



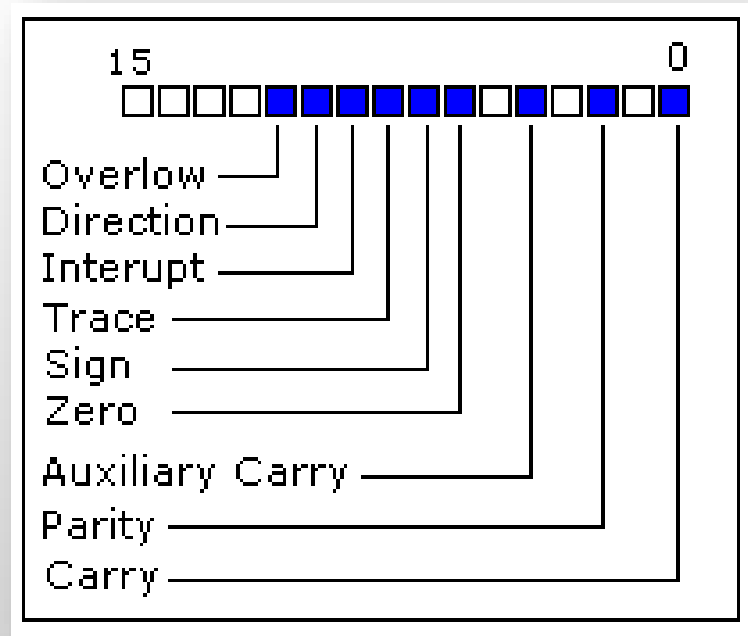
Bölüm 6: Aritmetik ve Mantık

Mikroişlemciler



Aritmetik ve Mantık Komutları

- Sonuçları işlemci durum bayraklarını etkiler.
- İşlemci Durum Bayrağı:
 - 16 bit bulunur, her biri bir bayrak, 1 ve 0 değerini alabilir.





Aritmetik ve Mantık Komutları

- İşlemci Durum Bayrağı:
 - 16 bit bulunur, her biri bir bayrak, 1 ve 0 değerini alabilir.
- Taşıma Bayrağı (*CF - Carry flag*):
 - İşaretsiz bir taşma olduğunda 1'e ayarlanır.
 - Örneğin, $255 + 1$, sonuç 0...255 aralığında değil.
 - Taşma olmadığında 0'a ayarlanır.
- Sıfır Bayrağı (*ZF - Zero flag*):
 - Sonuç sıfır ise 1'e ayarlanır.
 - Sıfır olmayan bir sonuç için 0'a ayarlanır.



Aritmetik ve Mantık Komutları

- İşaret Bayrağı (*SF - Sign flag*):
 - Sonuç negatif ise 1'e ayarlanır.
 - Sonuç pozitif olduğunda 0'a ayarlanır.
 - Aslında bu bayrak, en önemli bitin (*MSB*) değerini alır.
- Taşma Bayrağı (*OF - Overflow flag*):
 - İmzalı bir taşma olduğunda 1'e ayarlanır.
 - Örneğin, $100 + 50$ eklediğinizde, sonuç $-128...127$ aralığında değil.
- Çiftlik Bayrağı (*PF - Parity flag*):
 - Sonuçta tek sayıda bit varsa 1'e ayarlanır,
 - Çift sayıda bit varsa 0'a ayarlanır.
 - Sonuç bir kelime ise yalnızca düşük (*low*) 8 bite bakılır.



Aritmetik ve Mantık Komutları

- Yardımcı Bayrağı (*AF - Auxiliary flag*):
 - Düşük nibble (4 bit) için işaretless bir taşma olduğunda 1'e ayarlanır.
- Kesme Etkin Bayrağı (*IF - Interrupt enable flag*):
 - 1'e ayarlandığında CPU, harici aygıtlardan gelen kesmelere tepki verir.
- Yön Bayrağı (*DF - Direction flag*):
 - Bazı komutlar tarafından veri zincirlerini işlemek için kullanılır;
 - 0'a ayarlandığında işlem ileri doğru yapılır,
 - 1'e ayarlandığında işlem geriye doğru yapılır.



Komutlar Üç Gruba Ayrılır

- Birinci Grup:
 - Artırma ve Azaltma
- İkinci Grup:
 - Çarpma ve Bölme
- Üçüncü Grup:
 - Tek Değişkenli İşlemler



Birinci Grup: Artırma ve Azaltma

- ADD (Toplama): İki değeri toplar ve sonucu hedefe yazar.
- SUB (Çıkarma): Bir değeri diğerinden çıkarır ve sonucu hedefe yazar.
- CMP (Karşılaştırma): İki değeri karşılaştırır, ancak sonucu saklamaz.
- AND (VE): İki değeri mantıksal olarak AND işlemine tabi tutar.
- TEST (Sınama): İki değeri bit seviyesinde sınar.
- OR (VEYA): İki değeri mantıksal olarak OR işlemine tabi tutar.
- XOR (Dışlayıcı VEYA): İki değeri mantıksal olarak XOR işlemine tabi tutar.



İkinci Grup: Çarpma ve Bölme

- MUL (Çarpma): İki değeri çarpar.
- IMUL (İşaretli Çarpma): İki işaretli değeri çarpar.
- DIV (Bölme): Bir değeri diğerine böler ve sonucu hedefe yazar.
- IDIV (İşaretli Bölme): İki işaretli değeri böler ve sonucu hedefe yazar.



Üçüncü Grup: Tek Değişkenli İşlemler

- INC (Artırma): Bir değeri bir artırır.
- DEC (Azaltma): Bir değeri bir azaltır.
- NOT (Mantıksal NOT): Bir değer bitlerini tersine çevirir.
- NEG (Negatif Alma): Bir değeri negatif hale getirir.



Birinci Grup: ADD, SUB, CMP, AND, TEST, OR, XOR

- İşlenenler:
 - REG, memory
 - memory, REG
 - REG, REG
 - memory, immediate
 - REG, immediate
- REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.
- memory: [BX], [BX+SI+7], variable, gibi..
- immediate: 5, -24, 3Fh, 10001101b, gibi..



Birinci Grup: ADD, SUB, CMP, AND, TEST, OR, XOR

- İki işlenenli işlemler sonrasında, sonuç her zaman ilk işlenende saklanır.
- CMP ve TEST komutları sadece bayrakları etkiler ve bir sonuç saklamaz.
- Etkilenen Bayraklar:
 - CF, ZF, SF, OF, PF, AF.
- ADD (Toplama)/SUB (Çıkarma): İkinci işleneni birinci işlenene ekler/çıkartır.
- CMP (Karşılaştırma): İkinci işleneni birinciden çıkarır, sonucu saklamaz.
 - sadece bayrakları etkiler.
- AND (VE)/OR (VEYA): İki işlenenin bitleri arasında VE/VEYA işlemi yapar.
- TEST (Test): AND ile aynıdır, sadece bayrakları etkiler.
- XOR (Dışlayan VEYA): İki işlenenin tüm bitleri arasında XOR işlemi yapar.



Mantıksal İşlem Kuralları

- AND:
 - $1 \text{ AND } 1 = 1$, $1 \text{ AND } 0 = 0$,
 - $0 \text{ AND } 1 = 0$, $0 \text{ AND } 0 = 0$.
- OR:
 - $1 \text{ OR } 1 = 1$, $1 \text{ OR } 0 = 1$,
 - $0 \text{ OR } 1 = 1$, $0 \text{ OR } 0 = 0$.
- XOR:
 - $1 \text{ XOR } 1 = 0$, $1 \text{ XOR } 0 = 1$,
 - $0 \text{ XOR } 1 = 1$, $0 \text{ XOR } 0 = 0$.



İkinci Grup: MUL, IMUL, DIV, IDIV

- İşlenenler:
 - REG
 - memory
- REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.
- memory: [BX], [BX+SI+7], variable, gibi..



İkinci Grup: MUL, IMUL, DIV, IDIV

- MUL ve IMUL komutları yalnızca CF ve OF bayraklarını etkiler.
- Sonuç, işlenen boyutunu aştığında bu bayraklar 1'e ayarlanır.
- DIV ve IDIV komutlarında bayraklar tanımsızdır.



MUL, IMUL

- MUL (İşaretsiz Çarpma):
 - İşlenen bir byte ise: $AX = AL * \text{işlenen}$.
 - İşlenen bir kelime ise: $(DX\ AX) = AX * \text{işlenen}$.
- IMUL (İşaretli Çarpma):
 - İşlenen bir byte ise: $AX = AL * \text{işlenen}$.
 - İşlenen bir kelime ise: $(DX\ AX) = AX * \text{işlenen}$.



DIV, IDIV

- DIV (İşaretsiz Bölme):
 - İşlenen bir byte ise: $AL = AX / \text{işlenen}$, $AH = \text{kalan (modulus)}$.
 - İşlenen bir kelime ise: $AX = (DX \text{ } AX) / \text{işlenen}$, $DX = \text{kalan (modulus)}$.
- IDIV (İşaretili Bölme):
 - İşlenen bir byte ise: $AL = AX / \text{işlenen}$, $AH = \text{kalan (modulus)}$.
 - İşlenen bir kelime ise: $AX = (DX \text{ } AX) / \text{işlenen}$, $DX = \text{kalan (modulus)}$.



Üçüncü Grup: INC, DEC, NOT, NEG

- İşlenenler:
 - REG
 - memory
- REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.
- memory: [BX], [BX+SI+7], variable, gibi..



INC, DEC, NOT, NEG

- INC ve DEC komutları yalnızca ZF, SF, OF, PF, AF bayraklarını etkiler.
- NOT komutu hiçbir bayrağı etkilemez!
 - İşlenenin her bir bitini ters çevirir.
- NEG komutu yalnızca CF, ZF, SF, OF, PF, AF bayraklarını etkiler.
 - İşleneni negatif yapar (ikili tümlleme).
 - Her bir bitini ters çevirir ve ardından 1 ekler.
 - Örneğin, 5 -5'e ve -2 2'ye dönüşecektir.



