

NAME :

PURVAL MADHUKAR BHUDE

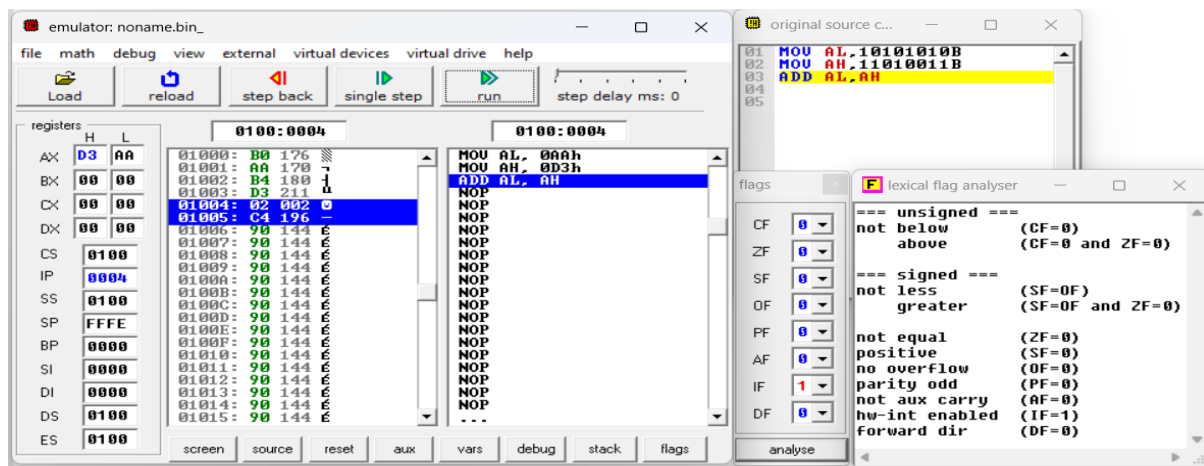
BRANCH :-

CSE

SECTION NO. 4

ROLL NO. S20230010193

LOAD NUMBER



The screenshot displays the x86-64 emulator interface. The top menu bar includes file, math, debug, view, external, virtual devices, virtual drive, and help. Below the menu is a toolbar with buttons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms.

The registers panel on the left shows the state of various registers. The AX register is highlighted with a value of 0000007D. The CS register is 0100, IP is 0006, SS is 0100, SP is FFFE, BP is 0000, SI is 0000, DI is 0000, DS is 0100, and ES is 0100.

The instruction window on the right shows a list of instructions. The instruction at address 0100:0006 is highlighted: `MOV AL, 00Ah`. The instruction at address 0100:0007 is `MOV AH, 0D3h`. The instruction at address 0100:0008 is `ADD AL, AH`.

The flags panel on the right shows the state of various flags. The CF flag is 1, ZF is 0, SF is 0, OF is 1, PF is 1, AF is 0, IF is 1, and DF is 0.

Q1 B) subtraction : doing next step in below process.

LOAD NUMBER

emulator: noname.bin_

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	AA
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0002	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0002

Address	Value	Comment
01000: B0	176	
01001: AA	170	
01002: B4	180	
01003: D3	211	
01004: 02	002	
01005: C4	196	
01006: 90	144	
01007: 90	144	
01008: 90	144	
01009: 90	144	
0100A: 90	144	
0100B: 90	144	
0100C: 90	144	
0100D: 90	144	
0100E: 90	144	
0100F: 90	144	
01010: 90	144	
01011: 90	144	
01012: 90	144	
01013: 90	144	
01014: 90	144	
01015: 90	144	

MOV AL, 0AAh
MOV AH, 0D3h
ADD AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

original source c...

```
01 MOV AL,10101010B
02 MOV AH,11010011B
03 ADD AL,AH
04
05
```

flags

lexical flag analyser

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

=== unsigned ===
not below (CF=0)
above (CF=0 and ZF=0)
=== signed ===
not less (SF=0F)
greater (SF=0F and ZF=0)
not equal (ZF=0)
positive (SF=0)
no overflow (OF=0)
parity odd (PF=0)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

emulator: noname.bin_

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	D3	AA
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0004	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0004

