

CS231 ระบบคอมพิวเตอร์และภาษาแอสเซมบลี

บทที่ 5: การประมวลผลระดับบิตและตรรกศาสตร์



ผู้ช่วยศาสตราจารย์ ดร. ปวีณ เชื้อนแก้ว
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้



การประมวลผลระดับบิต

AND operand1, operand2
OR operand1, operand2
XOR operand1, operand2
TEST operand1, operand2

Operand ที่ใช้ได้:

REG, memory
memory, REG
REG, REG
memory, immediate
REG, immediate

ผลการทำงานจะเก็บที่ operand ตัวแรก

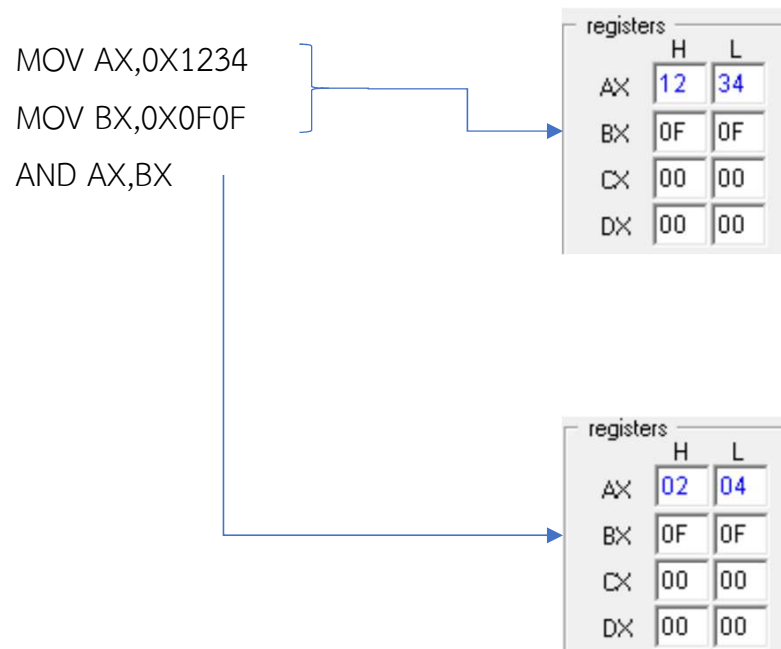
NOT operand1

Operand ที่ใช้ได้:

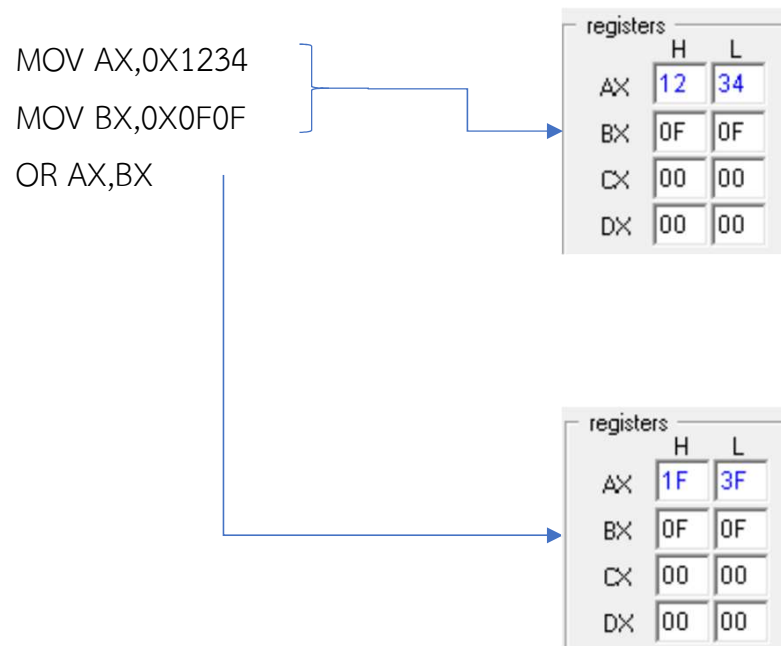
REG
Memory

ผลการทำงานจะเก็บที่ operand

AND operand1, operand2



OR operand1, operand2



XOR operand1, operand2

MOV AX,1010101001111111B

MOV BX,1001000100000001B

XOR AX,BX

registers	
	H L
AX	12 34
BX	0F 0F
CX	00 00
DX	00 00

1010101001111111

1001000100000001

registers	
	H L
AX	1F 3F
BX	0F 0F
CX	00 00
DX	00 00

0011101101111110

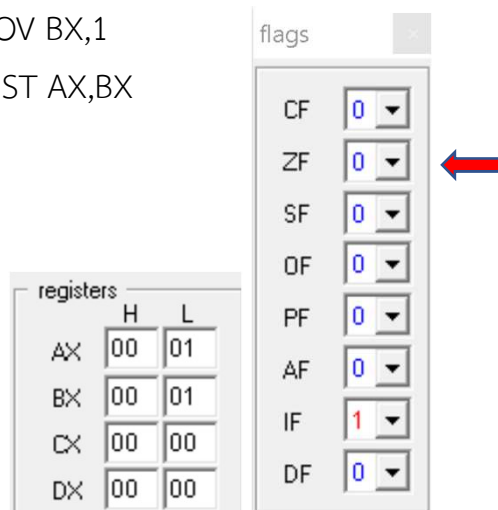
TEST operand1, operand2

Test ทำงานเหมือนกับ AND ต่างกันที่จะเปลี่ยนแปลง flag ตามผลการทำงาน แต่จะไม่เก็บคำตอบใน operand

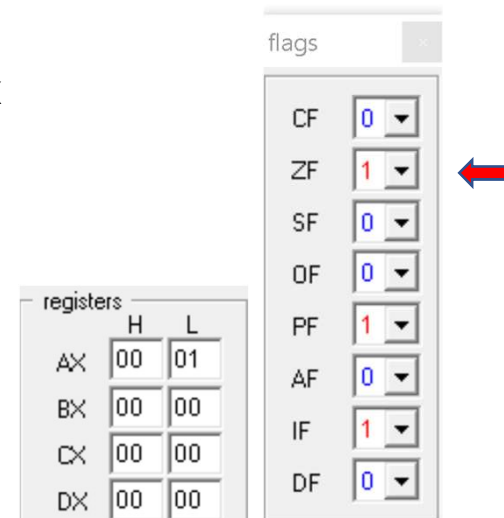
ข้อดี: ค่าใน operand ตัวแรกไม่หาย

ข้อเสีย: ถ้า operand มีมากกว่า 1 บิต การตีความจาก flag จะกำกวม (ตอบ 0 ZF จะติด, ตอบ 1 ZF จะดับ)

```
MOV AX,1  
MOV BX,1  
TEST AX,BX
```



```
MOV AX,1  
MOV BX,0  
TEST AX,BX
```