Dizi Elemanları Toplama

```
org 100h
jmp start

    vec1 db 1, 2, 5, 6
    vec2 db 3, 5, 6, 1
    vec3 db ?, ?, ?, ?

start:
    lea si, vec1
    lea bx, vec2
    lea di, vec3
    mov cx, 4
```



Dizi Elemanları Toplama

sum:

```
mov al, [si]
add al, [bx]
mov [di], al
inc si
inc bx
inc di
loop sum
```

ret



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	26
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100

07100:	EB	235	6
07101:	0C	012	♀
07102:	01	001	⊙
07103:	02	002	⊙
07104:	05	005	♣
07105:	06	006	♣
07106:	03	003	♥
07107:	05	005	♣
07108:	06	006	♣
07109:	01	001	⊙
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	00	000	NULL
0710D:	00	000	NULL
0710E:	BE	190	♠
0710F:	02	002	⊙
07110:	01	001	⊙
07111:	BB	187	♠
07112:	06	006	♣
07113:	01	001	⊙
07114:	BF	191	♠
07115:	0A	010	NEWL

0700:0100

```
JMP 010Eh
ADD [BP + SI], AX
ADD AX, 00306h
ADD AX, 00106h
ADD [BX + SI], AL
ADD [BX + SI], AL
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	26
DX	00	00
CS	0700	
IP	010E	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010E

07100:	EB	235	6
07101:	0C	012	♀
07102:	01	001	⊙
07103:	02	002	⊙
07104:	05	005	♣
07105:	06	006	♣
07106:	03	003	♥
07107:	05	005	♣
07108:	06	006	♣
07109:	01	001	⊙
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	00	000	NULL
0710D:	00	000	NULL
0710E:	BE	190	↓
0710F:	02	002	⊙
07110:	01	001	⊙
07111:	BB	187	♠
07112:	06	006	♣
07113:	01	001	⊙
07114:	BF	191	♠
07115:	0A	010	NEWL

0700:010E

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	26
DX	00	00
CS	0700	
IP	0111	
SS	0700	
SP	FFFE	
BP	0000	
SI	0102	
DI	0000	
DS	0700	
ES	0700	

0700:0111

07100:	EB	235	6
07101:	0C	012	♀
07102:	01	001	⊙
07103:	02	002	⊙
07104:	05	005	♣
07105:	06	006	♣
07106:	03	003	♥
07107:	05	005	♣
07108:	06	006	♣
07109:	01	001	⊙
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	00	000	NULL
0710D:	00	000	NULL
0710E:	BE	190	↓
0710F:	02	002	⊙
07110:	01	001	⊙
07111:	BB	187	↑
07112:	06	006	♣
07113:	01	001	⊙
07114:	BF	191	↑
07115:	0A	010	NEWL

0700:0111

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	01	06
CX	00	26
DX	00	00
CS	0700	
IP	0114	
SS	0700	
SP	FFFE	
BP	0000	
SI	0102	
DI	0000	
DS	0700	
ES	0700	

0700:0114

07100:	EB	235	6
07101:	0C	012	♀
07102:	01	001	⊙
07103:	02	002	⊙
07104:	05	005	♣
07105:	06	006	♣
07106:	03	003	♥
07107:	05	005	♣
07108:	06	006	♣
07109:	01	001	⊙
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	00	000	NULL
0710D:	00	000	NULL
0710E:	BE	190	↓
0710F:	02	002	⊙
07110:	01	001	⊙
07111:	BB	187	7
07112:	06	006	♣
07113:	01	001	⊙
07114:	BF	191	7
07115:	0A	010	NEWL

0700:0114

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	01	06
CX	00	04
DX	00	00
CS	0700	
IP	011A	
SS	0700	
SP	FFFE	
BP	0000	
SI	0102	
DI	010A	
DS	0700	
ES	0700	

0700:011A

07117:	B9	185	!
07118:	04	004	♦
07119:	00	000	NULL
0711A:	8A	138	è
0711B:	04	004	♦
0711C:	02	002	0
0711D:	07	007	BEEP
0711E:	88	136	è
0711F:	05	005	♠
07120:	46	070	F
07121:	43	067	C
07122:	47	071	G
07123:	E2	226	Γ
07124:	F5	245	J
07125:	C3	195	†
07126:	90	144	É
07127:	90	144	É
07128:	90	144	É
07129:	90	144	É
0712A:	90	144	É
0712B:	90	144	É
0712C:	90	144	É

0700:011A

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	01
BX	01	06
CX	00	04
DX	00	00
CS	0700	
IP	011C	
SS	0700	
SP	FFFE	
BP	0000	
SI	0102	
DI	010A	
DS	0700	
ES	0700	

0700:011C

```
07117: B9 185 H
07118: 04 004 
07119: 00 000 NULL
0711A: 8A 138 e
0711B: 04 004 
0711C: 02 002 
0711D: 07 007 BEEP
0711E: 88 136 e
0711F: 05 005 
07120: 46 070 F
07121: 43 067 C
07122: 47 071 G
07123: E2 226 r
07124: F5 245 J
07125: C3 195 t
07126: 90 144 e
07127: 90 144 e
07128: 90 144 e
07129: 90 144 e
0712A: 90 144 e
0712B: 90 144 e
0712C: 90 144 e
```

0700:011C

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	04
BX	01	06
CX	00	04
DX	00	00
CS	0700	
IP	011E	
SS	0700	
SP	FFFE	
BP	0000	
SI	0102	
DI	010A	
DS	0700	
ES	0700	

0700:011E

```
07117: B9 185 H
07118: 04 004 
07119: 00 000 NULL
0711A: 8A 138 e
0711B: 04 004 
0711C: 02 002 
0711D: 07 007 BEEP
0711E: 88 136 e
0711F: 05 005 
07120: 46 070 F
07121: 43 067 C
07122: 47 071 G
07123: E2 226 r
07124: F5 245 J
07125: C3 195 t
07126: 90 144 e
07127: 90 144 e
07128: 90 144 e
07129: 90 144 e
0712A: 90 144 e
0712B: 90 144 e
0712C: 90 144 e
```

0700:011E

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	04
BX	01	07
CX	00	04
DX	00	00
CS	0700	
IP	0122	
SS	0700	
SP	FFFE	
BP	0000	
SI	0103	
DI	010A	
DS	0700	
ES	0700	

0700:0122

```
07117: B9 185 H
07118: 04 004 
07119: 00 000 NULL
0711A: 8A 138 è
0711B: 04 004 
0711C: 02 002 
0711D: 07 007 BEEP
0711E: 88 136 ê
0711F: 05 005 
07120: 46 070 F
07121: 43 067 C
07122: 47 071 G
07123: E2 226 r
07124: F5 245 J
07125: C3 195 t
07126: 90 144 é
07127: 90 144 é
07128: 90 144 é
07129: 90 144 é
0712A: 90 144 é
0712B: 90 144 é
0712C: 90 144 é
```

0700:0122

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	04
BX	01	07
CX	00	04
DX	00	00
CS	0700	
IP	0123	
SS	0700	
SP	FFFE	
BP	0000	
SI	0103	
DI	010B	
DS	0700	
ES	0700	

0700:0123

07117:	B9	185	!
07118:	04	004	♦
07119:	00	000	NULL
0711A:	8A	138	è
0711B:	04	004	♦
0711C:	02	002	⊙
0711D:	07	007	BEEP
0711E:	88	136	ê
0711F:	05	005	♠
07120:	46	070	F
07121:	43	067	C
07122:	47	071	G
07123:	E2	226	Γ
07124:	F5	245	J
07125:	C3	195	†
07126:	90	144	É
07127:	90	144	É
07128:	90	144	É
07129:	90	144	É
0712A:	90	144	É
0712B:	90	144	É
0712C:	90	144	É

0700:0123

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	04
BX	01	07
CX	00	03
DX	00	00
CS	0700	
IP	011A	
SS	0700	
SP	FFFE	
BP	0000	
SI	0103	
DI	010B	
DS	0700	
ES	0700	

0700:011A

07117:	B9	185	!
07118:	04	004	♦
07119:	00	000	NULL
0711A:	8A	138	è
0711B:	04	004	♦
0711C:	02	002	0
0711D:	07	007	BEEP
0711E:	88	136	è
0711F:	05	005	♠
07120:	46	070	F
07121:	43	067	C
07122:	47	071	G
07123:	E2	226	Γ
07124:	F5	245	J
07125:	C3	195	†
07126:	90	144	É
07127:	90	144	É
07128:	90	144	É
07129:	90	144	É
0712A:	90	144	É
0712B:	90	144	É
0712C:	90	144	É

0700:011A

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	07
BX	01	0A
CX	00	01
DX	00	00
CS	0700	
IP	0122	
SS	0700	
SP	FFFE	
BP	0000	
SI	0106	
DI	010D	
DS	0700	
ES	0700	

0700:0122

```
07117: B9 185 H
07118: 04 004 
07119: 00 000 NULL
0711A: 8A 138 è
0711B: 04 004 
0711C: 02 002 
0711D: 07 007 BEEP
0711E: 88 136 ê
0711F: 05 005 
07120: 46 070 F
07121: 43 067 C
07122: 47 071 G
07123: E2 226 r
07124: F5 245 J
07125: C3 195 t
07126: 90 144 é
07127: 90 144 é
07128: 90 144 é
07129: 90 144 é
0712A: 90 144 é
0712B: 90 144 é
0712C: 90 144 é
```

0700:0122

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	07
BX	01	0A
CX	00	01
DX	00	00
CS	0700	
IP	0123	
SS	0700	
SP	FFFE	
BP	0000	
SI	0106	
DI	010E	
DS	0700	
ES	0700	

0700:0123

07117:	B9	185	!
07118:	04	004	♦
07119:	00	000	NULL
0711A:	8A	138	è
0711B:	04	004	♦
0711C:	02	002	0
0711D:	07	007	BEEP
0711E:	88	136	ê
0711F:	05	005	♠
07120:	46	070	F
07121:	43	067	C
07122:	47	071	G
07123:	E2	226	Γ
07124:	F5	245	J
07125:	C3	195	†
07126:	90	144	É
07127:	90	144	É
07128:	90	144	É
07129:	90	144	É
0712A:	90	144	É
0712B:	90	144	É
0712C:	90	144	É

0700:0123

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Dizi Elemanları Toplama

emulator: add-2.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	07
BX	01	0A
CX	00	00
DX	00	00
CS	0700	
IP	0125	
SS	0700	
SP	FFFE	
BP	0000	
SI	0106	
DI	010E	
DS	0700	
ES	0700	

0700:0125