Address	Value	Comment
01000: B0	176	
01001: AA	170	
01002: B4	180	
01003: D3	211	
01004: 02	002	
01005: C4	196	
01006: 90	144	
01007: 90	144	
01008: 90	144	
01009: 90	144	
0100A: 90	144	
0100B: 90	144	
0100C: 90	144	
0100D: 90	144	
0100E: 90	144	
0100F: 90	144	
01010: 90	144	
01011: 90	144	
01012: 90	144	
01013: 90	144	
01014: 90	144	
01015: 90	144	

MOV AL, 0AAh
MOV AH, 0D3h
ADD AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

original source c...

```
01 MOV AL,10101010B
02 MOV AH,11010011B
03 ADD AL,AH
04
05
```

flags

lexical flag analyser

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

=== unsigned ===
not below (CF=0)
above (CF=0 and ZF=0)
=== signed ===
not less (SF=0F)
greater (SF=0F and ZF=0)
not equal (ZF=0)
positive (SF=0)
no overflow (OF=0)
parity odd (PF=0)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

SUBTRACT NUMBER

emulator: noname.bin_

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	D3	07
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0006	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0006

Address	Value	Comment
01000: B0	176	
01001: AA	170	
01002: B4	180	
01003: D3	211	
01004: 2A	042	*
01005: C4	196	
01006: 90	144	
01007: 90	144	
01008: 90	144	
01009: 90	144	
0100A: 90	144	
0100B: 90	144	
0100C: 90	144	
0100D: 90	144	
0100E: 90	144	
0100F: 90	144	
01010: 90	144	
01011: 90	144	
01012: 90	144	
01013: 90	144	
01014: 90	144	
01015: 90	144	

MOV AL, 0AAh
MOV AH, 0D3h
SUB AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

original source c...

```
01 MOV AL,10101010B
02 MOV AH,11010011B
03 SUB AL,AH
04
05
```

flags

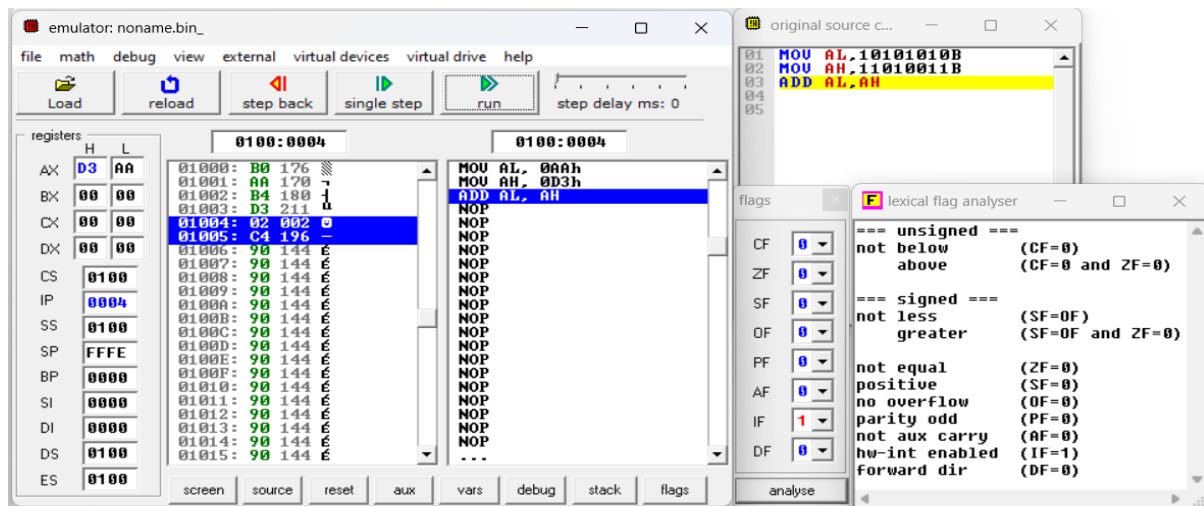
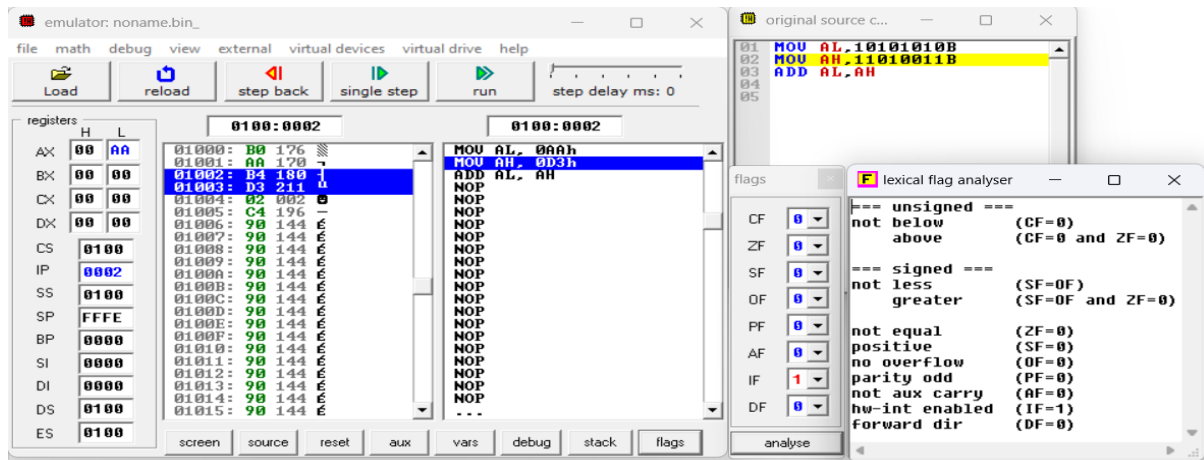
lexical flag analyser

