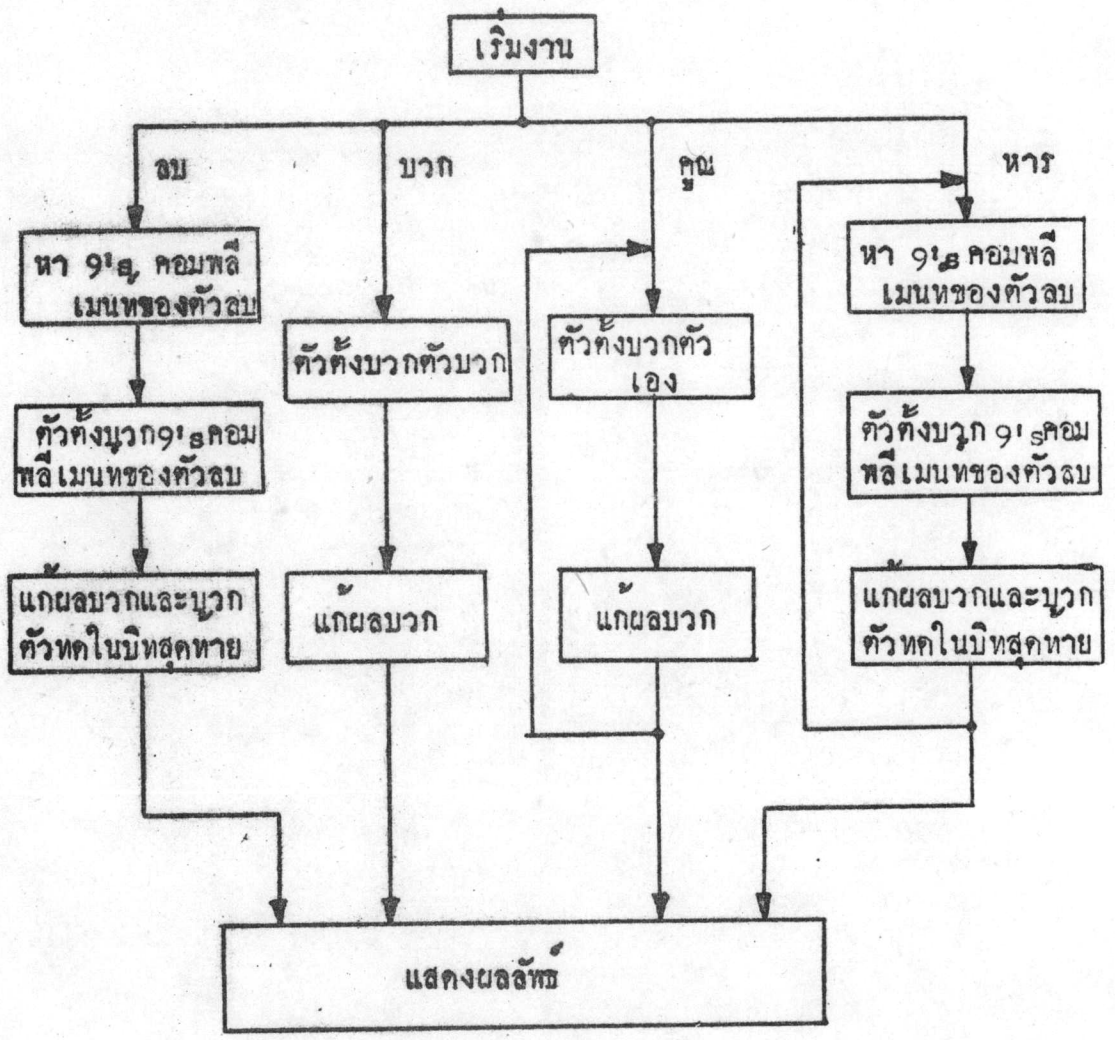
บทที่ 3
ตัวเรื่อง

3.1 วิธีดำเนินการวิจัย

วิธีดำเนินการวิจัย คือวิธีออกแบบวงจรเครื่องคำนวณอีเล็กโทรนิค ขนาดเล็ก ใช้หลักการจากไดอะแกรมของการทำงาน (Flow Diagram) ดังรูปที่ 28



รูปที่ 28 ไดอะแกรมของการทำงานในการบวกลบ คูณและหาร

หลักการทำงานตามไดอะแกรม แบ่งได้เป็น 2 แบบ คือ

1. การบวก (Addition) เนื่องจากการบวกเลข 2 จำนวน เราทราบเครื่องหมายของผลลัพธ์อยู่แล้ว หลังจากเปลี่ยนเลขทศนิยมเป็นเลขโค๊ด บี.ซี.ดี. แล้ว เราสามารถนำตัวบวกมาบวกกับตัวตั้ง แบบการบวกเลขฐานสองได้เลย ตัวอย่าง เช่น

22	0010	0010		
+				
13	0001	0011		
35	0011	0101	คำตอบ คือ 35	

หรือ

125	0001	0010	0101	
+				
313	0011	0001	0011	
438	0100	0011	1000	คำตอบ คือ 438

ตัวอย่าง การบวกข้างบนนี้จะเห็นว่าไม่มีตัวทดไปยังหลักต่อไป บางครั้งผลบวกในเลขโค๊ด บี.ดี.ซี. จะเกิน 1001 (9) ซึ่งไม่สามารถแสดงผลลัพธ์เป็นเลขทศนิยมและไม่ตรงคำตอบ ต้องใช้วิธีแก้ผลบวก โดยบวกด้วย 0110 (6) ดังกล่าวมาแล้ว ในหัวข้อ 2.6 ผลลัพธ์ที่ได้จะเป็นคำตอบที่ถูกต้อง ตัวอย่าง เช่น

25	0010	0101	
+			
37	0011	0111	
62	0101	1100	
		0110	แก้ผลบวกโดยบวกด้วย 6
	0101	0010	
	1		บวกด้วยตัวทด
	0110	0010	คำตอบ คือ 62

หรือ

365	0011	0110	0101	
+				
289	0010	1000	1001	
654	0101	1110	1110	
		0110	0110	แก้ผลบวกโดยบวกด้วย 6
	0101	0100	0100	
	1	1		บวกด้วยตัวทด
	0110	0101	0100	คำตอบ คือ 654

2. การลบ (Subtraction) การลบใช้วิธีนำจำนวนที่มากกว่าเป็นตัวตั้ง ผลลัพธ์จะเป็นบวกเสมอ หลักการลบที่ใช้ในการวิจัยนี้ คือใช้วิธีหาค่า 9's คอมพลีเมนท์ของตัวลบ แล้วจึงนำไปบวกกับตัวตั้ง

9 's คอมพลีเมนท์ (Natural Binary Coded Decimal Complement) ของเลขจำนวนใดได้จากการนำเลขจำนวนนั้นไปลบออกจากเลข 9 เช่น

9 's คอมพลีเมนท์ของ 7 คือ 9 - 7 = 2

9 's คอมพลีเมนท์ของ 34 คือ 99 - 34 = 65 เป็นต้น

เมื่อนำ 9 's คอมพลีเมนท์ของตัวลบไปบวกกับตัวตั้ง แบบการบวกเลขฐานสองโดยใช้วงจรบวกในเรื่องของการบวกแล้ว ผลบวกที่ได้ในแต่ละหลักถ้าเกิน 9 (1001) ก็ต้องแก้ผลบวกเช่นเดียวกับการบวก และในการลบนี้จะมีตัวทดในบิทสุดท้าย (End around carry) เกิดขึ้น ซึ่งต้องนำมารวมกับผลบวกในบิทแรกเสมอ จึงจะได้ผลลัพธ์ที่ถูกต้อง

ตัวอย่างเช่น

25	0010	0101		
-				
13	1000	0110		9's คอมพลีเมนท์ของ 13 คือ 86
12	1010	1011		
	0110	0110		แก้ผลบวกโดยบวกด้วย 6 (0110)
	0001	0001		
		1		ตัวทดจากบิทสุดท้าย
	0001	0010		คำตอบ คือ 12

