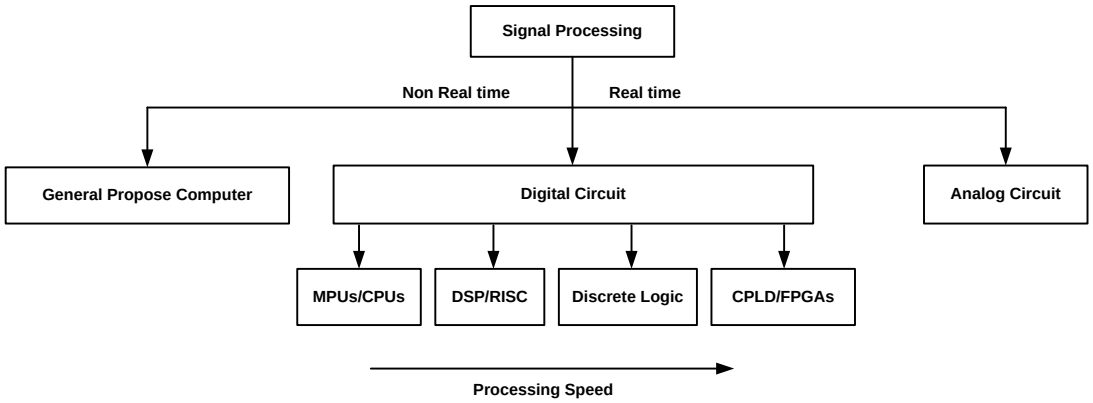
บทที่ 2
การประมวลผลสัญญาณ (Signal Processing)

2.1 บทนำ

การประมวลผลสัญญาณ หมายถึง การกระทำบางอย่างกับสัญญาณที่มีผลให้เกิด การปรับปรุงหรือมีการเปลี่ยนแปลงรูปร่างไปด้วย กระบวนการขยาย (Amplification) การลดทอน (Attenuation) และการกรอง (Filtering) เป็นต้น โดยผ่านการควบคุมตัวแปรของระบบ (system) ที่อยู่ในรูปของฟังก์ชันถ่ายโอน (transfer function) โดยอาศัยพื้นฐานของการรวม การคูณและการหน่วงสัญญาณ โดยสามารถแบ่งการประมวลผลออกเป็น 2 ประเภทได้แก่ การประมวลผลสัญญาณอนาลอก (Analog Signal Processing) และการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing) ดังแสดงไว้ในรูปที่ 2.1



รูปที่ 2.1 การประมวลผลสัญญาณแบบอนาลอก และการประมวลผลเชิงเลข

2.2 ข้อดีและข้อเสียของการประมวลผลเชิงเลข

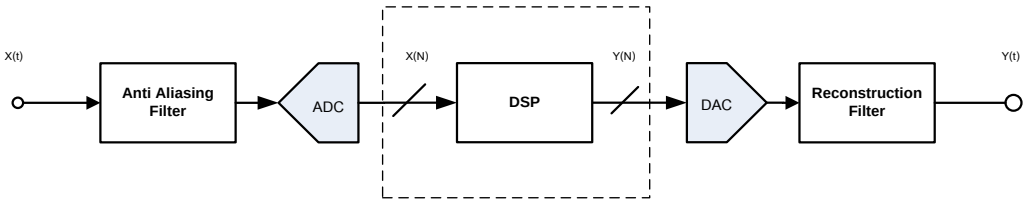
รูปแบบการประมวลผลสัญญาณอนาลอกและการประมวลผลสัญญาณเชิงเลขต่างก็มีคุณสมบัติเฉพาะที่แตกต่างกันออกไป ซึ่งสามารถเลือกพิจารณาการใช้งานระบบใด ๆ ได้ตามความเหมาะสม ซึ่งสามารถแบ่งข้อดี ข้อเสียของการประมวลผลเชิงเลขได้ดังนี้

ตารางที่ 2.1 ข้อดีข้อเสียของการประมวลผลเชิงเลข

ข้อดีของการประมวลผลเชิงเลข	ข้อเสียของการประมวลผลเชิงเลข
1. การประมวลผลเหมาะสมกับข้อมูลที่อยู่ในรูปแบบเชิงเลขเหมือนกันเช่น ภาพดิจิตอล 2. อุปกรณ์ทางด้านดิจิตอลมีแนวโน้มราคาต่ำลง ขนาดเล็กลง ประสิทธิภาพและความเร็วที่สูงขึ้น ความเที่ยงตรงเพิ่มมากขึ้น 3. สามารถทำการรับส่งข้อมูลได้แม่นยำกว่าสัญญาณอนาลอก เนื่องจากสัญญาณเชิงเลขมีสถานะเพียงศูนย์ “0” และหนึ่ง “1” เท่านั้น 4. การประมวลผลเชิงเลขสามารถทำได้โดยง่ายเพราะอัลกอริทึม (Algorithm) มีเพียงการบวก การลบ การคูณ และการหน่วงเวลา 5. ในการประมวลผลเชิงเลขสามารถทำได้พร้อม ๆ กันหลายช่องสัญญาณในลักษณะการแบ่งช่วงเวลาทำงาน (Time Sharing) 6. มีเสถียรภาพที่ดีต่อการเปลี่ยนแปลงของอุณหภูมิ หรือแม้กระทั่งการเปลี่ยนแปลงพารามิเตอร์ของอุปกรณ์ 7. ความแม่นยำของการควบคุมสามารถกำหนดได้จากจำนวนบิตที่ใช้ (Word Length) 8. ตัวประมวลผลเชิงเลขสามารถทำงานซ้ำหน้าที่ (Function) เดิมได้โดยไม่จำกัดจำนวนครั้ง	1. ต้องมีสัญญาณในการซิงโครไนซ์ (Synchronize) การจัดเวลา (Timing) ซึ่งจะมีผลต่อระบบการทำงานอย่างมาก 2. การเชื่อมต่อ (Interface) กับระบบการประมวลผลสัญญาณอนาลอก ทำให้วงจรมีความซับซ้อนมากขึ้น 3. การออกแบบระบบประมวลผลเชิงเลขจะมีความซับซ้อนมากกว่า โดยเฉพาะกับระบบที่มีขนาดใหญ่ 4. แถบปฏิบัติงาน ของระบบประมวลผลเชิงเลขจะต่ำกว่าระบบประมวลผลแบบอนาลอกมากเนื่องจากข้อจำกัดของอุปกรณ์ที่ใช้ เช่น วงจรเกต วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล(analog to digital converter ADC) และวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอก (digital to analog converter DAC) เป็นต้น 5. ระบบประมวลผลเชิงเลขจำเป็นต้องมีไฟเลี้ยงอยู่ตลอดเวลา 6. ในการแปลงสัญญาณอนาลอกให้อยู่ในรูปแบบสัญญาณเชิงเลขแล้วจะทำให้ความถูกต้องของสัญญาณบางส่วนขาดหายไป และไม่สามารถทำคืนให้ถูกต้องเหมือนเดิมได้ เนื่องจากผลของการกำหนดขนาดในการเก็บข้อมูล

2.3 โครงสร้างของระบบประมวลสัญญาณเชิงเลข (Structure of digital Signal Processing)

การพิจารณาถึงโครงสร้างของระบบประมวลผลสัญญาณเชิงเลขโดยแสดงองค์ประกอบทางฮาร์ดแวร์ (Hardware) เป็นหลักดังรูปที่ 2.2 จะแยกส่วนประกอบต่าง ๆ ภายในระบบได้ดังต่อไปนี้ [1]



รูปที่ 2.2 บล็อกไดอะแกรมแสดงโครงสร้างระบบ DSP ที่กระทำกับสัญญาณอนาลอก

2.3.1 ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter: ADC)

วงจรเชื่อมต่อที่ทำการแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอล ดังนั้นเพื่อให้สัญญาณอนาลอกที่จะถูกแปลงโดย ADC จะต้องสุ่มวางค่าขนาดของ แอมพลิจูด (Amplitude) ค้างไว้ในช่วงเวลาสั้น ๆ ช่วงหนึ่งจนกว่าการแปลงจะเสร็จสมบูรณ์ จึงจะทำให้ความผิดพลาดมีค่าน้อยที่สุดสามารถทำได้โดยวงจร sample & Hold (S&H) ซึ่งข้อมูลที่ได้จะถูกจัดระดับสัญญาณ (Quantized) ให้เป็นระดับอ้างอิงที่ใกล้เคียงกันที่สุด โดยระดับอ้างอิงนี้จะแบ่งเป็น 2^n-1 ระดับเมื่อ n เป็นจำนวนบิตของการเปลี่ยนแปลง หลังจากนั้นจึงถูกเข้ารหัสสัญญาณข้อมูลเลขฐานสองขนาด n บิต แล้วจึงทำการสุ่มสัญญาณที่เข้ามาใหม่อีกครั้ง เช่นนี้ไปเรื่อย ๆ

2.3.2 ตัวประมวลผลสัญญาณเชิงเลข (Digital Signal Processor: DSP)

ในส่วนนี้จะทำการประมวลผลสัญญาณเชิงเลขโดยมีรูปแบบอัลกอริทึมเฉพาะหน้าที่ ซึ่งขึ้นอยู่กับเงื่อนไขของสัญญาณทางออกว่าต้องการให้มีลักษณะเช่นไร เช่น การทำการกรองสัญญาณความถี่สูง การกรองสัญญาณรบกวน หรือการมอดูเลตสัญญาณเป็นต้น โดยรูปทรงแบบการประมวลผลเชิงเลขจะมีตัวดำเนินการ (Operator) คือ ตัวบวก (Adder) และตัวคูณ (Multiplier) ที่มักจะถูกใช้ดำเนินการกับสัญญาณเชิงเลขอยู่บ่อย ๆ

2.3.3 ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก (Digital to Analog Converter: DAC)

