

การทดลองที่ 6 การพัฒนาโปรแกรมด้วยภาษา Assembly

11. ป้อนคำสั่งเหล่านี้ในไฟล์ main.s

```
.global main
main:
MOV R0, #0
MOV R1, #2
MOV R2, #4
ORR R0, R1, R2
BX LR
```

Handwritten binary calculation for the ORR instruction:

$R0 = 0 \rightarrow 0000_2$

$R1 = 2 \rightarrow 0010_2$

$R2 = 4 \rightarrow 0100_2$

ORR operation:

0010 <sub>2</sub>	OR	0100 <sub>2</sub>	=	0110 <sub>2</sub>
0010 <sub>2</sub>		0100 <sub>2</sub>		0110 <sub>2</sub>
				↓
				6

An arrow points from the ORR instruction in the assembly code to the final result '6'.

12. เลือกเมนู Build->Build เพื่อแปล (Assemble) โปรแกรมที่เขียนให้เป็นโปรแกรมภาษาเครื่อง
13. เลือกเมนู Build->Run เพื่อรันโปรแกรม
14. อ่านและบันทึกประโยคที่เกิดขึ้นในหน้าต่าง Terminal ที่ปรากฏขึ้นมา

2. จงปรับแก้โปรแกรมใน main.s เป็นดังนี้ จดบันทึกผลการทดสอบและอธิบาย

1 main	MOV	R5, #1	R0	0x1	Dec	Bin	Hex
2 loop	CMP	R4, #0	R1	0x0	Dec	Bin	Hex
3	BLE	lend	R2	0x0	Dec	Bin	Hex
4 else	MOV	R5, #2	R3	0x0	Dec	Bin	Hex
5 lend	MOV	R0, R5	R4	0x0	Dec	Bin	Hex
6			R5	0x1	Dec	Bin	Hex

ตอบ คำสั่งไล่ที่ละบรรทัดเป็นดังนี้

1. กำหนด Label ชื่อ main และทำคำสั่งโหลดค่า 1 เข้าไปใน register R5
2. กำหนด Label ชื่อ loop และทำคำสั่งเปรียบเทียบใน register R4 กับค่าคงที่ 0
3. ถ้าผลลัพธ์จากการเปรียบเทียบในบรรทัดที่แล้วมีค่า  $\leq 0$  ให้ข้ามไปทำงานที่ Label lend
4. กำหนด Label ชื่อ else และทำคำสั่งโหลดค่า 5 เข้าไปใน register R5
5. กำหนด Label ชื่อ lend และทำคำสั่งโหลดค่าจาก register R5 เข้าไปใน register R0

ลำดับการทำงานคือบรรทัดที่ 1 -> 2 -> 3 -> 5

3. จงปรับแก้โปรแกรมใน main.s เป็นดังนี้ จดบันทึกผลการทดสอบและอธิบาย

Reset to continue loading code			
1	var1	DCD	1
2	var1addr	DCD	var1
3			
4	main	MOV	R1, #2
5		LDR	R2, =var1
6		STR	R1, [R2]
7		LDR	R0, [R2]
8			

R0	0x2	Dec	Bin
R1	0x2	Dec	Bin
R2	0x100	Dec	Bin
R3	0x0	Dec	Bin
R4	0x0	Dec	Bin

ตอบ คำสั่งไล่ที่ละบรรทัดเป็นดังนี้

1. กำหนดตัวแปรชื่อ var1 ให้เก็บค่าคงตัว 1
2. กำหนดตัวแปรชื่อ var1addr ให้เก็บ address ของตัวแปร var1 ไว้
3. เว้นไว้เพื่อความอ่านง่าย
4. กำหนด Label ชื่อ main และทำคำสั่งโหลดค่า 2 เข้าไปใน register R1
5. โหลดค่า address ของตัวแปร var1 เข้าไปใน register R2
6. นำค่าจาก register R1 ไปเก็บไว้ที่ address ที่ register R2 เก็บไว้อยู่
7. โหลดค่าจาก address ที่ register R2 เก็บไว้อยู่ ไปไว้ที่ register R0