

The assignment itself spells out all of the details. I've run the NAND, RSHIFT example completely below and commented each important step with <<<<==== comment directly in the screen output. Begin by creating a directory for assignment 4 in your environment:

```
[bill@mercury2 ~]$ mkdir my_a4 <<<<==== make a local directory
[bill@mercury2 ~]$ cd ~bill/cs305/NandRshiftExample/ <<<<==== move to my assignment
directory
```

```
[bill@mercury2 NandRshiftExample]$ ls -l <<<<==== check the content
total 295
-rw-r--r-- 1 bill fac 351 Mar 8 2023 Makefile_nand_rshift
-rw-r--r-- 1 bill fac 548 Mar 8 2023 README_assign4
-rw-r--r-- 1 bill fac 809 Oct 10 12:20 Some_Test_Results.txt
-rwx----- 1 bill fac 63232 Mar 16 2023 masm_nr
-rwxr-xr-x 1 bill fac 64032 Feb 14 2023 masm_orig
-rwxr-xr-x 1 bill fac 64384 Feb 14 2023 mcc
-rwxr-xr-x 1 bill fac 69384 Feb 14 2023 mic1
-rw-r--r-- 1 bill fac 17532 Mar 8 2023 mic1symasm_nand_rshift.c
-rw-r--r-- 1 bill fac 2565 Mar 8 2023 mic1symasm_nand_rshift.ll
-rw-r--r-- 1 bill fac 3729 Mar 8 2023 microcode_nand_rshift.mc
-rw-r--r-- 1 bill fac 645 Oct 10 08:32 nand_rshift_test.asm
-rw----- 1 bill fac 3234 Oct 10 11:11 nr.dat
drwx----- 2 bill fac 4096 Mar 16 2023 nr_check
-rw----- 1 bill fac 3234 Oct 10 09:36 prom_nr.dat
```