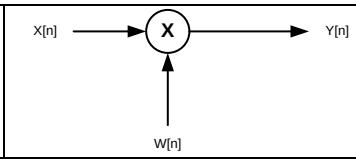
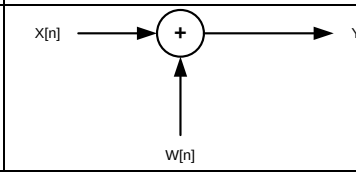
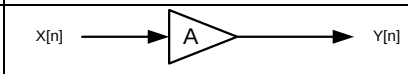
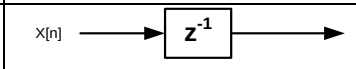
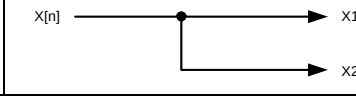
DAC ทำหน้าที่แปลงสัญญาณติจิตอลกลับมาเป็นสัญญาณอนาลอกกรณีที่เอาต์พุตของ DAC ลักษณะเหมือนขั้นบันไดอาจจำต้องทำการปรับค่าจาก DAC ให้มีสัญญาณราบเรียบขึ้นสามารถทำได้โดยวงจร Reconstruction (Smoothing) Filter เช่น first-order LPF interpolation

แต่ในการใช้งานจริงนั้นการประมวลผลทั้งหมดจะถูกจำกัดแบนด์วิดธ์การประมวลผลด้วยความถี่การสุ่มสัญญาณของ ADC ดังนั้นจึงต้องมีการจำกัดแบนด์วิดธ์ที่เข้ามาเกินความถี่สูงสุดในการประมวลผลสัญญาณจำเป็นต้องใช้วงจรกรองอนาลอก (Anti-aliasing Filter) ต่อไว้ก่อนภาค ADC

2.4 โอโปรเรชั่นของลำดับสัญญาณ (Operation on Sequence)

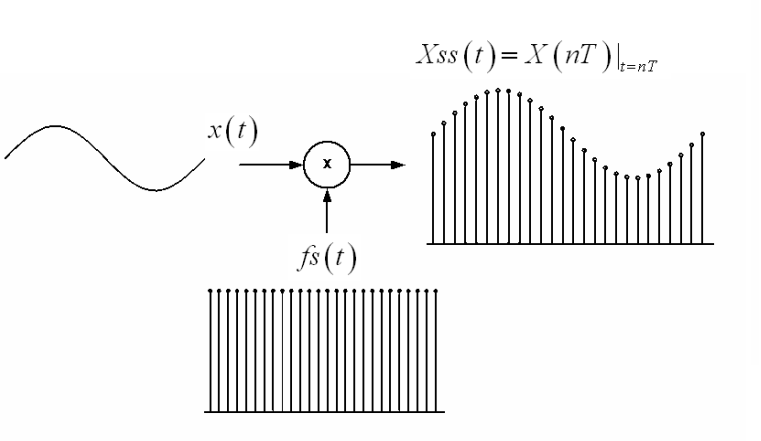
ในระบบการประมวลผลเชิงเลขหนึ่ง ๆ จะมีโอเปอร์เรชันของสัญญาณ หรือการกระทำต่อลำดับสัญญาณ ซึ่งโอเปอร์เรชันจะมีระบบการประมวลผลเชิงเลขดังนี้ [2]

ตารางที่ 2.2 การกระทำต่อลำดับสัญญาณในระบบการประมวลผลเชิงเลข

การคูณกันของสัญญาณ		$Y[n] = X[n] * W[n]$
การบวกกันของสัญญาณ		$Y[n] = X[n] + W[n]$
การคูณสัญญาณด้วยค่าคงที่		$Y[n] = A * X[n]$
การหน่วงเวลา		$Y[n] = X[n-1]$
การแตกกิ่ง		$X1[n], X2[n] = X[n]$

2.5 ทฤษฎีการสุ่มตัวอย่าง (Sampling Theory)

ในการประมวลผลสัญญาณเชิงเลข สัญญาณอินพุตจะถูกเปลี่ยนให้สัญญาณเชิงเลขโดยมีการแทนสัญญาณต่อเนื่องช่วงซึ่งในแต่ละช่วงห่างกันที่เวลา Ts คงที่ โดยหากช่วงเวลาในการสุ่มตัวอย่าง Ts มีค่าที่เหมาะสมพอแล้วนั้นการแทนค่าดังกล่าวก็จะยังคงความถูกต้องไว้ได้ ดังนั้นผลการสุ่มตัวอย่างก็คือการคูณสัญญาณต่อเนื่องด้วยสัญญาณอิมพัลซ์ที่เลื่อนไปเรื่อย ๆ โดยมีเวลาห่างกันเป็นเวลา $T = Ts = 1/fs$ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 การสุ่มสัญญาณ (Sampling Signal)

จากรูปที่ 2.3 อิมพัลซ์แต่ละตัวกำหนดให้มีความห่างเท่ากับ T วินาที โดยทั่วไปลำดับอิมพัลซ์หนึ่งหน่วยของ $fs(t)$ เขียนแทนด้วยสมการ

$$fs(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.1}$$

เนื่องจาก $fs(t)$ เป็นสัญญาณที่มีลักษณะที่มีสัญญาณเป็นคาบเพราะฉะนั้นจึงสามารถเขียนแทน $fs(t)$ ได้ด้วยอนุกรมฟูริเยอร์คือ

$$fs(t) = \sum_{n=-\infty}^{\infty} Cn \cdot e^{(j2n\pi/T)} = \sum_{n=-\infty}^{\infty} Cn \cdot e^{(jn\omega_s t)} \tag{2.2}$$

โดยที่ $\omega_s = 2\pi / T$ และ Cn คือค่าขนาดเชิงซ้อน (complex amplitude) ของสัญญาณความถี่เชิงซ้อน (complex frequency component) ω_o และ Cn หาค่าได้ตามลำดับคือ

$$Cn = \frac{1}{T} \int_{-\infty}^{\infty} fs(t) \cdot e^{(-jn\omega_s t)} dt$$

$$Cn = \frac{1}{T} \int_{-\infty}^{\infty} \delta(t) \cdot e^{(-jn\omega_s t)} dt$$

$$Cn = \frac{1}{T} \left\{ e^{(-jn\omega_s t)} \right\} \Big|_{-\infty}^{\infty}$$

$$Cn = \frac{1}{T} \tag{2.3}$$

เมื่อแทนค่า Cn จาก (2.3) กลับใน (2.2) จะได้

$$fs(t) = \frac{1}{T}\sum_{n=-\infty}^{\infty} e^{(jn\omega_s t)} \tag{2.4}$$

เมื่อทำการแปลงฟูริเยอร์เพื่อทำการหาค่าสเปกตรัมความถี่ของ $fs(t)$ จะได้

$$Fs(\omega) = \zeta\left\{fs(t)\right\} = \tfrac{1}{2\pi}\sum_{n=-\infty}^{\infty}\delta(\omega - n\omega_s) \tag{2.5}$$

ถ้าพิจารณาจากสมการ(2.4) และ (2.5) จะเห็นได้ว่า

$$\sum_{n=-\infty}^{\infty}\delta(t-nT) \rightarrow \omega_o\sum_{n=-\infty}^{\infty}\delta(\omega - n\omega_s) \tag{2.6}$$

จะเห็นได้ว่าเมื่อพิจารณาในโดเมนความถี่สเปกตรัมความถี่ของตัวสุ่มสัญญาณ $fs(t)$ เป็นอิมพัลซ์ที่วางห่างเท่า ๆ กันดังแสดงในรูปที่ 2.4 (b)

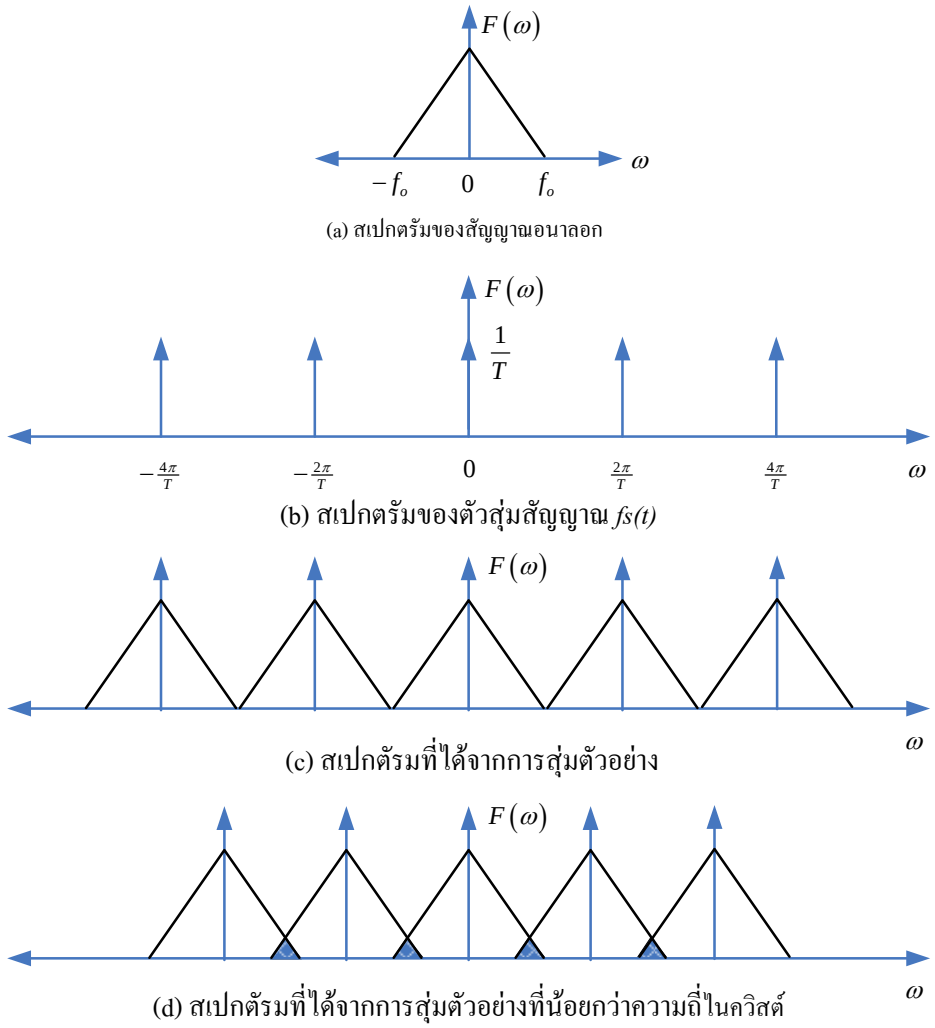
ถ้าให้ $Xss(t)$ เป็นสัญญาณที่ได้จากการสุ่มตัวอย่างดังนั้น

