บทที่ 2

งานวิจัยที่เกี่ยวข้องและทฤษฎีพื้นฐาน

จากแนวความคิดในการออกแบบหน่วยประมวลผลทางคณิตศาสตร์แบบเชื่อมตรง สิ่งแรกที่จะกล่าวถึงคือ ระบบการคำนวณทางคณิตศาสตร์แบบเชื่อมตรง และจากแนวความคิดดังกล่าว การคำนวณจำเป็นจะต้องอาศัยระบบจำนวนที่มีคุณสมบัติการซ้ำซ้อน ดังนั้น ในงานวิจัยนี้จำเป็นต้องศึกษาระบบจำนวนที่เหมาะสมกับการทำงาน จากนั้นจะศึกษาระบบการคำนวณที่ถูกสร้างขึ้นมาภายใต้แนวคิดของการทำงานแบบเชื่อมตรง โดยสนใจเฉพาะการคำนวณที่เป็นพื้นฐานทางคณิตศาสตร์ นั่นคือ การบวก การลบ การคูณ เท่านั้น

2.1 **การคำนวณแบบเชื่อมตรง** (on-line computation)

การคำนวณแบบเชื่อมตรงถูกนำเสนอขึ้นครั้งแรกในปี ค.ศ. 1977 โดย เออเซอโกแวก และทริเวอร์ดี (Ercegovac and Trivedi) ในงานวิจัย [1] แนวความคิดของการทำงานอาศัยการคำนวณแบบลำดับ (serial computation) ซึ่งมีความซับซ้อนของการคำนวณที่น้อยกว่าการคำนวณแบบขนาน (parallel computation) ซึ่งต่อมาได้มีการนำมารวมกับโครงสร้างแบบท่อตรง (pipeline structure) เพื่อทดแทนเวลาที่เสียไปกับการคำนวณแบบลำดับ หมายความว่าถ้าการคำนวณทั้งหมดสามารถทำงานได้แบบลำดับ คือแทนที่จะต้องรอให้การคำนวณในคำสั่งก่อนหน้าเสร็จก่อนแล้วค่อยเริ่มคำนวณในคำสั่งถัดไป การทำงานแบบท่อตรงจะเริ่มต้นการคำนวณในคำสั่งถัดไปได้เลย โดยเร็วที่สุดเท่าที่จะเป็นไปได้ นั่นคือการคำนวณจะทำบนผลลัพธ์

สำหรับการคำนวณแบบลำดับ โดยปกติแล้วการบวก การลบ และการคูณจะเริ่มการคำนวณจากหลักหรือดิจิตที่มีเลขนัยสำคัญต่ำสุดก่อน (least significant digit first, LSDF) แต่สำหรับการหารจะเริ่มจากดิจิตที่มีเลขนัยสำคัญมากสุดก่อน (most significant digit first, MSDF) ดังนั้นจึงทำให้การคำนวณแบบลำดับ ไม่สามารถดำเนินการแบบท่อตรงไปพร้อมกันทุกตัวดำเนินการได้ เพราะการหารนั้นจำเป็นต้องรอให้การทำงานของตัวดำเนินการก่อนหน้าเสร็จก่อน ทั้งนี้การดำเนินการแบบท่อตรงนั้น ตัวดำเนินการทุกตัวจำเป็นต้องมีทิศทางของการคำนวณไปในทิศทางเดียวกันทั้งหมด จากงานวิจัยของอเวเซียนีส (Aviezienis) ในปี ค.ศ. 1961 [2] ทำให้เห็นได้ว่า การคำนวณแบบลำดับของตัวดำเนินการพื้นฐานทางเลขคณิตทุกตัวดำเนินการสามารถทำงานได้ในทิศทางเดียวกันโดยเริ่มจากดิจิตที่มีนัยสำคัญสูงสุด ด้วยเหตุนี้จึงเป็นที่มาของการคำนวณทางคณิตศาสตร์แบบเชื่อมตรง

ในการคำนวณทางคณิตศาสตร์แบบเชื่อมตรงนั้น เมื่อเริ่มต้นการคำนวณ ตัวดำเนินการคำนวณจะทำการประมวลผลโดยรับข้อมูลนำเข้า (input) ทีละดิจิต และให้ผลลัพธ์หรือข้อมูลออก (output) ทีละดิจิตเช่นกัน นั้นคือการคำนวณแบบตำแหน่งต่อตำแหน่ง