```
07117: B9 185 H
07118: 04 004 
07119: 00 000 NULL
0711A: 8A 138 è
0711B: 04 004 
0711C: 02 002 
0711D: 07 007 BEEP
0711E: 88 136 ê
0711F: 05 005 
07120: 46 070 F
07121: 43 067 C
07122: 47 071 G
07123: E2 226 r
07124: F5 245 J
07125: C3 195 |
07126: 90 144 é
07127: 90 144 é
07128: 90 144 é
07129: 90 144 é
0712A: 90 144 é
0712B: 90 144 é
0712C: 90 144 é
```

0700:0125

```
MOV SI, 00102h
MOV BX, 00106h
MOV DI, 0010Ah
MOV CX, 00004h
MOV AL, [SI]
ADD AL, [BX]
MOV [DI], AL
INC SI
INC BX
INC DI
LOOP 011Ah
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





Binary Coded Decimal

; first number '9':

```
mov    ah, 09h
```

; second number '5':

```
mov    al, 05h
```

; $al = al + ah = 09h + 05h = 0eh$

```
add    al, ah
```

; clear tens byte of bcd

```
xor    ah, ah
```

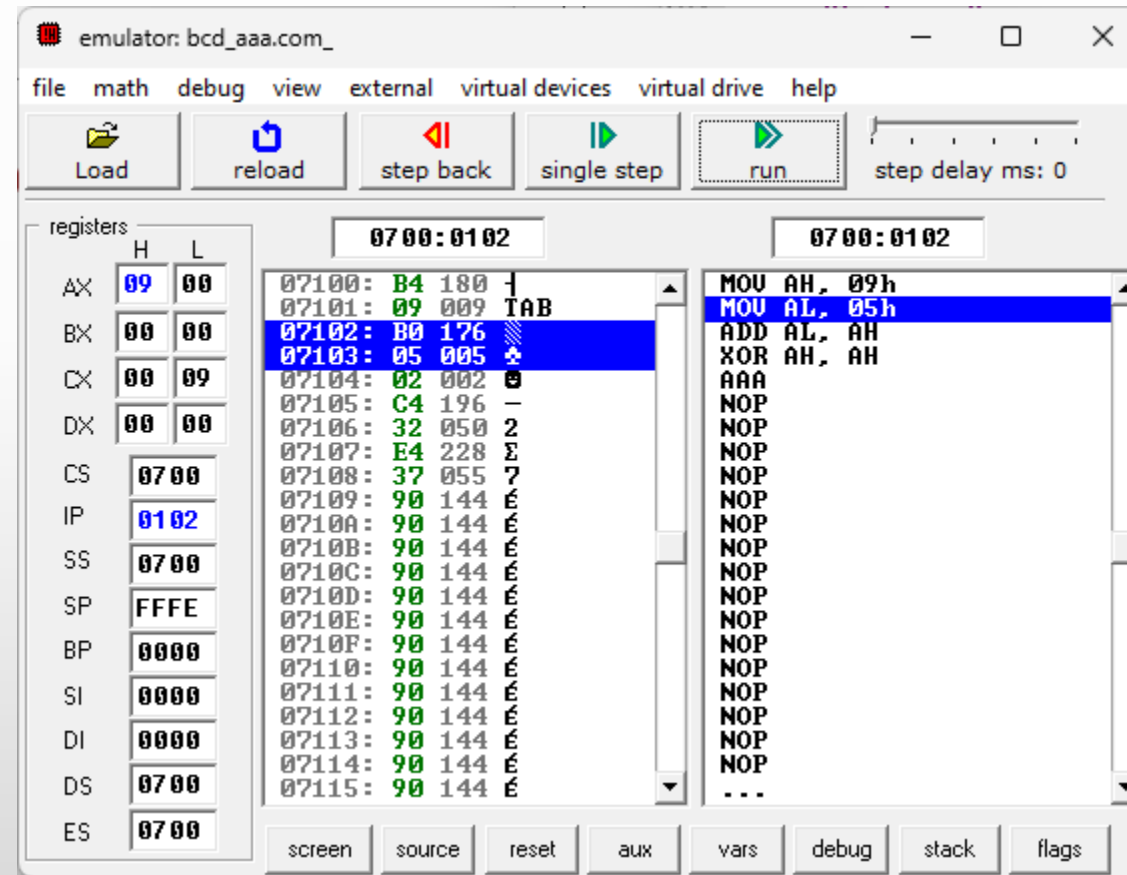
; adjust result to bcd form, ; $ah = 1, al = 4 \rightarrow '14'$

```
aaa
```



