การทดลองที่ 9: ALU: Arithmetic Logic Unit
กลุ่มลงทะเบียน กลุ่มการทดลอง รหัส ชื่อ-สกุล

ตารางตรวจ การทดลอง: ให้พิมพ์ 2 หน้าแรกแบบหน้าหลัง มาพร้อมเข้าเรียน

การทดลอง	Score	**ลายเซ็นต์**
การทดลอง 01/4 วงจรข้อ 4	2	LW5: ALU 2bit 8Instruction
การทดลอง 02/4 วงจรข้อ 3,5,7	2	LW5: 74181 Simulator
Exercise-01	2	LW5: 74181 Simulator
Exercise-02	5	IC: 74181 + Digital Trainer Board
Exercise-03	5	IC: 74181 + Digital Trainer Board
Exercise-04	5	IC: 74181 + Digital Trainer Board

สรุปผลการทดลอง

(1) ALU คือ

(2) Top-Down Design คือ

(3) Button-Up Design คือ

Lab09 ALU: Arithmetic Logic Unit

(1) บันทึกเป็น pdf ไฟล์ และกำหนดชื่อไฟล์เป็น รหัส-ชื่อ สกุล เช่น B3601234-นายวิชัย ศรีสุรักษ์
(2) เตรียมการทดลอง ตามคำถามก่อนการทดลอง
(3) ส่งเตรียมการทดลอง พร้อมกันทุกกลุ่ม ก่อน 20250114-0600 ที่ https://forms.gle/ebw5v2QRTQpD5Qic6
(4) เติมคำตอบ คำอธิบาย สรุปผล ด้วยลายมือตัวเอง ใน pdf ไฟล์
(5) ส่ง Full Report Pdf File ก่อน 20250120-0600 ที่ https://forms.gle/DbY1hxXmEA7tEUAn9

1. จุดประสงค

1. เขาใจการทำงานของ ALU: Arithmetic Logic Unit
2. ฝึกการใชโปรแกรมจำลองการทำงาน LogicWork5
3. ฝึกการใชงานไอซี CPLD

2. อุปกรณการทดลอง

1. ไอซี 74181: A 4-bit ALU (Arithmetic and Logic Unit) 1 ตัว
2. คอมพิวเตอร์ที่ติดตั้งโปรแกรม LogicWork5 1 ชุด
3. บอรดทดลองทางดิจิตอล CPLD Board พร้อมสายไฟ 1 ชุด

3. ทฤษฎี

The 74181 chip is the forerunner of today's microcomputer math coprocessor. This arithmetic Logic unit is an MSI integrated circuit, which can perform 16 arithmetic operations and 16 logic Operations on two 4-bit input numbers. It can also compare the magnitudes of the input numbers.

An ALU performs many different operations. Selection of an operation: The ***Select*** inputs and the ***Carry- In*** are used to select the operation.

An operation may require "**two** operands as the input" and "**one** output". The two 4-bit input-operands and the 4-bit output are to be stored in three 4-bit registers. The registers are to be used in the LOAD mode so that the output of each of the registers is equal to its input.

74181 *(ALU):*

INPUTS of 74181:

- *Two 4-bit inputs (A & B)*
- *4 select inputs S3, S2, S1, S0 (should be taken in the same order)*
- *Mode Control Input:* *M = HIGH → Logical Operations*
  M = LOW → Arithmetic Operations
- *Carry Input:* *CN = HIGH →No Carry*
  CN = LOW → Carry enabled

OUTPUTS of 74181:

- *4 Active HIGH outputs (F0..F3);I Other Outputs: No significance for this lab.*

Extracts from the FUNCTION TABLE of 74181:

Logical operations: *M=H, CN=H*

S3	*S2*	*S1*	*S0*	*Operation*
H	*L*	*H*	*H*	*A* Λ *B*
H	*H*	*H*	*L*	*A V B*
L	*L*	*L*	*L*	*A'*
L	*H*	*L*	*L*	*(A* Λ *B)'*
L	*L*	*L*	*H*	*(A V B)'*

Arithmetic operations: *M=L*

S3	*S2*	*S1*	*S0*	*Cn*	*Operation*
H	*L*	*L*	*H*	*H*	*A + B*
L	*H*	*H*	*L*	*L*	*A - B*
H	*H*	*H*	*H*	*H*	*A - 1*

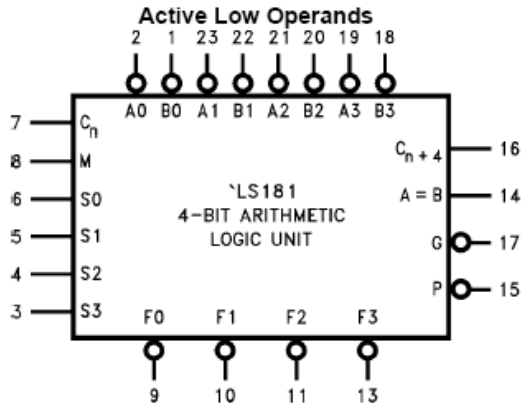
Function table for output F

In the following table, AND is denoted as a product, OR with a + sign, XOR with $\oplus$, logical NOT with an overbar and arithmetic plus and minus using the words plus and minus.

Selection				**Active-low data**			**Active-high data**		
				Logic M = H	**Arithmetic** M = L		**Logic** M = H	**Arithmetic** M = L	
S3	S2	S1	S0		Cn = L (no carry)	Cn = H (carry)		Cn = L (no carry)	Cn = H (carry)
L	L	L	L	$\overline{A}$	A minus 1	A	$\overline{A}$	A	A plus 1
L	L	L	H	$\overline{AB}$	AB minus 1	AB	$\overline{A+B}$	$A+B$	$(A+B)$ plus 1
L	L	H	L	$\overline{A}+B$	$A\overline{B}$ minus 1	$A\overline{B}$	$\overline{AB}$	$A+\overline{B}$	$(A+\overline{B})$ plus 1
L	L	H	H	Logical 1	-1 (two's complement)	0 (zero)	Logical 0	-1 (two's complement)	0 (zero)
L	H	L	L	$\overline{A+B}$	A plus $(A+\overline{B})$	A plus $(A+\overline{B})$ plus 1	$\overline{AB}$	A plus $A\overline{B}$	A plus $(A\overline{B})$ plus 1
L	H	L	H	$\overline{B}$	AB plus $(A+\overline{B})$	AB plus $(A+\overline{B})$ plus 1	$\overline{B}$	$(A+B)$ plus $A\overline{B}$	$(A+B)$ plus $A\overline{B}$ plus 1
L	H	H	L	$\overline{A\oplus B}$	A minus B minus 1	A minus B	$A\oplus B$	A minus B minus 1	A minus B
L	H	H	H	$A+\overline{B}$	$A+\overline{B}$	$A+\overline{B}$ plus 1	$A\overline{B}$	$A\overline{B}$ minus 1	$A\overline{B}$
H	L	L	L	$\overline{A}B$	A plus $(A+B)$	A plus $(A+B)$ plus 1	$\overline{A}+B$	A plus AB	A plus AB plus 1
H	L	L	H	$A\oplus B$	A plus B	A plus B plus 1	$\overline{A\oplus B}$	A plus B	A plus B plus 1
H	L	H	L	B	$A\overline{B}$ plus $(A+B)$	$A\overline{B}$ plus $(A+B)$ plus 1	B	$(A+\overline{B})$ plus AB	$(A+\overline{B})$ plus AB plus 1
H	L	H	H	$A+B$	$A+B$	AB plus 1	AB	AB minus 1	AB
H	H	L	L	Logical 0	A plus A	A plus A plus 1	Logical 1	A plus A	A plus A plus 1
H	H	L	H	$A\overline{B}$	AB plus A	AB plus A plus 1	$A+\overline{B}$	$(A+B)$ plus A	$(A+B)$ plus A plus 1
H	H	H	L	AB	$A\overline{B}$ plus A	$A\overline{B}$ plus A plus 1	$A+B$	$(A+\overline{B})$ plus A	$(A+\overline{B})$ plus A plus 1
H	H	H	H	A	A	A plus 1	A	A minus 1	A