อาจไม่เป็นความจริงเสมอที่ผลลัพธ์ในดิจิตแรกจะสามารถแสดงผลได้จากการประมวลผลของข้อมูลนำเข้าเพียงดิจิตเดียว ตัวดำเนินการส่วนใหญ่ผลลัพธ์ในดิจิตแรกสามารถคำนวณได้ต้องอาศัยข้อมูลนำเข้ามากกว่าหนึ่งดิจิต แต่เมื่อผลลัพธ์ดิจิตแรกถูกผลิตออกมาแล้ว ผลลัพธ์ดิจิตถัดไปจะถูกผลิตออกมาทีละดิจิตที่รับข้อมูลนำเข้าทีละดิจิต จำนวนของดิจิตของข้อมูลนำเข้าที่น้อยที่สุดที่ต้องถูกใช้ในการประมวลผลเพื่อให้สามารถผลิตผลลัพธ์ในดิจิตแรกได้นั้นเรียกว่า *ค่าความหน่วงเชื่อมตรง* (on-line delay, δ) ดังนั้นอาจกล่าวได้ว่า ผลลัพธ์จำนวน n ตำแหน่งแรกนั้นสามารถคำนวณออกมาได้จากตัวถูกดำเนินการ $n + \delta$ ตำแหน่งแรก จากงานวิจัย [x] พบว่า ค่าความหน่วงเชื่อมตรงสำหรับตัวดำเนินการพื้นฐานทางเลขคณิตเป็นเลขจำนวนเต็มที่มีค่าน้อยมาก เช่น การบวกแบบเชื่อมตรงบนเลขฐานสองซ้ำซ้อนของอเวเซียนีสมีค่าความหน่วงเชื่อมตรงเป็นสองเท่านั้น การคูณแบบเชื่อมตรงบนเลขฐานสองซ้ำซ้อนของอเวเซียนีสมีค่าความหน่วงเชื่อมตรงเป็นสองเช่นกัน เป็นต้น

ตัวดำเนินการแบบเชื่อมตรงสามารถให้นิยามได้ชัดเจนดังนี้

กำหนดให้ β เป็นค่าฐานของระบบจำนวน

D และ E เป็นเซตจำกัดของตัวเลข (finite digit sets)

ตัวดำเนินการแบบเชื่อมตรง φ คือ การประมวลผลจากข้อมูลนำเข้า X ในระบบจำนวน (β, D) และให้ผลลัพธ์เป็น Y ในระบบจำนวน (β, E) สามารถให้อยู่ในรูปของฟังก์ชันได้ดังนี้

$$\varphi : D^{N} \rightarrow E^{N}$$

โดยที่ D^{N} และ E^{N} หมายถึงเซตของลำดับของดิจิตที่มีความยาวไม่จำกัดบน Dและ E ตามลำดับ ถ้ากำหนดให้ $X = (x_m x_{m-1} x_{m-2} \ldots)_\beta$ และ $Y = (y_m y_{m-1} y_{m-2} \ldots)_\beta$ โดยที่ สำหรับทุกค่าของ $j \leq m$ และ x_j และ y_j เป็นสมาชิกใน D และ E ตามลำดับ นั่นคือ

$$\varphi(x_m x_{m-1} x_{m-2} \ldots) = y_m y_{m-1} y_{m-2} \ldots$$

เนื่องจากฟังก์ชัน φ เป็นฟังก์ชันการคำนวณแบบเชื่อมตรง และถ้ากำหนดให้ δ เป็นค่าความหน่วงเชื่อมตรง สำหรับทุกค่าของ $k \leq m - \delta$ จะมีฟังก์ชัน ϕ_k ที่มีคุณสมบัติดังนี้

$$\phi_k : D^{k+\delta} \rightarrow E$$

และ $$\phi_k(x_m x_{m-1} x_{m-2} \ldots x_k) = y_{k+\delta}$$

อัลกอริธึมของการคำนวณในรูปทั่วไปของการคำนวณแบบเชื่อมตรง ได้ถูกเสนอในรายละเอียดแล้วโดย ฟรูนี (Frougny) ในงานวิจัย [6,7]

2.2 **ระบบจำนวน** (number system)