$$Xss(t) = fs(t)\cdot x(t) \tag{2.7}$$

และถ้าให้ $X(\omega)$ เป็นสเปกตรัมความถี่ของ $x(t)$ เนื่องจากในโดเมนเวลาสัญญาณที่ได้จากการสุ่มตัวอย่างเป็น การคูณกันของสองสัญญาณดังนั้นในโดเมนความถี่จึงเป็นการคอนโวลูชันกันระหว่างสเปกตรัมของ $fs(t)$ และ $x(t)$ หรือ

$$Xss(\omega) = Fs(\omega) * X(\omega) \tag{2.8}$$

โดยที่สัญลักษณ์ $*$ แทนการคอนโวลูชันผลที่ได้แสดงดังรูป 2.4(c) ซึ่งจะเห็นได้ว่าจากกราฟสัญญาณอนาลอกที่มีแถบความถี่สมมุติให้มีค่าเท่ากับ $2\omega_o$ เมื่อสัญญาณนั้นถูกสุ่มตัวอย่าง $fs(t)$ ผลลัพธ์ที่ได้จะเห็นว่าสเปกตรัมของสัญญาณที่ได้จากการสุ่มตัวอย่าง $Xss(\omega)$ เป็นการนำเอาสเปกตรัมของสัญญาณ $X(\omega)$ มาวางเรียงห่างเท่ากันไปตลอดแกนความถี่ ω



(a) สเปกตรัมของสัญญาณอนาลอก

(b) สเปกตรัมของตัวสุ่มสัญญาณ *fs(t)*

(c) สเปกตรัมที่ได้จากการสุ่มตัวอย่าง

(d) สเปกตรัมที่ได้จากการสุ่มตัวอย่างที่น้อยกว่าความถี่ไนควิสต์

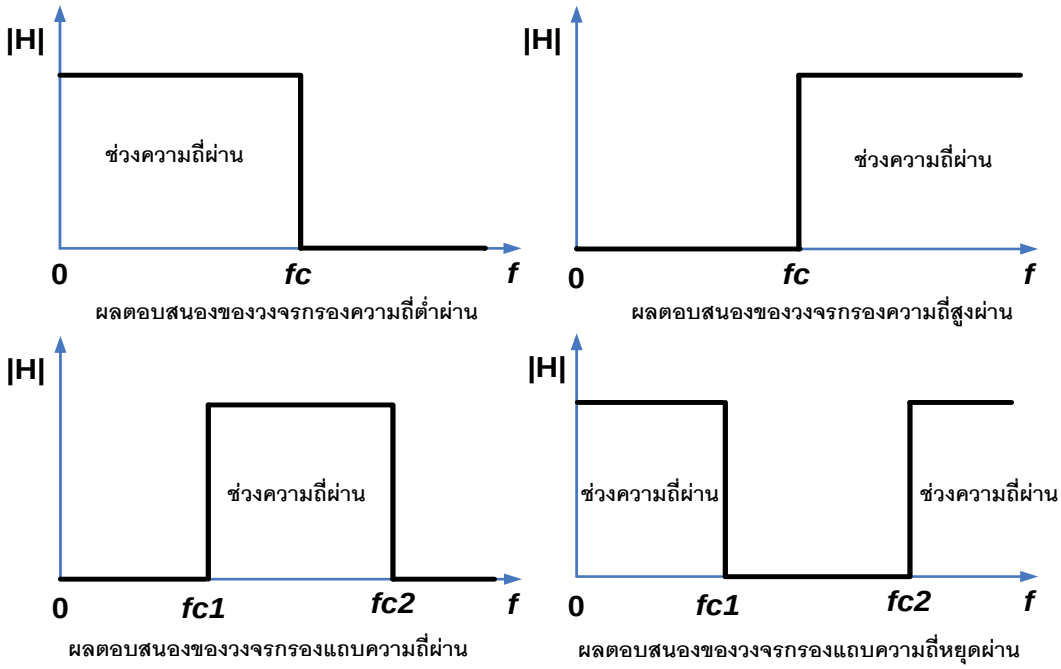
รูปที่ 2.4 สเปกตรัมของสัญญาณจากการสุ่มตัวอย่าง

จากรูปที่ 2.4 (d) ถ้าความถี่ของการสุ่มสัญญาณ $fss(t)$ มีค่าต่ำหรือ ω_s มีค่าต่ำหรือมี T มาก จะทำให้ช่วงห่างของแต่ละกลุ่มสเปกตรัมของ $Xss(\omega)$ เข้ามาทับกันผลนี้ทำให้เกิดความผิดเพี้ยนไปของสเปกตรัมเดิมได้ผลนี้เรียกว่า เอเลสซิง (aliasing) หมายความว่าความถี่การสุ่มสัญญาณ (fs) ซึ่งไม่ทำให้สัญญาณสูญเสียข้อมูลที่สำคัญไป ทฤษฎีการสุ่มตัวอย่าง (Sampling Theory) ของแชนนอล (Shannon) กล่าวไว้ว่า “หากสัญญาณต่อเนื่อง $x(t)$ ที่มีความถี่ไม่เกิน $\omega_{\max} = 2\pi f_{\max}$ ข้อมูลของสัญญาณต่อเนื่องนั้นสามารถอธิบายด้วย $x(nT)$ ก็ต่อเมื่อความถี่ในการสุ่มตัวอย่าง fs มีค่ามากกว่าหรือเท่ากับสองเท่าของความถี่ใช้งานสูงสุด ($f_{\max}$) หรือ $fs \geq 2f_{\max}$ “ [4] ซึ่งโดยทั่วไปจะทำการสุ่มตัวอย่างด้วยความถี่ $f_{sN} = 2f_{\max}$ พอดีค่าความถี่นี้มีชื่อเรียกว่าความถี่ไนควิสต์ (Nyquist frequency) และคาบเวลา $T_N = 1/(1f_{\max})$ นี้เรียกว่าช่วงเวลาสุ่มตัวอย่างไนควิสต์ (Nyquist interval)

2.6 พื้นฐานวงจรกรองอนาลอกต้นแบบ

เนื่องจากในการออกแบบวงจรกรองดิจิตอลนั้นมีการอาศัยวงจรกรองอนาลอกต้นแบบ ดังนั้นในการเริ่มต้นออกแบบวงจรดิจิตอลจึงควรทราบการคำนวนฟังก์ชันถ่ายโอนของวงจรอนาลอกให้ได้ตามข้อกำหนดที่ต้องการเสียก่อน จากนั้นจึงทำการเขียนให้อยู่ในรูปแบบการชื่อมต่ออุปกรณ์อนาลอกชนิดต่าง ๆ หรืออาจแปลงสู่โดเมนดิจิตอลจากฟังก์ชันถ่ายโอนโดยตรงเพื่อสร้างวงจรกรองแบบต่าง ๆ ขึ้นมา

วงจรกรองความถี่ทำหน้าที่จำแนกความถี่ตามความต้องการของผู้ใช้ แบ่งตามคุณลักษณะของผลตอบสนองความถี่ (frequency response) ดังรูปที่ 2.5 ได้ 4 ชนิดด้วยกันคือ [5] วงจรกรองความถี่ต่ำผ่าน (low-pass filter : LPF) , วงจรกรองความถี่สูงผ่าน (high-pass filter : HPF) , วงจรกรองแถบความถี่ผ่าน (band-pass filter ; BPF) และวงจรกรองแถบความถี่หยุดผ่าน (band-stop filter : BSF)



รูปที่ 2.5 ชนิดของวงจรกรองความถี่แบบต่าง ๆ

จากผลตอบสนองความถี่ในรูปที่ 2.5 เมื่อให้ |H| คือขนาดของแรงดันทางด้านเอาต์พุต วงจรกรองความถี่ต่ำผ่าน จะยอมให้ความถี่ตั้งแต่ 0 Hz ถึงความถี่ f_c ผ่านไปยังเอาต์พุตของวงจรได้ ส่วนความถี่ที่สูงกว่า f_c ความถี่จะไม่ผ่านไปยังเอาต์พุตของวงจร สำหรับวงจรกรองความถี่สูงผ่านจะยอมให้ความถี่สูงกว่าความถี่ f_c ผ่านไปยังเอาต์พุตของวงจรได้ ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ f_c จะไม่

ผ่านไปยังเอาต์พุตของวงจร สำหรับวงจรกรองแถบความถี่ผ่าน จะยอมให้ความถี่ตั้งแต่ f_{c1} ถึงความถี่ f_{c2} ผ่านไปยังเอาต์พุตของวงจร ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ f_{c1} กับความถี่ที่สูงกว่า f_{c2} จะไม่ผ่านไปยังเอาต์พุตของวงจร และวงจรกรองแถบความถี่หยุดผ่าน จะไม่ยอมให้ช่วงความถี่ f_{c1} ถึงความถี่ f_{c2} ผ่านไปยังเอาต์พุตของวงจร ส่วนความถี่อื่น ๆ วงจรยอมให้ผ่านไปยังเอาต์พุตได้