459	0100	0101	1001	
-				
283	0111	0001	0110	9's คอมพลีเมนท์ของ 283
176	1011	0110	1111	คือ 716
	0110		0110	แก้ผลบวกโดยบวกด้วย 6 (0110)
	0001	0111	0101	
			1	ตัวทดจากบิทสุดท้าย
	0001	0111	0110	คำตอบคือ 176

ถ้าตัวตั้งน้อยกว่าตัวลบ หลักการนี้ก็คงยังใช้ได้ แต่ต้องเปลี่ยนผลบวกที่ได้เป็น 9's คอมพลีเมนท์อีกครั้งหนึ่ง ตัวอย่างเช่น

132	0001	0011	0010	
-				
897	0001	0000	0010	9 's คอมพลีเมนท์ของ
- 765	0010	0011	0100	897 คือ 102 ผลบวกคือ 234
	0111	0110	0101	9 's คอมพลีเมนท์ของ 234 คือ 765

กรณีนี้จะต้องมีพัลซ์ที่แสดงว่าผลลัพธ์เป็น ลบ เมื่อเปลี่ยนผลบวกที่ได้เป็น 9's คอมพลีเมนท์ อีกครั้งหนึ่ง

3. การคูณ (Multiplication) การคูณใช้วิธีการบวกซ้ำ ๆ กัน (Repeated - Addition) เป็นจำนวนครั้งเท่ากับตัวคูณ และถ้าผลบวกของแต่ละครั้งเกิน 1001 (9) ก็ต้องแก้ผลบวก เพื่อจะแสดงผลลัพธ์ (display) ออกมาได้ เช่นเดียวกับการบวกลบ ตัวอย่างเช่น

พระบรมราชูปถัมภ์ สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

7	0111		
x 4	0111		บวกด้วย 7
28	1110		
	0110		แก้ผลบวกโดยบวกด้วย 6
0001	0100		
	0111		บวกด้วย 7
0001	1011		
	0110		แก้ผลบวกด้วย 6
0010	0001		
	0111		บวกด้วย 7
0010	1000		คำตอบ คือ 28

12		0001	0010	
x 23		0010	0011	
36		0011	0110	บวกตัวตั้ง 3 ครั้ง
24	0010	0100		บวกตัวตั้ง 2 ครั้ง
276	0010	0111	0110	คำตอบ คือ 276

คำตอบได้จากการ shift ผลลัพธ์ครั้งที่สอง 4 ครั้ง บวกกับผลลัพธ์ครั้งแรก

4. การหาร (Division) การหารใช้วิธีการลบซ้ำ ๆ กัน (Repeated - Subtraction) โดยใช้ 9's คอมพลีเมนท์ของตัวลบไปบวกกับตัวตั้งและจำนวนครั้งที่ทำการลบ คือผลลัพธ์ ผลจากการลบครั้งสุดท้าย คือเศษ ถ้าผลบวกของการลบในแต่ละครั้งเกิน 1001 (9) ก็ต้องแก้ผลบวกเช่นเดียวกัน ตัวอย่างเช่น

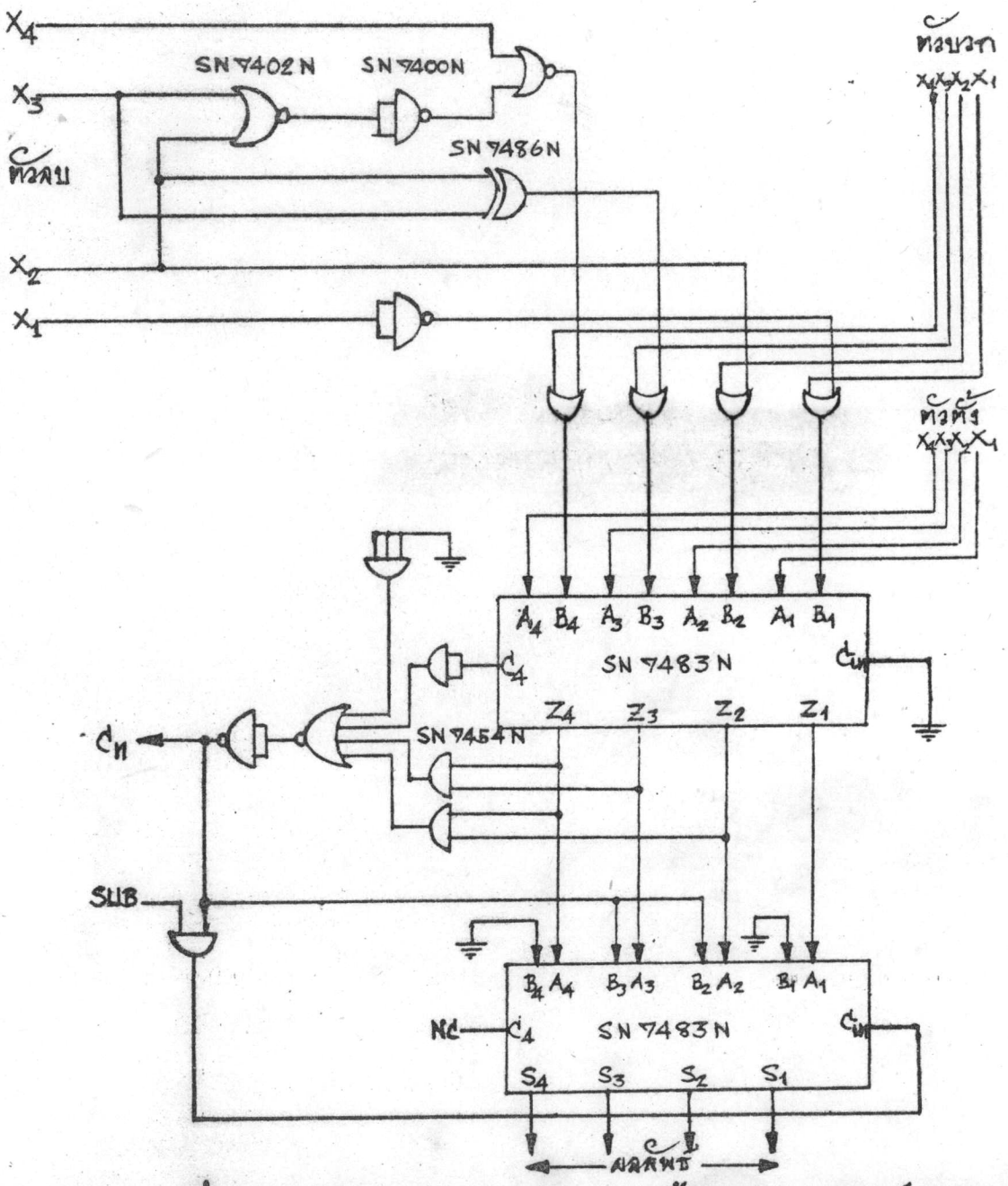
9 หารด้วย 2 = 4 เศษ 1

```
 1001
 0111      บวก 9's คอมพลีเมนท์ของ 2 คือ 7
10000
 0110      แก้ผลบวกโดยบวกด้วย 6
10110
  └→1      บวกด้วย end - around carry
 0111
 0111      บวก 9's คอมพลีเมนท์ของ 2 คือ 7
 1110
 0110      แก้ผลบวกโดยบวกด้วย 6
10100
  └→1      บวกด้วย end - around carry
 0101
 0111      บวก 9's คอมพลีเมนท์ของ 2 คือ 7
 1100
 0110      แก้ผลบวกโดยบวกด้วย 6
10010
  └→1      บวกด้วย end - around carry
 0011
 0111      บวก 9's คอมพลีเมนท์ของ 2 คือ 7
 1010
 0110      แก้ผลบวกโดยบวกด้วย 6
10000
  └→1      บวกด้วย end - around carry
 0001      เศษ คือ 1
```