ระบบจำนวนมีหลายรูปแบบ ทั้งนี้ขึ้นอยู่กับความเหมาะสมของการนำไปใช้งาน สำหรับในงานวิจัยนี้จะขอกล่าวถึงเฉพาะระบบจำนวนเลขฐาน (radix number systems) เท่านั้น ซึ่งระบบจำนวนแบบนี้เขียนแทนด้วย (β, D) ซึ่งประกอบด้วยเลขฐาน β โดยที่ β สามารถเป็นได้ทั้งจำนวนจริงและจำนวนเชิงซ้อน ที่มีขนาด $\|\beta\| > 1$ และประกอบด้วยเซตจำกัดของตัวเลขหรือดิจิต D ที่ดิจิตเป็นได้ทั้งจำนวนจริงและจำนวนเชิงซ้อน รูปแบบการแทนจำนวน X ในระบบจำนวน (β, D) หรือบางครั้งถูกเรียกว่า *รูปแบบแทนแบบเบต้า* (β-representation) ของจำนวน X ใน D แสดงได้ดังนี้

$$X = (x_m x_{m-1} x_{m-2} \ldots x_1 x_0 . x_{-1} x_{-2} \ldots)_\beta$$

โดยที่ x_j สมาชิกใน D สำหรับทุกค่าของ $j \le m$ และ *ค่าเชิงเลขคณิต* (numerical value) ของจำนวน X ในฐาน β แสดงได้โดย

$$\|X\| = \sum_{j=m}^{-\infty} x_j \beta^j$$

ในกรณีที่เซตจำกัดของตัวเลข D สามารถถูกบรรยายให้อยู่ในรูปของ $\{0, 1, 2, \ldots, \beta-1\}$ เซตจำกัดของตัวเลขนี้จะถูกเรียกว่า เซตจำกัดของตัวเลขแบบคาโนนิค (canonical digit set)

2.3 **ระบบจำนวนแบบมีเครื่องหมาย** (Signed digit number systems)

ในปี ค.ศ. 1961 อเวเซียนีส [2] ได้เสนอแนวคิดเรื่อง *ระบบจำนวนแบบมีเครื่องหมาย* (Signed digit number systems) โดยมีวัตถุประสงค์เพื่อจำกัดการแพร่ของตัวทด (carry propagation) ของการคำนวณ ซึ่งมีผลทำให้การคำนวณแบบขนานสามารถทำงานได้ แนวคิดของระบบจำนวนแบบนี้คือ ดิจิตสามารถมีเครื่องหมายได้ทั้งบวกและลบ รูปแบบทั่วไปของระบบแทนจำนวนแบบมีเครื่องหมายสามารถนิยามได้ดังนี้

กำหนดให้ β เป็นเลขฐานจำนวนเต็มบวกที่มากกว่า 1

D เป็นเซตจำกัดของตัวเลขหรือดิจิตที่เป็นจำนวนเต็ม

ระบบจำนวน (β, D) จะถูกเรียกว่า *ระบบจำนวนแบบมีเครื่องหมายแบบปกติ* (ordinary signed digit number system, OSD) ถ้า เซตของตัวเลข D สามารถเขียนแทนได้ด้วย

$$D = \{-a, -a+1, -a+2, \ldots, 0, \ldots, a\}$$

โดยที่ $\left\lceil \beta/2 \right\rceil \le x \le \beta - 1$

ในกรณีที่ จำนวนดิจิตทั้งหมดใน *D* มีมากกว่าเลขฐาน ระบบจำนวนที่เกิดขึ้นจะมีคุณสมบัติเป็น *ระบบจำนวนซ้ำซ้อน* (redundant number system) ซึ่งหมายถึง มีอย่างน้อยหนึ่งจำนวนที่มี *รูปแบบแทนจำนวนแบบจำกัด* (finite number representation) มากกว่าหนึ่งแบบ

ตัวอย่างที่ 2.1 *ในระบบจำนวนฐานสอง และเซตของตัวเลข { -1, 0, 1 } จำนวนเต็ม 5 สามารถเขียนด้วยรูปแบบแทนจำนวนได้มากกว่าหนึ่งแบบ*