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

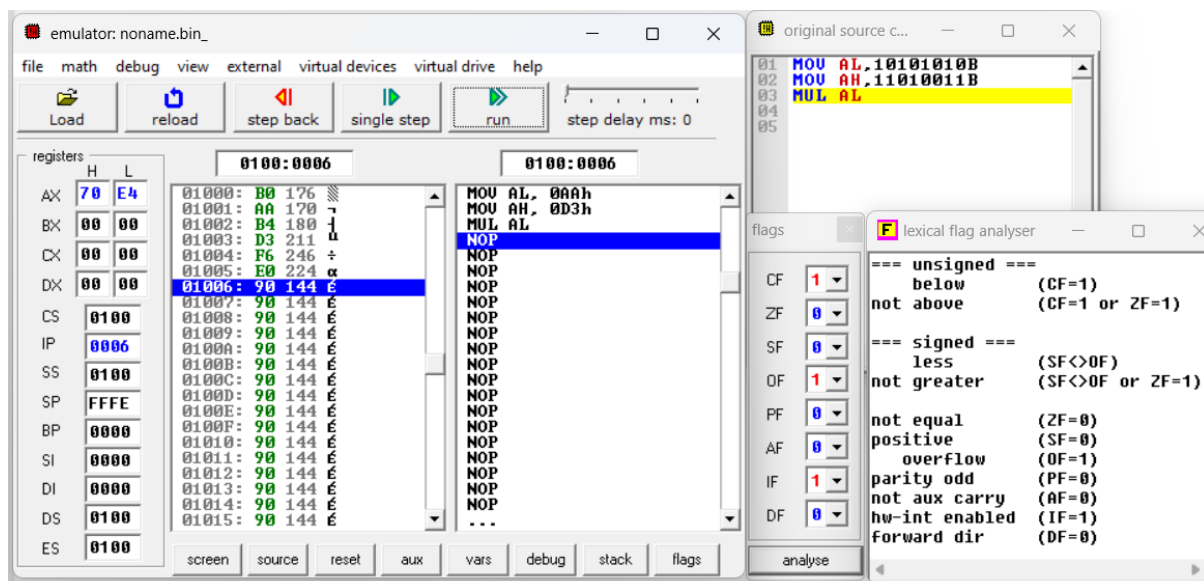
=== unsigned ===
below (CF=1)
not above (CF=1 or ZF=1)
=== signed ===
less (SF<0F)
not greater (SF<0F or ZF=1)
not equal (ZF=0)
negative (SF=1)
no overflow (OF=0)
parity even (PF=1)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

Q1) C) Multiplication : doing next step

LOADING NUMBER



MULTIPLICATION



Q2) A) AND : DOING NEXT STEP

LOADING NUMBERS

emulator: noname.bin_

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	FA
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0002	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0002

01000:	B0	176
01001:	FA	250
01002:	B4	180
01003:	97	151
01004:	22	034
01005:	C4	196
01006:	90	144
01007:	90	144
01008:	90	144
01009:	90	144
0100A:	90	144
0100B:	90	144
0100C:	90	144
0100D:	90	144
0100E:	90	144
0100F:	90	144
01010:	90	144
01011:	90	144
01012:	90	144
01013:	90	144
01014:	90	144
01015:	90	144

0100:0002

```
MOV AL, 0FAh
MOV AH, 097h
AND AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

original source c...