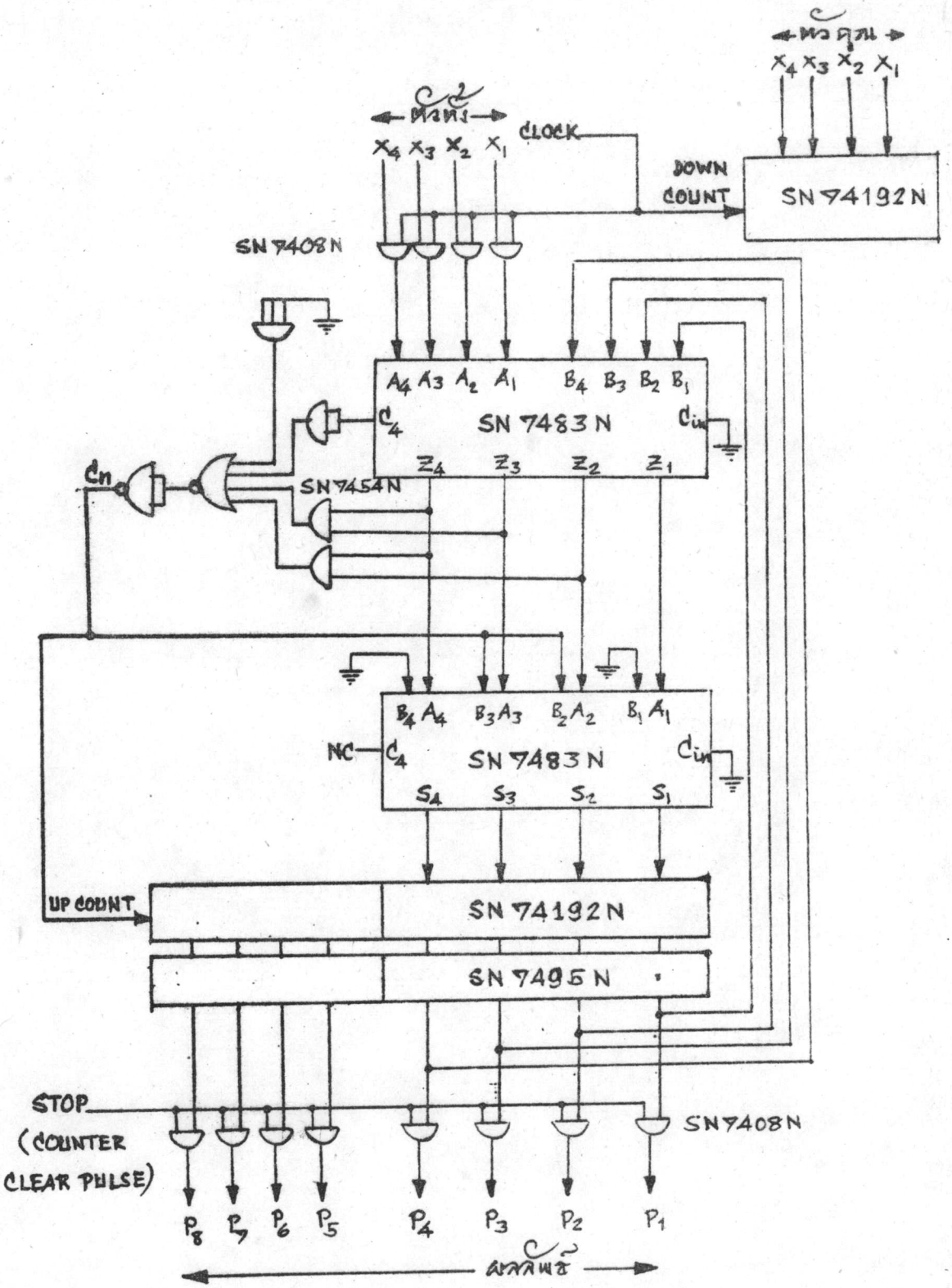
เมื่อไม่มี end - around carry จึงจะหยุดการลบ

3.2 ผลการวิจัย

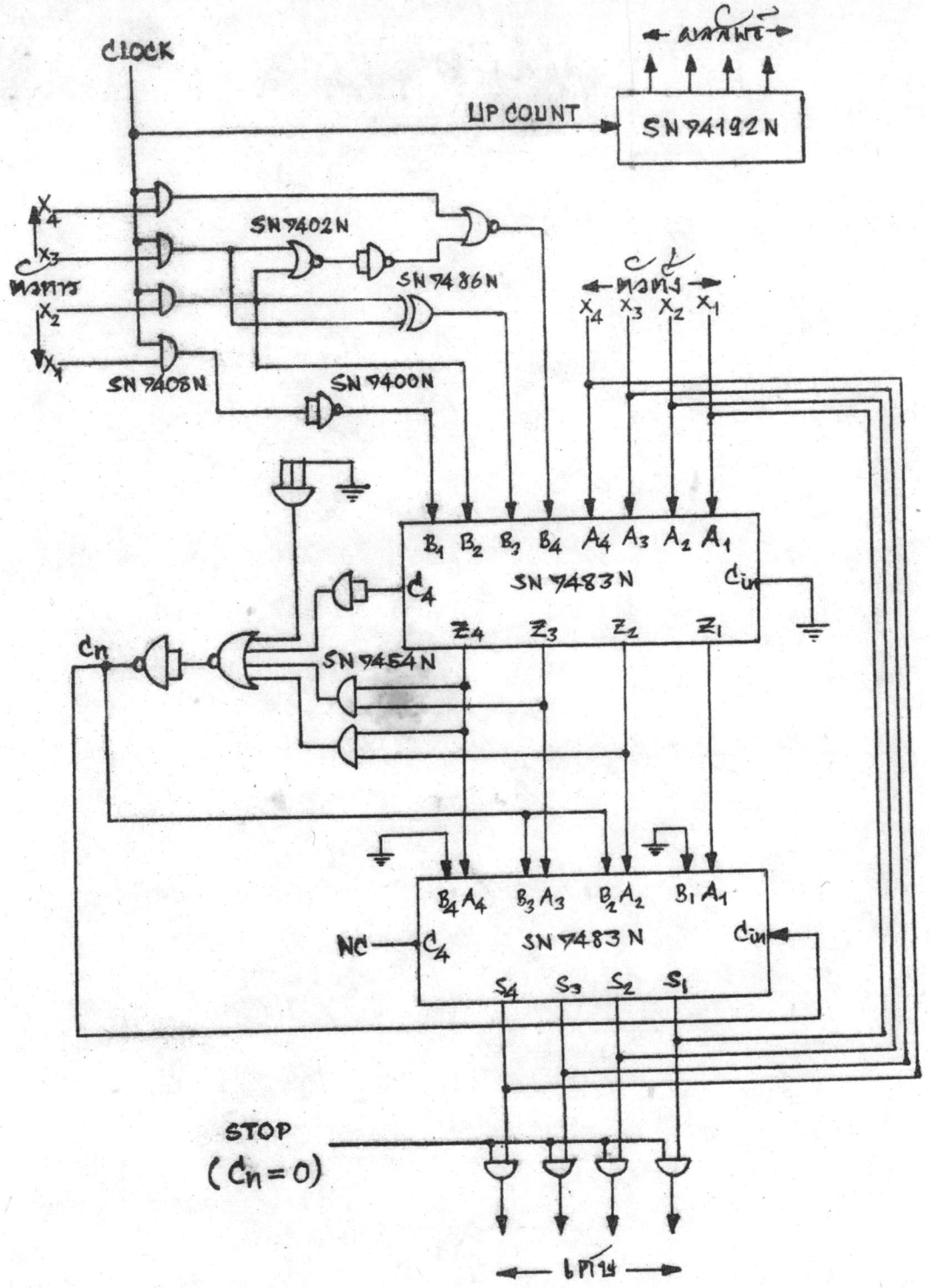
จากหลักการทำงานในหัวข้อของวิธีดำเนินการวิจัย เราสามารถเขียนแบบวงจรในส่วนคำนวณได้ดังรูปที่ 29 เบอร์ของ ICs ที่ใช้ ได้เขียนกำกับไว้ในรูปของวงจร รายละเอียดต่าง ๆ ของ ICs จะดูได้จากภาคผนวก