- https://en.wikipedia.org/wiki/74181

All function Table of 74181 < http://irlenys.tripod.com/digitalesi/arit/suma.htm >



Pin Names	Description
$\overline{A0}$–$\overline{A3}$	Operand Inputs (Active LOW)
$\overline{B0}$–$\overline{B3}$	Operand Inputs (Active LOW)
S0–S3	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\overline{F0}$–$\overline{F3}$	Function Outputs (Active LOW)
A = B	Comparator Output
$\overline{G}$	Carry Generate Output (Active LOW)
$\overline{P}$	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output

V_{CC} = Pin 24

GND = Pin 12

Mode Select Inputs				Active LOW Operands & F_n Outputs		Active HIGH Operands & F_n Outputs	
				Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
S3	S2	S1	S0	(M = H)	(M = L) (C_n = L)	(M = H)	(M = L) (C_n = H)
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\overline{A} + B$	A + B
L	L	H	L	$\overline{A} + \overline{B}$	$A\overline{B}$ minus 1	$\overline{A}\,B$	$A + \overline{B}$
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\overline{A} + \overline{B}$	A plus $(A + \overline{B})$	$\overline{AB}$	A plus $A\overline{B}$
L	H	L	H	$\overline{B}$	AB plus $(A + \overline{B})$	$\overline{B}$	$(A + B)$ plus $A\overline{B}$
L	H	H	L	$\overline{A} \oplus \overline{B}$	A minus B minus 1	$A \oplus B$	A minus B minus 1
L	H	H	H	$A + \overline{B}$	$A + \overline{B}$	$A\overline{B}$	AB minus 1
H	L	L	L	$\overline{A}\,B$	A plus (A + B)	$\overline{A} + B$	A plus AB
H	L	L	H	$A \oplus B$	A plus B	$\overline{A} \oplus \overline{B}$	A plus B
H	L	H	L	B	$A\overline{B}$ plus (A + B)	B	$(A + \overline{B})$ plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	$A\overline{B}$	AB plus A	$A + \overline{B}$	$(A + B)$ plus A
H	H	H	L	AB	$A\overline{B}$ minus A	A + B	$(A + \overline{B})$ plus A
H	H	H	H	A	A	A	A minus 1

Note 1: Each bit is shifted to the next most significant position.

Note 2: Arithmetic operations expressed in 2s complement notation.

4. คำถามก่อนการทดลอง

1. ศึกษาการใช้งานโปรแกรม LW5 https://www.youtube.com/watch?v=_wro2OEMQPY
2. ศึกษาการทำงานของ http://www.esi.uclm.es/www/isanchez/apuntes/ci/74181.pdf

3. เติมค่าในการทดลองที่ 01/4 ข้อ 2 ให้แสดงการลดรูปฟังก์ชัน 0S1,0S0-(add), 1S1,1S0-(minus), 5S1,5S0(A>>1), 6S1,6S0(A<<1) ให้สมบูรณ์ (ด้วยการเขียนด้วยลายมือ)
4. เติมค่าในการทดลองที่ 01/4 ข้อ 3 เขียนวงจรด้วย LogicWork5 ตามรูปแล้วพิมพ์เป็น pdf ไฟล์ แทรกหลังวงจรข้อ 3
5. เติมค่าในการทดลองที่ 01/4 ข้อ 4 ทำการทดสอบป้อนอินพุตแล้วเติมค่าในตาราง 1 (ด้วยการเขียนด้วยลายมือ)

6. เติมค่าในการทดลองที่ 02/4 ข้อ 3 เขียนวงจรด้วย LogicWork5 ตามรูปแล้วพิมพ์เป็น pdf ไฟล์ แทรกหลังวงจรข้อ 3
7. เติมค่าในการทดลองที่ 02/4 ข้อ 5 ทำการทดสอบป้อนอินพุตแล้วเติมค่าในตาราง 2 (ด้วยการเขียนด้วยลายมือ)
8. เติมค่าในการทดลองที่ 02/4 ข้อ 7 ทำการทดสอบป้อนอินพุตแล้วเติมค่าในตาราง 3 (ด้วยการเขียนด้วยลายมือ)
9. เติมค่าในการทดลองที่ 02/4 ข้อ 9 ทำการทดสอบป้อนอินพุตแล้วเติมค่าในตาราง 4 (ด้วยการเขียนด้วยลายมือ)

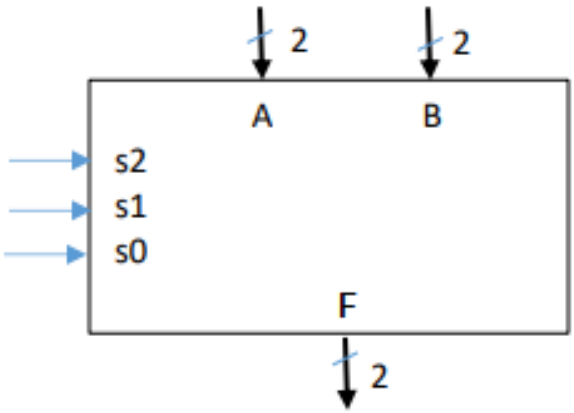
10. เตรียมวงจรใน Exercise-01 ด้วย LogicWork5 ตามรูปแล้วพิมพ์เป็น pdf ไฟล์ แทรกด้านหลัง Excercis-01
11. เตรียมวงจรใน Exercise-02 ด้วย LogicWork5 แล้วพิมพ์เป็น pdf ไฟล์ แทรกด้านหลัง Excercis-02
12. เตรียมวงจรใน Exercise-03 ด้วย LogicWork5 แล้วพิมพ์เป็น pdf ไฟล์ แทรกด้านหลัง Excercis-03
13. เตรียมวงจรใน Exercise-04 ด้วย LogicWork5 แล้วพิมพ์เป็น pdf ไฟล์ แทรกด้านหลัง Excercis-04

5. การทดลอง

Experiment 01/4. จำลองการทำงานในโปรแกรม LogicWork5 – ALU 2bit 8Instruction

1. วงจร 2bit ALU การทำงานจะมีอินพุต A กับ B โดยจำนวน A,B จะมีขนาด 2bit มีค่า {0,1,2,3 = 00,01,10,11} ผลการทำงานจะได้คำตอบ F 2bit กำหนดให้ การทำงานเป็นดังตารางโดยเลือกค่าคำตอบ 8 กรณี ยกตัวอย่าง หาก S=000 Output F = A+B เป็นต้น