NOT operand

Operand ที่ใช้ได้:

REG

Memory

ผลการทำงานจะเก็บที่ operand

MOV AX,0xF0F0

MOV BL,10010110B

registers		
	H	L
AX	F0	F0
BX	00	96
CX	00	00
DX	00	00

registers		
	H	L
AX	0F	0F
BX	00	96
CX	00	00
DX	00	00

NOT AX

registers		
	H	L
AX	0F	0F
BX	00	69
CX	00	00
DX	00	00

NOT BL

registers		
	H	L
AX	0F	0F
BX	FF	96
CX	00	00
DX	00	00

NOT BX

CMP operand1,operand2 เปรียบเทียบจำนวน 2 ค่า

Operand ที่ใช้ได้:

REG, memory

memory, REG

REG, REG

memory, immediate

REG, immediate

ผลการทำงานจะเก็บที่ flag เท่านั้น

การแปลผลคำตอบ

	CF	ZF	SF
Operand1 == Operand2	0	1	0
Operand1 > Operand2	0	0	0
Operand1 < Operand2	1	0	1

MOV AX,1234

registers		H	L
AX		04	D2
BX		00	00
CX		00	00
DX		00	00

flags	
CF	0
ZF	1
SF	0
OF	0
PF	1
AF	0
IF	1
DF	0

CMP AX,1234

flags	
CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

CMP AX,1233

flags	
CF	1
ZF	0
SF	1
OF	0
PF	1
AF	1
IF	1
DF	0

CMP AX,1235

การเลื่อนบิต

SHR : Shift Right

SAR : Shift Arithmetic Right

SHL : Shift Left

SAL : Shift Arithmetic Left

ROL : Rotate Left

ROR : Rotate Right

RCL : Rotate Carry Left

RCR : Rotate Carry Right

คำสั่งเหล่านี้เป็น Opcode Extensions กลุ่มเดียวกับ MUL และ DIV จึงไม่มีใน opcode map

SHR : Right Shift ใช้เลื่อนบิตไปทางด้านขวา

SHR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

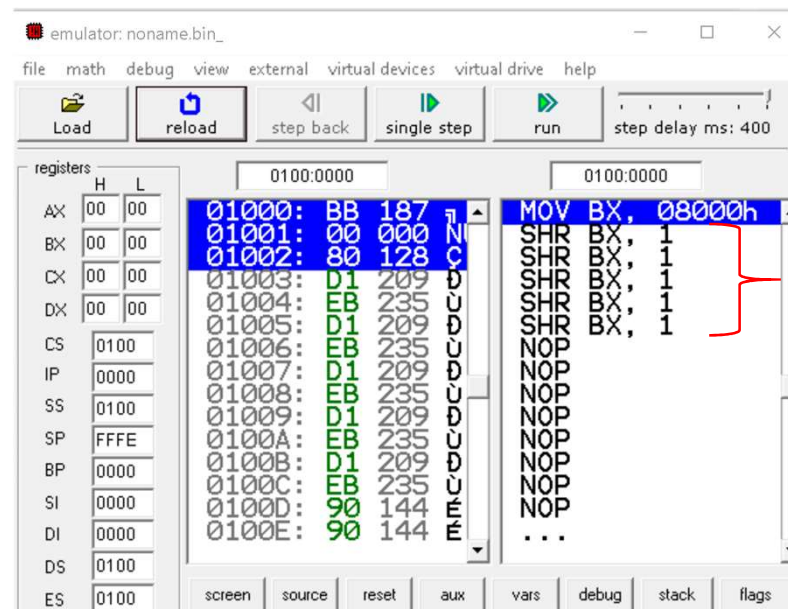
memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