รูปที่ 29 โลยิคไดอะแกรมของการบวกลบใช้ 9's คอมพลีเมนท์ของตัวลบ



รูปที่ 30 โลยิคไดอะแกรมของการคูณวิธี Repeated - Addition



รูปที่ 31 โลยิคไดอะแกรมของการหารวิธี Repeated - Subtraction

ผลการวิจัยแบ่งเป็น 2 ส่วน คือ

1. ส่วนรับผลและส่วนคำนวณ

ส่วนรับผลและคำนวณมีแสดงในรูปที่ 32 การทำงานใช้วงจรนับโมดูลัส-6 เป็นวงจรควบคุมลำดับของการทำงาน

เมื่อกดคีย์ CLEAR จะทำให้รีจิสเตอร์ทั้ง 3 คือ I/O รีจิสเตอร์, A- รีจิสเตอร์, B - รีจิสเตอร์ และวงจรนับ อยู่ในภาวะ "รีเซ็ท" หรือเคลียร์

เมื่อ Key-in เลขทศนิยมจำนวนแรกที่เป็นตัวตั้ง เลขทศนิยมแต่ละตัวจะถูกเปลี่ยนเป็นโคด บีซีดี ใน DECIMAL TO BCD ENCODER และ load เข้าใน I/O รีจิสเตอร์ ตามลำดับของวงจร นับตั้งแต่ลำดับการนับที่ 1 (CS0) ถึงลำดับการนับที่ 6 (CS5) และ จะแสดงตัวเลข (Display) ในส่วนแสดงผลด้วย

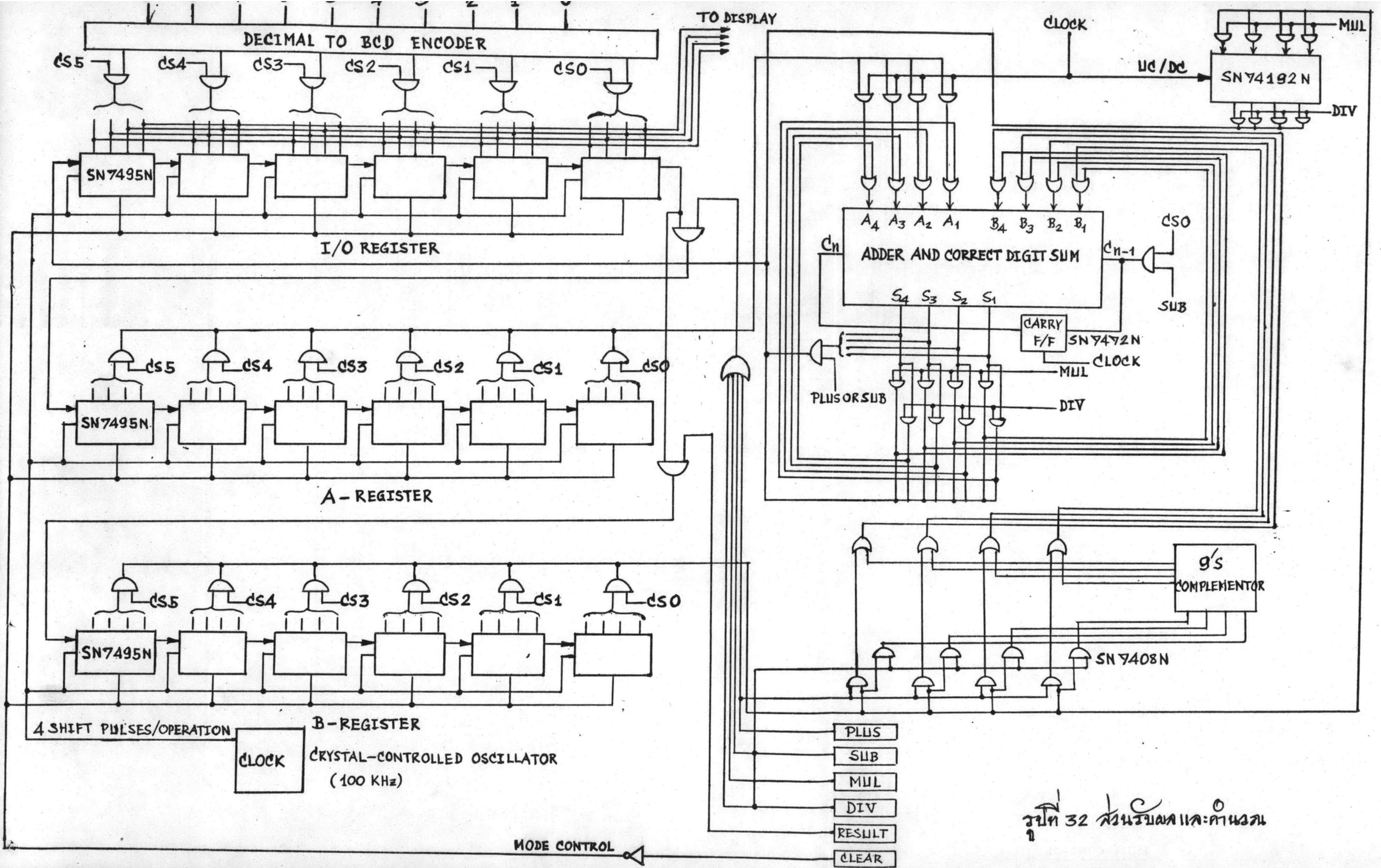
เนื่องจากตัวเลขแต่ละจำนวนที่ load เข้า I/O รีจิสเตอร์ อาจจะมีไม่ถึง 6 ตำแหน่ง เราจึงต้องเคลียร์วงจรนับทุกครั้งก่อนที่จะนำมาใช้อีก

ลำดับต่อไป เมื่อกดคีย์ PLUS หรือ SUB หรือ MUL หรือ DIV วงจรนับจะถูกเคลียร์ และทำให้ข้อมูลใน I/O รีจิสเตอร์ shift right เข้าใน A- รีจิสเตอร์ โดยคล็อคครั้งละ 4 พัลซ์ (4 - pulses per operation) ตั้งแต่ CS0 ถึง CS5

เมื่อ Key-in เลขทศนิยมจำนวนที่สอง ที่อาจจะเป็นตัวบวก, ตัวลบ, ตัวคูณ หรือตัวหาร จะถูกเปลี่ยนเป็นเลขโคด บีซีดี load เข้า I/O รีจิสเตอร์ ตามลำดับของวงจรนับ และแสดงตัวเลขในส่วนแสดงผล เช่นเดียวกัน

ต่อไปเมื่อกดคีย์ RESULT วงจรนับจะถูกเคลียร์ และข้อมูลใน I/O รีจิสเตอร์ จะ shift right เข้าใน B- รีจิสเตอร์ ตามลำดับการนับ ครั้งละ 4 บิท ตั้งแต่ CS0 ถึง CS5 เช่นเดียวกับตอนแรก

ถ้าเป็นการบวก ข้อมูลใน B - รีจิสเตอร์ (B_1, B_2, B_3, B_4) และข้อมูลใน A - รีจิสเตอร์ (A_1, A_2, A_3, A_4) จะ transfer เข้าไป บวกกันในส่วน



รูปที่ 32 ส่วนของแอคคิวมูเลเตอร์

คำนวณ ตั้งแต่ CS0 ถึง CS5 และส่งผลลัพธ์ (S_1, S_2,S_3,S_4) เข้า I/O รีจิสเตอร์ แสดงตัวเลขในส่วนแสดงผล

ถ้าเป็นการลบข้อมูลใน B - รีจิสเตอร์ จะเปลี่ยนเป็น 9's คอมพลีเมนท์ ในวงจร 9's คอมพลีเมนท์ แล้วจึงบวกกับข้อมูลใน A - รีจิสเตอร์ ตัวทดในบิทสุดท้าย (end - around carry) ได้ใช้เป็นตัวทดเข้า (carry -in)ของการบวกครั้งแรกอยู่แล้ว

ถ้าเป็นการคูณ ตัวคูณจะ transfer เข้าวงจรนับ บีซีดี (Synchronous 4 - bit up/down Counter) ซึ่งเรียกว่า Multiplier Counter ตัวตั้งจะ transfer เข้าวงจรบวก เมื่อมีพัลซ์ของการคูณเกิดขึ้น ตั้งแต่ CS0 ถึง CS5.