Input s2 s1 s0	ALU operation
0 0 0	S = A + B
0 0 1	S = A - B
0 1 0	S = A AND B
0 1 1	S = A OR B
1 0 0	S = A' (Not A)
1 0 1	S = A >> 1
1 1 0	S = A << 1
1 1 1	S = B



Input-A		Input-B		A plus B		A minus B		A>>1		A<<1	
A1	A0	B1	B0	0S1	0S0	1S1	1S0	5S1	5S0	6S1	6S0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1	0	0	0	0
0	0	1	0	1	0	1	0	0	0	0	0
0	0	1	1	1	1	0	1	0	0	0	0
0	1	0	0	0	1	0	1	0	0	1	0
0	1	0	1	1	0	0	0	0	0	1	0
0	1	1	0	1	1	1	1	0	0	1	0
0	1	1	1	0	0	1	0	0	0	1	0
1	0	0	0	1	0	1	0	0	1	0	0
1	0	0	1	1	1	0	1	0	1	0	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	1	1	1	0	1	0	0
1	1	0	0	1	1	1	1	0	1	1	0
1	1	0	1	0	0	1	0	0	1	1	0
1	1	1	0	0	1	0	1	0	1	1	0
1	1	1	1	1	0	0	0	0	1	1	0

0S1 = (A0 and B0) xor A1 xor B1
1S1 = (A0' and B0) xor A1 xor B1
2S1 = A1 and B1
3S1 = A1 or B1
4S1 = A1'
5S1 = 0
6S1 = A0
7S1 = B1

0S0 = A0 xor B0
1S0 = A0 xor B0
2S0 = A0 and B0
3S0 = A0 or B0
4S0 = A0'
5S0 = A1
6S0 = 0
7S0 = B0

2. จากตารางค่าความจริง ให้ลดรูปเพื่อออกแบบบวงจร OS1,OSO-(add), 1S1,1SO-(minus), 5S1,5SO(A>>1), 6S1,6SO(A<<1) https://sublime.tools/karnaugh-map

<table>
<tr><th>[OS1 OSO] = [A1 AO] add [B1 BO]</th><th>[1S1 1SO] = [A1 AO] minus [B1 BO]</th></tr>
<tr><td>

B1B0 / A1A0	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

B1B0 / A1A0	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

⊕

B1B0 / A1A0	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	1	0	0
10	1	1	0	0

OSO = (AO●A1) ⊕ (A1 ⊕ B1)

B1B0 / A1A0	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

OSO = (AO ⊕ BO)

</td><td></td></tr>
<tr><th>[5S1 5SO] = [A1 AO] >> 1 = [O A1]</th><th>[6S1 6SO] = [A1 AO] << 1 = [AO O]</th></tr>
<tr><td>

B1B0 / A1A0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

5S1 = O

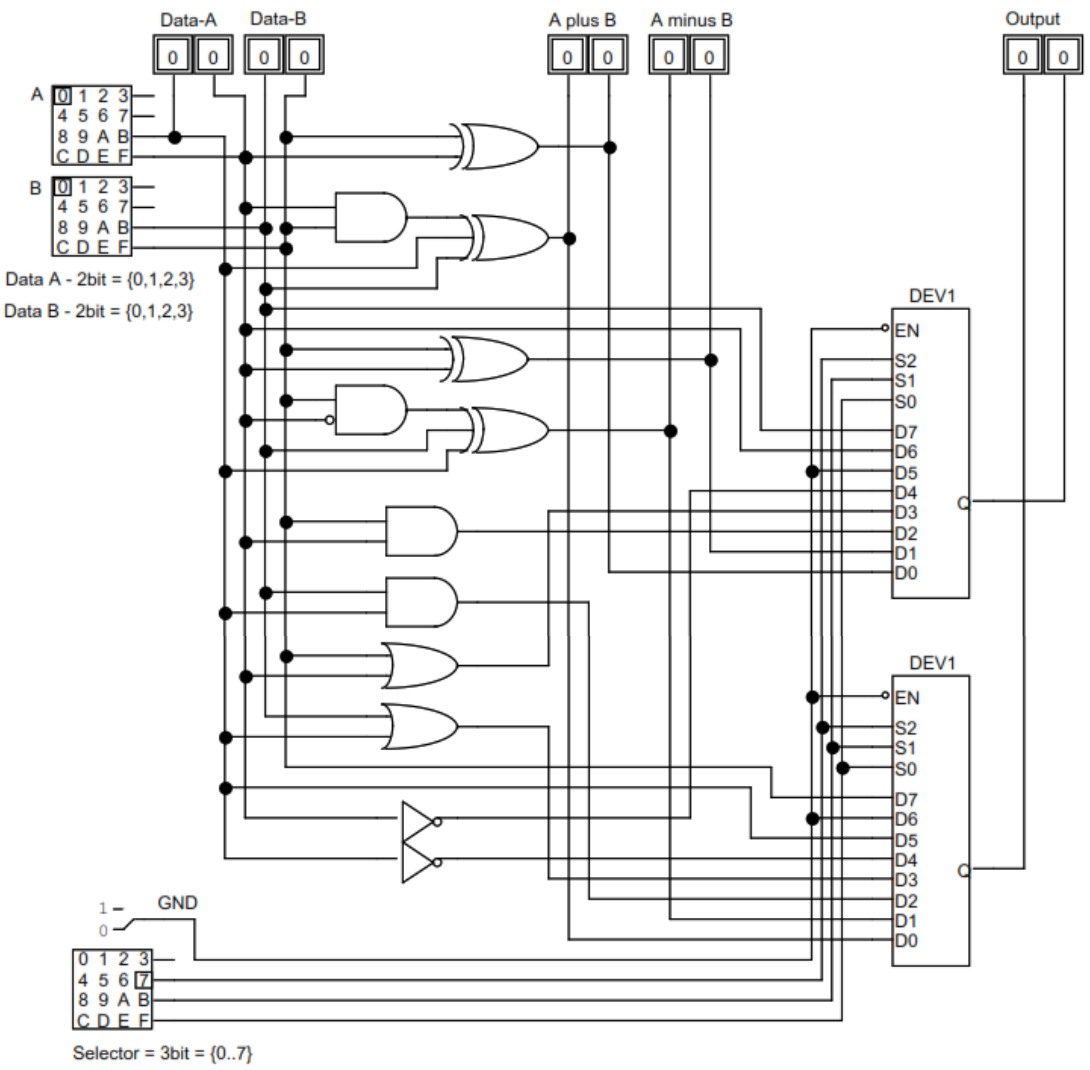
B1B0 / A1A0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

5SO = A1

</td><td></td></tr>
</table>

3. ใช้โปรแกรม LW5 จำลองการทำงาน โดยวาดวงจรดังรูป

LogicWork5 – ALU 2bit 8Instruction



Select = 000 --> F = A plus B	0S1 = (A0 and B0) xor A1 xor B1	0S0 = A0 xor B0
Select = 001 --> F = A minusB	1S1 = (A0' and B0) xor A1 xor B1	1S0 = A0 xor B0
Select = 010 --> F = A and B	2S1 = A1 and B1	2S0 = A0 and B0
Select = 011 --> F = A or B	3S1 = A1 or B1	3S0 = A0 or B0
Select = 100 --> F = A'	4S1 = A1'	4S0 = A0'
Select = 101 --> F = A >> 1	5S1 = 0	5S0 = A1
Select = 110 --> F = A << 1	6S1 = A0	6S0 = 0
Select = 111 --> F = B	7S1 = B1	7S0 = B0