MOV BX,0X8000

SHR BX,5



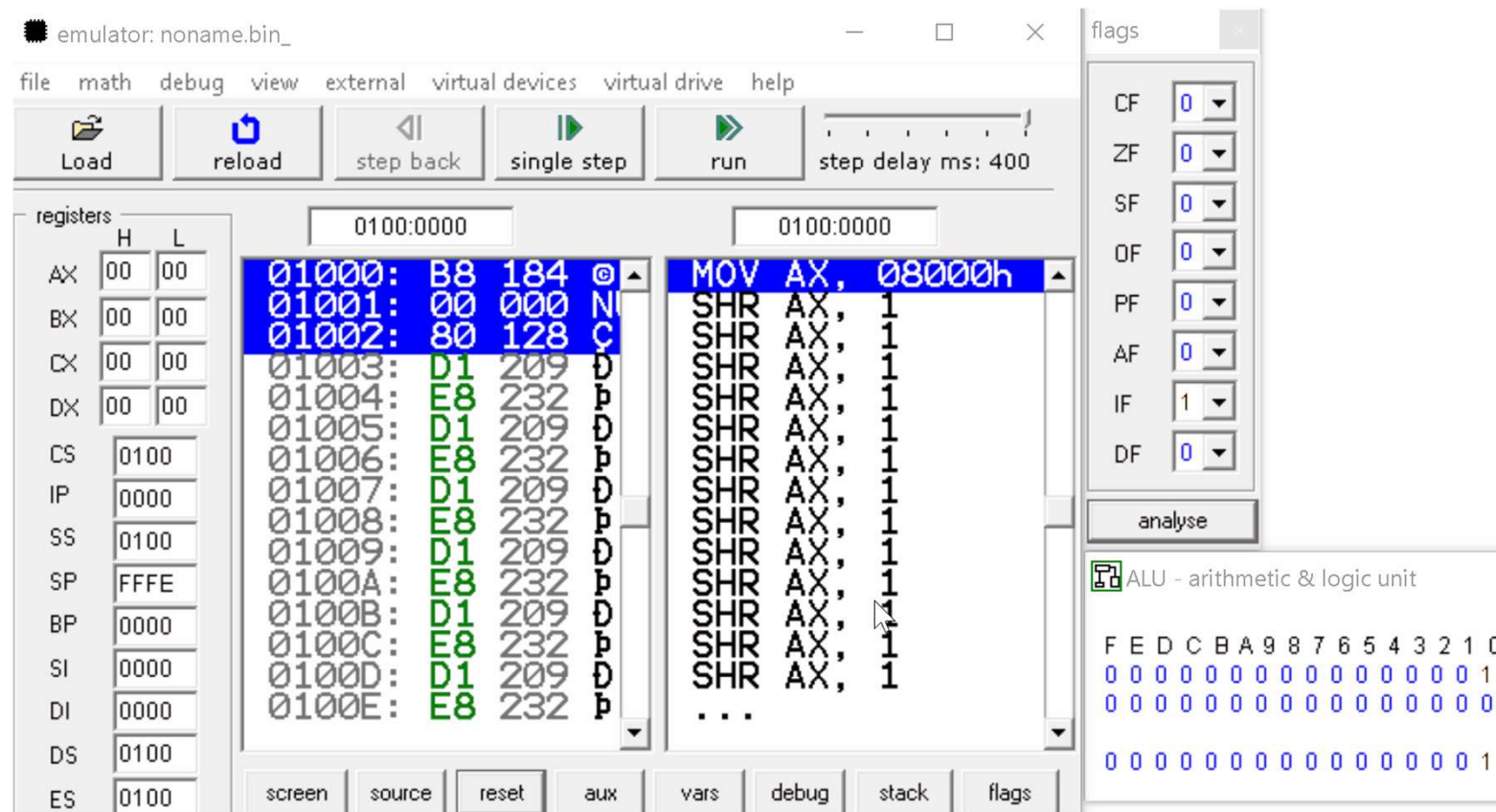
SHR BX,5

SHR : Right Shift ใช้เลื่อนบิตไปทางด้านขวา

SHR operand1,operand2

MOV AX,0X8000

SHR AX,15



SHL : Right Left ใช้เลื่อนบิตไปทางด้านซ้าย

SHR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

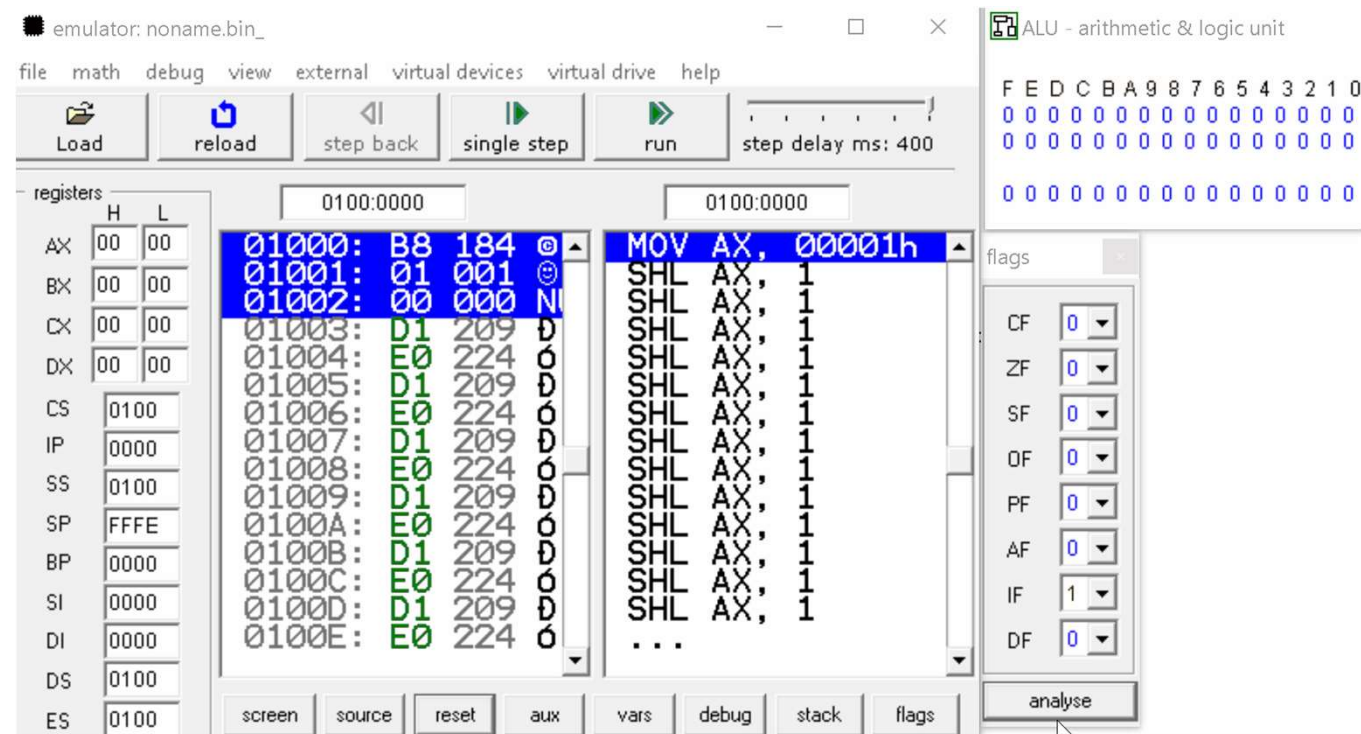
REG, immediate

memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

MOV AX,1
SHL AX,18



การเลื่อนบิตไปทางด้านขวา หากข้อมูลเป็น 2's complement ความหมายจะเปลี่ยนไปเพราะมีการนำ 0 มาเติมด้านซ้าย

หากต้องการให้คงค่าเป็นติดลบไว้ ต้องนำ 1 มาเติมด้านซ้ายแทนเลข 0

หากค่าไม่ใช่ 2's complement (บิตซ้ายสุดเป็น 0)ให้นำ 0 เติมด้านซ้ายเหมือนเดิม

คำสั่งที่ทำงานแบบนี้คือ SAR

SAR : Shift Arithmetic Right ใช้เลื่อนบิตไปทางด้านขวา

SAR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate REG, immediate **memory, CL** **register, CL**