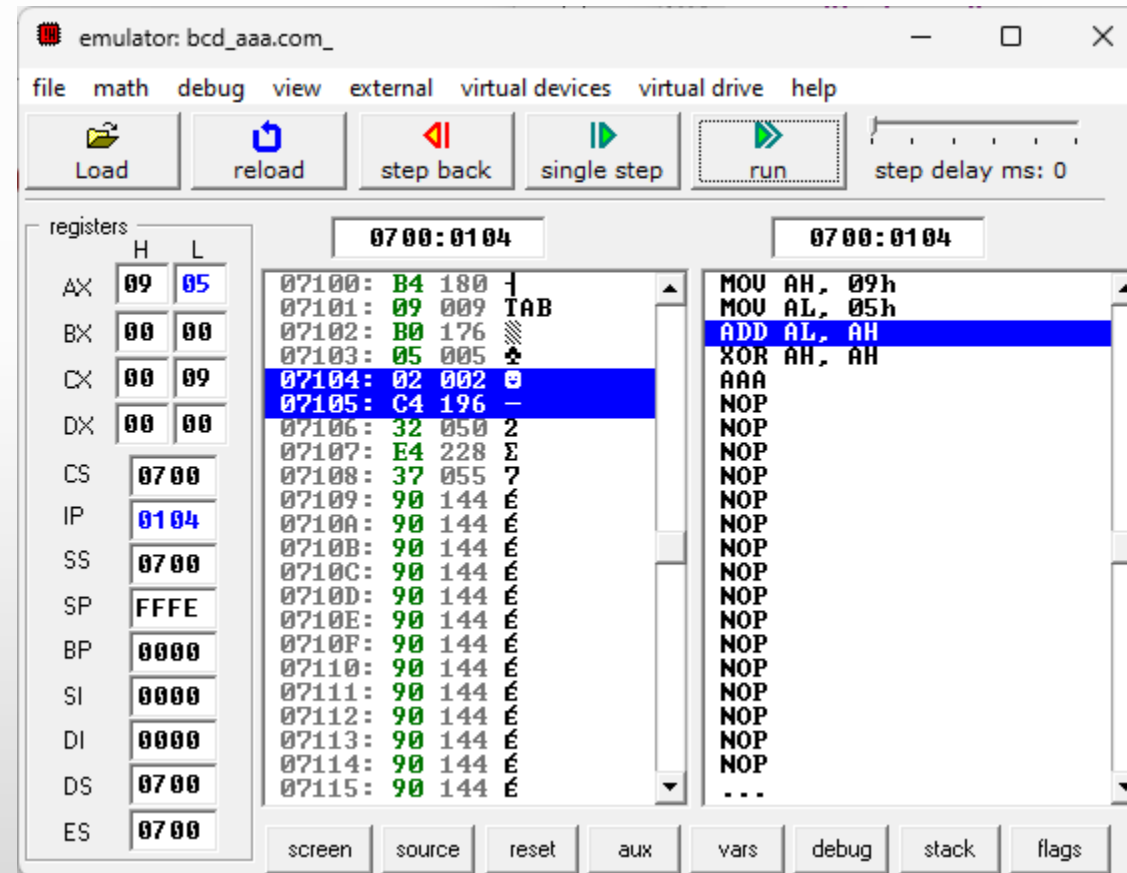


Binary Coded Decimal





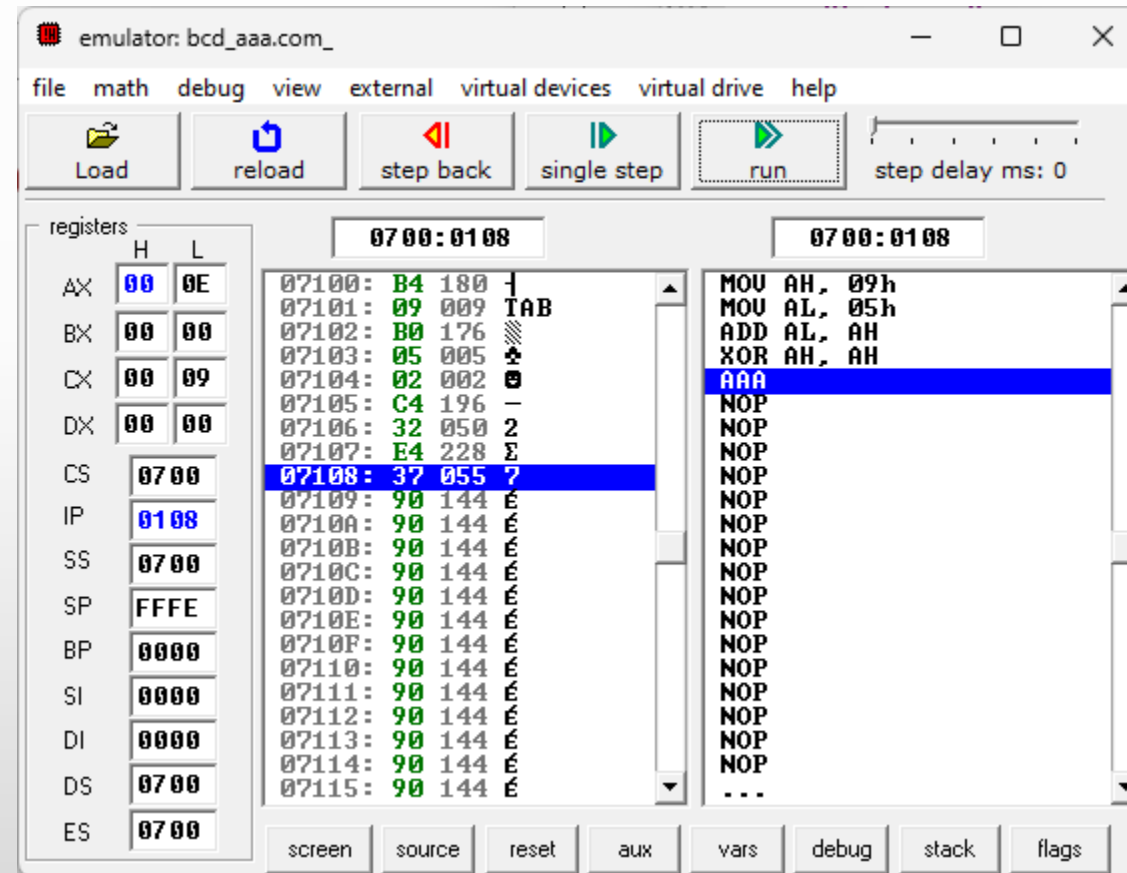
Binary Coded Decimal







Binary Coded Decimal







Binary Coded Decimal - Negatif Sayı

; make 5 - 8, al = 0fdh (not in binary coded decimal form)

```
mov  al, 05h
```

```
mov  bl, 08h
```

```
sub  al, bl
```

; convert to binary coded decimal, al = 7

; and 1 is borrowed from ah, like calculating 15 - 8:

```
aas
```



Binary Coded Decimal - Negatif Sayı

emulator: bcd_aas.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	12
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100

07100:	B0	176	
07101:	05	005	
07102:	B3	179	
07103:	08	008	BACK
07104:	2A	042	*
07105:	C3	195	†
07106:	3F	063	?
07107:	0C	012	♀
07108:	30	048	0
07109:	B4	180	†
0710A:	0E	014	†
0710B:	CD	205	=
0710C:	10	016	†
0710D:	B4	180	†
0710E:	00	000	NULL
0710F:	CD	205	=
07110:	16	022	=
07111:	C3	195	†
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

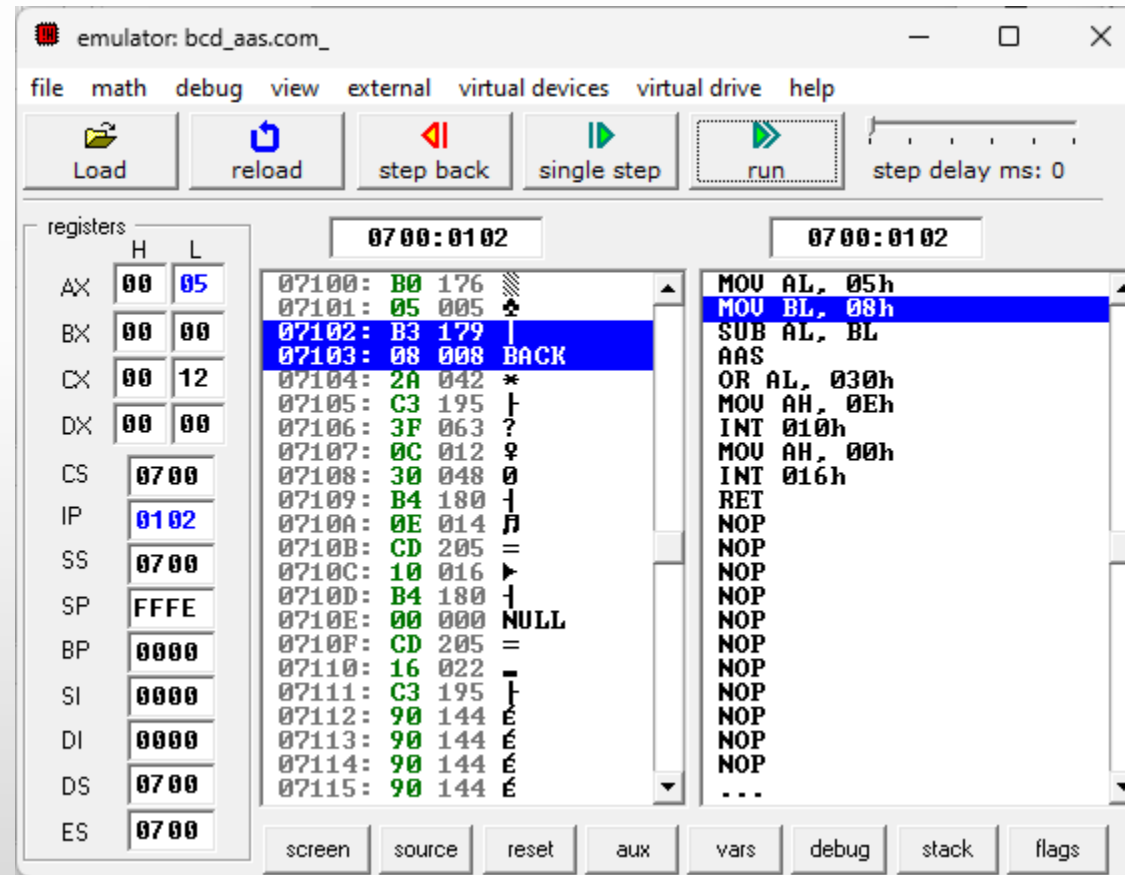
0700:0100

```
MOV AL, 05h
MOV BL, 08h
SUB AL, BL
AAS
OR AL, 030h
MOV AH, 0Eh
INT 010h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags

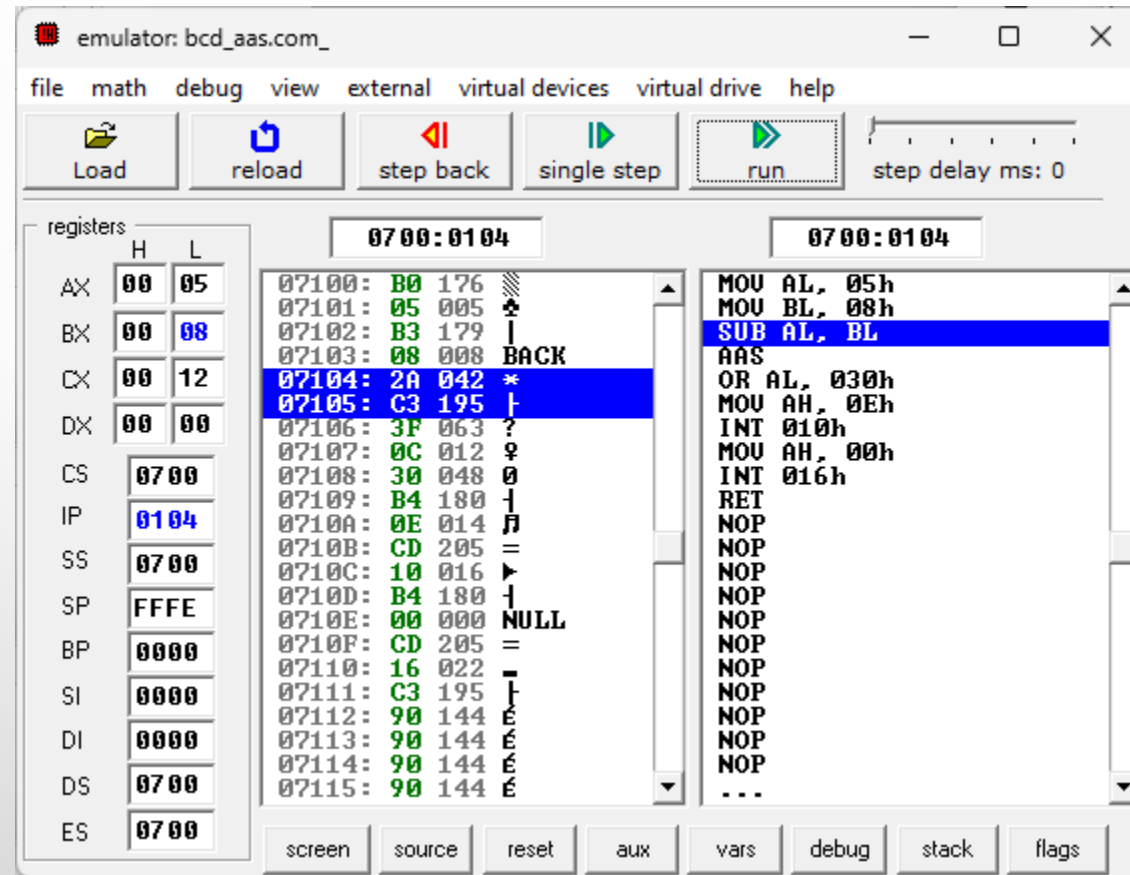


Binary Coded Decimal - Negatif Sayı



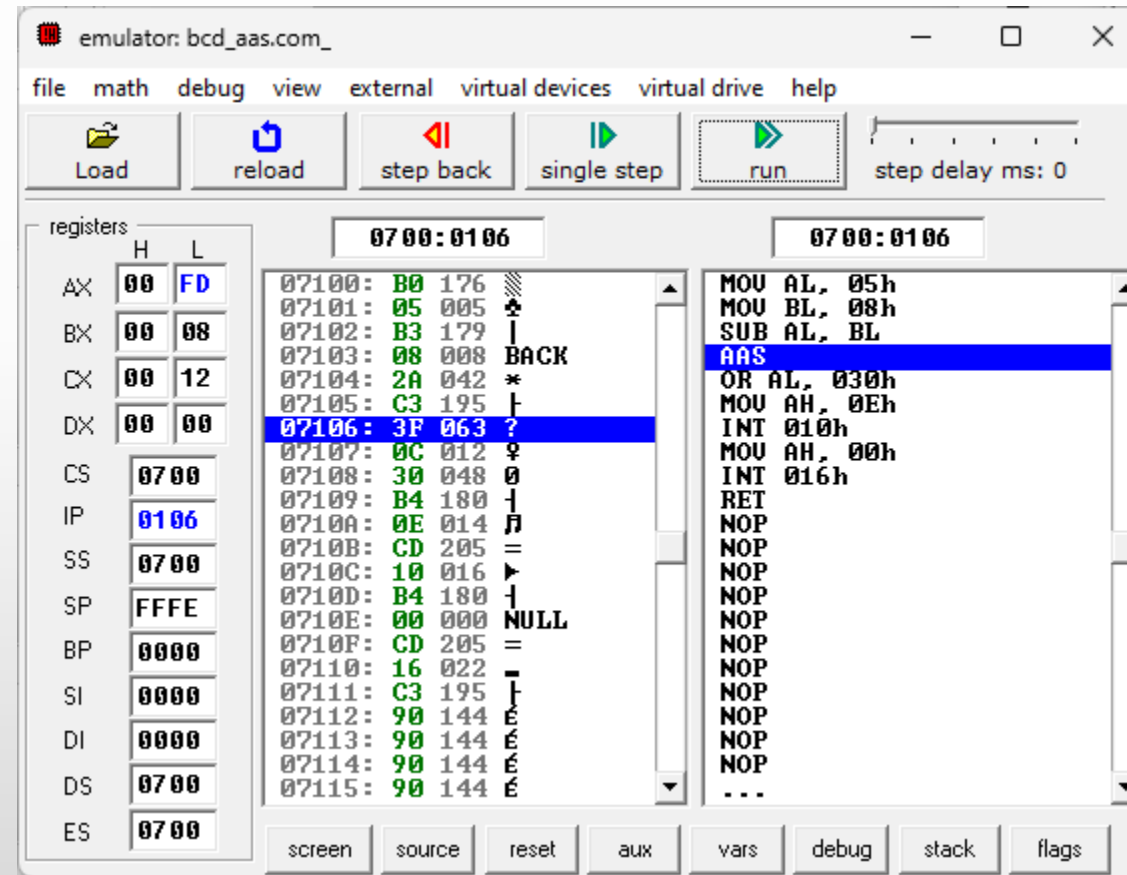


Binary Coded Decimal - Negatif Sayı



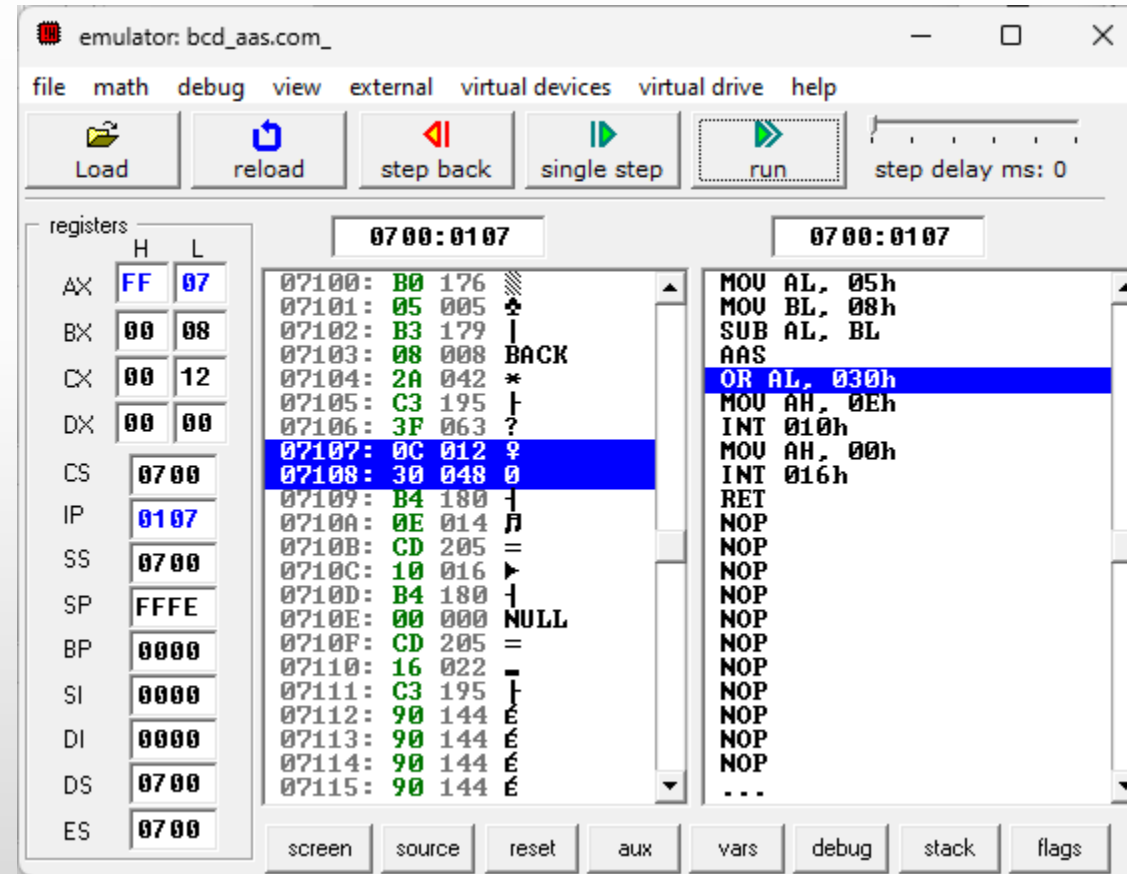


Binary Coded Decimal - Negatif Sayı





Binary Coded Decimal - Negatif Sayı





Santigrad Fahrenayt Dönüşümü