LW5 Part

- Binary Probe, Binary Switch
- Hex Keboard wo STB
- Mux-8
- AND-2, AND-2(1-Inv), XOR-2, XOR-3

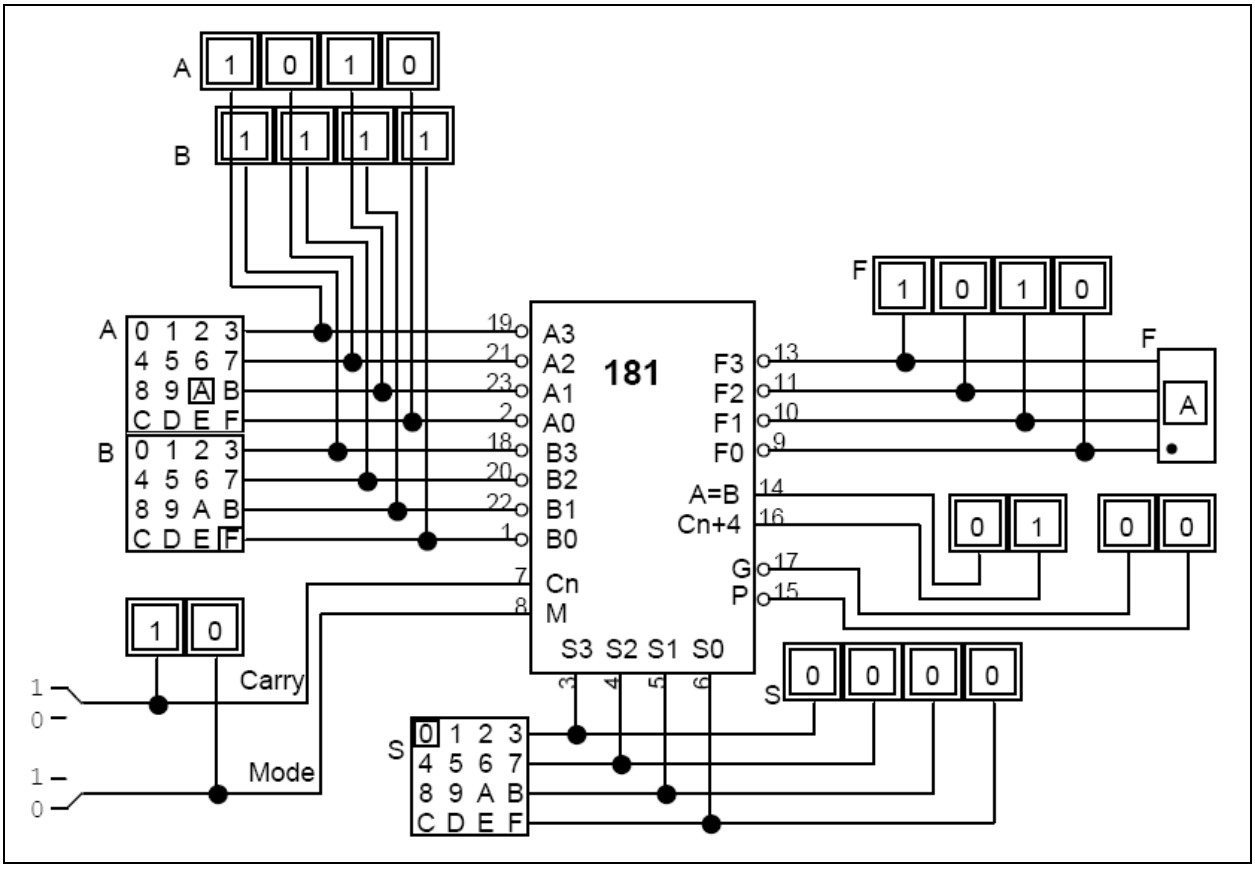
4. ทดสอบการทำงาน ด้วยโปรแกรม LW5

ตารางบันทึกผลที่ 1:

Input		**Output**							
A	**B**	**F = A plus B**	**F = A minus B**	**F = A and B**	**F = A or B**	**F = A'**	**F = A>>1**	**F = A<<1**	**F = B**
		S=000	**S=001**	**S=010**	**S=011**	**S=100**	**S=101**	**S=110**	**S=111**
00	**00**								
00	**01**								
01	**10**								
01	**11**								
10	**11**								
10	**10**								
11	**01**								
11	**00**								

Experiment 02/4. ไอซี 74181-4Bit ALU

1. ศึกษาการทำงานของ http://www.esi.uclm.es/www/isanchez/apuntes/ci/74181.pdf
2. Read https://apollo181.wixsite.com/apollo181/about
3. ต่อวงจรดังรูปให้จำลองวงจรดังรูปซึ่งเป็นการต่อแบบ Active High ในโปรแกรม Logic Work 5



Testing the Logic Functions:

4. จากตารางหน้า 3 การทดลองนี้เป็นโหมด Testing Logic Function จะกำหนดให้ขา M มีลอจิกเป็น 1
5. ทดสอบการทำงานโดยต่อ A=0101, B=0011 และ S ตามตารางสังเกตผลที่ได้ และบันทึกผลในตาราง บันทึกผลที่ 1 ช่อง CPLD Test พร้อมทั้งเปรียบเทียบกับช่อง LW5 Simulation

ตารางบันทึกผลที่ 2:

Instruction Select				**with M = 1 (Logic mode) and A = 0101_2 and B = 0011_2**			
				Outputs			
S3	**S2**	**S1**	**S0**	**F3**	**F2**	**F1**	**F0**
0	**0**	**0**	**0**				
0	**0**	**0**	**1**				
0	**0**	**1**	**0**				
0	**0**	**1**	**1**				
0	**1**	**0**	**0**				
0	**1**	**0**	**1**				
0	**1**	**1**	**0**				
0	**1**	**1**	**1**				
1	**0**	**0**	**0**				
1	**0**	**0**	**1**				
1	**0**	**1**	**0**				
1	**0**	**1**	**1**				
1	**1**	**0**	**0**				
1	**1**	**0**	**1**				
1	**1**	**1**	**0**				
1	**1**	**1**	**1**				

Active HIGH Operation

Selection				Active-high data		
				Logic M = H	**Arithmetic** M = L	
S3	S2	S1	S0		Cn = H (no carry)	Cn = L (carry)
L	L	L	L	$\overline{A}$	A	A plus 1
L	L	L	H	$\overline{A+B}$	$A+B$	$(A+B)$ plus 1
L	L	H	L	$\overline{A}B$	$A+\overline{B}$	$(A+\overline{B})$ plus 1
L	L	H	H	Logical 0	-1 (two's complement)	0 (zero)
L	H	L	L	$\overline{AB}$	A plus $A\overline{B}$	A plus $(A\overline{B})$ plus 1
L	H	L	H	$\overline{B}$	$(A+B)$ plus $A\overline{B}$	$(A+B)$ plus $A\overline{B}$ plus 1
L	H	H	L	$A \oplus B$	A minus B minus 1	A minus B
L	H	H	H	$A\overline{B}$	$A\overline{B}$ minus 1	$A\overline{B}$
H	L	L	L	$\overline{A}+B$	A plus AB	A plus AB plus 1
H	L	L	H	$\overline{A \oplus B}$	A plus B	A plus B plus 1
H	L	H	L	B	$(A+\overline{B})$ plus AB	$(A+\overline{B})$ plus AB plus 1
H	L	H	H	AB	AB minus 1	AB
H	H	L	L	Logical 1	A plus A	A plus A plus 1
H	H	L	H	$A+\overline{B}$	$(A+B)$ plus A	$(A+B)$ plus A plus 1
H	H	H	L	$A+B$	$(A+\overline{B})$ plus A	$(A+\overline{B})$ plus A plus 1
H	H	H	H	A	A minus 1	A

