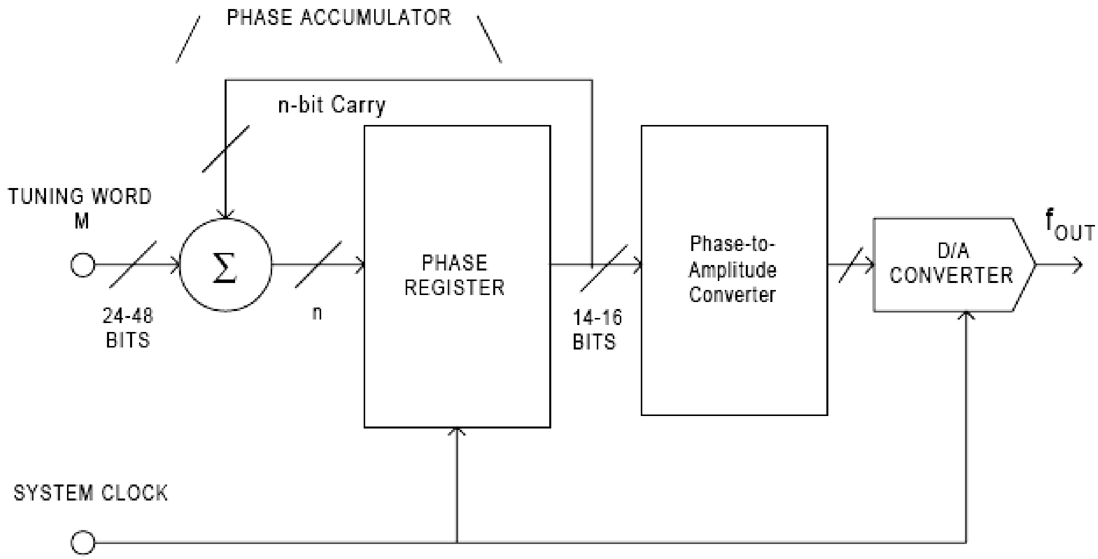


รูปที่ 3.16 แสดงระบบดีดีเอสอย่างง่าย

จากรูปที่ 3.16 ข้อมูลขนาดของสัญญาณไซน์หนึ่งไซเคิล (Cycle) จะถูกเก็บไว้ในหน่วยความจำพีรอม (PROM) ด้วยเหตุนี้ทำให้มันทำหน้าที่เป็นตารางเปิดดูของสัญญาณไซน์ (Sine Lookup Table: Sine LUT) ในการสร้างสัญญาณ ค่าเฟสของสัญญาณจะถูกสร้างจากวงจรนับตำแหน่ง (Address Counter) และส่งค่าตำแหน่งให้กับพีรอมเพื่อเปลี่ยนค่าเฟสให้เป็นค่าขนาดเชิงเลขของสัญญาณไซน์และส่งต่อให้รีจิสเตอร์ (Register) เพื่อทำการซิงโครไนส์ ขั้นตอนสุดท้ายของระบบคือแปลงสัญญาณเชิงเลขให้เป็นสัญญาณเชิงอุปมาน จากระบบอย่างง่ายนี้ ค่าความถี่ของสัญญาณขาออกที่ได้ขึ้นอยู่กับ 2 ค่าได้แก่ ค่าความถี่ของสัญญาณนาฬิกาอ้างอิง (Reference Clock) และค่าช่วงการเปลี่ยนขนาดของสัญญาณไซน์ที่เก็บในพีรอม ดังนั้นการเปลี่ยนค่าความถี่สามารถทำได้ 2 วิธีเช่นกันโดย วิธีแรกเปลี่ยนค่าความถี่ของสัญญาณนาฬิกาอ้างอิงหรือวิธีที่สองเขียนค่าพีรอมใหม่ เนื่องจากทั้งสองวิธีนี้ไม่สนับสนุนฟังก์ชั่นการย้ายความถี่อย่างเร็ว (High-speed Frequency Hopping) ของสัญญาณขาออก จึงได้มีการปรับปรุงระบบโดยการเพิ่มเฟสแอคคิวมูเลเตอร์ (Phase Accumulator) ไปแทนที่วงจรนับตำแหน่ง สถาปัตยกรรมแบบนี้เรียกว่า Numerical Control Oscillator (NCO) ซึ่งเป็นส่วนที่สำคัญมากในระบบดีดีเอสและในวิทยานิพนธ์นี้ก็ได้นำสถาปัตยกรรมแบบนี้มาประยุกต์ใช้งาน ระบบดีดีเอสที่ได้รับการปรับปรุงให้มีความสามารถในการปรับความถี่ได้ (Frequency Tunable) แสดงดังรูปที่ 3.17