2.6.1 ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอก

การสร้างฟังก์ชันถ่ายโอนต้นแบบอาศัยการประมาณพหุนามตามวิธีการแบบ บัตเตอร์เวิร์ธ เชบีเชฟ และอิลลิปติค ซึ่งอาจใช้วิธีการเปิดตาราง หรือใช้โปรแกรมช่วยในการคำนวณจากข้อกำหนดที่ต้องการ รูปแบบของฟังก์ชันถ่ายโอนต้นแบบในโดเมน s ดังสมการที่ (2.9)-(2.11)

$$H(s)=\frac{k(s-z_1)(s-z_2)\ldots}{(s-p_0)(s-p_1)(s-p_1^*)\ldots} \tag{2.9}$$

$$H(s)=\frac{N(s)}{(s+p_0)\prod(s+p_i)(s+p_i^*)}=\frac{N(s)}{D(s)} \tag{2.10}$$

$$H(s)=\frac{\sum_{i=0}^{m} b_i\cdot s^i}{\sum_{i=0}^{n} a_i\cdot s_i} \tag{2.11}$$

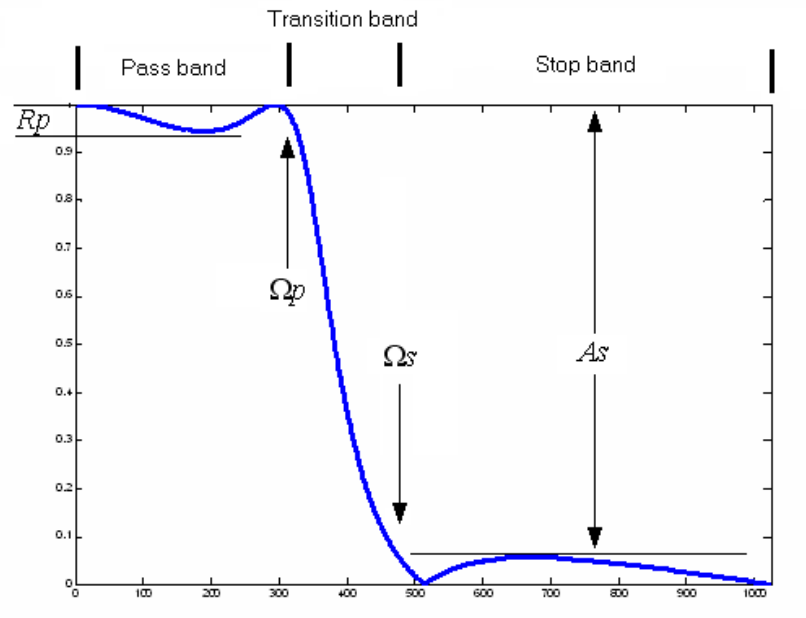
เมื่อ $z_1, z_2, \ldots, z_m$ ซีโรของวงจรกรอง $p_0, p_1, \ldots, p_n$ เป็นโพลของวงจรกรอง k อัตราขยาย และ a_i, b_i เป็นสัมประสิทธิ์ของพหุนาม s

2.6.2 วงจรกรองอนาลอกแบบต่าง ๆ

ในการออกแบบวงจรกรองความถี่ [6] โดยทั่วไปจะต้องทำการกำหนดคุณสมบัติต่าง ๆ ซึ่งประกอบไปด้วย ชนิดและอันดับของวงจรกรอง (order: n) จุดตัดความถี่ (frequency cutoff : ω_c) การกระเพื่อมในแถบผ่าน (Passband ripple: R_p) และค่าลดทอนในช่วงความถี่หยุด (stopband ripple: R_s)

ในรูปที่ 2.6 แสดงผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำ Rp คือการกระเพื่อมในแถบผ่าน (dB) จะกำหนดให้มีระดับของขนาดจากยอดถึงยอดซึ่งโดยทั่วผู้ออกแบบจะกำหนดมีปริมาณเพียงเล็กน้อยช่วงความถี่ผ่านเท่านั้น (ในทางอุดมคติจะมีค่าเป็น 0) As คือการลดทอนในแถบหยุด (dB) Ωs คือความถี่ที่ขอบแถบหยุด (rad/s) Ωp เป็นความถี่ที่ขอบแถบผ่าน (rad/s) ความถี่ระหว่าง Ωp และ Ωs เรียกว่าความถี่ทรานซิชั่นแบนด์ซึ่งความชันนั้นแปรผันโดยตรงกับอันดับของวงจรกรองขึ้นอยู่กับผู้ออกแบบกำหนด อย่างไรก็ตาม ผลของริปเปิ้ลและทรานซิชั่น

แบนด์จะมีผลต่อกันโดยตรงซึ่งหากที่จะต้องการความชันที่มากก็จะทำให้มีค่าริปเปิ้ลมากตามไปด้วยเช่นเดียวกันซึ่งในทางอุดมคติค่า Ωp และ Ωs จะมีค่าเท่ากัน $(\Omega p = \Omega s)$



รูปที่ 2.6 คุณลักษณะผลตอบสนองความถี่ของวงจรกรองความถี่ต่ำ

ซึ่งหากจะทำการออกแบบวงจรกรองความถี่แบบอื่น ๆ ไม่ว่าจะเป็นวงจรกรองแถบความถี่ผ่าน วงจรกรองแถบความถี่หยุด หรือวงจรกรองความถี่สูง สามารถนำเอาวงจรกรองความถี่ต่ำที่กล่าวไว้ในข้างต้นไปทำการออกแบบได้โดยวิธีการทรานฟอร์มทางความถี่ (frequency transform) ให้อยู่ในรูปแบบของวงจรกรองความถี่อื่น ๆ ซึ่งจะไม่ขอกล่าวถึง

ปัจจุบันการออกแบบวงจรกรองความถี่จะกำหนดค่าตัวแปรต่าง ๆ เหล่านี้ลงในซอฟท์แวร์ที่ช่วยในการออกแบบซึ่งจะได้พารามิเตอร์ต่าง ๆ ไปใช้งานต่อไปเช่นค่า R, L, C สำหรับการนำไปสร้างวงจรกรองแบบอนาลอก หรือค่าสัมประสิทธิ์ที่อยู่ในรูปของสมการผลต่างสำหรับนำไปสร้างไม่ว่าจะอยู่ในรูปแบบของ Hardware หรือ Software อื่น ๆ ก็ตาม โดยที่ชนิดของวงจรกรองความถี่จะเป็นไปตามวิธีการประมาณทางคณิตศาสตร์ซึ่งมีหลายแบบดังนี้

2.6.2.1 วงจรกรองอนาลอกแบบ บัทเตอร์เวิร์ธ (Butterworth filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกแบบ บัทเตอร์เวิร์ธ สร้างจาก อันดับ และ n ใดๆ ความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{\log\left\{\left(10^{\frac{Rp}{10}} - 1\right)\left(10^{\frac{As}{10}} - 1\right)\right\}}{2\log\left(\Omega_p / \Omega_s\right)} \tag{2.12}$$

$$\Omega c = \frac{\Omega p}{\sqrt[2N]{10^{\frac{Rp}{10}} - 1}} \tag{2.13}$$

เมื่อ N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.6.2.2 วงจรกรองอนาลอกแบบ เชบีเชฟ (Chebyshev filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกต้นแบบ เชบีเชฟ สร้างจาก อันดับ n ใดๆ และความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{\log\left(g + \sqrt{g^2 - 1}\right)}{\log\left(\Omega r + \sqrt{\Omega_r^2 - 1}\right)} \tag{2.14}$$

เมื่อ $g = \sqrt{\frac{10^{\frac{As}{10}} - 1}{10^{\frac{Rp}{10}} - 1}}$ และ $\Omega_r = \frac{\Omega s}{\Omega p}$ (2.15)

$$\Omega c = \Omega p$$

N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.6.2.3 วงจรกรองอนาลอกแบบ อิลิปติค (Elliptic filter)

ฟังก์ชันถ่ายโอนของวงจรกรองอนาลอกต้นแบบ อิลิปติค สร้างจาก อันดับ n ใดๆ และความถี่ตัดของวงจรกรองแสดงดังสมการ

$$N = \frac{K(k)K\left(\sqrt{1-k_1^2}\right)}{K(k_1)K\left(\sqrt{1-k^2}\right)} \tag{2.16}$$

เมื่อ $k = \frac{\Omega p}{\Omega s}, k_1 = \sqrt{\frac{10^{\frac{Rp}{10}} - 1}{10^{\frac{As}{10}} - 1}}$ และ $K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - x^2 \sin^2\theta}}$ (2.17)

และมีความถี่ตัดเช่นเดียวกับวงจรของเชบีเชฟคือ

$$\Omega c = \Omega p$$

N คือ อันดับของวงจรกรอง

Rp คือการกระเพื่อมในแถบผ่าน (dB)

As คือการลดทอนในแถบหยุด (dB)

Ωp คือความถี่ที่ขอบแถบผ่าน (rad/s)

Ωs คือความถี่ที่ขอบแถบหยุด (rad/s)

Ωc คือความถี่ตัดเชิงมุม (rad/s)

2.7 วงจรกรองความถี่แบบดิจิตอล

ในปัจจุบันเทคโนโลยีในการประมวลผลสัญญาณดิจิตอลมีความสามารถสูงขึ้นความละเอียดและความเร็วในการประมวลผลมากขึ้น จึงเป็นที่นิยมการประมวลผลในรูปแบบสัญญาณดิจิตอลมากขึ้นซึ่งระบบดิจิตอลนั้นมีความยืดหยุ่นมากกว่าเพราะอยู่ในรูปของซอฟท์แวร์เป็นส่วนใหญ่ซึ่งสามารถปรับปรุงและแก้ไขได้ง่ายกว่าโดยการเปลี่ยนค่าสัมประสิทธิ์โดยในขณะที่ในวงจรกรองความถี่ของอนาลอกสามารถทำได้ซับซ้อนกว่า

ในการออกแบบดิจิตอลฟิลเตอร์นั้นสามารถแบ่งออกได้สองประเภท [7] ได้แก่ การออกแบบโดยใช้วงจรกรองความถี่แบบ เอฟ ไอ อาร์ (Finite Impulse Response: FIR) และวงจรกรองความถี่แบบ ไอ ไอ อาร์ (Infinite Impulse Response: IIR)