5 เขียนได้เป็น $(101)_2$ หรือ $(1\text{-}101)_2$ เป็นต้น

ระบบจำนวนแบบมีเครื่องหมายนี้ ได้ถูกศึกษาอย่างละเอียดดังปรากฏในงานวิจัยของ พาร์ฮามี (Parhami) [3] ในปี ค.ศ. 1990 ซึ่งได้มีการเสนอ ระบบจำนวนแบบทั่วไปที่มีเครื่องหมาย (general signed digit,GSD) โดยเซตของตัวเลข *D* ไม่จำเป็นต้องมีรูปแบบสมมาตร โดยสามารถแสดงได้ด้วย $D = \{ a, -a+1, -a+2, \ldots, 0, \ldots, b-1, b \}$ เมื่อ a เป็นจำนวนเต็มที่น้อยกว่า b

ตัวอย่างที่ 2.2 *ในระบบจำนวนฐานห้า และเซตของตัวเลข { -2, -1, 0, 1, 2, 3 } จำนวนเต็ม 13 สามารถเขียนด้วยรูปแบบแทนจำนวนได้มากกว่าหนึ่งแบบ*

13 เขียนได้เป็น $(3\text{-}2)_2$ หรือ $(23)_2$ เป็นต้น

การแสดงผลจำนวนที่มีเครื่องหมาย

ในกรณีนี้กำหนดให้เลขฐานมีค่าเท่ากับสอง และเซตจำกัดของตัวเลข คือ {-1, 0, 1} โดยการแสดงดิจิตแต่ละตัวในระดับบิต เรียกว่า *การแทนค่าแบบจำกัดการทด* (borrow – save) ซึ่งหนึ่งดิจิตจะประกอบด้วยสองบิตคือ a_p เรียกว่าบิตบวก และ a_m เรียกว่าบิตลบ โดยค่าของดิจิตได้จากการคำนวณ $a_p - a_m$ นั้นคือ

ดิจิต 1 แสดงด้วยตัวเลขในระดับบิต จะได้ (1, 0)

ดิจิต -1 แสดงด้วยตัวเลขในระดับบิต จะได้ (0, 1)

ดิจิต 0 สามารถแสดงด้วยตัวเลขในระดับบิตได้สองแบบ คือ (0, 0) หรือ (1, 1)

นอกจากนี้ แนวทางในการประยุกต์สร้างระบบจำนวนซ้ำซ้อนนั้นยังได้ถูกนำไปใช้กับระบบจำนวนเชิงซ้อนด้วยเช่น ระบบจำนวนเชิงซ้อนของเพนเนย์ (Penney) [4] ที่ใช้เลขฐาน $\beta = -1+i$ และเซตของตัวเลข $D = \{ 0, 1 \}$ ซึ่งได้รับการพิสูจน์แล้วว่า จำนวนเชิงซ้อนแต่ละจำนวนมีรูปแบบแทนจำนวนแบบจำกัดได้ในระบบนี้และมีเพียงรูปแบบเดียวเท่านั้น แต่เมื่อเปลี่ยนเซตของตัวเลขให้

เป็นดิจิตแบบมีเครื่องหมาย E = { -1, 0, 1 } ระบบจำนวนใหม่นี้จะมีคุณสมบัติซ้ำซ้อนเช่นกันดังแสดงได้ในตัวอย่างที่ 2.3

ตัวอย่างที่ 2.3 *ในระบบจำนวนฐานห้า และเซตของตัวเลข { -2, -1, 0, 1, 2, 3 } จำนวนเชิงซ้อน 7 – 2i สามารถเขียนด้วยรูปแบบแทนจำนวนได้มากกว่าหนึ่งแบบ*

รูปแบบแทนจำนวนของ 7 – 2i ด้วย $(101001)_{-1+i}$ สามารถพิสูจน์ได้ดังนี้

ตารางที่ 2.1แสดงการแทนค่า 7-2i ด้วย $(101001)_{-1+i}$

N	$(-1+i)^N$	X_i	$X_i(-1+i)^N$
0	1	1	1
1	-1+i	0	0
2	-2i	0	0
3	2(i+1)	1	2(i+1)
4	-4	0	0
5	4(1-i)	1	4(1-i)

$$\|x\| = \sum_{j=m}^{-\infty} x^j \beta^j = 1 + 2(i+1) + 4(i-1) = 7 - 2i$$

ระบบจำนวนเชิงซ้อนของคณุท (Knuth) [5] ก็เป็นอีกระบบจำนวนที่สามารถนำประยุกต์ให้เป็นระบบจำนวนแบบมีเครื่องหมายได้ ทั้งนี้การแสดงตัวเลขด้วย เลขฐาน $\beta = i\sqrt{r}$ โดยที่ r เป็นจำนวนเต็มที่มากกว่าหรือเท่ากับ 2 และมีเซตจำกัดของตัวเลขแบบคาโนนิคอล นั่นคือ $C = \{c \in I \mid 0 \leq c \leq r-1\}$ ซึ่งคณุทได้พิสูจน์ว่าจำนวนเชิงซ้อนทุกจำนวนสามารถเขียนให้อยู่ในระบบนี้ได้