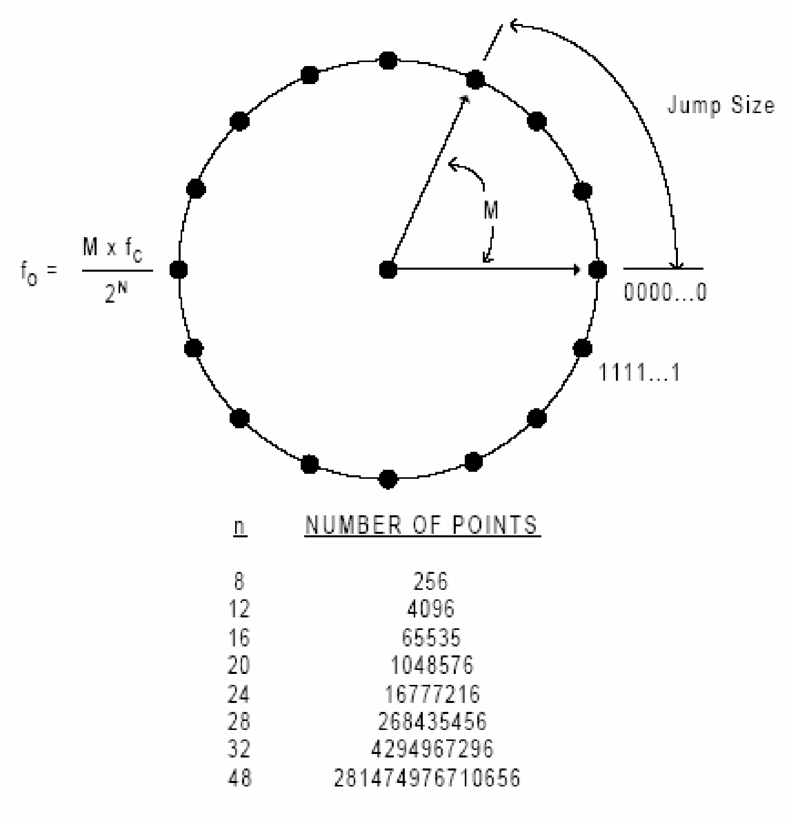
หน้าที่ของส่วนเฟสแอคคิวมูเลเตอร์เปรียบได้ว่าเป็นวงล้อของเฟส (Phase Wheel) ในระบบดีดีเอส เพื่อทำความเข้าใจกับฟังก์ชั่นการทำงานจะสมมติให้สัญญาณไซน์เกิดจากการหมุนของเวคเตอร์รอบวงกลมของเฟส (ดูรูที่ 3.18 ประกอบ) แต่ละจุดบนวงกลมของเฟสก็คือจุดที่เป็นขนาดเชิงเลขของสัญญาณไซน์ หนึ่งรอบการหมุนของเวคเตอร์ด้วยความเร็วคงที่จะทำให้ได้สัญญาณไซน์หนึ่งลูก (Cycle) เฟสแอคคิวมูเลเตอร์ถูกใช้งานให้เสมือนว่าเป็นเวคเตอร์ที่หมุนอยู่รอบวงล้อเฟสอย่างเชิงเส้นแต่สัญญาณขาออกของมันไม่สามารถนำไปใช้ผลิตสัญญาณได้โดยตรงเนื่องจากสัญญาณที่จุดนี้เกิดจาก

การบวกเพิ่มมันจึงได้สัญญาณอยู่ในรูปแรมป์ (Ramp) ที่ข้อมูลถูกตัดปลาย (Truncate) แต่จะใช้เพื่อเป็นตัวชี้ (Pointer) ให้กับพีรอมที่เก็บขนาดเชิงเลขของสัญญาณไซน์