Testing the Arithmetic Functions:

6. จากตารางหน้า 3 การทดลองนี้เป็นโหมด Arithmetic Function จะกำหนดให้ขา M มีลอจิกเป็น 0
7. ทดสอบการทำงานโดยต่อ A=0101, B=0011 และ S ตามตารางสังเกตผลที่ได้ และบันทึกผลในตารางบันทึกผลที่ 2 ช่อง CPLD Test พร้อมทั้งเปรียบเทียบกับช่อง LW5 Simulation

ตารางบันทึกผลที่ 3:

Instruction Select				**with M = 0 (Arithmetic mode) C=1, A = 0101_2 and B = 0011_2**				
				Outputs				
S3	**S2**	**S1**	**S0**	**Cn+4**	**F3**	**F2**	**F1**	**F0**
0	**0**	**0**	**0**					
0	**0**	**0**	**1**					
0	**0**	**1**	**0**					
0	**0**	**1**	**1**					
0	**1**	**0**	**0**					
0	**1**	**0**	**1**					
0	**1**	**1**	**0**					
0	**1**	**1**	**1**					
1	**0**	**0**	**0**					
1	**0**	**0**	**1**					
1	**0**	**1**	**0**					
1	**0**	**1**	**1**					
1	**1**	**0**	**0**					
1	**1**	**0**	**1**					
1	**1**	**1**	**0**					
1	**1**	**1**	**1**					

Active HIGH Operation

Selection				**Active-high data**		
				Logic M = H	**Arithmetic** M = L	
S3	S2	S1	S0		Cn = H (no carry)	Cn = L (carry)
L	L	L	L	$\overline{A}$	A	A plus 1
L	L	L	H	$\overline{A+B}$	$A+B$	$(A+B)$ plus 1
L	L	H	L	$\overline{A}B$	$A+\overline{B}$	$(A+\overline{B})$ plus 1
L	L	H	H	Logical 0	-1 (two's complement)	0 (zero)
L	H	L	L	$\overline{AB}$	A plus $A\overline{B}$	A plus $(A\overline{B})$ plus 1
L	H	L	H	$\overline{B}$	$(A+B)$ plus $A\overline{B}$	$(A+B)$ plus $A\overline{B}$ plus 1
L	H	H	L	$A \oplus B$	A minus B minus 1	A minus B
L	H	H	H	$A\overline{B}$	$A\overline{B}$ minus 1	$A\overline{B}$
H	L	L	L	$\overline{A}+B$	A plus AB	A plus AB plus 1
H	L	L	H	$\overline{A \oplus B}$	A plus B	A plus B plus 1
H	L	H	L	B	$(A+\overline{B})$ plus AB	$(A+\overline{B})$ plus AB plus 1
H	L	H	H	AB	AB minus 1	AB
H	H	L	L	Logical 1	A plus A	A plus A plus 1
H	H	L	H	$A+\overline{B}$	$(A+B)$ plus A	$(A+B)$ plus A plus 1
H	H	H	L	$A+B$	$(A+\overline{B})$ plus A	$(A+\overline{B})$ plus A plus 1
H	H	H	H	A	A minus 1	A

Testing the Comparison Function:

8. จาก Datasheet หากต้องการเปรียบเทียบค่า A กับ B ต้องกำหนดค่า M, S3-S0, และ Cn เป็นอย่างไร เติมค่า M, S3-S0, และ Cn ในตารางบันทึกผลที่ 3
9. ทดสอบการทำงานโดยป้อน A, B, M, S3-S0, และ Cn ตามตารางสังเกตผลที่ได้ และบันทึกผลในตารางบันทึกผลที่ 3 ช่อง CPLD Test พร้อมทั้งเปรียบเทียบกับช่อง LW5 Simulation

FUNCTIONAL DESCRIPTION

The SN54/74LS181 is a 4-bit high speed parallel Arithmetic Logic Unit (ALU). Controlled by the four Function Select Inputs ($S_0 \ldots S_3$) and the Mode Control Input (M), it can perform all the 16 possible logic operations or 16 different arithmetic operations on active HIGH or active LOW operands. The Function Table lists these operations.

When the Mode Control Input (M) is HIGH, all internal carries are inhibited and the device performs logic operations on the individual bits as listed. When the Mode Control Input is LOW, the carries are enabled and the device performs arithmetic operations on the two 4-bit words. The device incorporates full internal carry lookahead and provides for either ripple carry between devices using the C_{n+4} output, or for carry lookahead between packages using the signals P (Carry Propagate) and G (Carry Generate), P and G are not affected by carry in. When speed requirements are not stringent, the LS181 can be used in a simple ripple carry mode by connecting the Carry Output (C_{n+4}) signal to the Carry Input (C_n) of the next unit. For high speed operation the LS181 is used in conjunction with the 9342 or 93S42 carry lookahead circuit. One carry lookahead package is required for each group of the four LS181 devices. Carry lookahead can be provided at various levels and offers high speed capability over extremely long word lengths.

The A = B output from the LS181 goes HIGH when all four $\overline{F}$ outputs are HIGH and can be used to indicate logic equivalence over four bits when the unit is in the subtract mode. The A = B output is open collector and can be wired-AND with other A = B outputs to give a comparison for more then four bits. The A = B signal can also be used with the C_{n+4} signal to indicate A>B and A<B.

The Function Table lists the arithmetic operations that are performed without a carry in. An incoming carry adds a one to each operation. Thus, select code LHHL generates A minus B minus 1 (2s complement notation) without a carry in and generates A minus B when a carry is applied. Because subtraction is actually performed by complementary addition (1s complement), a carry out means borrow; thus a carry is generated when there is no underflow and no carry is generated when there is underflow.

As indicated, the LS181 can be used with either active LOW inputs producing active LOW outputs or with active HIGH inputs producing active HIGH outputs. For either case the table lists the operations that are performed to the operands labeled inside the logic symbol.

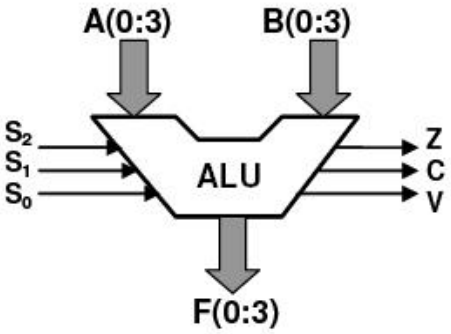
ตารางบันทึกผลที่ 4:

M = _____ S3-S0 = _____ Cn = _____							
Test Case	**A3-A0 inputs**	**B3-B0 inputs**	**Output**				
			C_{n+4}	F3'F2'F1'F0'	A=B	G'	P'
A > B	1011 – 3	0001 – 1					
A > B	1011 – 3	0010 – 2					
A = B	0011 – 3	0011 – 3					
A < B	0011 – 3	0100 – 4					
A < B	0011 – 3	0101 – 5					
A > B	1100 – 12	1010 – 10					
A > B	1011 - 11	1010 – 10					
A = B	1010 – 10	1010 – 10					
A < B	1001 - 9	1010 – 10					
A < B	1000 – 8	1010 – 10					

Experiment 03/4. ออกแบบวงจร 4Bit-ALU 8 Instruction < ไม่ต้องทำ >

11. ศึกษาการทำงาน https://forums.xilinx.com/t5/Archived-ISE-issues-Archived/4-bit-ALU/td-p/9280
12. ศึกษาการทำงาน Overflow --> http://www.righto.com/2013/01/a-small-part-of-6502-chip-explained.html
13. ออกบบวงจร 4Bit ALU

Design a 4-bit Arithmetic and Logic Unit (ALU) according to the following specification.