```
01 MOV AL,1111010B
02 MOV AH,1001011B
03 AND AL,AH
04
05
```

flags

lexical flag analyser

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

=== unsigned ===
not below (CF=0)
above (CF=0 and ZF=0)

=== signed ===
not less (SF=0F)
greater (SF=0F and ZF=0)

not equal (ZF=0)
positive (SF=0)
no overflow (OF=0)
parity odd (PF=0)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

emulator: noname.bin_

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	97	FA
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0004	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0004

01000:	B0	176
01001:	FA	250
01002:	B4	180
01003:	97	151
01004:	22	034
01005:	C4	196
01006:	90	144
01007:	90	144
01008:	90	144
01009:	90	144
0100A:	90	144
0100B:	90	144
0100C:	90	144
0100D:	90	144
0100E:	90	144
0100F:	90	144
01010:	90	144
01011:	90	144
01012:	90	144
01013:	90	144
01014:	90	144
01015:	90	144

0100:0004

```
MOV AL, 0FAh
MOV AH, 097h
AND AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

original source c...

```
01 MOV AL,1111010B
02 MOV AH,1001011B
03 AND AL,AH
04
05
```

flags

lexical flag analyser

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

=== unsigned ===
not below (CF=0)
above (CF=0 and ZF=0)

=== signed ===
not less (SF=0F)
greater (SF=0F and ZF=0)

not equal (ZF=0)
positive (SF=0)
no overflow (OF=0)
parity odd (PF=0)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

AND

emulator: noname.bin_

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	97	92
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0006	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0006

01000:	B0	176
01001:	FA	250
01002:	B4	180
01003:	97	151
01004:	22	034
01005:	C4	196
01006:	90	144
01007:	90	144
01008:	90	144
01009:	90	144
0100A:	90	144
0100B:	90	144
0100C:	90	144
0100D:	90	144
0100E:	90	144
0100F:	90	144
01010:	90	144
01011:	90	144
01012:	90	144
01013:	90	144
01014:	90	144
01015:	90	144

0100:0006

```
MOV AL, 0FAh
MOV AH, 097h
AND AL, AH
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

original source c...