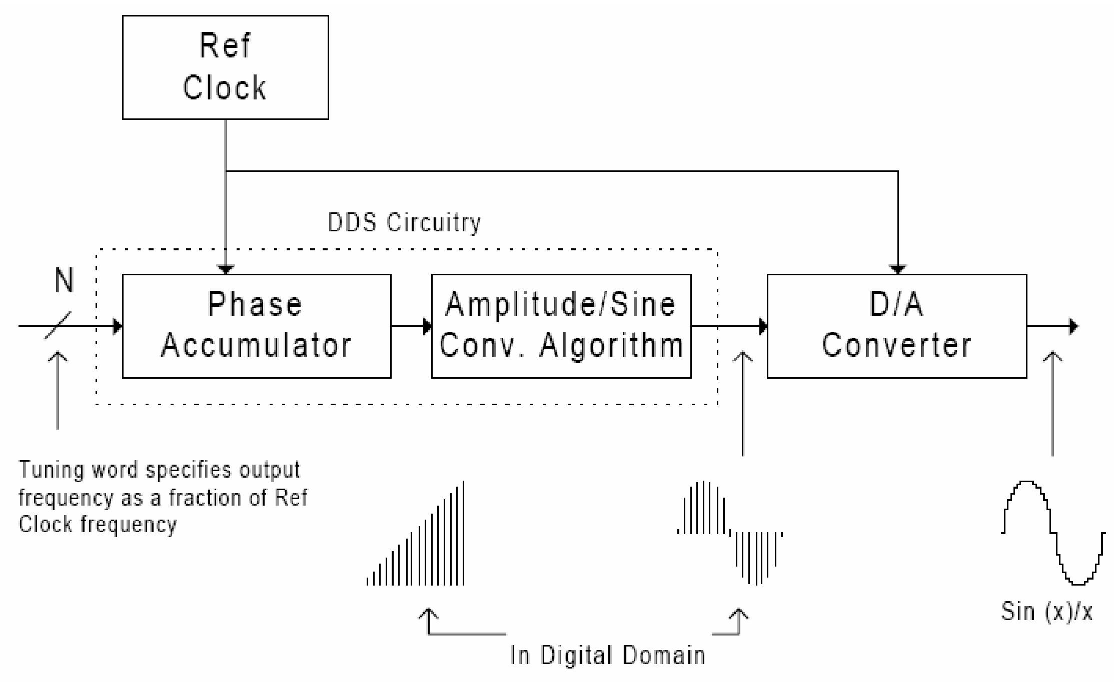


รูปที่ 3.17 แสดงระบบดีดีเอสที่สามารถปรับความถี่ได้

ในความเป็นจริงเฟสแอคคิวมูเลเตอร์ก็คือตัวนับแบบโมดูลัส M (Modulus) ซึ่งมันจะบวกค่าเพิ่มจากค่าที่เก็บไว้ทุกๆคาบของสัญญาณนาฬิกาอ้างอิง ค่าของการบวกเพิ่มกำหนดโดยค่า M ซึ่งเป็นตัวกำหนดค่าการกระโดดของเวคเตอร์ในแต่ละคาบของสัญญาณนาฬิกาหรืออีกนัยหนึ่งก็คือมันเป็นค่าที่บอกจำนวนที่ถูกข้ามในวงล้อเฟส ค่าการกระโดดที่มากจะทำให้การโอเวอร์โฟล์วเกิดเร็วขึ้นหรือใช้เวลาต่อรอบน้อยผลคือได้สัญญาณขาออกไซน์ที่มีความถี่สูง ยกตัวอย่างเช่นสมมติให้เฟสแอคคิวมูเลเตอร์มีขนาด N = 32 บิต และให้ค่า M = 1 กรณีนี้การเกิดโอเวอร์โฟล์วจะต้องใช้เวลาเท่ากับ 2^{32} ลูกของสัญญาณนาฬิกา แต่ถ้าใช้ค่า M = $2^{31}-1$ จะใช้เวลาเพียงสองลูกของสัญญาณนาฬิกาเท่านั้นก็จะทำให้เกิดโอเวอร์โฟล์ว ค่าของการกระโดดนี้เองเป็นตัวบอกถึงความละเอียดของการปรับความถี่ของสถาปัตยกรรมดีดีเอส



รูปที่ 3.18 แสดงวงล้อเฟสเชิงเลข (Digital Phase Wheel)



รูปที่ 3.19 แสดงสัญญาณที่จุดต่างๆของระบบดีดีเอส

การหาค่าความถี่ที่ได้จากระบบดีดีเอสอาศัยความสัมพันธ์ของเฟสแอคคิวมูเลเตอร์และค่า M ดังนี้

$$Fout = \frac{M * REFCLOCK}{2^N} \tag{3.13}$$

โดยที่

Fout คือ ความถี่สัญญาณขาออกที่ได้จากระบบดีดีเอส

M คือ ค่าของการควบคุมการปรับ (The binary tuning word)

REFCLOCK คือ ความถี่ของสัญญาณนาฬิกาอ้างอิง

N คือ จำนวนบิตของเฟสแอคคิวมูเลเตอร์