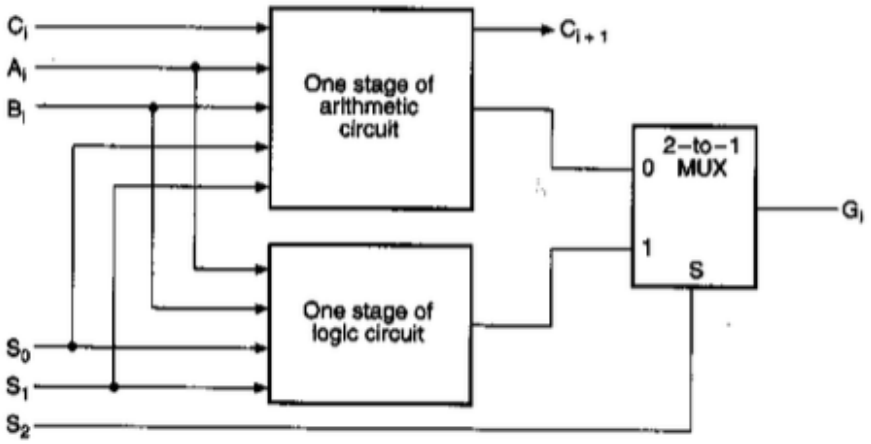
Z, C and V are status flags
Z = 1 if F=0
C = Carry or Borrow
V = Overflow

S_2	S_1	S_0	Function (F)	
0	0	0	A+B	A plus B
0	0	1	A-B	A minus B
0	1	0	A-1	A minus 1
0	1	1	A+1	A plus 1
1	0	0	A $\wedge$ B	A and B
1	0	1	A $\vee$ B	A or B
1	1	0	NOT A	not A
1	1	1	A $\oplus$ B	A xor B

Experiment 04/4. ออกแบบวงจร 2Bit-ALU 16Instruction < ไม่ต้องทำ >

14. Design 2-Bit ALU As Shown in Figure Below. ALU Design Should Have Min Of 16 Instructions.

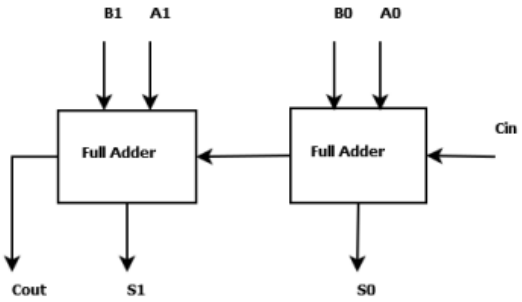
2. Implement 2-bit ALU as shown in Figure below. (30 points)



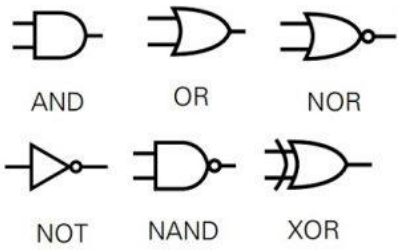
ALU design should have min of 16 Instructions.

Operation Select					
S_2	S_1	S_0	C_{in}	Operation	Function
0	0	0	0	$G = A$	Transfer A
0	0	0	1	$G = A + 1$	Increment A
0	0	1	0	$G = A + B$	Addition
0	0	1	1	$G = A + B + 1$	Add with carry input of 1
0	1	0	0	$G = A + \overline{B}$	A plus 1's complement of B
0	1	0	1	$G = A + \overline{B} + 1$	Subtraction
0	1	1	0	$G = A - 1$	Decrement A
0	1	1	1	$G = A$	Transfer A
1	0	0	X	$G = A \wedge B$	AND
1	0	1	X	$G = A \vee B$	OR
1	1	0	X	$G = A \oplus B$	XOR
1	1	1	X	$G = \overline{A}$	NOT (1's complement)

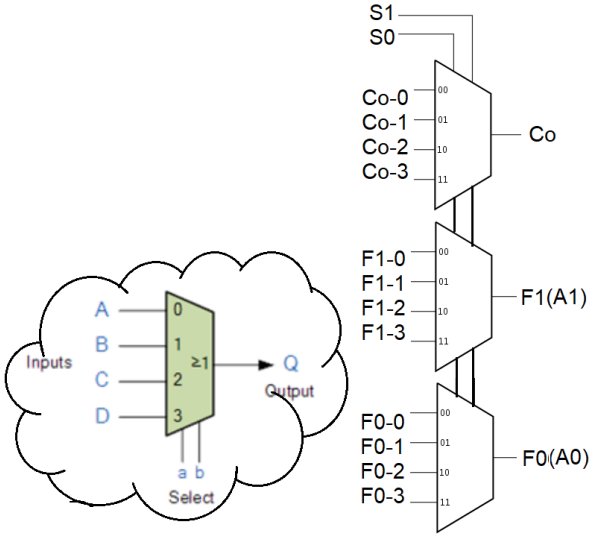
15. Top-Down Digital Design



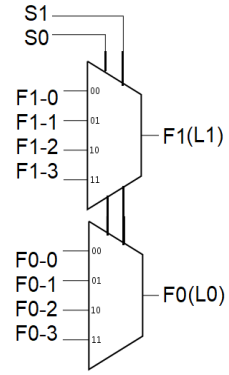
Sub-Arithmetic Unit - from full Adder



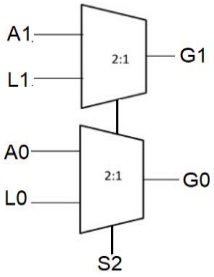
Sub-Arithmetic Unit - From Truth Table



Arithmetic Unit {Co, A1, A0}



Logic Unit {L1, L0}



ALU {G1, G0}

16. Designed with Adder

- G=A+0+Ci
- G=A+B+Ci
- G=A+B'+Ci
- G=A+(-1)+Ci
- From full adder ➔ S = A⊕B⊕Ci, Co = AB+ACi+BCi