2.7.1 วงจรกรองความถี่แบบ เอฟ ไอ อาร์ (FIR Filters)

วงจรกรองความถี่แบบ เอฟ ไอ อาร์ หรือ วงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวจำกัด ซึ่งเป็นการนำเอาเฉพาะสัญญาณอินพุตในแซมเปิ้ลปัจจุบันและแซมเปิ้ลที่ถูกหน่วงเวลามาทำการประมวลผลซึ่งสามารถเขียนสมการได้ดังนี้

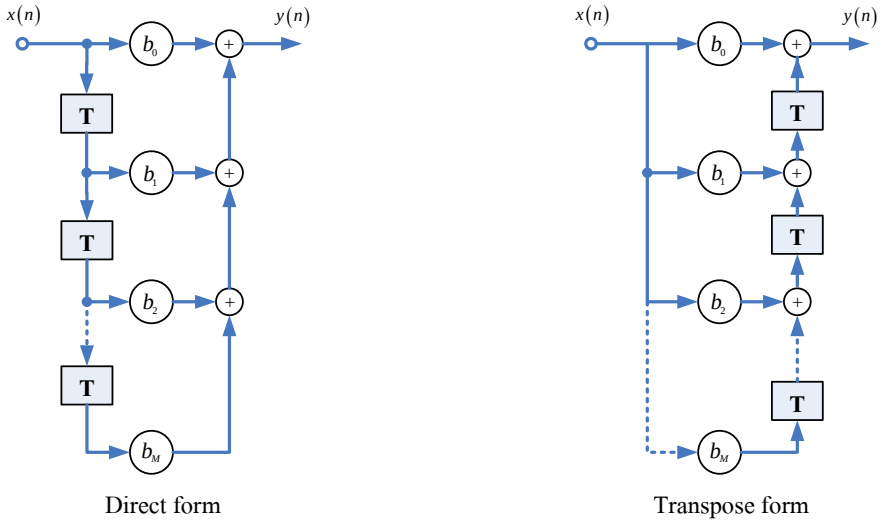
$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{2.18}$$

โดยที่ b_k คือค่าสัมประสิทธิ์ของวงจรกรองความถี่จะมีค่าจำนวนอยู่จำนวน M ค่าโดยที่จำนวนของค่าสัมประสิทธิ์นั้นจะขึ้นอยู่กับความต้องการของผู้ออกแบบซึ่งสามารถเขียนฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังนี้

$$H(z) = \sum_{k=0}^{M} b_k \cdot Z^{-k} \tag{2.19}$$

ซึ่งในการออกแบบจะพยายามลดค่าจำนวนสัมประสิทธิ์ให้มีจำนวนน้อยที่สุดในขณะที่ยังคงรักษาคุณสมบัติที่ต้องการไว้ได้อยู่ โดยปกติแล้วในการออกแบบใช้งานทั่วไปนั้นจะมีจำนวน

สัมประสิทธิ์ตั้งแต่ไม่กี่ตัวไปจนถึงร้อยตัวซึ่งทำให้มีโครงสร้างและวงจรที่มีขนาดใหญ่มาก แต่วงจรกรองดังกล่าวนั้นจะให้การตอบสนองทางเฟสที่เป็นเชิงเส้นโดยมีโครงสร้างดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างของวงจรกรองแบบ FIR

2.7.2 วงจรกรองความถี่แบบ ไอ ไอ อาร์ (IIR Filters)

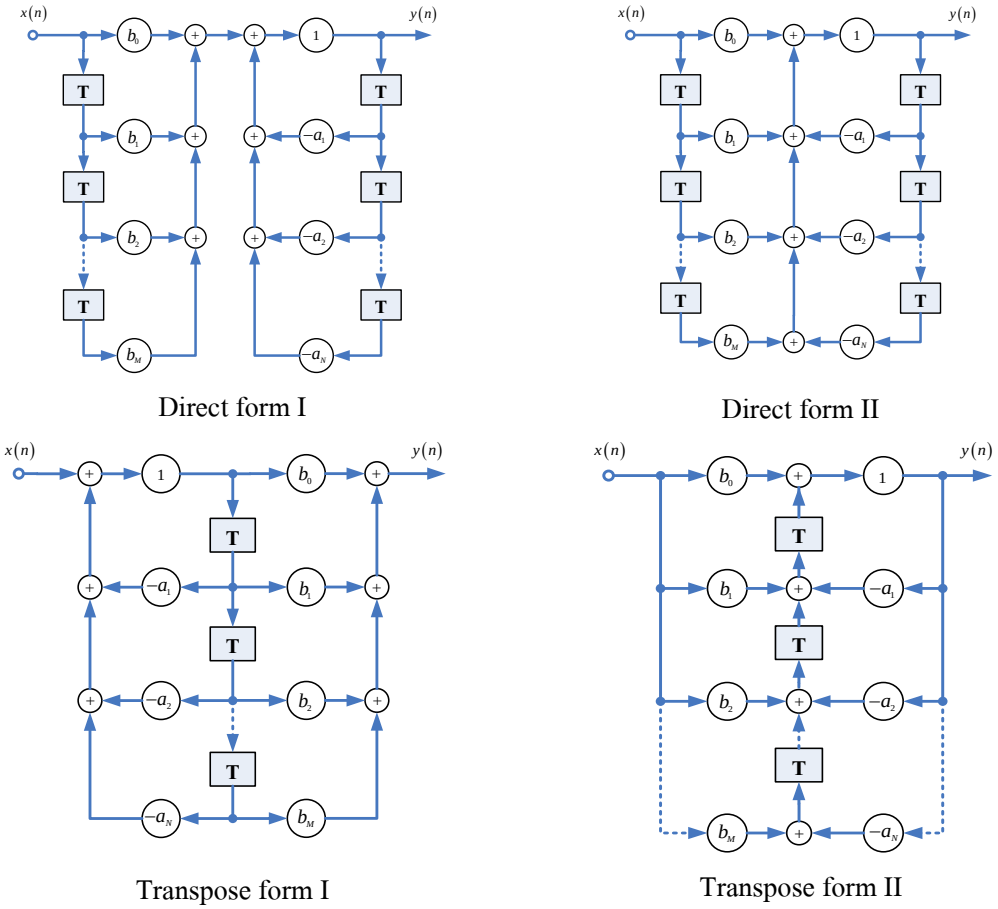
วงจรกรองความถี่แบบ ไอ ไอ อาร์ หรือ วงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวไม่จำกัด ซึ่งเป็นการนำเอาสัญญาณอินพุตในแซมเปิ้ลปัจจุบันและแซมเปิ้ลที่ถูกหน่วงเวลาและเอาต์พุตแซมเปิ้ลที่ถูกหน่วงเวลามาทำการประมวลผลซึ่งสามารถเขียนสมการได้ดังนี้

$$\sum_{k=0}^{N} a_k \cdot y(n-k) = \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{2.20}$$

โดยที่ b_k และ a_k คือค่าสัมประสิทธิ์ของวงจรกรองความถี่จะมีค่าจำนวนอยู่จำนวน N และ M ค่าตามลำดับโดยที่จำนวนของค่าสัมประสิทธิ์นั้นจะมีจำนวนน้อยมากเมื่อเทียบกับวงจรกรองแบบ เอฟ ไอ อาร์ ภายใต้เงื่อนไขการออกแบบเดียวกัน ซึ่งสามารถเขียนฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังนี้

$$H(z) = \frac{\sum_{k=0}^{M} b_k \cdot Z^{-k}}{\sum_{k=0}^{N} a_k \cdot Z^{-k}} \tag{2.21}$$

วงจรกรองแบบ ไอ ไอ อาร์ นี้จะมีข้อดีก็คือมีจำนวนของสัมประสิทธิ์ที่น้อยกว่าวงจรกรองแบบ เอฟ ไอ อาร์ ทำให้ลดเวลาในการประมวลผลลงอย่างมากแต่จะมีข้อเสียเช่นกันก็คือในการออกแบบระบบนี้จะมีความไวในเรื่องของการปัดเศษค่าสัมประสิทธิ์ เพราะผลจากการปัดเศษค่าสัมประสิทธิ์อาจจะทำให้เกิดการไม่เสถียรภาพของวงจรได้และมีการตอบสนองทางเฟสที่ไม่เป็นเชิงเส้นซึ่งข้อดีข้อเสียของวงจรกรองความถี่ทั้งสองแบบแสดงดังตารางที่ 2.3



รูปที่ 2.8 โครงสร้างของวงจรกรองความถี่แบบ IIR

โครงสร้างของวงจรกรองแบบ IIR จากสมการที่ (2.21) มีโครงสร้างแบ่งออกเป็น 4 แบบ ดังรูปที่ 2.8 ได้แก่โครงสร้างแบบตรง (Direct form) และแบบกลับ (Transpose form) ซึ่งจะมีความซับซ้อนของวงจรที่น้อยกว่า

ตารางที่ 2.3 ข้อดีและข้อเสียของวงจรกรองความถี่แต่ละแบบ

ไอ ไอ อาร์ (IIR)	เอฟ ไอ อาร์ (FIR)
1. วงจรอาจไม่เสถียรได้	1. วงจรเสถียรเสมอ
2. ให้ช่วงทรานซิชั่นแบนด์ที่แคบกว่า	2. ให้ช่วงทรานซิชั่นแบนด์ที่กว้างกว่า
3. มีการหน่วงสัญญาณที่เอาต์พุตน้อยกว่า	3. มีการหน่วงสัญญาณมากกว่า
4. มีความเพี้ยนของผลตอบสนองทางเฟสสูง	4. มีผลตอบสนองทางเฟสเป็นเชิงเส้น
5. มีสัญญาณรบกวนจากการปัดเศษมาก	5. มีสัญญาณรบกวนจากการปัดเศษน้อย
6. สามารถออกแบบได้จากวงจรอนาลอกต้นแบบ	6. การออกแบบทำได้โดยใช้วิธีการประมาณค่า