B- รีจิสเตอร์ แต่ละตัวจะทำหน้าที่เป็น Buffer Register ของ Multiplier Counter เมื่อมีพัลซ์คล็อคเกิดขึ้น Counter จะนับถอยหลัง (down count) บอกจำนวนครั้งที่ตัวตั้งบวกตัวมันเอง จนกระทั่งค่าของ Counter เป็น 0 จึงหยุด ผลคูณที่ได้จะ transfer เข้า Shift Register แล้วจึง shift right เข้า I/O รีจิสเตอร์ และแสดงผลเช่นเดียวกัน

ถ้าเป็นการหาร ตัวหารจะเปลี่ยนเป็น 9's คอมพลีเมนท์ และบวกกับตัวตั้ง พัลซ์ของคล็อคจะทำให้ Counter นับขึ้น (up count) แสดงจำนวนครั้งของการลบ และคือผลหาร ซึ่งจะ shift เข้า I/O รีจิสเตอร์ และแสดงผลออกมา เศษจะถูกปัดทิ้ง

ในการหารที่ตัวตั้งและตัวหาร มีมากกว่า 1 ตำแหน่ง จะต้องเพิ่มจำนวนวงจรบวกเต็ม และวงจรนับ บีซีดี ตามจำนวนตำแหน่งของตัวตั้งและตัวหารด้วย

2. ส่วนแสดงผล

ในการแสดงตัวเลขทศนิยม ใช้วงจร Light Emitting Diode Display ซึ่งประกอบด้วย Diode P-N Junction 7 Segments

โดยการเลือกคอมบิเนชั่น (Combinations) ของส่วนของภาพทั้ง 7 (a,b,c,d,e,f,g,) จะสามารถแสดงตัวเลขทศนิยมจาก 0 ถึง 9 ดังได้กล่าวมาแล้วในหัวข้อของการถอดโค๊ด และการแสดงตัวเลขโดยใช้ภาพ 7 ส่วน

การทำงานของ Multiplexing of Displays

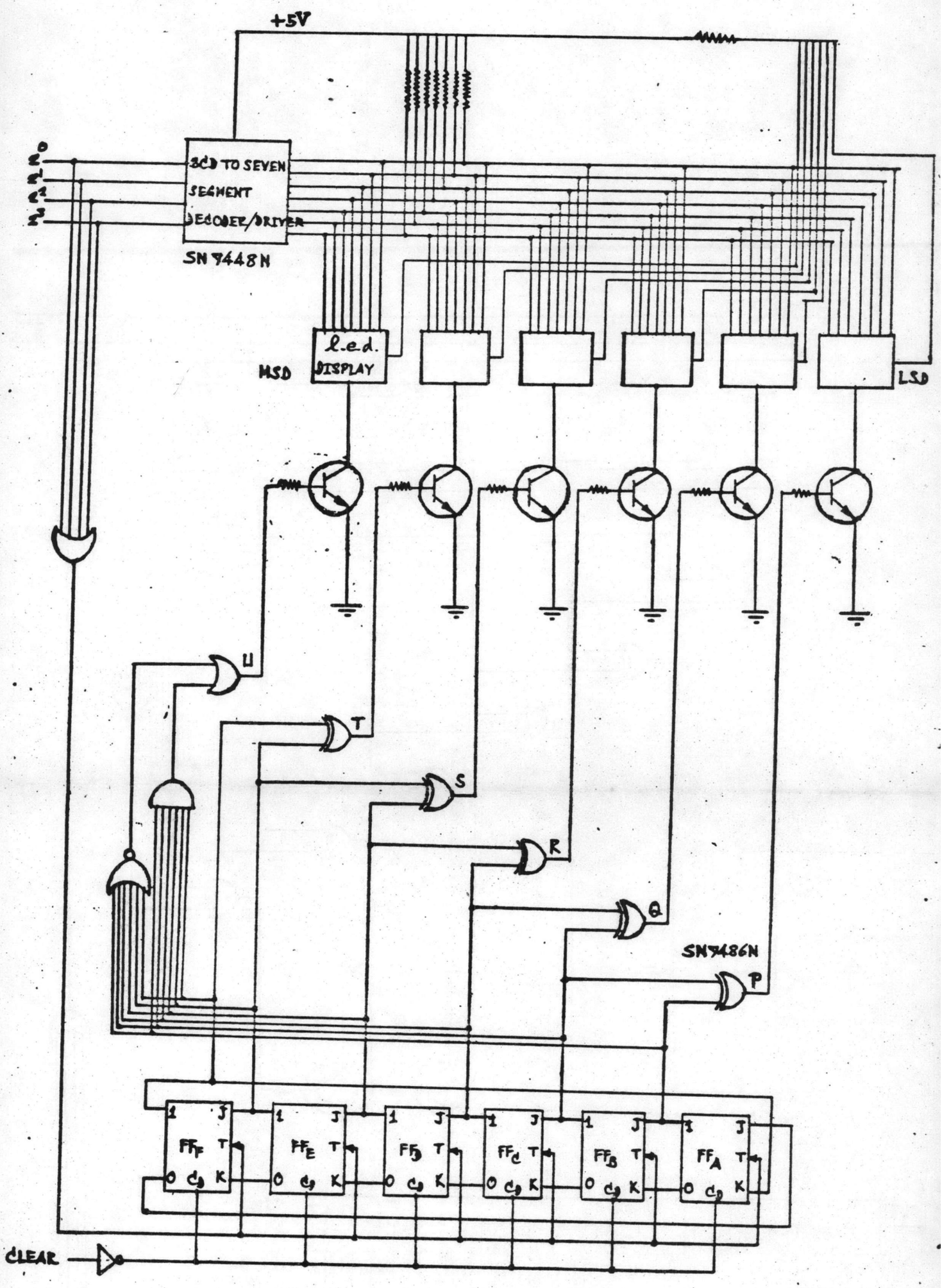
อินพุท บีซีดี ที่ป้อนเข้า BCD-to-Seven-Segment Decoder/Driver เอาพุทที่ได้จะมี 7 ไลน์ (Active High Outputs) และต่อเข้ากับ l.e.d.Display แต่ละตัว ดังรูปที่ 33

ตามรูปที่ 33 มี l.e.d.Display 6 ตัว เมื่อมีอินพุท บีซีดี ผ่านเข้า BCD-to-Seven-Segment Decoder/Driver, l.e.d. Display ตัวขวาสุดจะทำงานหรือคอนดัคท์ (Conduct) ก่อนตัวอื่นได้ต่อเมื่อมีพัลซ์ป้อนเข้าที่เบสของทรานซิสเตอร์ที่ต่อจาก BCD-to-Seven-Segment Decoder/Driver แล้วต่อลงกราวน์ (Ground) ทำให้กระแสไหลครบวงจร ส่วน l.e.d. Display ตัวอื่น ยังไม่ทำงาน แต่เมื่อมีอินพุท บีซีดี ต่อไป เข้ามาอีกจะทำให้ l.e.d. Display ถัดมาทำงาน หรือคอนดัคท์เรียงกันต่อไปเรื่อย ๆ ทั้งนี้ต้องให้อินพุท บีซีดี ที่เข้าไปใน BCD-to Seven-Segment Decoder/Driver ซินโครไนซ์ (Syncronize) กับพัลซ์ที่ป้อนเข้าเบสของทรานซิสเตอร์แต่ละตัวด้วย