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

SAR : Shift Arithmetic Right ใช้เลื่อนบิตไปทางด้านขวาแบบคิดเครื่องหมาย

SAR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

memory, CL

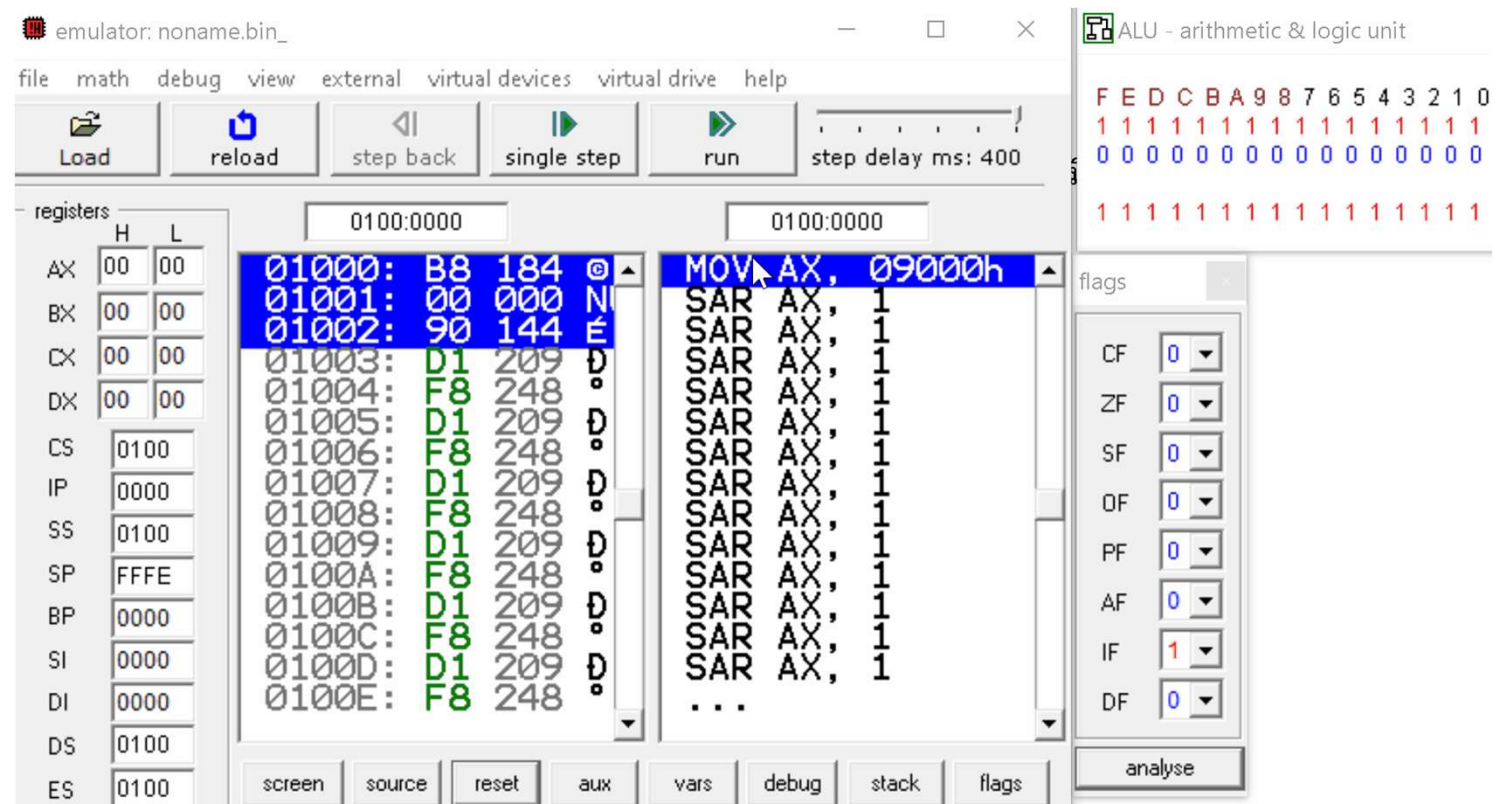
register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

เลขติดลบ

MOV AX,0x9000

SAR AX,18



SAR : Shift Arithmetic Right ใช้เลื่อนบิตไปทางด้านขวาแบบคิดเครื่องหมาย

SAR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate	REG, immediate	memory, CL	register, CL
-------------------	----------------	------------	--------------

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

เลขไม่ติดลบ

MOV AX,0x7000

SAR AX,18

The screenshot shows the emulor interface with the following components:

- Top Bar:** File, math, debug, view, external, virtual devices, virtual drive, help.
- Buttons:** Load, reload, step back, single step, run, step delay ms: 400.
- Registers:**

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	
- Assembly Code:**

```

0100:0000 MOV AX, 07000h
0100:0001 SAR AX, 1
0100:0002 SAR AX, 1
0100:0003 SAR AX, 1
0100:0004 SAR AX, 1
0100:0005 SAR AX, 1
0100:0006 SAR AX, 1
0100:0007 SAR AX, 1
0100:0008 SAR AX, 1
0100:0009 SAR AX, 1
0100:000A SAR AX, 1
0100:000B SAR AX, 1
0100:000C SAR AX, 1
0100:000D SAR AX, 1
0100:000E SAR AX, 1

```
- Flags:**

Flag	Value
CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

SAL : Shift Arithmetic Left ใช้เลื่อนบิตไปทางด้านซ้ายแบบคิดเครื่องหมาย

SAL operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate REG, immediate memory, CL register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

การเลื่อนบิตไปทางด้านซ้าย ส่งผลให้ข้อมูลมีค่าสูงขึ้น หากบิตแรกเป็น 1 จะเกิด overflow
หลังจากเกิด overflow แล้วหากยังเลื่อนบิตต่อไปอีก ค่าจะติดลบน้อยลงเรื่อย ๆ

การเลื่อนบิตไปด้านซ้าย ต้องนำ 0 มาเติมด้านขวาเท่านั้น

การตีความผลการคำนวณ ALU

0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	a	10
11	b	11
12	c	12
13	d	13
14	e	14
15	f	15
16	10	16
17	11	17
18	12	18
19	13	19
20	14	20
21	15	21
22	16	22
23	17	23
24	18	24
25	19	25
26	1a	26
27	1b	27
28	1c	28
29	1d	29
30	1e	30
31	1f	31
32	20	32
33	21	33
34	22	34
35	23	35
36	24	36
37	25	37
38	26	38
39	27	39
40	28	40
41	29	41
42	2a	42
43	2b	43
44	2c	44
45	2d	45
46	2e	46
47	2f	47
48	30	48
49	31	49
50	32	50
51	33	51
52	34	52
53	35	53
54	36	54
55	37	55
56	38	56
57	39	57
58	3a	58
59	3b	59
60	3c	60
61	3d	61
62	3e	62
63	3f	63

64	40	64
65	41	65
66	42	66
67	43	67
68	44	68
69	45	69
70	46	70
71	47	71
72	48	72
73	49	73
74	4a	74
75	4b	75
76	4c	76
77	4d	77
78	4e	78
79	4f	79
80	50	80
81	51	81
82	52	82
83	53	83
84	54	84
85	55	85
86	56	86
87	57	87
88	58	88
89	59	89
90	5a	90
91	5b	91
92	5c	92
93	5d	93
94	5e	94
95	5f	95
96	60	96
97	61	97
98	62	98
99	63	99
100	64	100
101	65	101
102	66	102
103	67	103
104	68	104
105	69	105
106	6a	106
107	6b	107
108	6c	108
109	6d	109
110	6e	110
111	6f	111
112	70	112
113	71	113
114	72	114
115	73	115
116	74	116
117	75	117
118	76	118
119	77	119
120	78	120
121	79	121
122	7a	122
123	7b	123
124	7c	124
125	7d	125
126	7e	126
127	7f	127