ในระยะเวลาหลายปีที่ผ่านมาได้มีนักวิจัยได้ทำการค้นคว้าหาทางแก้ข้อเสียต่าง ๆ ของ ไอ ไอ อาร์ ฟิลเตอร์ เช่น การมีความไวต่ำต่อการปัดเศษของค่าสัมประสิทธิ์ [8] โดยใช้โครงสร้างของ PAPF (Parallel all-pass Filter) โครงสร้างแบบอนุกรมคานอนิคัลที่มีค่าสัญญาณรบกวนต่ำ โครงสร้าง ไอ ไอ อาร์ ฟิลเตอร์ในรูปของวงจรกรองความถี่ผ่านตลอดย่าน โครงสร้างของ linear phase IIR filter ทำให้วงจรกรองแบบ IIR เป็นอีกทางเลือกหนึ่งในการนำไปใช้งาน

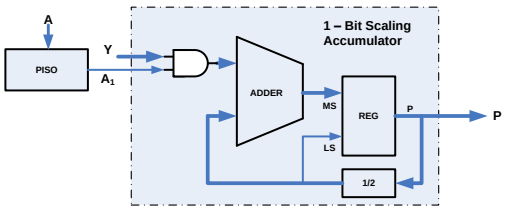
2.8 การประมวลผลสัญญาณเชิงเลขโดยวิธีการของคณิตศาสตร์แบบการกระจาย

คณิตศาสตร์แบบการกระจายหรือ (Distributed Arithmetic: DA) ได้ถูกนำเสนอขึ้นครั้งแรกในปี คศ. 1971 (US Patent 3,777,130) [9]โดย Croisier และ คณะ ซึ่งต่อมาได้ถูกนำไปประยุกต์ ใช้งานกับการประมวลผลสัญญาณดิจิตอลโดย Abrahan Peled และ Bede Liu [10] โดยรูปแบบการคำนวณของ DA เป็นรูปแบบที่สอดคล้องกับการประมวลผลสัญญาณดิจิตอลอีกทั้งยังสามารถนำไปสร้างในวงจรลอจิกได้อีกด้วย จึงทำให้โครงสร้างดังกล่าวถูกนำมาใช้งานอย่างแพร่หลาย โดยพื้นฐานของคณิตศาสตร์แบบการกระจายเป็นการเริ่มต้นที่การพิจารณาจากสมการ ผลบวกของผลคูณ (Sum of Product: SOP) หากกำหนดให้ Y เป็นข้อมูลเวกเตอร์ขนาดความยาว N ซึ่งทำการคูณอยู่กับค่าสัมประสิทธิ์เวกเตอร์ A ขนาดความยาว N เช่นเดียวกันจะได้ผลลัพธ์ P มีค่าดังนี้

$$P = \sum_{i=1}^{N} A_i \cdot Y_i \tag{2.22}$$

2.8.1 การคูณค่าและการบวกสะสมค่าผลคูณ (MAC)

ถ้าพิจารณาจากสมการที่ (2.14) เฉพาะค่าของผลคูณย่อยของ $A_i \cdot Y_i$ โดยแทน A_i และ Y_i เป็นข้อมูลขนาด 4 บิตดังแสดงในรูปที่ 2.6 ผลคูณของเลขฐานสอง $A_i \cdot Y_i$ จะเป็นการคูณค่าแล้วทำการบวกสะสมค่าผลคูณที่ได้ทั้ง 4 ค่า (Multiply and Accumulate: MAC) ซึ่งจะถูกเรียกเป็นบล๊อก MAC [11]



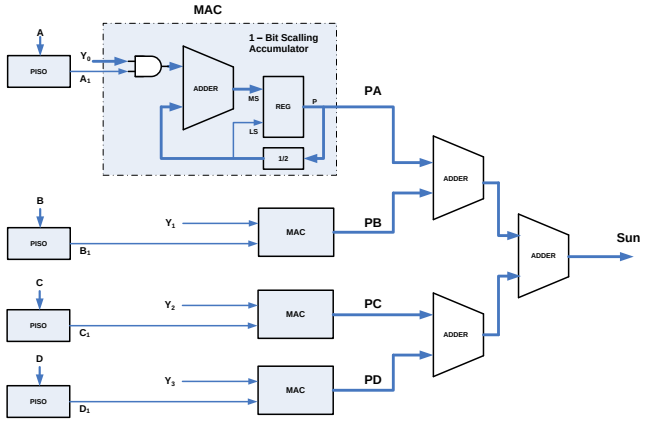
LET A = $A_3\ A_2\ A_1\ A_0$ and Y = $Y_3\ Y_2\ Y_1\ Y_0$ then A*Y = P is

		{Start}
$A_0*(Y_3\ Y_2\ Y_1\ Y_0)$	<1/2>	{Cycle = 0}
$+ A_1*(Y_3\ Y_2\ Y_1\ Y_0)$	<1/2>	{Cycle = 1}
$+ A_2*(Y_3\ Y_2\ Y_1\ Y_0)$	<1/2>	{Cycle = 2}
$+ A_3*(Y_3\ Y_2\ Y_1\ Y_0)$		{Cycle = 3}
$P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$		{End}

รูปที่ 2.9 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ขนาด 4 บิต

MAC จะทำการหาผลคูณย่อยในแต่ละครั้งระหว่างค่าสัมประสิทธิ์ Y และข้อมูล A ด้วยการ and กัน เทคนิคนี้จะทำการบวกค่าผลคูณย่อยเหล่านี้เข้าด้วยกัน โดยแอกคิวมูเลเตอร์ซึ่งจะทำการเลื่อนบิตไปทางขวา 1 บิต ซึ่งเหมือนกับการหารข้อมูลนั้นด้วย 2 (แทนสัญลักษณ์ $\frac{1}{2}$) ในแต่ละรอบของสัญญาณนาฬิกาซึ่งเป็นการชดเชยสำหรับการถ่วงน้ำหนักบิตของผลคูณย่อย

เราสามารถหาค่าผลลัพธ์ในรูแบบของ SOP ได้ดังนี้ $SUM = A \cdot Y_0 + B \cdot Y_1 + C \cdot Y_2 + D \cdot Y_3$ ดังรูปที่ 2.10

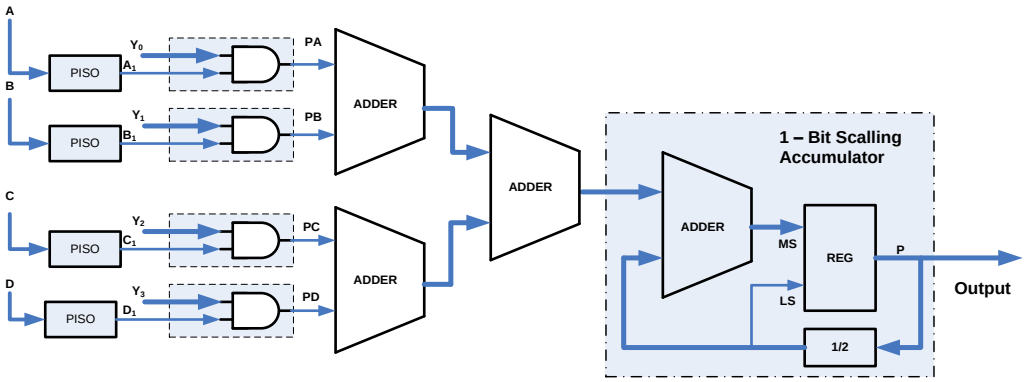


รูปที่ 2.10 ผลลัพธ์จากการรวมผลคูณย่อยจาก MAC ทั้ง 4 ชุด

ตัวคูณทั้ง 4 ตัวจะถูกทำตามลำดับและผลลัพธ์จะถูกรวมเข้าด้วยกัน เมื่อทำการคูณเสร็จการทำงานจะใช้จำนวนรอบของสัญญาณนาฬิกา n รอบสำหรับข้อมูลขนาด n บิต ดังนั้นอัตราของสัญญาณนาฬิกาจึงมีค่าเท่ากับอัตราของข้อมูลหารด้วยจำนวนบิตโดยที่ระหว่างแต่ละรอบสัญญาณนาฬิกาของข้อมูลตัวคูณ 4 ตัวจะถูกรวมเข้าด้วยกันเป็นผลลัพธ์ออกมา (ในรูป PA, PB, PC และ PD) ทั้ง 4 ตัว จะถูกรวมกันเป็นสมการเอาต์พุต ดังนั้นจะเห็นได้ว่าถ้าจำนวนเทอมของผลคูณย่อยที่มีปริมาณมากขึ้นทำให้จำนวนของผลคูณย่อย และจำนวนของแอกคิวมูเลเตอร์เพิ่มมากขึ้นด้วยทำให้ขนาดของวงจรมีขนาดใหญ่ขึ้น

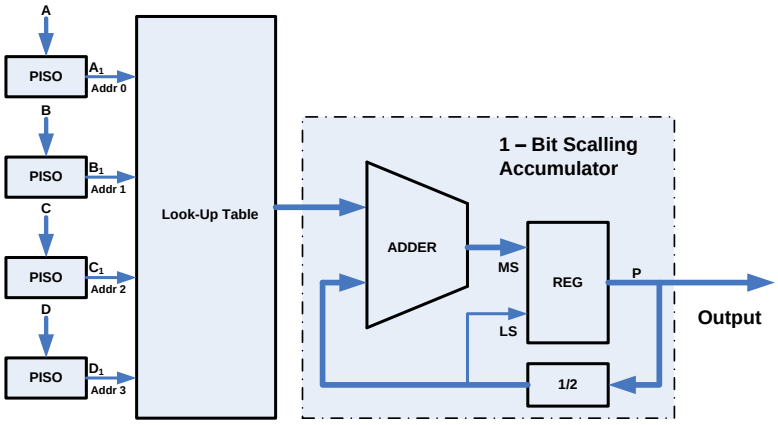
2.8.2 การลดขนาดของ MAC ด้วย DA [11]