17. Designed with Truth Table

Arithmetic Unit (S2=0)						
Input			S1S0 = 00	S1S0 = 01	S1S0 = 10	S1S0 = 11
Ci	A1A0	B1B0	A+Ci	A+B+Ci	A+B'+Ci	A-1+Ci
0	00	00	0-00	0-00		
0	00	01	0-00	0-01		
0	00	10	0-00	0-10		
0	00	11	0-00	0-11		
0	01	00	0-01	0-01		
0	01	01	0-01	0-10		
0	01	10	0-01	0-11		
0	01	11	0-01	1-00		
0	10	00	0-10	0-10		
0	10	01	0-10	0-11		
0	10	10	0-10	1-00		
0	10	11	0-10	1-01		
0	11	00	0-11	0-11		
0	11	01	0-11	1-00		
0	11	10	0-11	1-01		
0	11	11	0-11	1-10		
1	00	00	0-01	0-01		
1	00	01	0-01	0-10		
1	00	10	0-01	0-11		
1	00	11	0-01	1-00		
1	01	00	0-10	0-10		
1	01	01	0-10	0-11		
1	01	10	0-10	1-00		
1	01	11	0-10	1-01		
1	10	00	0-11	0-11		
1	10	01	0-11	1-00		
1	10	10	0-11	1-01		
1	10	11	0-11	1-10		
1	11	00	1-00	1-00		
1	11	01	1-00	1-01		
1	11	10	1-00	1-10		
1	11	11	1-00	1-11		

Logic Unit (S2=1)					
Input		S1S0 = 00	S1S0 = 01	S1S0 = 10	S1S0 = 11
A1A0	B1B0	AND	OR	XOR	A'
00	00	00	00	00	11
00	01	00	01	01	11
00	10	00	10	10	11
00	11	00	11	11	11
01	00	00	01		10
01	01	01	01		10
01	10	00	11		10
01	11	01	11		10
10	00	00			01
10	01	00			01
10	10	10			01
10	11	10			01
11	00				00
11	01				00
11	10				00
11	11				00

18. Exercise 01. ออกแบบโดยใช้ 74181 เพื่อทำวงจร Output F[16] = A[8] plus B[8]

แนวคิด

- ใช้ 74181 ตั้งค่า Select [S3S2S1S0] = 1001 (A plus B) เป็น 4-bit Adder >> M=0 และ Cin=1(no carry)
- ใช้ 4 bit Adder 2 ชุด สำหรับ 4บิตบน และ 4บิตล่าง (Low Nibble)
- ตัวทด Carry Out (C_{n+4})จาก ชุด 4 บิตล่าง [3-0] ป้อนเป็น Carry In ชุด 4 บิตบน [7-4]
- ตัวทด Carry Out จาก ชุด 4 บิตบน[7-4] ป้อนเป็น Output บิต 8 (Active Low >> Not)
- Output บิต 15-9 = 0

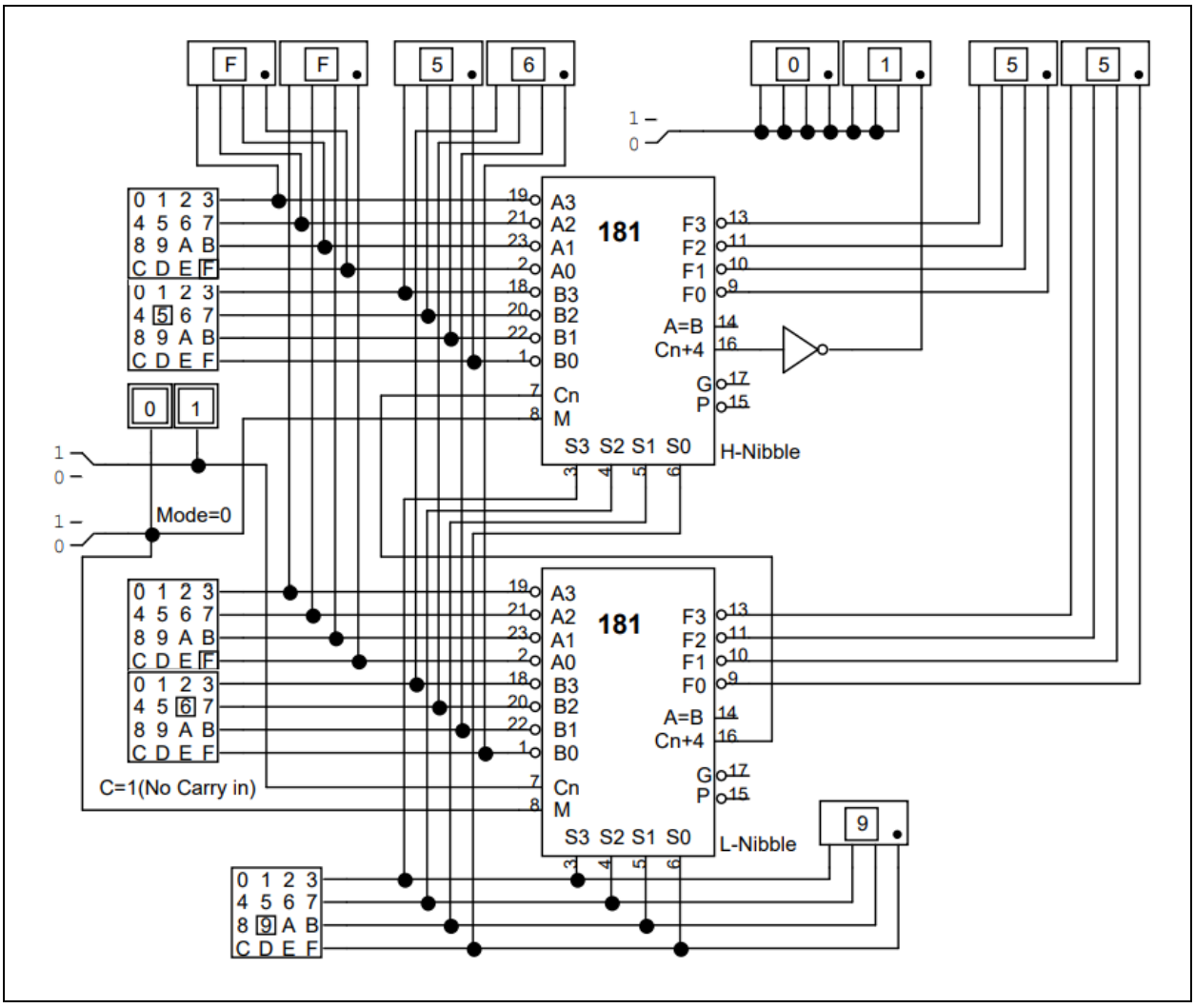
ผลการทดสอบ

M = _0 (Arithmetic Mode) ____ S3-S0 = _1001_____ Cn = _1 (no carry) ____

A	B	F	Note
00	FE		
11	DC		
22	BA		
33	98		
44	76		
55	54		
66	32		
77	10		

A	B	F	Note
88	01		
99	23		
AA	45		
BB	67		
CC	89		
DD	AB		
EE	CD		
FF	EF		

<u>วงจรที่ได้</u>


F
F
5
6
0
1
5
5
181
A3
A2
A1
A0
B3
B2
B1
B0
Cn
M
S3 S2 S1 S0
F3
F2
F1
F0
A=B
Cn+4
G
P
H-Nibble
L-Nibble
Mode=0
C=1(No Carry in)
9