128	80	-128
129	81	-127
130	82	-126
131	83	-125
132	84	-124
133	85	-123
134	86	-122
135	87	-121
136	88	-120
137	89	-119
138	8a	-118
139	8b	-117
140	8c	-116
141	8d	-115
142	8e	-114
143	8f	-113
144	90	-112
145	91	-111
146	92	-110
147	93	-109
148	94	-108
149	95	-107
150	96	-106
151	97	-105
152	98	-104
153	99	-103
154	9a	-102
155	9b	-101
156	9c	-100
157	9d	-99
158	9e	-98
159	9f	-97
160	a0	-96
161	a1	-95
162	a2	-94
163	a3	-93
164	a4	-92
165	a5	-91
166	a6	-90
167	a7	-89
168	a8	-88
169	a9	-87
170	aa	-86
171	ab	-85
172	ac	-84
173	ad	-83
174	ae	-82
175	af	-81
176	b0	-80
177	b1	-79
178	b2	-78
179	b3	-77
180	b4	-76
181	b5	-75
182	b6	-74
183	b7	-73
184	b8	-72
185	b9	-71
186	ba	-70
187	bb	-69
188	bc	-68
189	bd	-67
190	be	-66
191	bf	-65

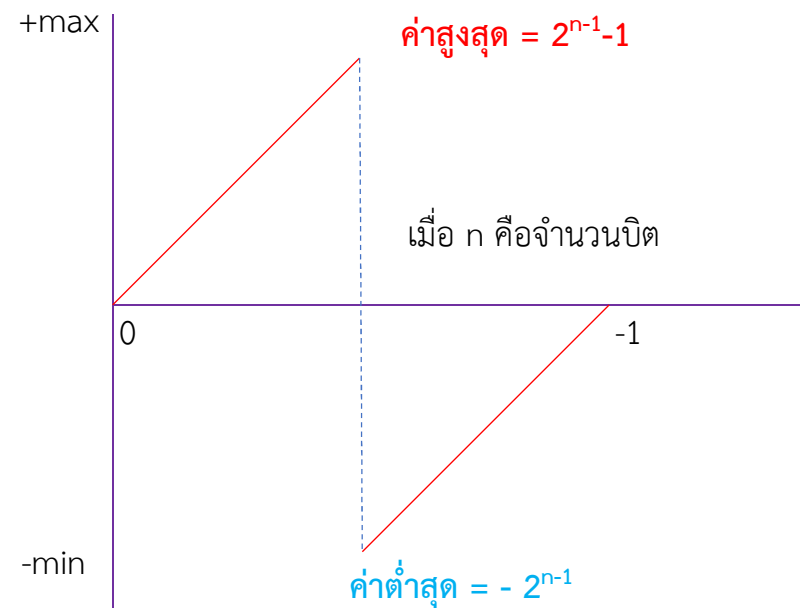
192	c0	-64
193	c1	-63
194	c2	-62
195	c3	-61
196	c4	-60
197	c5	-59
198	c6	-58
199	c7	-57
200	c8	-56
201	c9	-55
202	ca	-54
203	cb	-53
204	cc	-52
205	cd	-51
206	ce	-50
207	cf	-49
208	d0	-48
209	d1	-47
210	d2	-46
211	d3	-45
212	d4	-44
213	d5	-43
214	d6	-42
215	d7	-41
216	d8	-40
217	d9	-39
218	da	-38
219	db	-37
220	dc	-36
221	dd	-35
222	de	-34
223	df	-33
224	e0	-32
225	e1	-31
226	e2	-30
227	e3	-29
228	e4	-28
229	e5	-27
230	e6	-26
231	e7	-25
232	e8	-24
233	e9	-23
234	ea	-22
235	eb	-21
236	ec	-20
237	ed	-19
238	ee	-18
239	ef	-17
240	f0	-16
241	f1	-15
242	f2	-14
243	f3	-13
244	f4	-12
245	f5	-11
246	f6	-10
247	f7	-9
248	f8	-8
249	f9	-7
250	fa	-6
251	fb	-5
252	fc	-4
253	fd	-3
254	fe	-2
255	ff	-1

ตัวอย่างเส้นจำนวนของเลข 2's complement ขนาด 8 บิต

จำนวนตัวเลขทั้งหมด 256

ค่าสูงสุด 127

ค่าน้อยสุด -128



การป้อนข้อมูลตรวจสอบให้อยู่ในช่วง เพราะถ้าเกิดการล้น

หรือ Overflow

ข้อมูลอาจเปลี่ยนจาก 127 เป็น -128 ได้

SAL : Shift Arithmetic Left ใช้เลื่อนบิตไปทางด้านซ้ายแบบคิดเครื่องหมาย

SAL operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate REG, immediate memory, CL register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

การเลื่อนบิตไปทางด้านซ้าย ส่งผลให้ข้อมูลมีค่าสูงขึ้น หากบิตแรกเป็น 1 จะเกิด overflow
หลังจากเกิด overflow แล้วหากยังเลื่อนบิตต่อไปอีก ค่าจะติดลบน้อยลงเรื่อย ๆ