วิธีการประมวลผลแบบคณิตศาสตร์กระจายหรือ DA นั้น ต่างไปจากวิธีการทำงานของ MAC โดย DA จะทำการบวกค่าของผลคูณย่อยก่อน จากนั้นจะทำการชดเชยโดยการถ่วงน้ำหนักบิตของการบวกสะสมสามารถแสดงโครงสร้างการใช้ DA ได้ดังรูปที่ 2.11



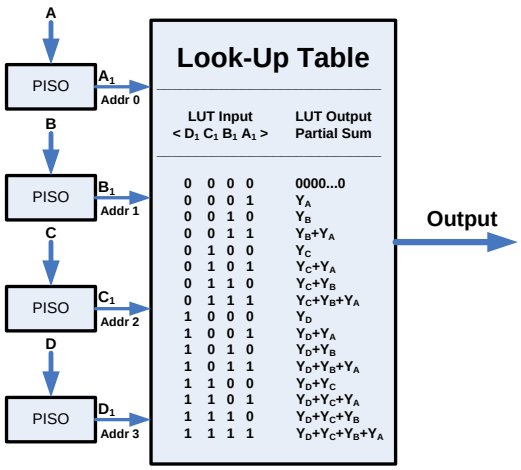
รูปที่ 2.11 ผลจากการรวมผลคูณย่อยจาก Serial Distributed Arithmetic แทน MAC ทั้ง 4 ชุด

การทำงานของโครงสร้างนี้จะเป็นการเรียงลำดับของข้อมูลใหม่ โดยเทคนิคนี้จะช่วยลดจำนวนวงจรเลื่อนบิตและบวก (Shift and Add) รวมทั้งตัวบวกทั้ง 3 ตัว เป็นข้อมูล เราเรียกวิธีนี้ว่าตารางดูค่า (Look up table: LUT) ขนาด 4 บิตให้ผลลัพธ์ ที่มีขนาดเล็กลง และเรียกรูปแบบนี้ว่าคณิตศาสตร์แบบการกระจายข้อมูลอนุกรม (Bit Serial Distributed Arithmetic MAC: SDA-MAC) ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของ SDA-MAC ที่สร้างจากตารางดูค่า

ข้อมูลของตารางดูค่า (LUT) ตามรูปที่ 2.13 ประกอบด้วยผลคูณย่อยของสัมประสิทธิ์ (Y_0 Y_1 Y_2 และ Y_3) LSB (output จากแต่ละ serial shift register) ของข้อมูลทั้ง 4 จะอ้างอิงตำแหน่งใน ตาราง LUT ถ้าทั้ง 4 บิตเป็นระดับสัญญาณ “1” เอาต์พุตของสัญญาณจากตารางดูค่าก็จะสอดคล้องกับสัมประสิทธิ์ทั้ง 4 ค่า ซึ่งจะมีค่าทั้งหมด 16 ค่า



รูปที่ 2.13 ค่าของ LUT ที่เกิดจากค่าสัมประสิทธิ์ทั้ง 4 ค่า

2.9 การสร้างดิจิตอลฟิลเตอร์แบบเลขคณิตแจกแจง [12]

ในวงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวจำกัด (FIR) ในสมการ (2.23) สามารถเขียนเป็นสมการผลต่างได้เป็น

$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{2.23}$$

ในวงจรที่มีการตอบสนองต่อสัญญาณอิมพัลซ์มีความยาวไม่จำกัด (IIR) ในสมการ (2.24) สามารถเขียนเป็นสมการผลต่างสืบเนื่องได้เป็น

$$y(n) = \sum_{k=0}^{M} b_k \cdot x(n-k) - \sum_{k=1}^{M} b_k \cdot y(n-k) \tag{2.24}$$

ซึ่งจะเห็นได้ว่าในการคำนวณ $y(n)$ นั้นจะต้องทำการคูณและการบวกโดยที่ทราบกันดีแล้วว่าการคูณในคอมพิวเตอร์นั้นจะใช้เวลาที่นานมากกว่าคำสั่งอื่น ๆ ทำให้ในการทำงานที่ความเร็วสูง ๆ อาจทำงานเป็นแบบเวลาจริงไม่ได้ (Real-time) ซึ่งโครงสร้างแบบเลขพีชคณิตแจกแจง (Distributed Arithmetic Structure) หรือที่เรียกว่าโครงสร้างแบบ Rom/Acc (Rom-Accumulator) สามารถเปลี่ยนการคูณให้อยู่ในรูปแบบของการบวก (Adding) และการเลื่อน (Shifting) โดยสามารถนำหน่วยความจำประเภท ROM มาประยุกต์ใช้งานร่วมกันได้

2.10 หลักการโครงสร้างของเลขคณิตแจกแจง

เมื่อพิจารณาวงจรกรอง IIR ที่มีฟังก์ชันถ่ายโอนอันดับ 3 (n=3) มีรูปทั่วไปคือ

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} \tag{2.25}$$

สามารถเขียนให้อยู่ในรูปแบบของสมการผลต่าง (Difference equation) ได้คือ

$$\begin{aligned} y(n) = a_0 \cdot x(n) + a_1 \cdot x(n-1) + a_2 \cdot x(n-2) + a3 \cdot x(n-3) \\ - b_1 \cdot y(n-1) - b_2 \cdot y(n-2) - b_3 \cdot y(n-3) \end{aligned} \tag{2.26}$$

โดยให้ $x(n), x(n-1), x(n-2)$ และ $x(n-3)$ เป็นลำดับของสัญญาณที่เข้า

$y(n), y(n-1), y(n-2)$ และ $y(n-3)$ เป็นลำดับของสัญญาณที่ออก

$a_0, a_1, a_2, a_3, b_1, b_2$ และ b_3 เป็นค่าสัมประสิทธิ์ของตัวกรอง

วิธีการของโครงสร้างเลขคณิตแจกแจงทำโดยการเขียนแทนลำดับสัญญาณเข้าและลำดับสัญญาณออกด้วยตัวเลขแบบเติมเต็มสอง (2's Complement) ที่มีจำนวนบิตรวมทั้งเครื่องหมายด้วยเป็น B บิตหรือเขียนกระจายเป็นเลขฐานสองคือ

$$x(n) = x_0(n) \cdot x_1(n) x_2(n) \cdots x_{B-1}(n) \tag{2.27}$$

$$y(n) = y_0(n) \cdot y_1(n) y_2(n) \cdots y_{B-1}(n) \tag{2.28}$$

โดยที่ $x_0(n)$ และ $y_0(n)$ เป็นบิตที่แสดงเครื่องหมายของตัวเลข ส่วน $x_1(n)x_2(n)\cdots x_{B-1}(n)$ และ $y_1(n)y_2(n)\cdots y_{B-1}(n)$ เป็นบิตที่ i ของลำดับสัญญาณมีค่าเป็น "0" หรือ "1" เท่านั้น สามารถเขียนในรูปแบบของ 2's Complement ได้ดังนี้

$$x(n) = -x_0(n) + \sum_{i=1}^{B} x_i(n) \cdot 2^{-i} \tag{2.29}$$

$$y(n) = -y_0(n) + \sum_{i=1}^{B} y_i(n) \cdot 2^{-i} \tag{2.30}$$

เมื่อแทนสมการ (2.29) และ (2.30) ลงในสมการผลต่างสืบเนื่องจะได้

$$\begin{aligned} y(n) = a_0 &\left\{\sum_{i=1}^{B} x_i(n)\cdot 2^{-i} - x_0(n)\right\} + a_1\left\{\sum_{i=1}^{B} x_i(n-1)\cdot 2^{-i} - x_0(n-1)\right\} \\ &+ a_2\left\{\sum_{i=1}^{B} x_i(n-2)\cdot 2^{-i} - x_0(n-2)\right\} + a_3\left\{\sum_{i=1}^{B} x_i(n-3)\cdot 2^{-i} - x_0(n-3)\right\} \\ &- b_1\left\{\sum_{i=1}^{B} y_i(n-1)\cdot 2^{-i} - y_0(n-1)\right\} - b_2\left\{\sum_{i=1}^{B} y_i(n-2)\cdot 2^{-i} - y_0(n-2)\right\} \\ &- b_3\left\{\sum_{i=1}^{B} y_i(n-3)\cdot 2^{-i} - y_0(n-3)\right\} \end{aligned} \tag{2.31}$$

เมื่อทำการจัดพจน์ใหม่โดยนำบิตที่สมนัยมาเขียนรวมกันจะได้

$$\begin{aligned} y(n) = \sum_{i=1}^{B} 2^{-i} \cdot &\begin{Bmatrix} a_0\cdot x_i(n) + a_1\cdot x_i(n-1) + a_2\cdot x_i(n-2) + a_3\cdot x_i(n-3) \\ -b_1\cdot y_1(n-1) - b_2\cdot y_2(n-2) - b_3\cdot y_3(n-3) \end{Bmatrix} \\ - &\begin{Bmatrix} a_0\cdot x_0(n) + a_1\cdot x_0(n-1) + a_2\cdot x_0(n-2) + a_3\cdot x_0(n-2) - b_1 y_0(n-1) \\ -b_2 y_0(n-2) - b_3\cdot y_0(n-3) \end{Bmatrix} \end{aligned} \tag{2.32}$$

ให้

$$F_i\{\cdot\} = F_i\{x_i(n), x_i(n-1), x_i(n-2), x_i(n-3), y_i(n-1), y_i(n-2), y_i(n-3)\} \tag{2.33}$$

และ

$$\begin{aligned} F_i\{\cdot\} = a_0\cdot x_i(n) + a_1\cdot x_i(n-1) + a_2\cdot x_i(n-2) + a_3\cdot x_i(n-3) \\ - b_1\cdot y_i(n-1) - b_2\cdot y_i(n-2) - b_3\cdot y_i(n-3) \end{aligned} \tag{2.34}$$