ตัวอย่างที่ 2.4 *ในระบบจำนวนฐาน $i\sqrt{4}$ และเซตของตัวเลข { 0, 1, 2, 3 } จำนวนเชิงซ้อน 5 + 17i สามารถเขียนด้วยรูปแบบแทนจำนวนได้*

จำนวน 5+17i สามารถแสดงด้วย $(102213.2)_{i2}$ ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงการแทนค่า 5+17i

N	$(i2)^N$	X_i	$X_i(i2)^N$
-1	(-1/2)i	2	-1i
0	1	3	3
1	2i	1	2i
2	-4	2	-8
3	-8i	2	-16i
4	16	0	0
5	32i	1	32i

ทั้งนี้เนื่องจาก $\sum_{j=-1}^{5} x_h\left(i\sqrt{4}\right)^j$ = -1i + 3 + 2i – 8 – 16i + 0 + 32i = 5 + 17i

2.4 **วงจรบวกแบบเชื่อมตรง** (On-line adder)

วงจรบวกแบบเชื่อมตรงได้ถูกนำเสนอโดย แอลง กูโยต์, ยวน เฮร์เรรอส และ ฌอง มิเชล มูลเลอร์ ดังปรากฏใน [10] วงจรการบวกแบบเชื่อมตรงนี้ ได้ถูกพัฒนาขึ้นมาจากแนวความคิดของการบวกแบบขนาน แต่ได้มีการลดขนาดลงให้ทำงานครั้งละดิจิต ดังนั้นค่าทดระหว่างดิจิตระหว่างการคำนวณจึงถือเสมือนเป็นการทดกลับเข้ามาในวงจรแบบวนซ้ำ

ส่วนประกอบหลักของวงจรบวก คือวงจรย่อยที่เรียกว่า พีพีเอ็มเซล (PPM cell) ซึ่งหมายถึงวงจรการบวกสามบิต แสดงได้ดังสมการต่อไปนี้

$$a + b - c = 2d - e$$

โดยที่ ข้อมูลนำเข้าคือ a b และ c ซึ่งมีค่าเป็น 0 หรือ 1 เท่านั้น และผลลัพธ์คือการบวกบนฐานสองโดย d หมายถึงค่าทด และ e หมายถึงผลลัพธ์ โดยที่ทั้ง d และ e มีค่าเป็น 0 หรือ 1 เท่านั้น ผลการคำนวณสามารถแสดงได้ดังตารางที่ 2.3

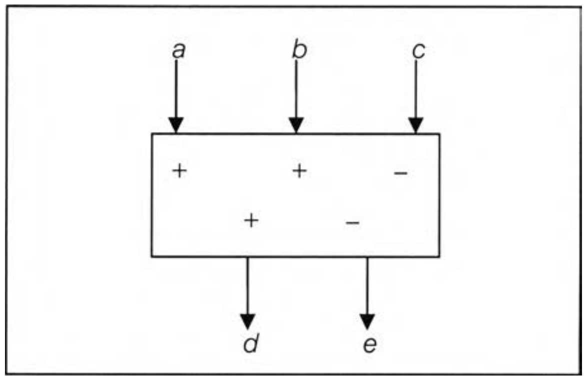
ตารางที่ 2.3 แสดงผลลัพธ์ของพีพีเอ็มเซล

ข้อมูลนำเข้า			ผลลัพธ์	
a	*b*	*c*	*d*	*e*
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

โดยพีพีเอ็มเซลประกอบด้วยวงจรตรรกสองส่วน คือคำนวณค่า d และ e ซึ่งสามารถแสดงด้วยสมการทางตรรกได้ดังนี้

$$d = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$$
$$e = a \oplus b \oplus c$$