การเลื่อนบิตไปด้านซ้าย ต้องนำ 0 มาเติมด้านขวาเท่านั้น

ดังนั้นการเลื่อนบิตไปทางด้านซ้ายแบบคิดเครื่องหมายแบบคิดเครื่องหมายจึงไม่จำเป็น

SAL == SHL

SAL : Shift Arithmetic Left ใช้เลื่อนบิตไปทางด้านซ้ายแบบคิดเครื่องหมาย

SAL operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

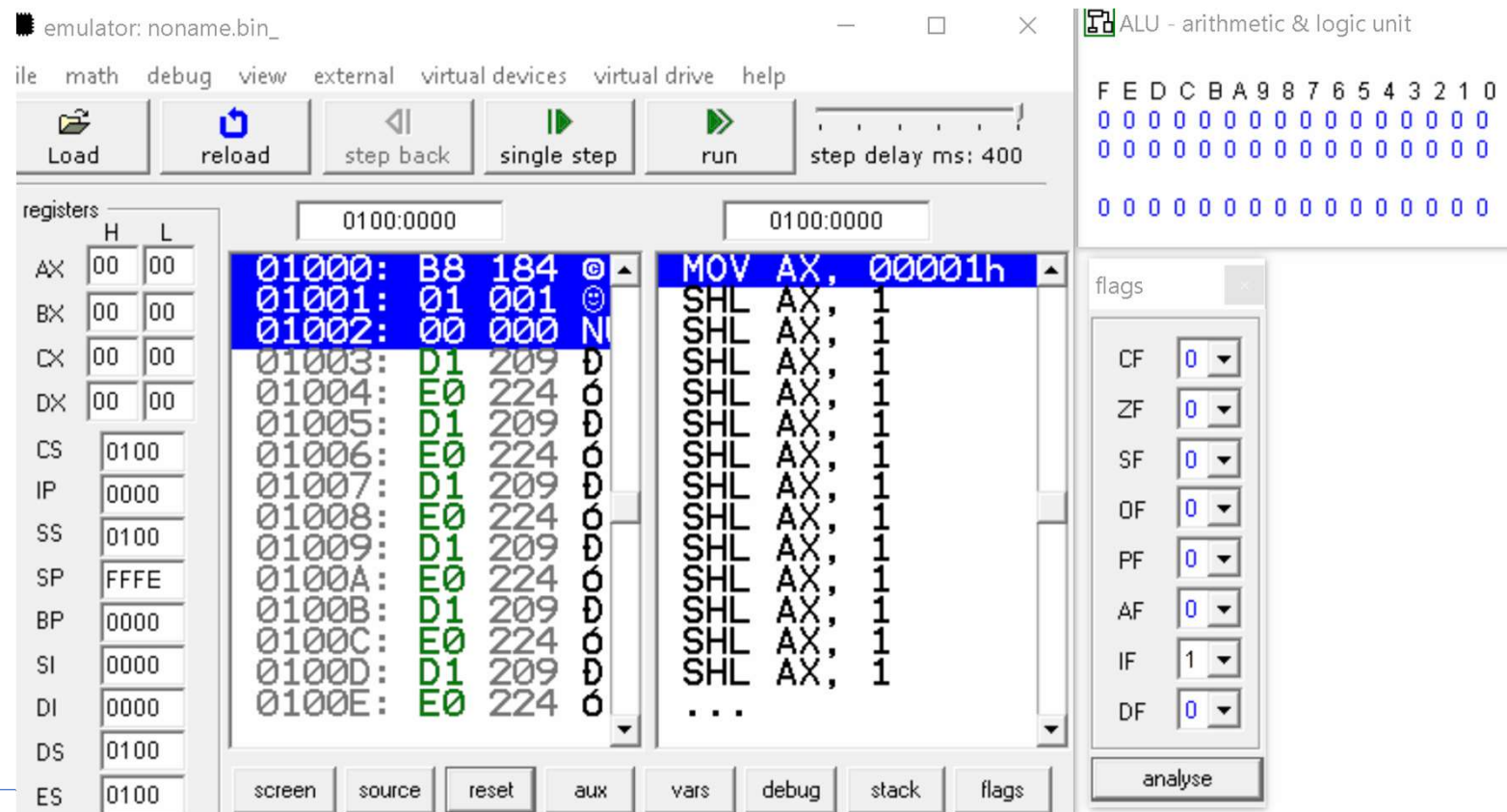
ค่าบวก

MOV AX,1B

SAL AX,20

SAL ถูกแปลเป็น

SHL



SAL : Shift Arithmetic Left ใช้เลื่อนบิตไปทางด้านซ้ายแบบคิดเครื่องหมาย

SAL operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

ค่าติดลบ

MOV AX,0xFFFF

SAL AX,20

SAL ถูกแปลเป็น

SHL

emulator: noname.bin_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 400

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000 0100:0000

01000: B8 184 @ MOV AX, 0FFFFh

01001: FF 255 R SHL AX, 1

01002: FF 255 R SHL AX, 1

01003: D1 209 D SHL AX, 1

01004: E0 224 6 SHL AX, 1

01005: D1 209 D SHL AX, 1

01006: E0 224 6 SHL AX, 1

01007: D1 209 D SHL AX, 1

01008: E0 224 6 SHL AX, 1

01009: D1 209 D SHL AX, 1

0100A: E0 224 6 SHL AX, 1

0100B: D1 209 D SHL AX, 1

0100C: E0 224 6 SHL AX, 1

0100D: D1 209 D SHL AX, 1

0100E: E0 224 6 ...

ALU - arithmetic & logic unit

F E D C B A 9 8 7 6 5 4 3 2 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

flags

CF 0

ZF 0

SF 0

OF 1

PF 0

AF 0

IF 1

DF 0

analyse

screen source reset aux vars debug stack flags

ROR : Rotate Right ใช้วนบิตไปทางด้านขวา

ROR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

memory, CL

register, CL

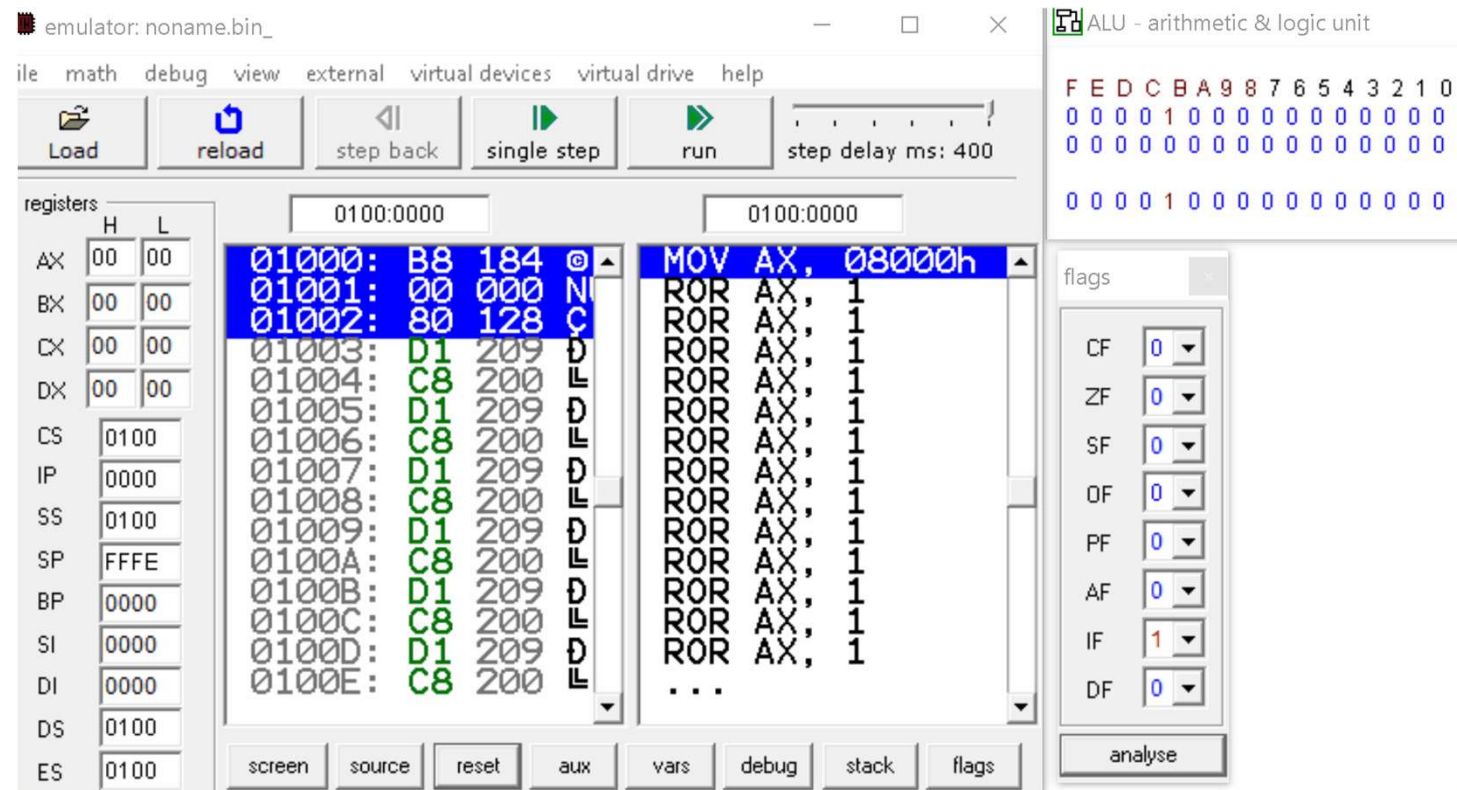
ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