จะได้

$$y(n) = \sum_{i=1}^{B-1} 2^{-i}\cdot F_i\{\cdot\} - F_0\{\cdot\} \tag{2.35}$$

ผลจากสมการ (2.33) และ (2.34) จะได้โครงสร้างแบบเลขคณิตแจกแจง โดยนำค่าฟังก์ชันทั้งหมดของ $F_i\{\cdot\}$ มาทำเป็นตารางเปิดดู (Look up table) โดยค่าในตารางเปิดดูสามารถคำนวณจากสมการ (3.34) และเนื่องจาก $F_i\{\cdot\}$ มีตัวแปรอยู่ 7 ตัว คือ $x_i(n)$, $x_i(n-1)$, $x_i(n-2)$, $x_i(n-3)$, $y_i(n-1)$, $y_i(n-2)$ และ $y_i(n-3)$ ทำให้ตารางเปิดดูมีค่าฟังก์ชันของ $F_i\{\cdot\}$ 2^7 ค่า โดยค่าของฟังก์ชัน $F_i\{\cdot\}$ ทั้ง 128 ค่านี้จะถูกคำนวณแล้วทำการปัดเศษให้เหลือ B บิต แล้วเก็บไว้ใน ROM เพื่อนำไปสร้างตัวกรองต่อไป

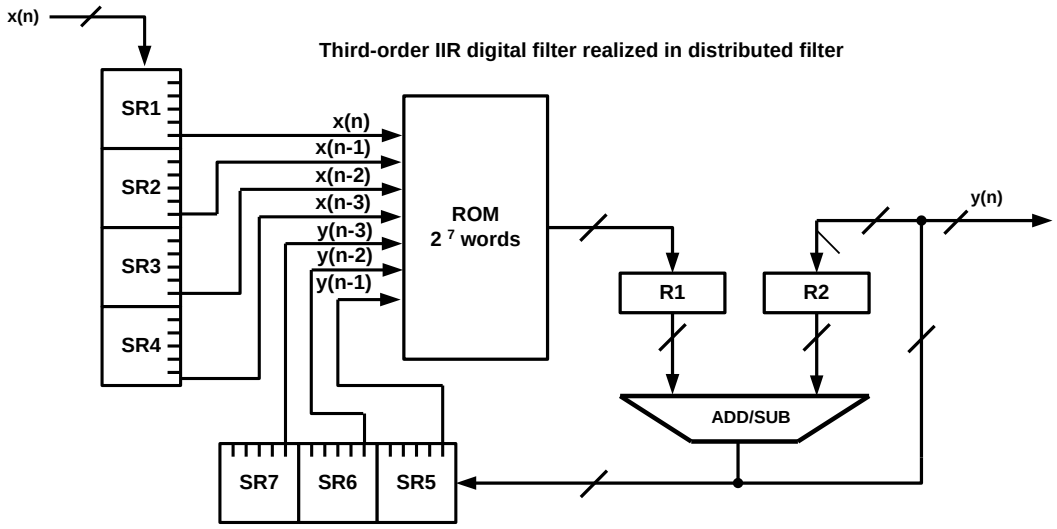
2.11 การสร้างตารางเปิดดูจากฟังก์ชัน $F_i\{.\}$ สำหรับวงจรกรองอันดับ 3

เมื่อได้สมการผลต่างสืบเนื่องอันดับ 3 แล้วนำค่าสัมประสิทธิ์คูณกับบิตแรกของแต่ละตัว แล้วบวกกันเก็บไว้ที่แอดเดรสที่ชี้โดยค่า $x_i(n)$, $x_i(n-1)$, $x_i(n-2)$, $x_i(n-3)$, $y_i(n-1)$, $y_i(n-2)$ และ $y_i(n-3)$ ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 ค่าของ $F_1\{.\}$ ภายใน ROM

ตำแหน่งที่	แอดเดรสของรอม							ค่าของ $F_1\{.\}$ ภายใน ROM
	$x_i(n)$	$x_i(n-1)$	$x_i(n-2)$	$x_i(n-3)$	$y_i(n-1)$	$y_i(n-2)$	$y_i(n-1)$	
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	$-b_3$
2	0	0	0	0	0	1	0	$-b_2$
3	0	0	0	0	0	1	1	$-b_2-b_3$
4	0	0	0	0	1	0	0	$-b_1$
5	0	0	0	0	1	0	1	$-b_1-b_3$
6	0	0	0	0	1	1	0	$-b_1-b_2$
7	0	0	0	0	1	1	1	$-b_1-b_2-b_3$
8	0	0	0	1	0	0	0	a_3
9	0	0	0	1	0	0	1	a_3-b_3
10	0	0	0	1	0	1	0	$a3-b_2$
11	0	0	0	1	0	1	1	$a_3-b_2-b_3$
123	1	1	1	1	0	1	1	$a_0+a_1+a_2+a_3-b_2-b_3$
124	1	1	1	1	1	0	0	$a_0+a_1+a_2+a_3-b_1$
125	1	1	1	1	1	0	1	$a_0+a_1+a_2+a_3-b_1-b_3$
126	1	1	1	1	1	1	0	$a_0+a_1+a_2+a_3-b_1-b_2$
127	1	1	1	1	1	1	1	$a_0+a_1+a_2+a_3-b_1-b_2-b_3$

จากนั้นทำการนำค่าของสัมประสิทธิ์ที่ได้จากการคำนวณมาหาค่าของแอดเดรสของรอมซึ่งมีทั้งหมด 128 ตำแหน่งโดยต่อร่วมกับรีจีสเตอร์และวงจรบวกตามรูปที่ 2.14



รูปที่ 2.14 โครงสร้างของ Distributed Arithmetic วงจรกรองความถี่แบบ IIR อันดับ 3

จากรูปที่ 1 SR1 และ SR5 เป็นรีจีสเตอร์แบบเข้าขนานออกอนุกรม (Serial In Parallel Out: SIPO) ใช้เก็บข้อมูลอินพุตในปัจจุบัน ส่วน SR2, SR3, SR4, SR6 และ SR7 เป็นรีจีสเตอร์ที่ใช้เก็บข้อมูลอินพุตในอดีต และเอาต์พุตในอดีต ซึ่งแบบเข้าอนุกรมออกอนุกรม (Serial In Serial Out: SISO) และ R1, R2, R3 เป็นรีจีสเตอร์ชั่วคราวใช้เก็บข้อมูลในการประมวลผลของวงจร ADD/SUB ซึ่งเป็นวงจรบวกลบเลข 2's complement โดยมีขั้นตอนการทำงานดังนี้

1. ทำการเคลียร์ข้อมูลในรีจีสเตอร์ R2 จากนั้นทำการโหลดข้อมูล x(n) เข้ามาเก็บไว้ในรีจีสเตอร์ SR1
2. ข้อมูลในบิตต่ำสุด (LSB) ของรีจีสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 เป็นลำดับสัญญาณของ x(n), x(n-1) , x(n-2) , x(n-3) , y(n-1) , y(n-2) และ y(n-3) ตามลำดับ โดยข้อมูลดังกล่าวจะใช้เป็นแอดเดรสให้กับ ROM เพื่อหาค่า F_i เมื่อ i เป็นข้อมูลบิตที่ L โดยข้อมูลที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการบวกค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้จะถูกเก็บไว้ใน R2 พร้อมกับเลื่อนข้อมูลไปทางขวา 1 บิต
3. ทำการเลื่อนข้อมูลในรีจีสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 ไป 1 บิต เพื่อกำหนดแอดเดรสของ ROM ใหม่ ได้ค่า F_i เมื่อ i เป็นบิตที่ L-1 จากนั้นนำค่าที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการบวกค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้จะถูกเก็บไว้ใน R2 พร้อมกับเลื่อนข้อมูลไปทางขวา 1 บิต

4. ทำซ้ำตามขั้นตอนที่ 3 สำหรับ i เป็นบิตที่ L-2, L-3, . . . , 1 ตามลำดับ
5. ทำการเลื่อนข้อมูลในรีจีสเตอร์ SR1, SR2, SR3, SR4, SR5, SR6 และ SR7 ไป 1 บิต เพื่อกำหนดแอดเดรสของ ROM ใหม่ ได้ค่า F_i เมื่อ i เป็นบิตที่ L-1 จากนั้นนำค่าที่ได้จาก ROM จะถูกเก็บไว้ที่ R1 จากนั้นทำการลบค่า R1 กับ R2 ด้วยวงจร ADD/SUB และผลลัพธ์ที่ได้คือลำดับสัญญาณ y(n)
6. ทำซ้ำตามขั้นตอนที่ 1-5 ใหม่สำหรับค่า y(n) ในลำดับถัดไป

2.12 บทสรุป

ในบทนี้ได้กล่าวถึงการประมวลผลสัญญาณที่อยู่ในรูปแบบของสัญญาณอนาลอกและสัญญาญดิจิตอล ซึ่งจะมีคุณสมบัติเด่นที่แตกต่างกันออกไปโดยในกระบวนการประมวลผลสัญญาณจะต้องทำการจัดรูปแบบให้เหมาะสมกับระบบนั้นรวมไปถึงเงื่อนไขที่จะต้องพิจารณา ไปจนถึงโครงสร้างของวงจรกรองแบบต่าง ๆ การประมวลผลเชิงเลขด้วยโครงสร้างของคณิตศาสตร์แบบการกระจายซึ่งโครงสร้างดังกล่าวเหมาะสมกับการออกแบบนำไปสร้างในวงจรดิจิตอล โดยในบทนี้ได้กล่าวถึงประเด็นที่สำคัญ ๆ ที่ควรทราบ ซึ่งในปัจจุบันนั้นได้มีเครื่องมือที่ช่วยในการออกแบบวงจรกรองความถี่ทั้งในรูปแบบอนาลอกและดิจิตอลโดยตรงทำให้ลดความยุ่งยากในการหาค่าของสัมประสิทธิ์ของวงจรกรองต้นแบบลงได้อย่างมากและยังสามารถนำไปประยุกต์ใช้ในการออกแบบวงจรครอสโอเวอร์เน็ตเวอร์คซึ่งจะได้กล่าวถึงในบทต่อไป