```
01 MOV AL,1111010B
02 MOV AH,1001011B
03 AND AL,AH
04
05
```

flags

lexical flag analyser

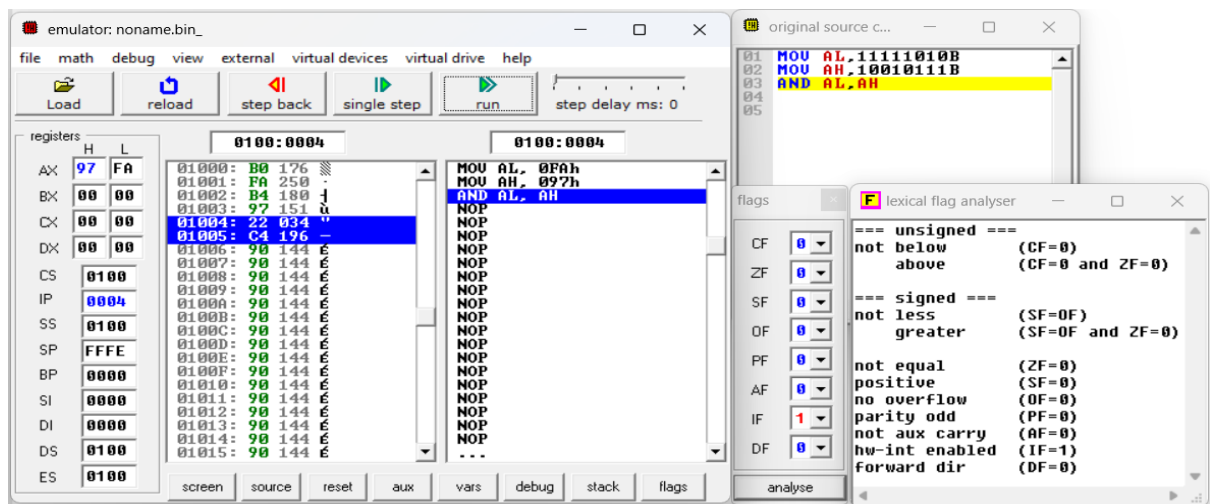
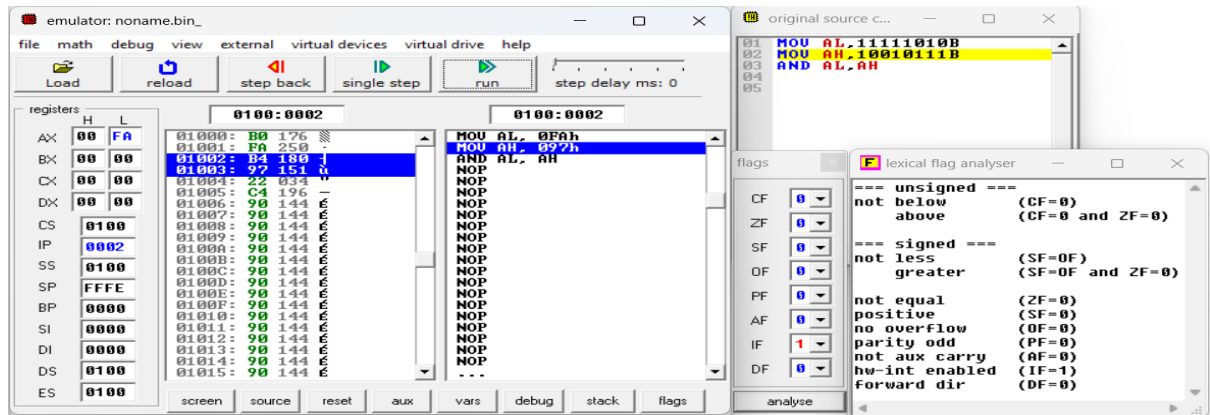
CF 0 ZF 0 SF 1 OF 0 PF 0 AF 0 IF 1 DF 0

=== unsigned ===
not below (CF=0)
above (CF=0 and ZF=0)

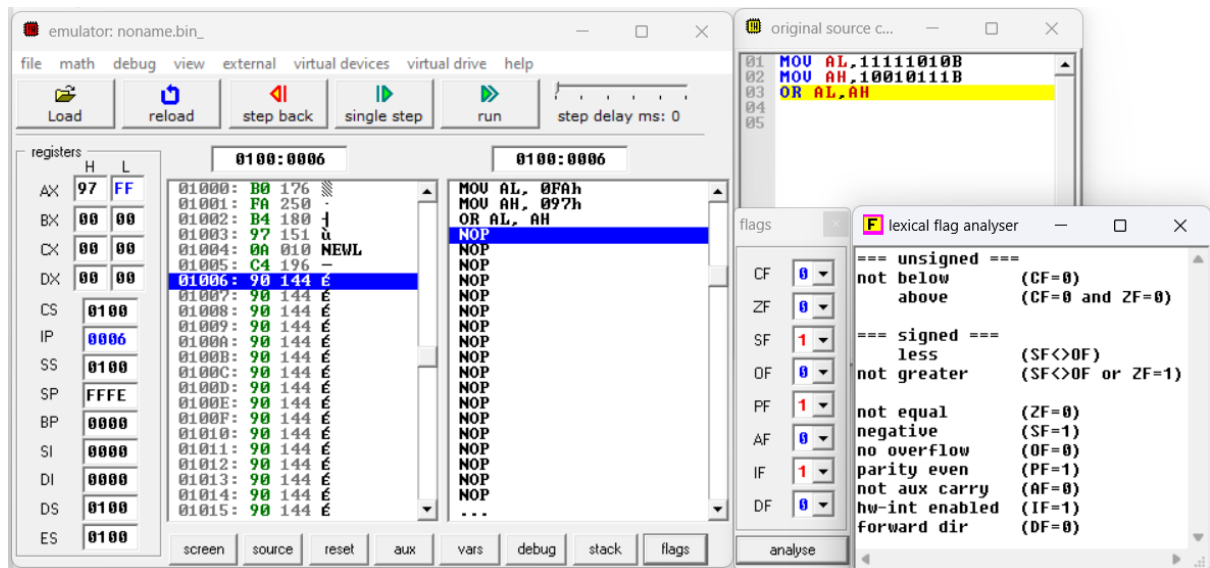
=== signed ===
less (SF<>OF)
not greater (SF<>OF or ZF=1)

not equal (ZF=0)
negative (SF=1)
no overflow (OF=0)
parity odd (PF=0)
not aux carry (AF=0)
hw-int enabled (IF=1)
forward dir (DF=0)

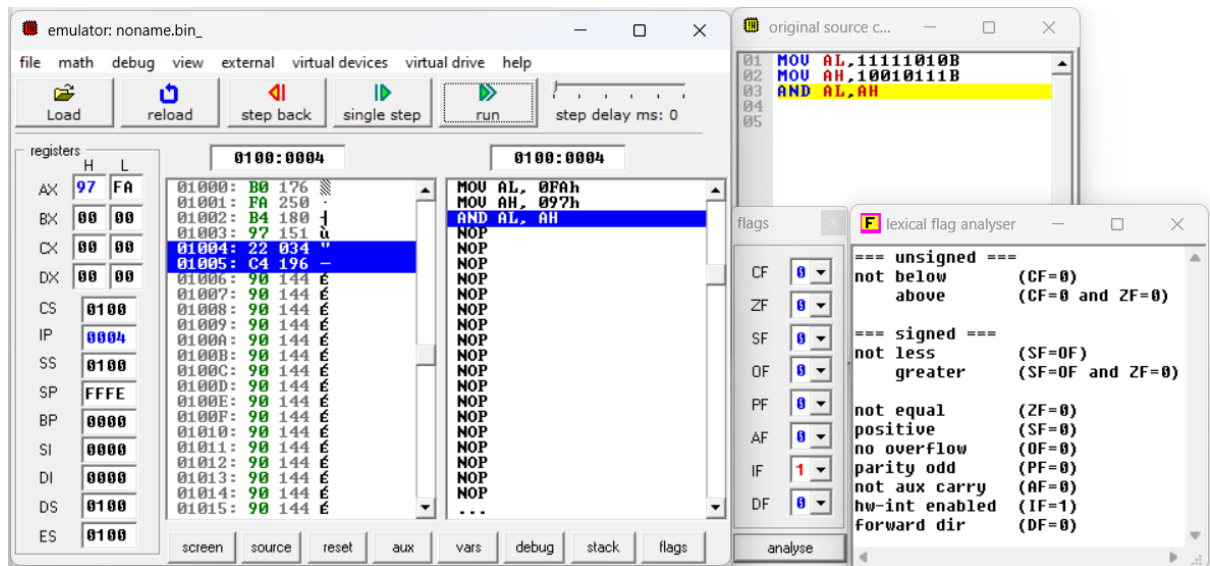
LOADING NUMBERS



OR



LOADING NUMBERS



The screenshot displays the x86-64 emulator interface with the following components:

- Top Bar:** Shows the emulator name "emulator: none.bin_" and standard window controls.
- Menu Bar:** Includes file, math, debug, view, external, virtual devices, virtual drive, and help.
- Toolbar:** Contains buttons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms.