เมื่อวนถึงบิตสุดท้าย CF จะมีค่าเป็น 1

MOV AX,0x8000

ROR AX,40

HLT



ROL : Rotate Left ใช้วนบิตไปทางด้านซ้าย

ROR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

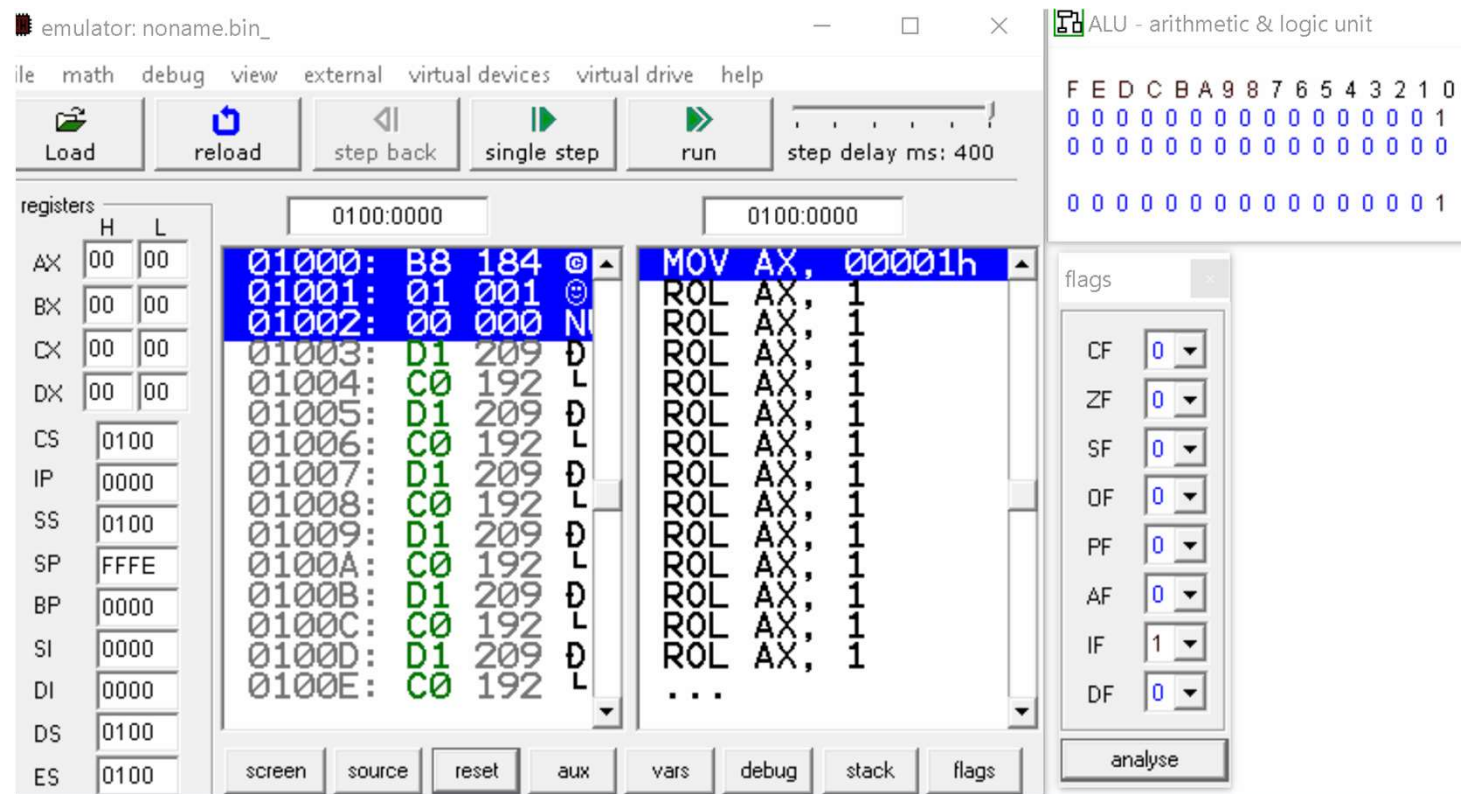
memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1