```
org 100h
```

```
start:
```

```
;celsius to fahrenheit according to  $f = c * 9 / 5 + 32$ 
```

```
    mov cl, 15
```

```
    mov al, 9
```

```
    imul cl
```

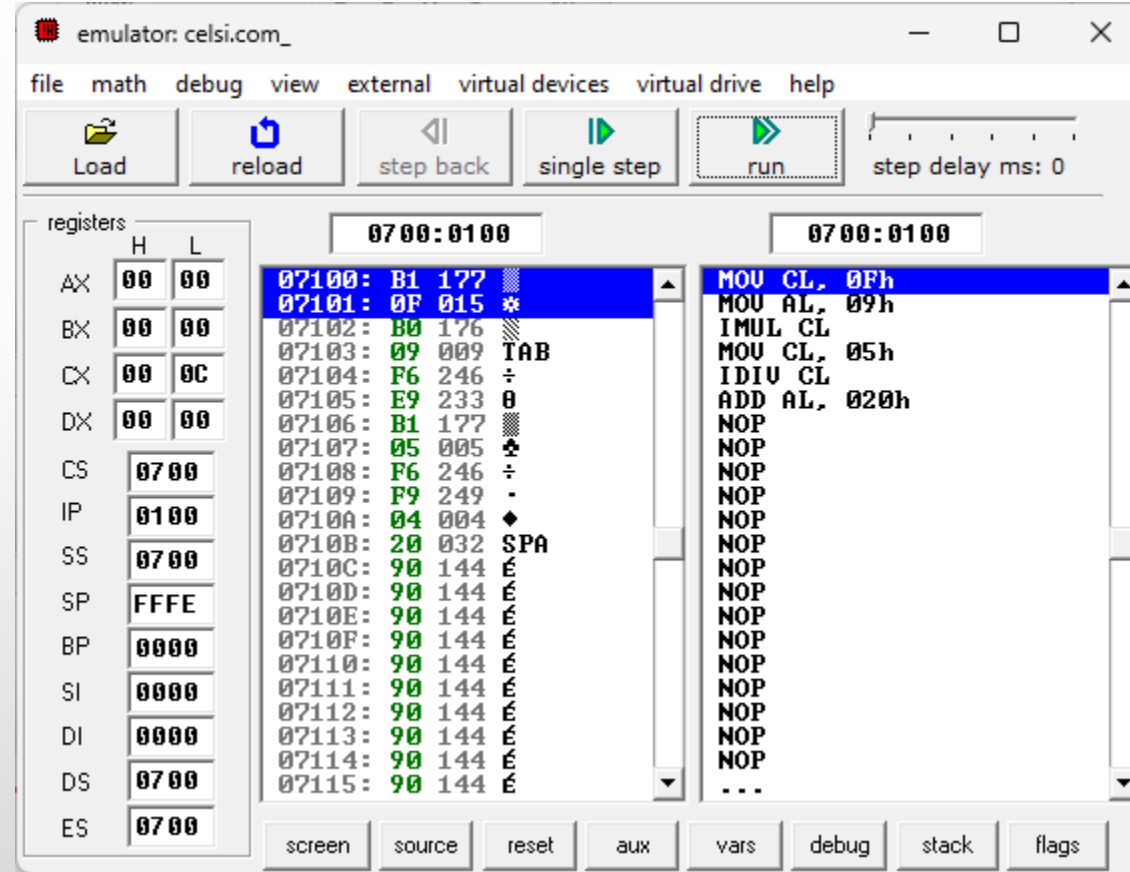
```
    mov cl, 5
```