และวงจรพีพีเอ็มเซลสามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 วงจรพีพีเอ็มเซล

เนื่องจากวงจรการบวกแบบเชื่อมตรงมีลำดับการทำงานทีละดิจิต เริ่มต้นจากดิจิตที่มีนัยสำคัญสูงสุด โดยแต่ละดิจิตของข้อมูลนำเข้าจะประกอบด้วยสองบิต ดังนั้นจะมีจำนวนบิตเข้าทั้งหมดสี่บิต และผลลัพธ์จะประกอบด้วยจำนวนบิตสองบิต โดยมีรายละเอียดดังนี้

On-line adder algorithm

Input addend $A = \sum_{j=k}^{-\infty} (a_{p,j} - a_{m,j}) 2^j$

sequence of ($a_{p,j}$ and $a_{m,j}$) started from $j = k$

adder $B = \sum_{j=k}^{-\infty} (b_{p,j} - b_{m,j}) 2^j$

sequence of ($b_{p,j}$ and $b_{m,j}$) started from $j = k$

Output Result $S = \sum_{j=k+2}^{-\infty} (s_{p,j} - s_{m,j}) 2^j = A + B$

sequence of ($s_{p,j}$ and $s_{m,j}$) started from $j = k + 2$

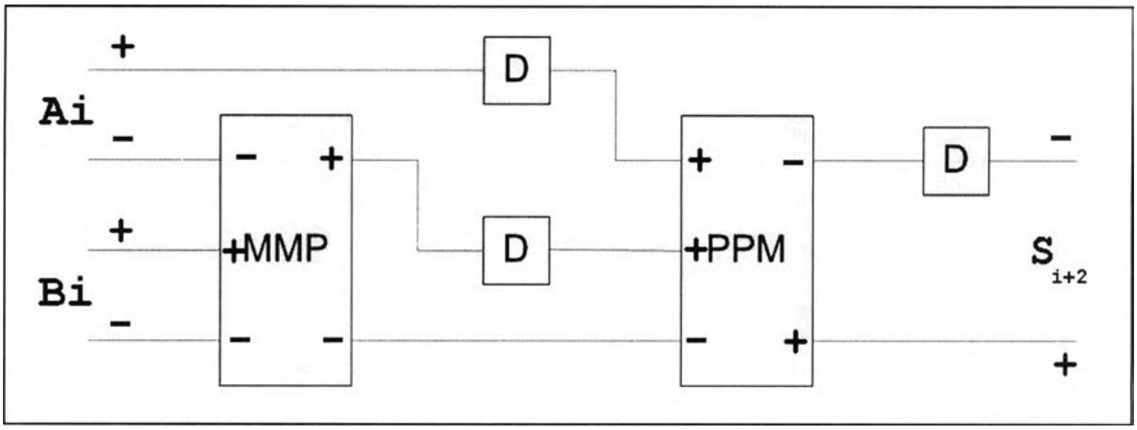
For each value of j started from k

Find x_j and y_{j-1} such that $y_{j-1} - 2x_j = b_{p,j} - b_{m,j} - a_{m,j}$

Find $s_{p,j+2}$ and $s_{m,j+1}$ such that $2s_{p,j+2} + s_{m,j+1} = y_j + a_{p,j+1} - x_j$

Decrease j by one

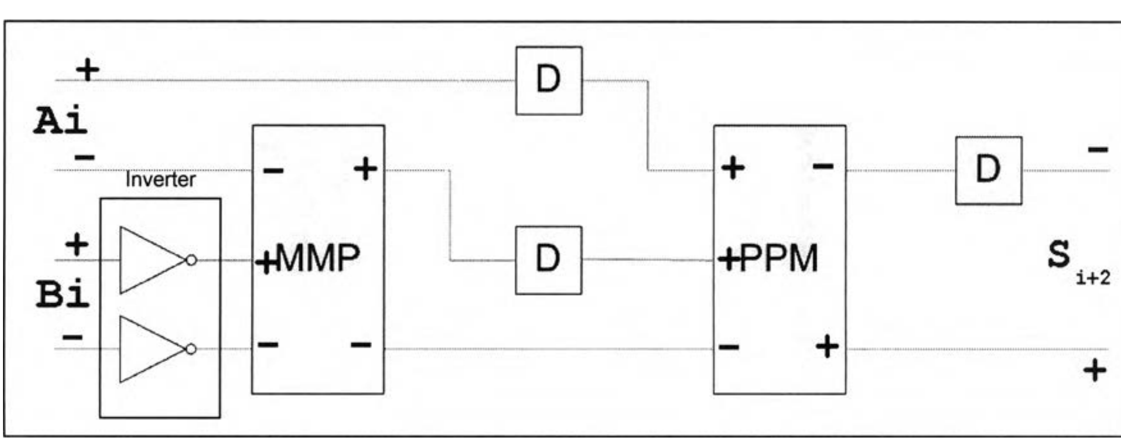
รูปแบบการคำนวณการบวกแบบเชื่อมตรงสามารถแสดงให้อยู่ในรูปของวงจรการบวกได้ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 วงจรบวกแบบเชื่อมตรง