- Registers Panel:**
 - AX: 97, BX: 00, CX: 00, DX: 00
 - CS: 0100, IP: 0006, SS: 0100, SP: FFFE, BP: 0000, SI: 0000, DI: 0000, DS: 0100, ES: 0100
- Memory Panel:**
 - Address 0100:0006: 00 176, 0100:0007: FA 250, 0100:0008: B4 180, 0100:0009: 97 151, 0100:000A: 32 050, 0100:000B: C4 196, 0100:000C: 90 144, 0100:000D: 90 144, 0100:000E: 90 144, 0100:000F: 90 144, 0100:0010: 90 144, 0100:0011: 90 144, 0100:0012: 90 144, 0100:0013: 90 144, 0100:0014: 90 144, 0100:0015: 90 144.
- Assembly Panel:**
 - 01: MOV AL, 11111010B
 - 02: MOV AH, 10010111B
 - 03: XOR AL, AH
 - 04: NOP
 - 05: NOP
- Flags Panel:**
 - CF: 0, ZF: 0, SF: 0, OF: 0, PF: 0, AF: 0, IF: 1, DF: 0
- Lexical Flag Analyser:**
 - === unsigned ===
 - not below (CF=0)
 - above (CF=0 and ZF=0)
 - === signed ===
 - not less (SF=0F)
 - greater (SF=0F and ZF=0)
 - not equal (ZF=0)
 - positive (SF=0)
 - no overflow (OF=0)
 - parity odd (PF=0)
 - not aux carry (AF=0)
 - hw-int enabled (IF=1)
 - forward dir (DF=0)