```
    idiv cl
```

```
    add al, 32
```

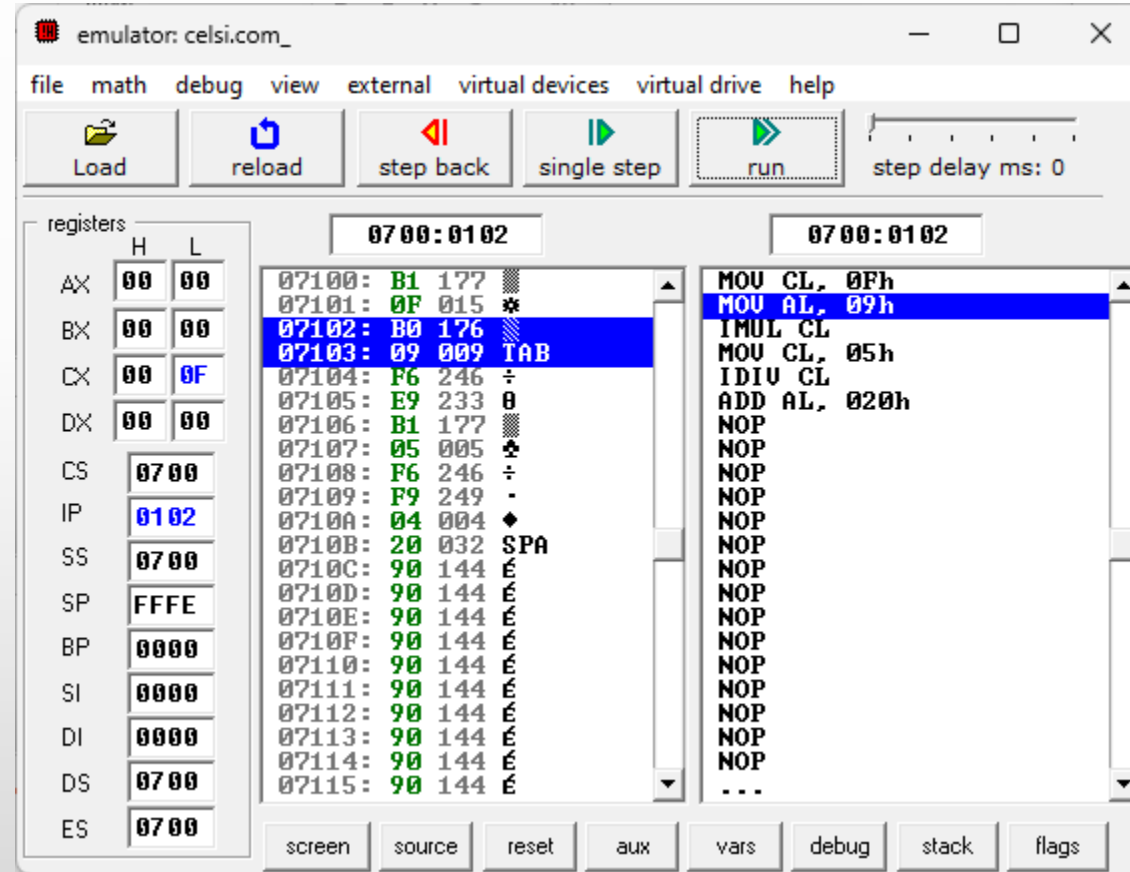



Santigrad Fahrenheit Dönüşümü



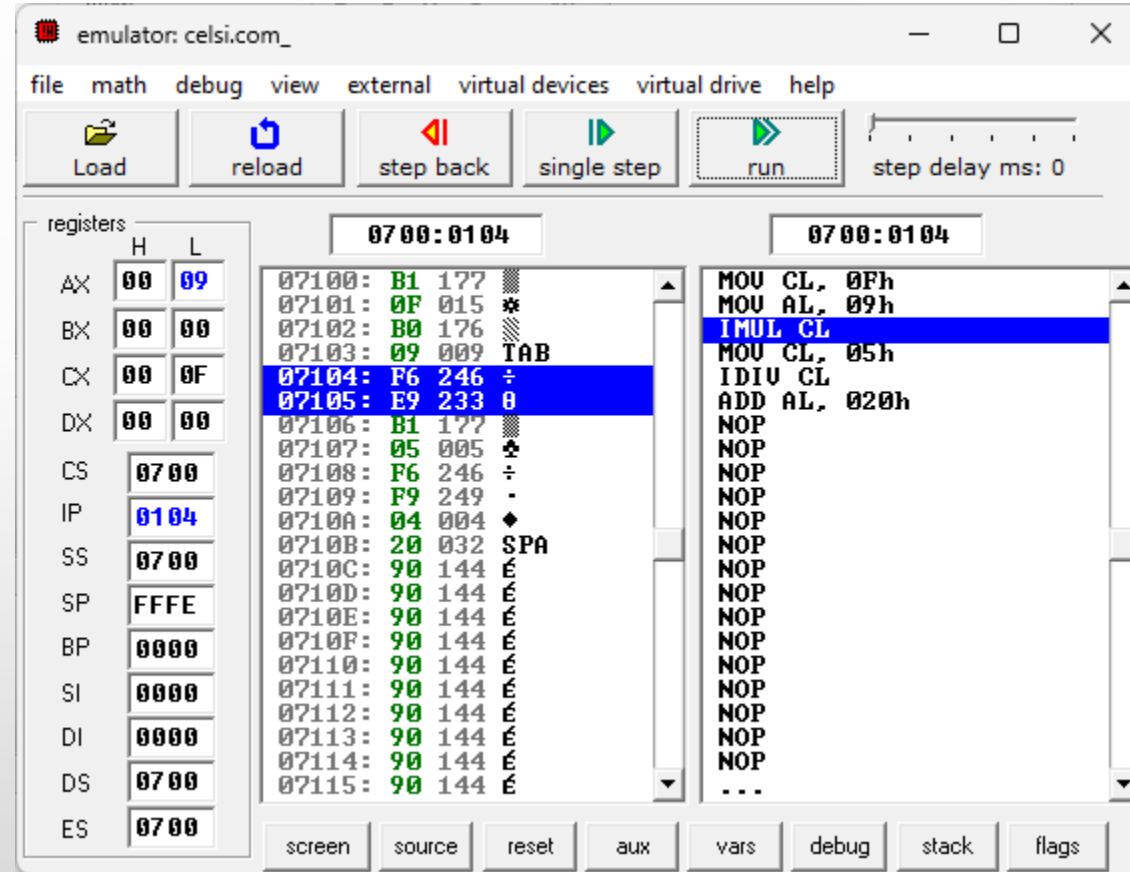


Santigrad Fahrenayt Dönüşümü



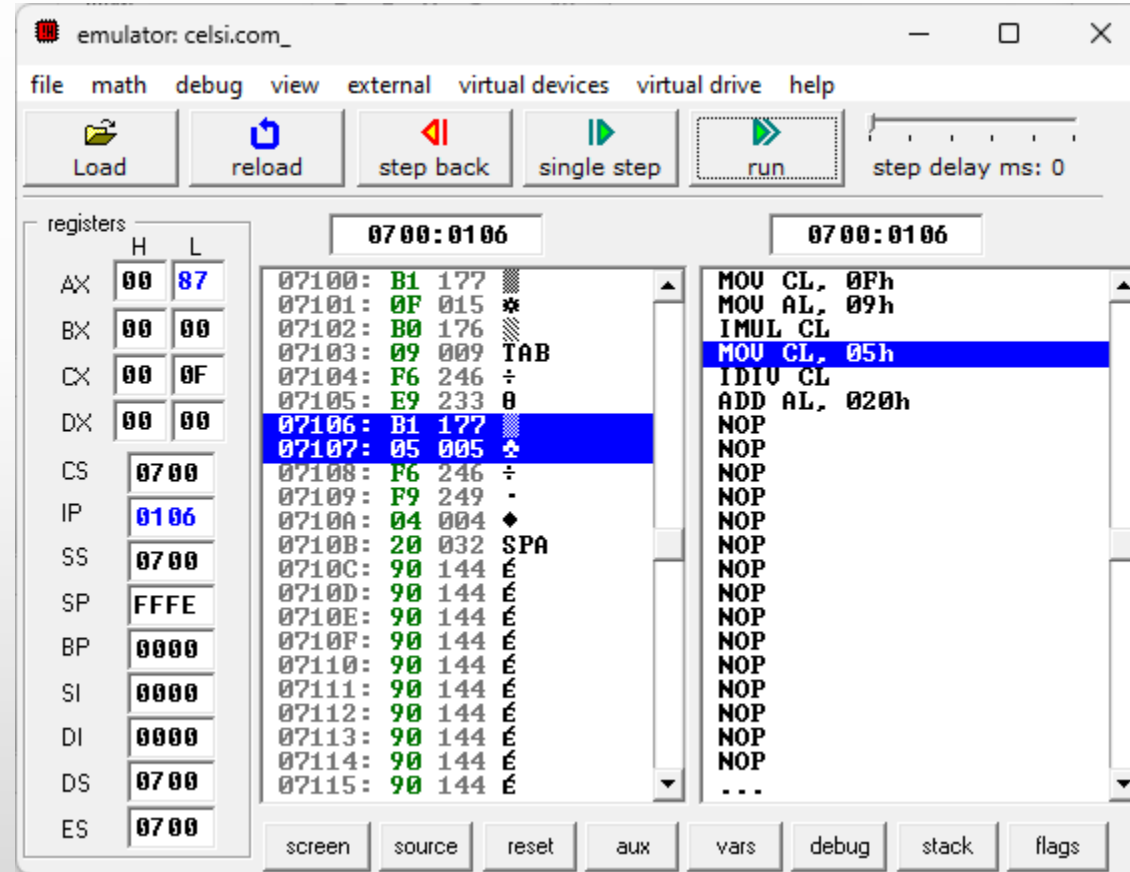


Santigrad Fahrenayt Dönüşümü





Santigrad Fahrenayt Dönüşümü





Santigrad Fahrenayt Dönüşümü

emulator: celsi.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	87
BX	00	00
CX	00	05
DX	00	00
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

07100:	B1	177	
07101:	0F	015	*
07102:	B0	176	
07103:	09	009	TAB
07104:	F6	246	÷
07105:	E9	233	0
07106:	B1	177	
07107:	05	005	+
07108:	F6	246	÷
07109:	F9	249	-
0710A:	04	004	♦
0710B:	20	032	SPA
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

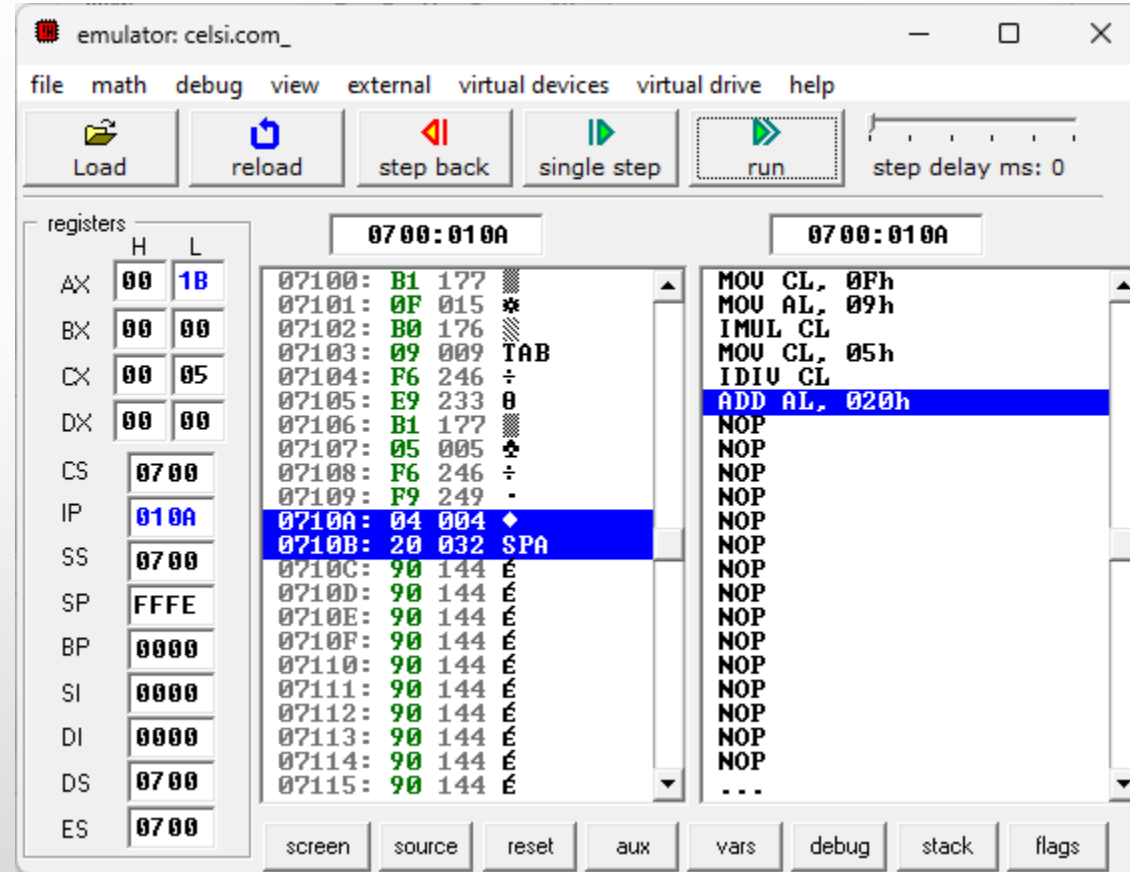
07 00: 01 08