19. Exercise 02. ออกแบบโดยใช้ 74181 เพื่อทำวงจร Output F[8] = A[4] minus B[4]

แนวคิด

ผลการทดสอบ

M = _0 (Arithmetic Mode) _____ S3-S0 = _____ Cn = _____

A	B	F	Note
0	F		0-F = F1
1	E		
2	C		
3	9		3-9 = FA
4	7		
5	5		
6	3		
7	1		

A	B	F	Note
8	0		8-0 = 08
9	2		
A	4		
B	6		
C	8		
D	A		
E	C		
F	E		

วงจรที่ได้

20. Exercise 03. ออกแบบโดยใช้ 74181 เพื่อทำวงจร Output X=1 ถ้า A[4] <= B[4]

แนวคิด

ผลการทดสอบ

M = _____ S3-S0 = _____ Cn = _____

A	B	F	Note

A	B	F	Note

วงจรที่ได้

21. Exercise 04. ออกแบบโดยใช้ 74181 เพื่อทำวงจร Output X=1 ถ้า A[4] >= B[4]

แนวคิด

ผลการทดสอบ

M = _____ S3-S0 = _____ Cn = _____

A	B	F	Note

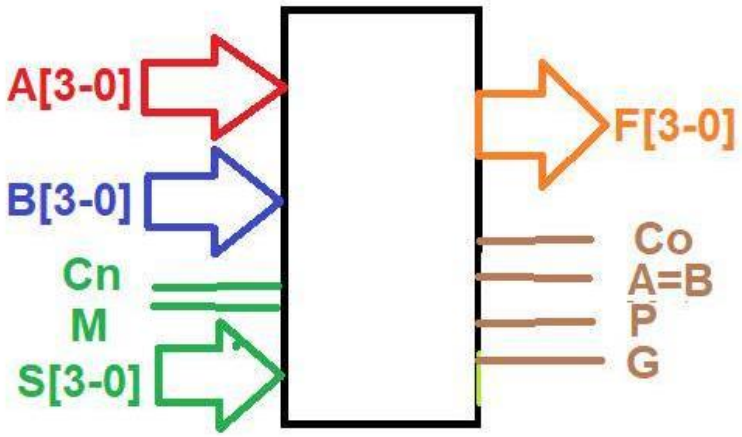
A	B	F	Note

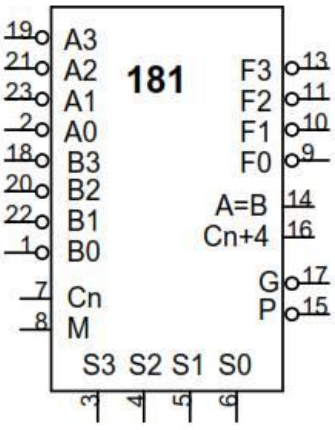
วงจรที่ได้

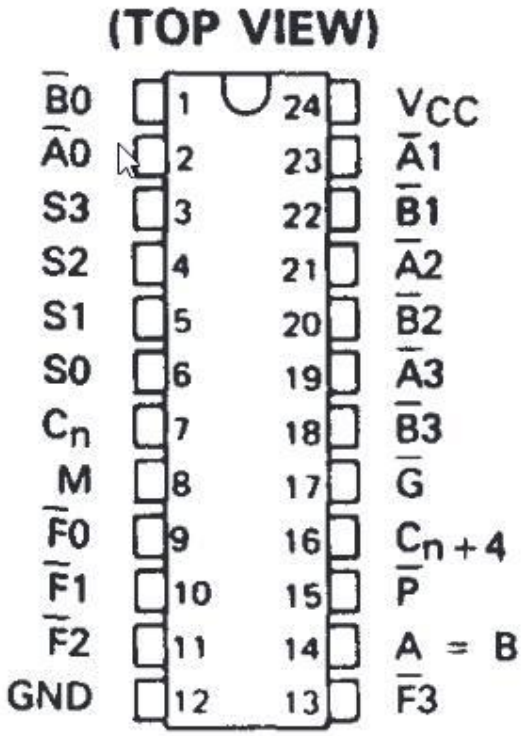
คำแนะนำอย่างย่อ เพื่อการทดลอง

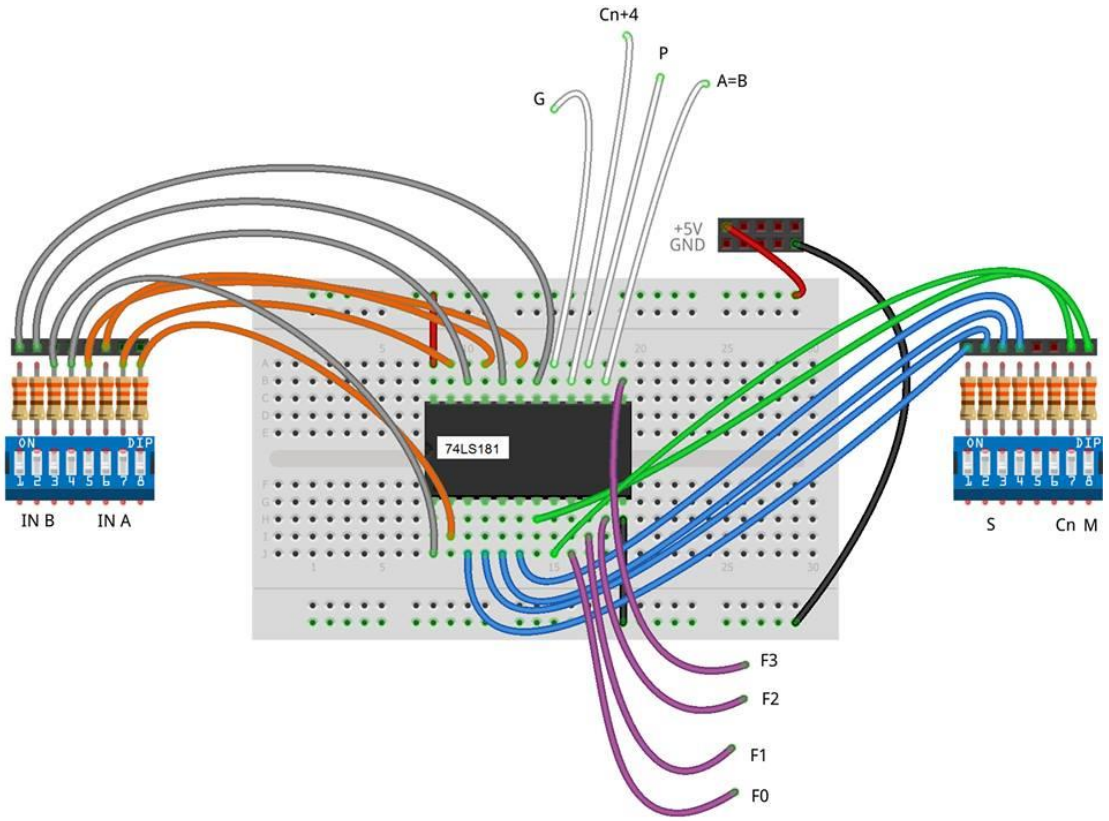
การทดลองข้อที่ 19, 20, 21 – ทดสอบด้วยไอซี 74181

1. การต่อวงจรด้วย 74181 ทดสอบการทำงาน









Logical Operation: M=1

S3	*S2*	*S1*	*S0*	*Operation*
H	*L*	*H*	*H*	*A ∧ B*
H	*H*	*H*	*L*	*A V B*
L	*L*	*L*	*L*	*A'*
L	*H*	*L*	*L*	*(A ∧ B)'*
L	*L*	*L*	*H*	*(A V B)'*

Arithmetic Operation: M=0

S3	*S2*	*S1*	*S0*	*Cn*	*Operation*
H	*L*	*L*	*H*	*H*	*A + B*
L	*H*	*H*	*L*	*L*	*A - B*
H	*H*	*H*	*H*	*H*	*A - 1*