การออกแบบพัลซ์ใช้ Switch - Tail Ring Counter พัลซ์อินพุทที่ป้อนเข้า Ring Counter คืออินพุท บีซีดี โดยใช้เกต OR ต่อเข้า Ring Counter เพื่อจะให้อินพุท บีซีดี ซินโครไนซ์ กับพัลซ์ที่ออกมา การเกิดเป็นเลขทศนิยมที่ l.e.d.Display จะเห็นชั่วครู เมื่อมีอินพุท บีซีดี ป้อนเข้าครั้งหนึ่ง แต่เมื่อเกิดเร็วขึ้น จะทำให้เราเห็นตัวเลขอยู่ตลอดเวลา

การทำงานของวงจรในรูปที่ 33 เป็นไปตามตารางที่ 10 ในตารางที่ 10 A,B,C, D, E และ F เป็นฟลิพ-พล็อพ ของ Switch-Tail Ring Counter ซึ่งมีการทำงานดังได้กล่าวมาแล้ว ในหัวข้อของวงจรนับ (Counters); P,Q,R,S, T และ U เป็นสัญญาณโลยิคของพัลซ์ที่ป้อนเข้าเบสของทรานซิสเตอร์ ตั้งแต่ตัวซ้ายสุดจนถึงตัวขวาสุด

รูปที่ 33 วงจรอิเล็กโทรนิกส์แสดงผลการทดลองที่ 4



A	B	C	D	E	F	P	Q	R	S	T	U
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0
1	1	1	1	0	0	0	0	0	1	0	0
1	1	1	1	1	0	0	0	0	0	1	0
1	1	1	1	1	1	0	0	0	0	0	1
0	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	0	1	0	0	0	0
0	0	0	1	1	1	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-

ตารางที่ 10 ตารางจริงของ Ring Counter และพัลซ์ที่ใช้จาก

ตารางที่ 10 จะเห็นว่า ค่าของ P จะเป็น 1 ได้ต้องขึ้นอยู่กับค่าของ A และ B

คือ P=1 เมื่ออินพุท A และ B มีค่าต่างกัน

และ P=0 เมื่ออินพุท A และ B มีค่าเหมือนกัน

ซึ่งเป็นรูปของฟังก์ชัน Exclusive-OR

$$P = A\bar{B} + \bar{A}B = A \oplus B$$

ทำนองเดียวกัน

Q จะเป็นฟังก์ชัน Exclusive-OR ของอินพุท B และ C

R จะเป็นฟังก์ชัน Exclusive-OR ของอินพุท C และ D

S จะเป็นฟังก์ชัน Exclusive-OR ของอินพุท D และ E

และ T จะเป็นฟังก์ชัน Exclusive-OR ของอินพุท E และ F

ส่วนการที่จะได้ค่า U เป็น 1 นั้น A, B, C, D, E และ F จะต้องเป็น 1 ทั้งหมดหรือเป็น 0 ทั้งหมด เขียนเป็นฟังก์ชันบูลีนได้เป็น

$$U = ABCDEF + \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}\,\bar{E}\,\bar{F}$$

$$= ABCDEF + \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}\,\bar{E}\,\bar{F}$$

$$= ABCDEF + \bar{\bar{A}} + \bar{\bar{B}} + \bar{\bar{C}} + \bar{\bar{D}} + \bar{\bar{E}} + \bar{\bar{F}}$$

$$= ABCDEF + A + B + C + D + E + F$$

การดัดแปลงวงจรเพื่อให้ใช้กับ ICs ที่มีอยู่ มีดังนี้

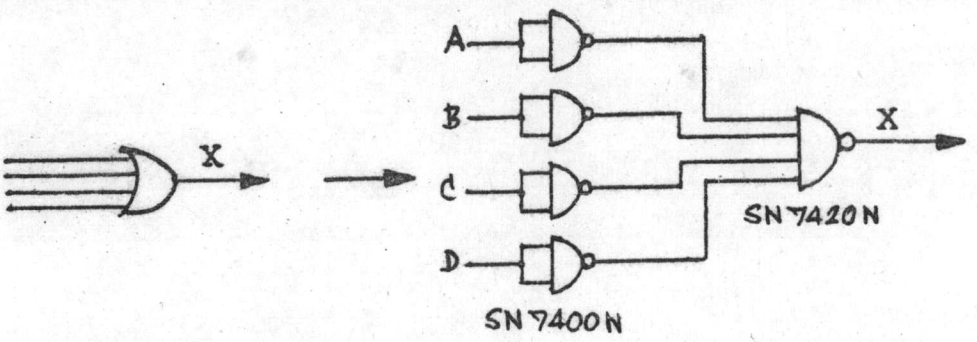
1. ใช้ 4- input NAND gate แทน 4- input OR gate

จาก $X = A+B+C+D$

$$= \overline{\overline{A+B+C+D}}$$

$$= \overline{\overline{(A+B)}\ \overline{(C+D)}}$$

$$= \overline{\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}}$$

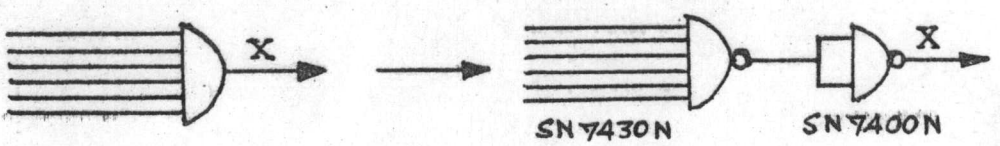


รูปที่ 34 การใช้ 4- input NAND gate แทน 4- input OR gate

2. ใช้ 6-input NAND gate แทน 6- input AND gate

จาก $X = ABCDEF$

$$= \overline{\overline{ABCDEF}}$$



รูปที่ 35 การใช้ 6- input NAND gate แทน 6- input AND gate

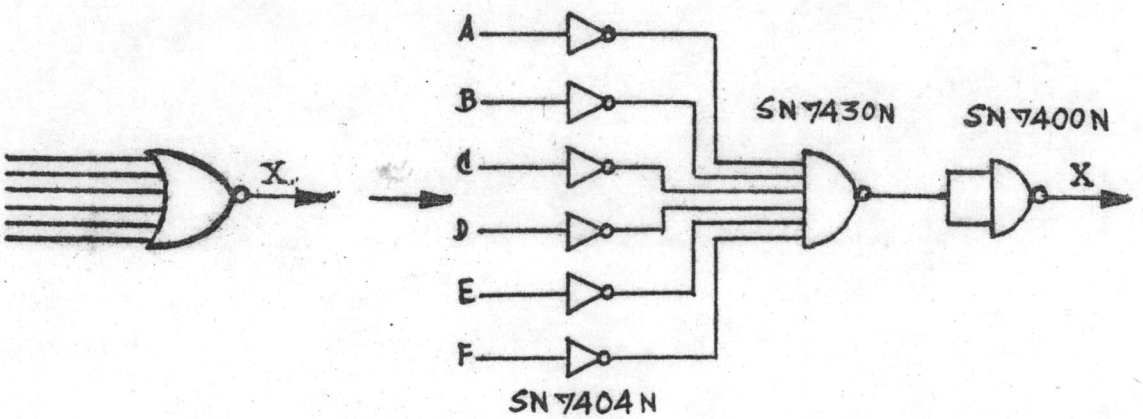
3. ใช้ 6- input NAND gate แทน 6- input NOR gate

จาก $X = \overline{A+B+C+D+E+F}$

$= \overline{(A+B+C)}\,\overline{(D+E+F)}$

$= \overline{(A+B)}\,\overline{C}\;\overline{(D+E)}\;\overline{F}$

$= A\;B\;C\;D\;E\;F$



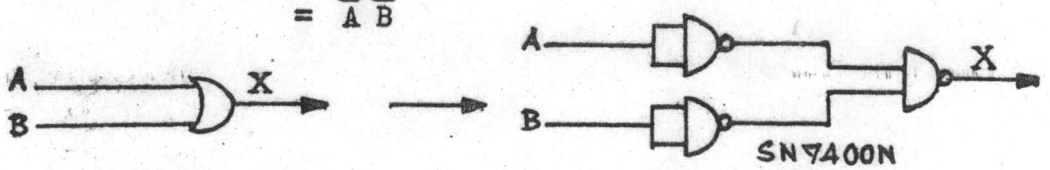
รูปที่ 36 การใช้ 6- input NAND gate แทน 6- input NOR gate