```
MOV CL, 0Fh
MOV AL, 09h
IMUL CL
MOV CL, 05h
IDIV CL
ADD AL, 020h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags

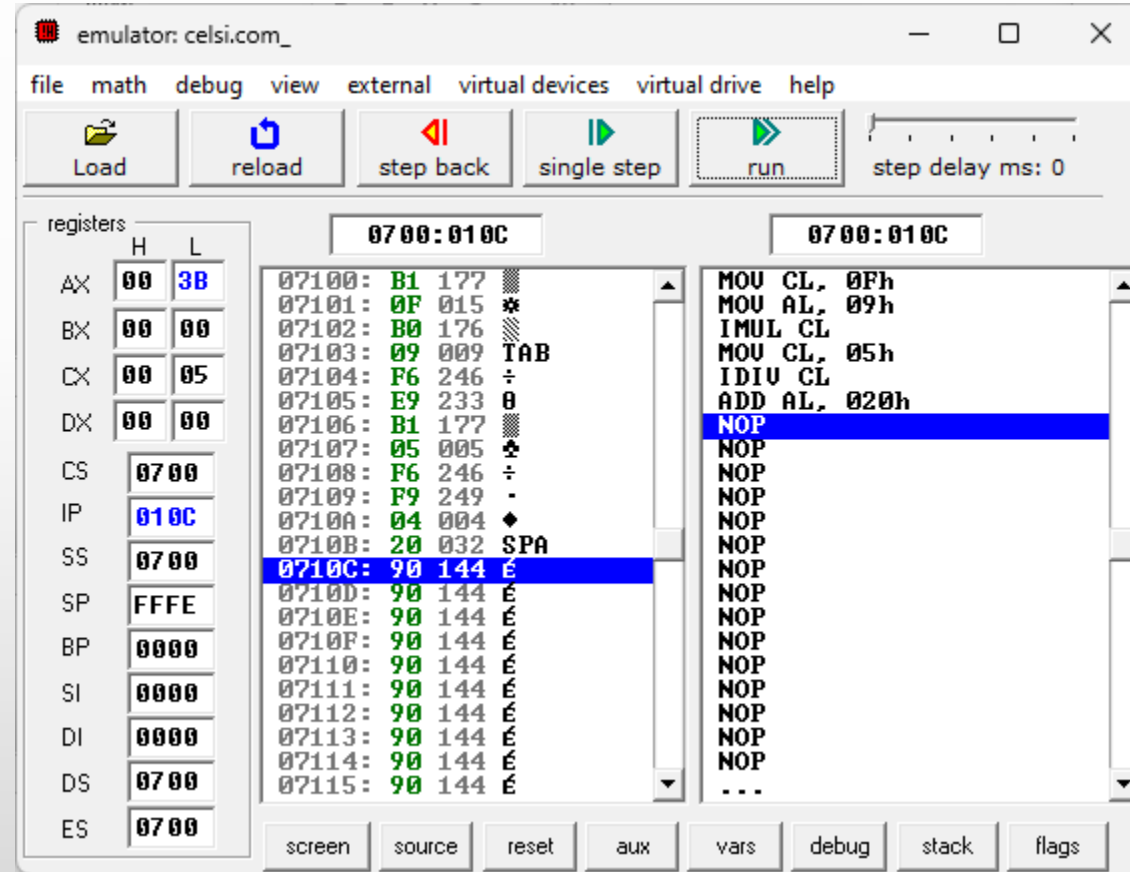


Santigrad Fahrenheit Dönüşümü





Santigrad Fahrenayt Dönüşümü





Fahrenayt Santigrad Dönüşümü

```
org 100h
```

```
start:
```

```
;fahrenheit to celsius according to:  $c = (f - 32) * 5 / 9$ 
```

```
    mov cl, 59
```

```
    sub cl, 32
```

```
    mov al, 5
```

```
    imul cl
```

```
    mov cl, 9
```

```
    idiv cl
```