2.5 **วงจรลบแบบเชื่อมตรง** (On-line subtract)

เนื่องจากระบบจำนวนแบบมีเครื่องหมายมีลักษณะพิเศษคือ จำนวนผบผันสำหรับการบวกของจำนวนใดๆ สามารถสร้างได้จากการเปลี่ยนดิจิตให้เป็นดิจิตที่มีเครื่องหมายตรงข้าม เช่น จาก -1 เป็น 1 เป็นต้น ดังนั้นการลบสามารถดำเนินการได้โดยทำตัวดำเนินการบวกด้วยจำนวนผบผันของตัวลบแทน ดังนั้นตัวดำเนินการลบแบบเชื่อมตรงจึงสามารถใช้โครงสร้างเดียวกับตัวดำเนินการบวก และเพิ่มในส่วนของการสร้างจำนวนผบผันสำหรับการบวกของตัวลบเท่านั้น นั่นคือ การใส่วงจรกลับค่า (inverter) เพิ่มเข้าไปเท่านั้น ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 วงจรลบแบบเชื่อมตรง

2.6 **วงจรคูณแบบเชื่อมตรง** (On-line multiplier)

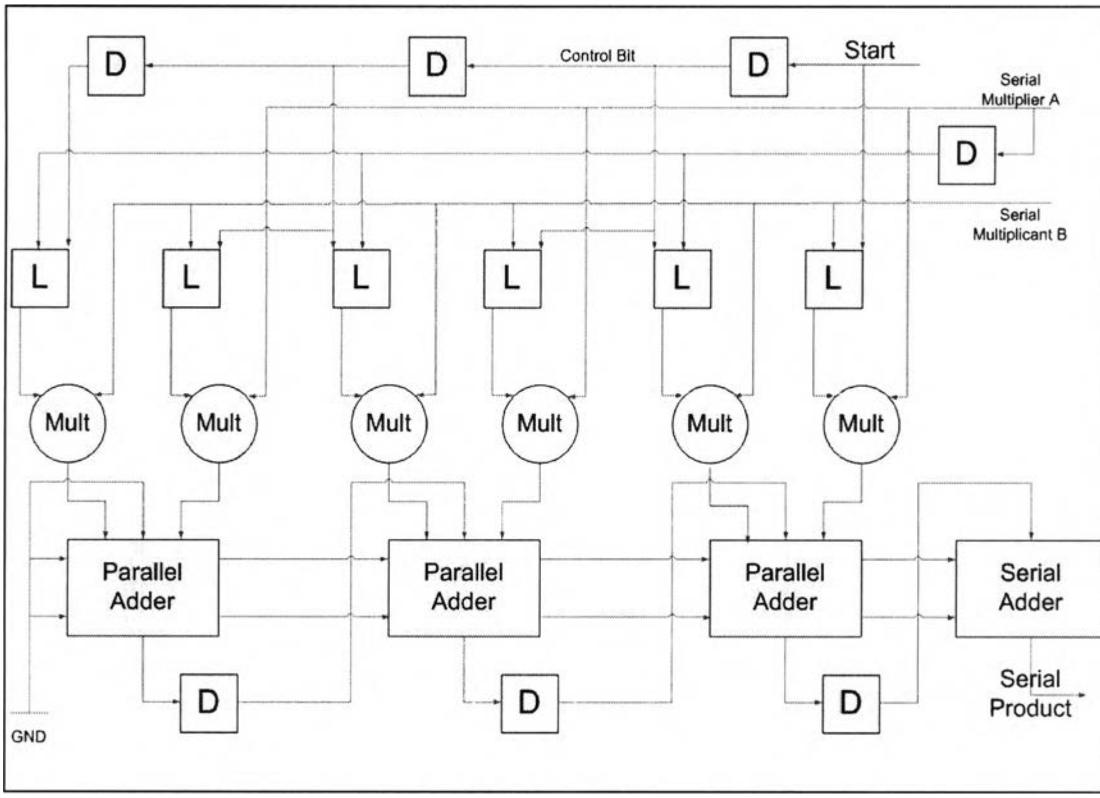
วงจรการคูณแบบเชื่อมตรงได้ถูกพัฒนาขึ้นโดย เออเซอโกแวก และ ทริเวอร์ดี ในปี ค.ศ. 1977 ดังกล่าวไว้ใน [1] โดยอัลกอริทึมการคำนวณแบบเชื่อมตรงมีหลักการทำงานเริ่มต้นจากรับข้อมูลตัวตั้งและตัวคูณครั้งละดิจิต นำมาคำนวณหาผลคูณที่ได้ทั้งหมด.เทียบกับข้อมูลที่อ่านมาทั้งหมด จากนั้นเลือกดิจิตของผลลัพธ์ที่ให้ค่าความแตกต่างระหว่างผลลัพธ์ที่อกทั้งหมดและค่าผลคูณจริงที่ได้น้อยที่สุด ทั้งนี้การคูณแบบเชื่อมตรงบนฐานสองจะมีค่าความหน่วงเชื่อมตรงเท่ากับสอง