4. การใช้ 2- input NAND gate แทน 2- input OR gate

จาก $X = A+B$

$= \overline{\overline{A+B}}$

$= \overline{\overline{A}\;\overline{B}}$



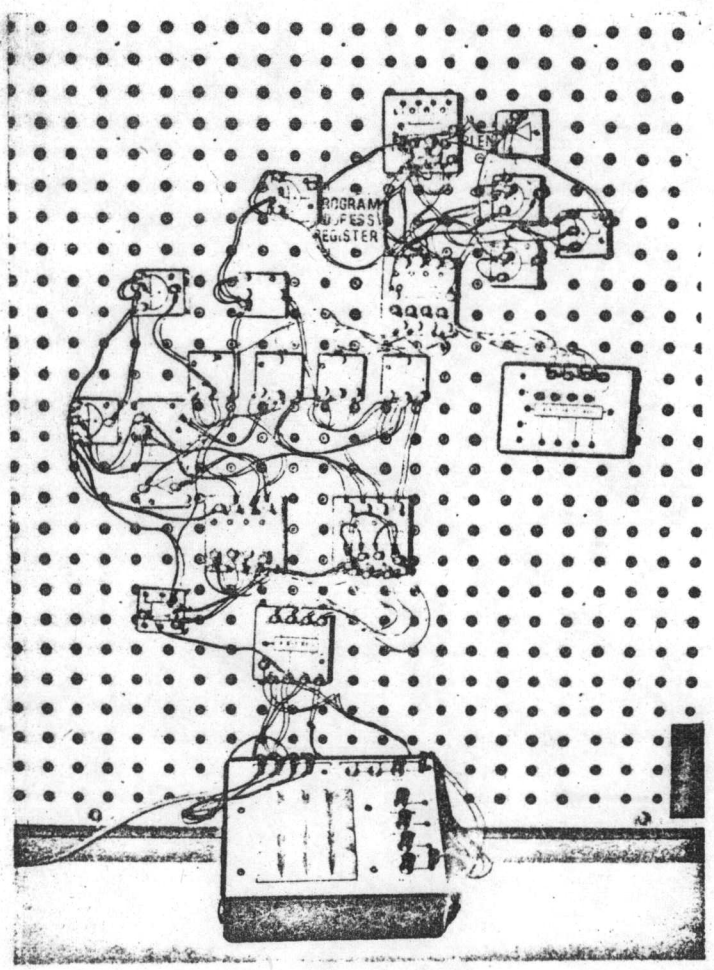
รูปที่ 37 การใช้ 2- input NAND gate แทน 2- input OR gate

3.3 การทดสอบวงจรที่ออกแบบในส่วนคำนวณ

เนื่องจากปัญหาในการสั่งซื้อ ICs จากต่างประเทศจึงไม่สามารถสร้างวงจรของจริงจาก ICs ได้ แต่ใช้วิธีการทดสอบวงจรที่ออกแบบบน Logic Board ของศูนย์คอมพิวเตอร์ศาสตร์ บัณฑิตวิทยาลัย จุฬาลงกรณมหาวิทยาลัย

ในการทดสอบ ได้ทดสอบทั้งการบวก,การบวกที่ต้องมีการแก้ผลบวก และการลบแบบที่ตัวตั้งมากกว่าตัวลบ

Logic Cubes ที่ใช้ในการทดสอบ ซึ่งแสดงในรูปที่ 38 มีรายการดังต่อไปนี้



รูปที่ 38 การทดสอบวงจรที่ออกแบบในการบวกลบ

ส่วนรับผลและแสดงคีย์บวกลบ

4 - Bit Register จำนวน 1

4 - Bit SL. SR. Register จำนวน 1

4 - Bit 2-input AND gate จำนวน 2

3 - input OR gate จำนวน 4

3 - input AND gate จำนวน 2

JK Master - Slave Flip-Flop จำนวน 1

ส่วนทำหน้าที่ 9's คอมพลีเมนท์

3 - input NOR gate จำนวน 1

2 - input Exclusive-OR gate จำนวน 1

Inverter gate จำนวน 1

ส่วนทำหน้าที่บวกและแก้ผลบวก

4 - Bit Full-Adder จำนวน 2

3 - input AND gate จำนวน 2

3 - input OR gate จำนวน 2

Inverter gate จำนวน 1

ผลการทดสอบได้ผลถูกต้องทั้งการบวก การบวกที่มีการแก้ผลบวก และการลบที่ตัวตั้งมากกว่าตัวลบ

การคูณใช้วิธี Repeated Addition, Logic Cubes ที่ใช้ในการทดสอบ ซึ่งแสดงในรูปที่ 39 มีรายการดังต่อไปนี้

ส่วนของตัวตั้ง

4 Bit Register จำนวน 1

Quad 2-input AND gate จำนวน 1

ส่วนตัวคูณซึ่งเป็น Down Counter

JK Master-Slave Flip-Flop จำนวน 4

3 - input AND gate จำนวน 2

Inverter gate จำนวน 1

ส่วนของการบวก, แก้ผลบวกและแสดงผล

4 Bit Full-Adder จำนวน 2

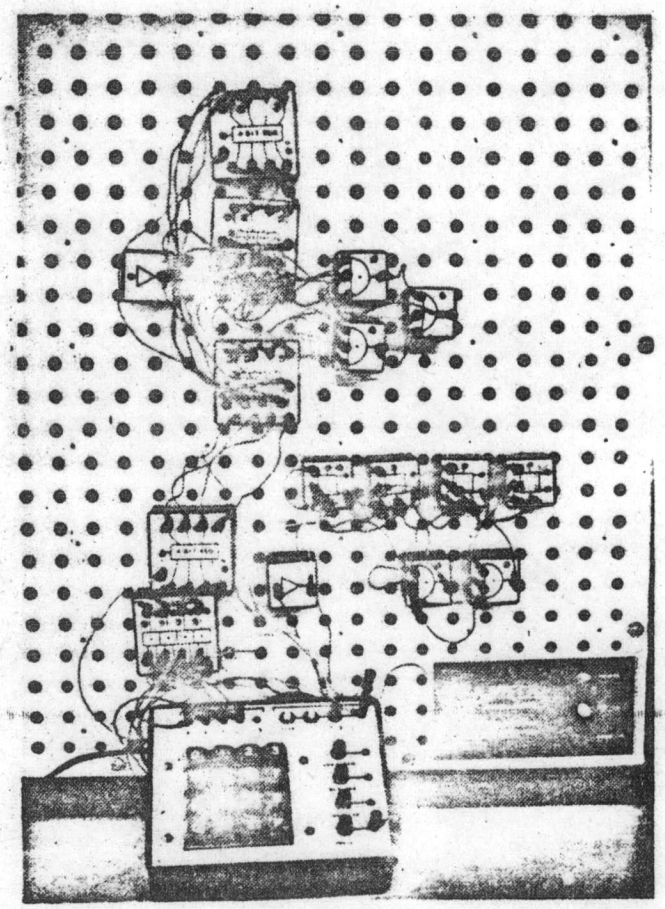
3 - Input AND gate จำนวน 2

3 - Input OR gate จำนวน 1

Inverter gate จำนวน 1

4 Bit Register จำนวน 1

ผลการทดสอบได้ผลถูกต้อง



รูปที่ 39 การทดสอบวงจรที่ออกแบบในการคูณ

การหารใช้วิธี Repeated - Subtraction, Logic Cubes ที่ใช้ในการทดสอบ ซึ่งแสดงในรูปที่ 40 มีดังต่อไปนี้