เมื่อวนถึงบิตสุดท้าย CF จะมีค่าเป็น 1

```
MOV AX,1
ROL AX,40
HLT
```



RCR : Rotate Carry Right ใช้วนบิตไปทางด้านขวารวมตัวทด

RCR operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

REG, immediate

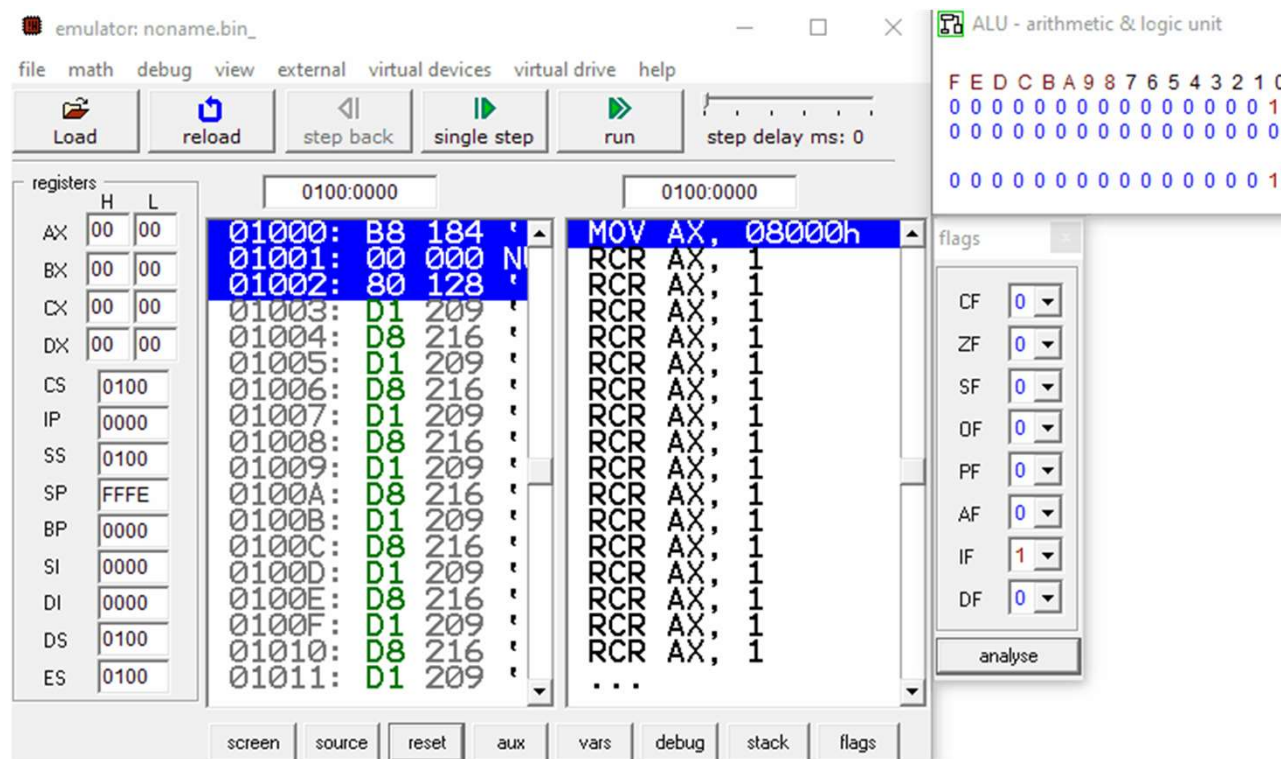
memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1 เมื่อวนถึงบิตสุดท้าย จะเลื่อนไปที่ CF แล้วจึงเลื่อนกลับมาที่บิตซ้ายสุด

MOV AX,0x8000

RCR AX,32



RCL : Rotate Carry Left ใช้วนบิตไปทางด้านซ้ายรวมตัวทด

RCL operand1,operand2

Operand1 คือข้อมูลที่ต้องการเลื่อน Operand2 คือจำนวนครั้งในการเลื่อน

Operand ที่ใช้ได้:

memory, immediate

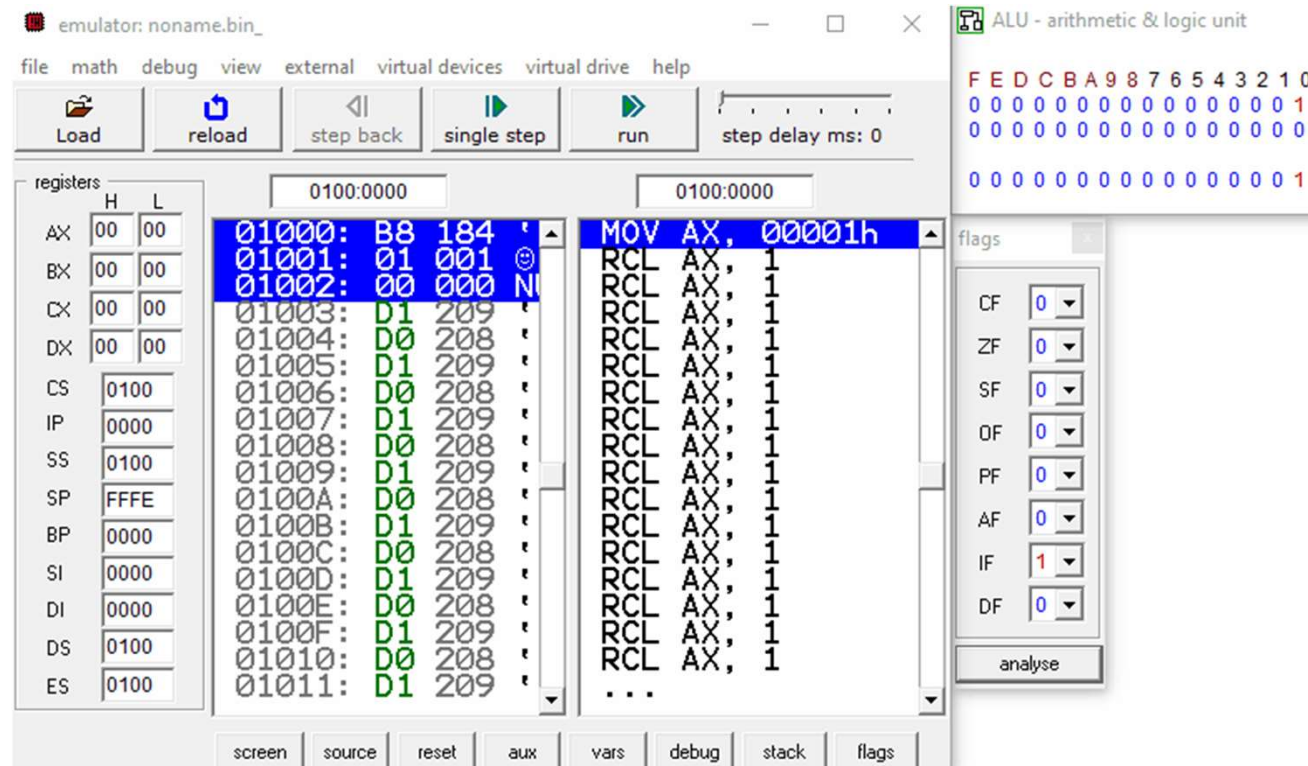
REG, immediate

memory, CL

register, CL

ผลการทำงานจะเก็บที่ operand1 และหากมีการเปลี่ยนเครื่องหมาย (signed bit) จะส่งผลให้ flag OF มีค่าเป็น 1 เมื่อวนถึงบิตสุดท้าย จะเลื่อนไปที่ CF แล้วจึงเลื่อนกลับมาที่บิตขวาสุด

```
MOV AX,1
RCL AX,40
HLT
```



การประมวลผลระดับบิต จะเก็บสถานะไว้ใน Flag ซึ่งค่า Flag สามารถนำไปใช้ในการควบคุมลำดับการทำงานของโปรแกรมได้

ในรูปแบบของ

IF
THEN
ELSE

ซึ่งเราจะศึกษาในบทถัดไป