<<<<==== now copy the files you need into your assignment directory

```
[bill@mercury2 NandRshiftExample]$ cp Makefile_nand_rshift mcc mic1
mic1symasm_nand_rshift.c mic1symasm_nand_rshift.ll microcode_nand_rshift.mc
nand_rshift_test.asm ~/my_a4
```

```
[bill@mercury2 NandRshiftExample]$ cd ~/my_a4 <<<<==== now go back to your assignment
directory
```

```
[bill@mercury2 my_a4]$ ls -l <<<<==== check the content of your assignment directory
total 157
-rw----- 1 bill fac 351 Oct 22 20:11 Makefile_nand_rshift <<<<==== the makefile for
the new assembler
-rwx----- 1 bill fac 64384 Oct 22 20:11 mcc <<<<==== the microcode compiler program
-rwx----- 1 bill fac 69384 Oct 22 20:11 mic1 <<<<==== the mic1 emulator program
-rw----- 1 bill fac 17532 Oct 22 20:11 mic1symasm_nand_rshift.c <<<<==== the new
assembler c code
-rw----- 1 bill fac 2565 Oct 22 20:11 mic1symasm_nand_rshift.ll <<<<==== the new
assembler flex specification file
-rw----- 1 bill fac 3729 Oct 22 20:13 microcode_nand_rshift.mc <<<<==== the new
microcode source with NAND and RSHIFT
-rw----- 1 bill fac 645 Oct 22 20:11 nand_rshift_test.asm <<<<==== the test program
for NAND and RSHIFT
```

```
[bill@mercury2 my_a4]$ ./mcc microcode_nand_rshift.mc > prom_nr.dat <<<<==== use mcc to
build microcode .dat file
Microcode Compiler - Version 1.0 - Richard Boccuzzi
```

```
[bill@mercury2 my_a4]$ make -f Makefile_nand_rshift <<<<==== use make to build a new
assembler
gcc -c -g mic1symasm_nand_rshift.c
flex mic1symasm_nand_rshift.ll
gcc -c -g lex.yy.c
gcc -o masm_nr mic1symasm_nand_rshift.o lex.yy.o
```

```
[bill@mercury2 my_a4]$ ls -l <<<<==== check the new contents of your assignment directory
total 354
-rw----- 1 bill fac 351 Oct 22 20:11 Makefile_nand_rshift
-rw----- 1 bill fac 53732 Oct 22 20:14 lex.yy.c
-rw----- 1 bill fac 45336 Oct 22 20:14 lex.yy.o
-rwx----- 1 bill fac 63168 Oct 22 20:14 masm_nr
-rwx----- 1 bill fac 64384 Oct 22 20:11 mcc
-rwx----- 1 bill fac 69384 Oct 22 20:11 mic1
-rw----- 1 bill fac 17532 Oct 22 20:11 mic1symasm_nand_rshift.c
-rw----- 1 bill fac 2565 Oct 22 20:11 mic1symasm_nand_rshift.ll
-rw----- 1 bill fac 34592 Oct 22 20:14 mic1symasm_nand_rshift.o
-rw----- 1 bill fac 3729 Oct 22 20:13 microcode_nand_rshift.mc
-rw----- 1 bill fac 645 Oct 22 20:11 nand_rshift_test.asm
-rw----- 1 bill fac 3234 Oct 22 20:14 prom_nr.dat
```

```

[bill@mercury2 my_a4]$ ./masm_nr < nand_rshift_test.asm > nand_rshift_test.obj
[bill@mercury2 my_a4]$
[bill@mercury2 my_a4]$ ls -l
total 355
-rw----- 1 bill fac 351 Oct 22 20:11 Makefile_nand_rshift
-rw----- 1 bill fac 53732 Oct 22 20:14 lex.yy.c <<<<<=== c file built by flex during
make
-rw----- 1 bill fac 45336 Oct 22 20:14 lex.yy.o <<<<<=== object file of aboce c file
-rwx----- 1 bill fac 63168 Oct 22 20:14 masm_nr <<<<<=== new assembler that knows NAND
and RSHIFT
-rwx----- 1 bill fac 64384 Oct 22 20:11 mcc
-rwx----- 1 bill fac 69384 Oct 22 20:11 mic1
-rw----- 1 bill fac 17532 Oct 22 20:11 mic1symasm_nand_rshift.c
-rw----- 1 bill fac 2565 Oct 22 20:11 mic1symasm_nand_rshift.ll
-rw----- 1 bill fac 34592 Oct 22 20:14 mic1symasm_nand_rshift.o
-rw----- 1 bill fac 3729 Oct 22 20:13 microcode_nand_rshift.mc
-rw----- 1 bill fac 645 Oct 22 20:11 nand_rshift_test.asm
-rw----- 1 bill fac 612 Oct 22 20:15 nand_rshift_test.obj <<<<<=== executable NAND,
RSHIFT test file
-rw----- 1 bill fac 3234 Oct 22 20:14 prom_nr.dat <<<<<=== new microcode built by mcc

```

<<<<<=== now run the emulator with the executable NAND, RSHIFT test file

```

[bill@mercury2 my_a4]$ ./mic1 prom_nr.dat nand_rshift_test.obj 0 2000

```

```

Read in 98 micro instructions
Read in 36 machine instructions
Starting PC is : 0000000000000000 base 10: 0
Starting SP is : 0000011111010000 base 10: 2000

```

```

ProgramCounter : 0000000000000111 base 10: 7 <<<<<=== first halt after NAND 3451,
8192
Accumulator : 1111111111111111 base 10: 65535
InstructionReg : 1111111110000000 base 10: 65472
TempInstr : 1100000000000000 base 10: 49152
StackPointer : 0000011111001110 base 10: 1998
ARegister : 0000000000000000 base 10: 0
BRegister : 0010000000000000 base 10: 8192
CRegister : 0000110101111011 base 10: 3451
DRegister : 0000000000000000 base 10: 0
ERegister : 0000000000000000 base 10: 0
FRegister : 0000000000000000 base 10: 0