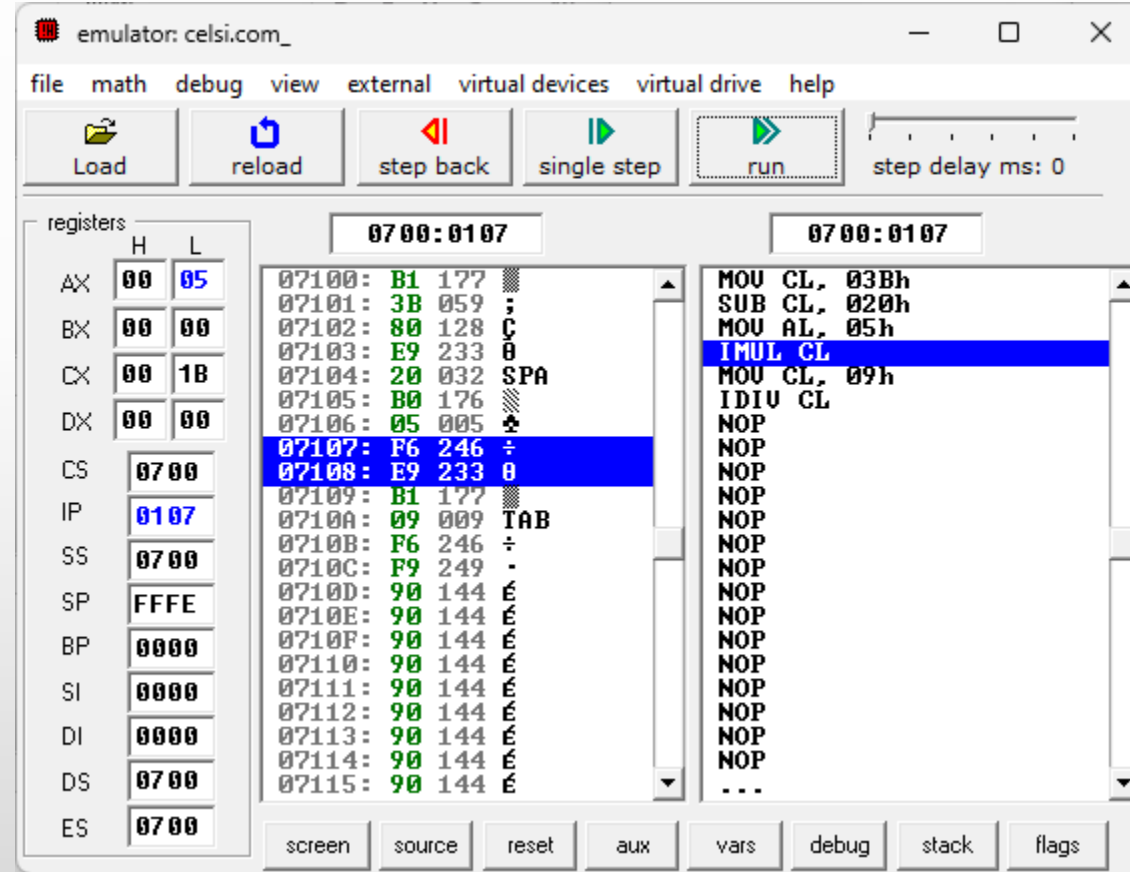






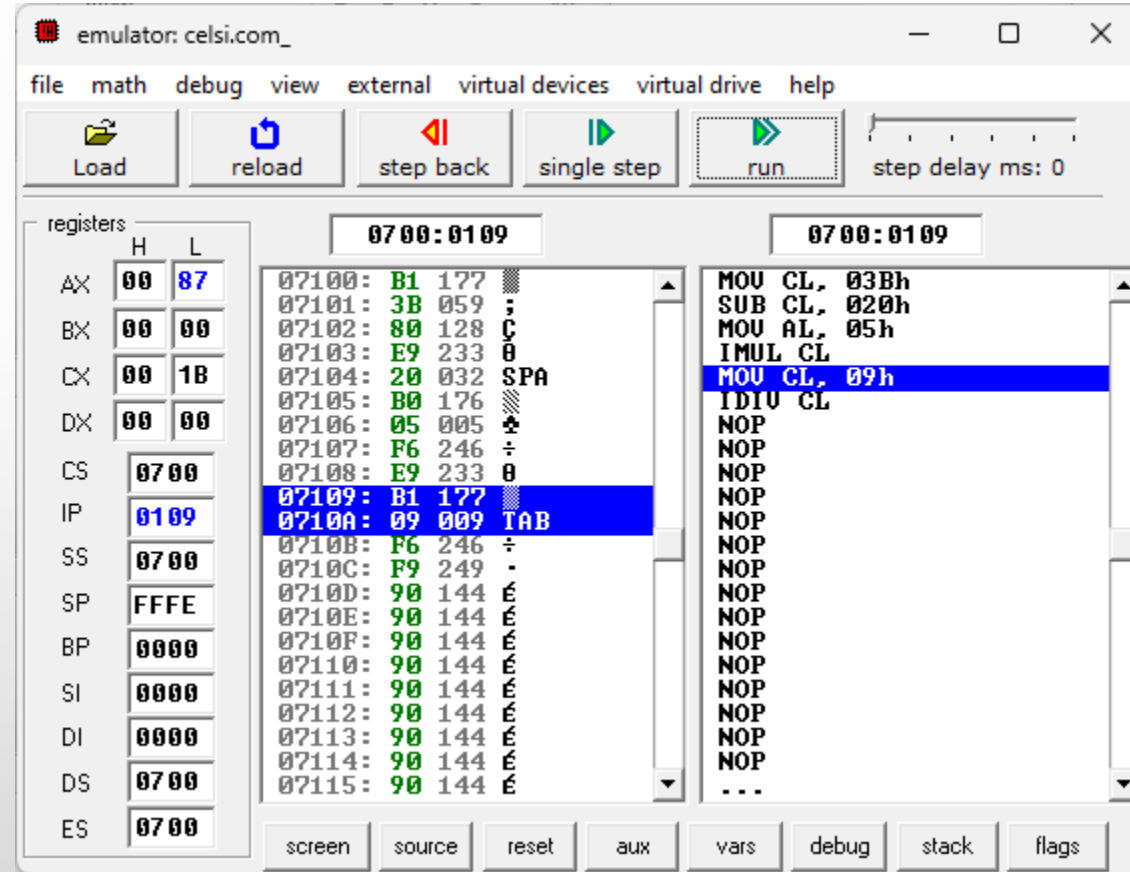



Fahrenayt Santigrad Dönüşümü



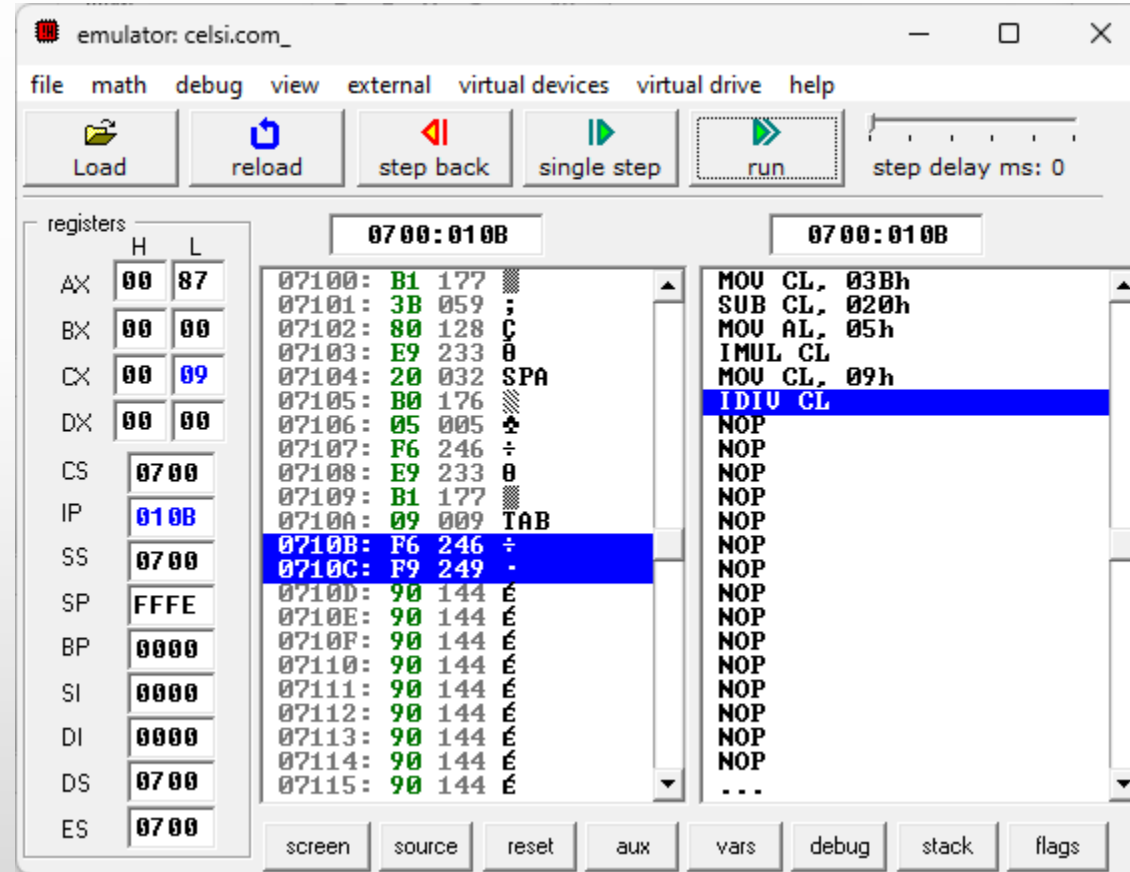


Fahrenayt Santigrad Dönüşümü



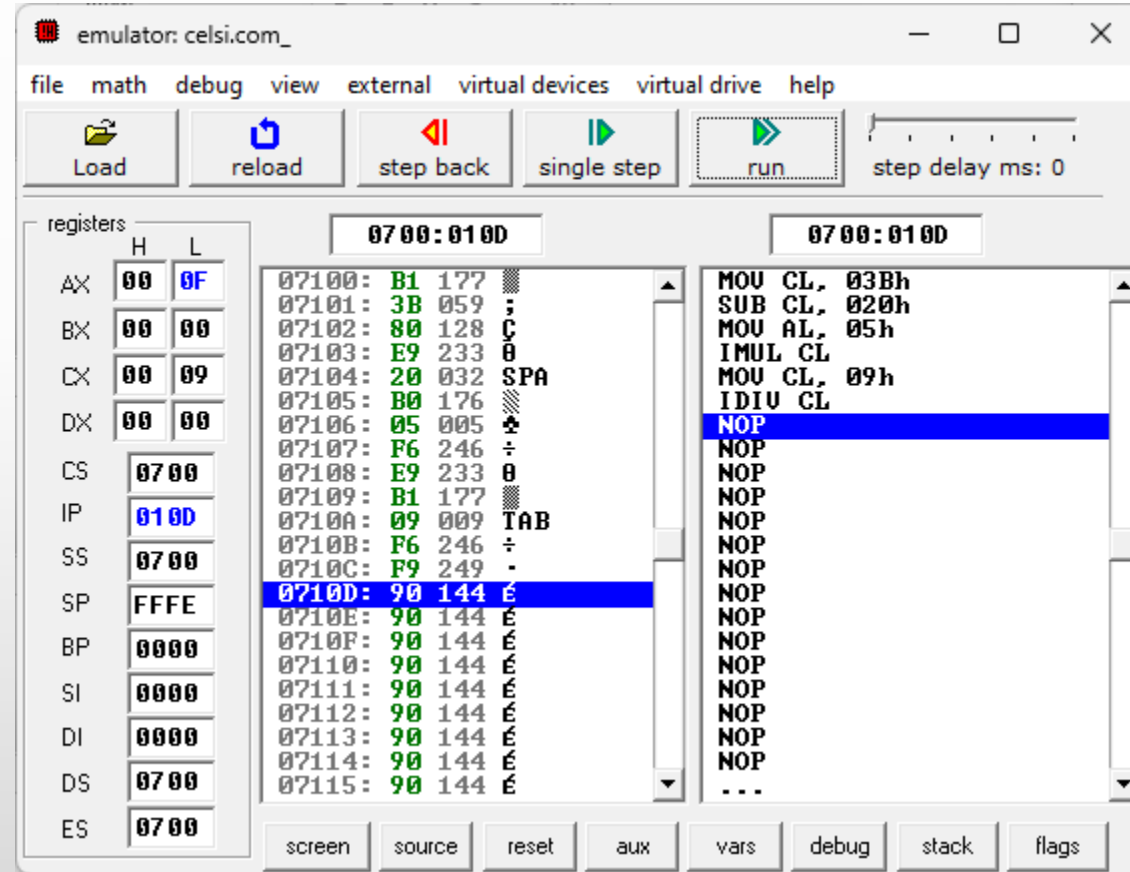


Fahrenayt Santigrad Dönüşümü





Fahrenayt Santigrad Dönüşümü





SON