ส่วนทำหน้าที่ 9's คอมพลีเมนท์ และ open gate

3 - input NOR gate จำนวน 1

2 - input Exclusive - OR gate จำนวน 1

Inverter gate จำนวน 1

Quad 2 - input AND gate จำนวน 1

ส่วนแสดงผลลัพธ์

BCD Counter (count up) จำนวน 1

ส่วนทำหน้าที่บวกและแก้ผลบวก

4 - Bit Full - Adder จำนวน 2

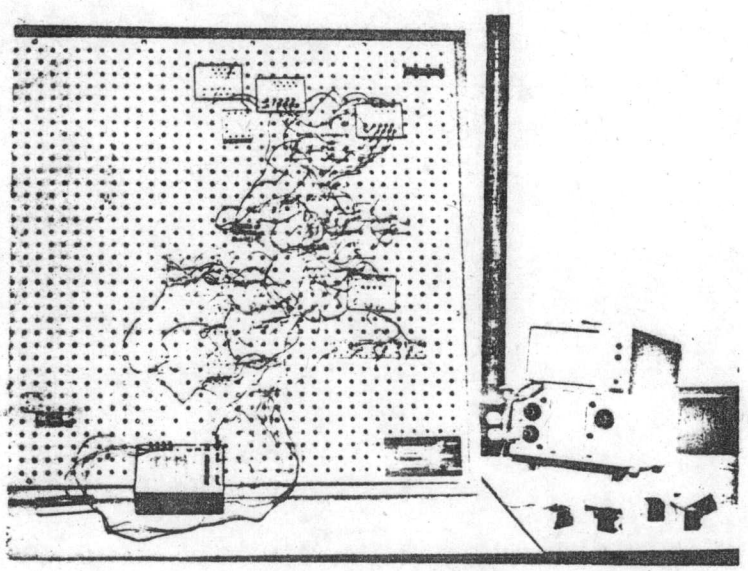
3 - input AND gate จำนวน 2

3 - input OR gate จำนวน 1

Inverter gate จำนวน 1

4 - Bit Register จำนวน 1

ผลการทดสอบได้ผลถูกต้อง



รูปที่ 40 การทดสอบวงจรที่ออกแบบในการหาร

ตารางข้อมูลที่ใช้ทดสอบในส่วนคำนวณ

1. ข้อมูลที่ใช้ทดสอบในการบวก

ลำดับที่	ตัวตั้ง	ตัวบวก	ผลลัพธ์	หมายเหตุ
1	0100	0101	1001	4 + 5 = 9
2	0110	0010	1000	6 + 2 = 8
3	0101	0111	0001 0010	5 + 7 = 12
4	1000	0110	0001 0100	8 + 6 = 14
5	1001	1000	0001 0111	9 + 8 = 17

เนื่องจาก clock ในห้องทดลองใช้ความถี่ = 500 KHz

ดังนั้น 1 period จะใช้เวลา = 0.2×10^{-5} sec.

เมื่อใช้ 1 clock pulse /digit จะใช้เวลา = 0.2×10^{-5} sec.

= 0.02 μsec.

2. ข้อมูลที่ใช้ทดสอบในการลบ

ลำดับที่	ตัวตั้ง	ตัวลบ	9's คอมพลีเมนทของตัวลบ	ผลลัพธ์	หมายเหตุ
1	0111	0011	0110	0100	7-3 = 4
2	1001	0111	0010	0010	9-7 = 2
3	1000	0101	0100	0011	8-5 = 3
4	0111	0010	0111	0101	7-2 = 5
5	1001	0110	0011	0011	9-6 = 3

การลบใช้ 1 clock pulse /digit จึงใช้เวลา 0.02 μsec. เช่นเดียวกับการบวก

3. ข้อมูลที่ใช้ทดสอบในการคูณ

ลำดับที่	ตัวตั้ง	ตัวคูณ	ผลลัพธ์	หมายเหตุ
1	0010	0011	0110	2 x 3 = 6
2	0010	0100	1000	2 x 4 = 8
3	0100	0011	0001 0010	4 x 3 = 12
4	0100	0100	0001 0110	4 x 4 = 16
5	0101	0010	0001 0000	5 x 2 = 10

การบวก 1 ครั้งใช้ 1 clock pulse ใช้เวลา = 0.2×10^{-5} sec.
แต่จำนวนครั้งที่จะบวกได้สูงสุด = 9 ครั้ง
ดังนั้นเวลาที่ใช้ในการคูณสูงสุด = 1.8×10^{-5} sec. = 0.18 μsec/digit

4. ข้อมูลที่ใช้ทดสอบในการหาร

ลำดับที่	ตัวตั้ง	ตัวหาร	ผลลัพธ์	เศษ	หมายเหตุ
1	0100	0010	0010	0000	4÷2 = 2 เศษ 0
2	0101	0010	0010	0001	5÷2 = 2 เศษ 1
3	0111	0011	0010	0001	7÷3 = 2 เศษ 1
4	0111	0100	0001	0011	7÷4 = 1 เศษ 3
5	1001	0010	0100	0001	9÷2 = 4 เศษ 1
6	1000	0100	0010	0000	8÷4 = 2 เศษ 0

การลบ 1 ครั้งใช้ 1 clock pulse ใช้เวลา = 0.2×10^{-5} sec.
แต่จำนวนครั้งที่จะลบได้สูงสุด = 9 ครั้ง (ตัวตั้ง 9 ตัวหาร 1)
ดังนั้นเวลาที่ใช้ในการหารสูงสุด = 0.18 μsec/digit

3.4 อภิปรายผลการวิจัย

แบบของวงจรที่ได้เป็นแบบที่ประหยัด คือใช้ โลยิคเกต จำนวนน้อยที่สุด อันเป็นวัตถุประสงค์ของการวิจัยนี้ โดยการใช้ ICs ที่เป็นวงจรบวกเต็ม 4 บิท เพียง 2 วงจร ในส่วนคำนวณ ยกเว้นการหารซึ่งต้องใช้วงจรบวกเต็ม และวงจรนับ บีซีดี มากขึ้นเมื่อตัวตั้งและตัวหารมีมากกว่า 1 ตำแหน่ง ในส่วนแสดงผลก็ใช้ Ics ที่เป็น BCD-to-Seven-Segment Decoder/Driver เพียงวงจรเดียว

ในการคูณนั้น จะต้องมี Shift Register อีกชุดหนึ่ง สำหรับรับข้อมูลที่เป็นผลคูณจำนวนหลาย ๆ บิท และเพื่อให้สามารถ shift ผลลัพธ์ที่ได้จากการบวกแต่ละครั้งมาบวกกันได้ แล้วจึงส่งผลคูณที่ได้เข้า I/O รียิสเตอร์

การหารในการวิจัยนี้ เป็นแบบปัดเศษ คือแสดงแต่ผลหารออกมาเท่านั้น จึงใช้ได้ไม่ดีนัก อย่างไรก็ดีแบบของวงจรนี้ก็สามารถจะทำการคำนวณติดต่อกันได้ คือเมื่อกดคีย์ PLUS,SUB, MUL หรือ DIV ต่อไป ผลลัพธ์จากการคำนวณครั้งแรกใน I/O รียิสเตอร์ จะ shift เข้า A - รียิสเตอร์ และสามารถจะ load เลขจำนวนต่อไปเข้า B - รียิสเตอร์เพื่อทำการบวกลบ คูณ หรือหารต่อไปได้อีก และถ้าต้องการให้รับตัวเลขได้มากตำแหน่งกว่านี้ คืออาจจะเป็น 8, 10 หรือ 12 ตำแหน่ง ก็สามารถทำได้ โดยการเพิ่มจำนวนรียิสเตอร์และเปลี่ยนวงจรนับที่ควบคุมลำดับของการทำงาน ให้เป็นไปตามจำนวนตำแหน่งที่ต้องการ หลักการทำงานอย่างอื่นจะเหมือนเดิม

เนื่องจาก Logic Cubes ในห้องทดลอง มีจำนวนจำกัด จึงทำการทดสอบในส่วนคำนวณได้เพียงตัวเลขตำแหน่งเดียว