รูปที่ 2.4 แสดงการคูณแบบเชื่อมตรง เมื่อข้อมูลนำเข้าคือ ตัวตั้ง $A = a_m a_{m-1} a_{m-2} \ldots$ และตัวคูณ $B = b_m b_{m-1} b_{m-2} \ldots$ จากนั้น ทุกรอบที่ดิจิตของข้อมูลนำเข้าทั้งสองจำนวน (a_j และ b_j) ถูกอ่านเข้ามา ผลคูณ $A_j b_j = a_m b_j + a_{m-1} b_j + \ldots + a_j b_j$ และ $B_{j+1} a_j = b_m a_j + b_{m-1} a_j + \ldots + b_{j+1} a_j$ จะถูกคำนวณ ต่อจากนั้น จะนำผลคูณทั้งสองไปรวมกับ ผลบ่วยย่อยจากรอบที่ผ่านมา เพื่อนำไปคำนวณหาผลลัพธ์ s_{j+2} ต่อไป โดยคำนวณจากค่าผลต่างของผลลัพธ์ทั้งหมดที่ผลิตออกทั้งหมด นั่นคือ $s_m 2^m + s_{m-1} 2^{m-1} + \ldots + s_{j+3} 2^{j+3}$ และ ผลบวกสะสมทั้งหมด จากนั้นผลบวกสะสมทั้งหมดจะนำไปเป็นผลรวมย่อยสำหรับรอบถัดไป

$$
\begin{array}{cccccccccc}
 & & & & & & a_m & a_{m-1} & a_{m-2} \;\ldots\; a_j & \ldots \\
 & & & & & \times & b_m & b_{m-1} & b_{m-2} \;\ldots\; b_j & \ldots \\
\hline
 & & & & a_m b_j & a_{m-1} b_j & \ldots & \ldots & a_j b_j & \\
 & & & \ldots & & & a_j b_{j-1} & & & \\
 & & a_m b_{m-2} & a_{m-1} b_{m-1} & \ldots & \ldots & & & & \\
 & a_m b_{m-1} & a_{m-1} b_{m-1} & & & \ldots & & & & \\
a_m b_m & a_{m-1} b_m & a_{m-2} b_m & & a_j b_m & & & & & \\
\hline
s_m \;\; s_{m-1} & s_{m-2} & \ldots & s_{j+2} & & & & & & \\
\hline
\end{array}
$$

รูปที่ 2.4 การคูณแบบเชื่อมตรง

การออกแบบการวงจรการคูณจึงประกอบด้วยสามส่วน คือ ส่วนการคูณ ส่วนการคำนวณผลบวกของแต่ละรอบ และส่วนการบวกผลคูณทั้งหมด ดังแสดงได้ในรูปที่ 2.5



รูปที่ 2.5 วงจรการคูณแบบเชื่อมตรง