```

Total cycles : 81

```

Type decimal address to view memory, q to quit or c to continue: 29 <<<<<=== data and
answers begin at 30
the location 29 has value 1111111111111111 , or 65535 or signed -1
Type <Enter> to continue debugging
Type q to quit
Type f for forward range
Type b for backward range: f
Type the number of forward locations to dump: 7
d1: the location 30 has value 0000110101111011 , or 3451 or signed 3451 <<<<<===
d2: the location 31 has value 0010000000000000 , or 8192 or signed 8192 <<<<<===
d3: the location 32 has value 1110000000000000 , or 57344 or signed -8192 <<<<<===
d4: the location 33 has value 1000000000000000 , or 32768 or signed -32768 <<<<<===
d5: the location 34 has value 0000001110110000 , or 944 or signed 944 <<<<<===
the location 35 has value 1111111111111111 , or 65535 or signed -1 <<<<<===
nandans: all 1s for NAND 3451, 8192
the location 36 has value 1111111111111111 , or 65535 or signed -1
Type decimal address to view memory, q to quit or c to continue: c <<<<<=== continue
to next halt

```

The new PC is : 0000000000000111

```

ProgramCounter : 0000000000001110 base 10: 14
Accumulator : 1111111011001111 base 10: 65231
InstructionReg : 1111111110000000 base 10: 65472
TempInstr : 1100000000000000 base 10: 49152
StackPointer : 0000011111001100 base 10: 1996
ARegister : 0000000100110000 base 10: 304
BRegister : 0000001110110000 base 10: 944

```

CRegister : 0000110101111011 base 10: 3451
DRegister : 0000000000000000 base 10: 0
ERegister : 0000000000000000 base 10: 0
FRegister : 0000000000000000 base 10: 0

Total cycles : 162

Type decimal address to view memory, q to quit or c to continue: 29
the location 29 has value 1111111111111111 , or 65535 or signed -1

Type <Enter> to continue debugging

Type q to quit

Type f for forward range

Type b for backward range: f

Type the number of forward locations to dump: 7

the location 30 has value 0000110101111011 , or 3451 or signed 3451

the location 31 has value 0010000000000000 , or 8192 or signed 8192

the location 32 has value 1110000000000000 , or 57344 or signed -8192

the location 33 has value 1000000000000000 , or 32768 or signed -32768

the location 34 has value 0000001110110000 , or 944 or signed 944

the location 35 has value 1111111011001111 , or 65231 or signed -305 <<<<====

nandans: -305 for NAND 3451, 944

the location 36 has value 1111111111111111 , or 65535 or signed -1

Type decimal address to view memory, q to quit or c to continue: c <<<<==== continue
to next halt

The new PC is : 0000000000001110

ProgramCounter : 000000000010001 base 10: 17
Accumulator : 0000110101111011 base 10: 3451 <<<<==== first RSHIFT 0 of 3451, no
change
InstructionReg : 111111111000000 base 10: 65472
TempInstr : 1100000000000000 base 10: 49152
StackPointer : 0000011111001100 base 10: 1996
ARegister : 0000000000000111 base 10: 15
BRegister : 1111111111111111 base 10: 65535
CRegister : 0000110101111011 base 10: 3451
DRegister : 0000000000000000 base 10: 0
ERegister : 0000000000000000 base 10: 0
FRegister : 0000000000000000 base 10: 0

Total cycles : 202

Type decimal address to view memory, q to quit or c to continue: c <<<<==== continue
to next halt

The new PC is : 000000000010001

ProgramCounter : 000000000010100 base 10: 20
Accumulator : 0000010000000000 base 10: 1024 <<<<==== second RSHIFT 3 of 8192
InstructionReg : 111111111000000 base 10: 65472
TempInstr : 1100000000000000 base 10: 49152
StackPointer : 0000011111001100 base 10: 1996
ARegister : 0000000000000111 base 10: 15
BRegister : 1111111111111111 base 10: 65535
CRegister : 0000110101111011 base 10: 3451
DRegister : 0000000000000000 base 10: 0
ERegister : 0000000000000000 base 10: 0
FRegister : 0000000000000000 base 10: 0

Total cycles : 248

Type decimal address to view memory, q to quit or c to continue: c <<<<==== continue
to next halt

The new PC is : 000000000010100

ProgramCounter : 000000000010111 base 10: 23
Accumulator : 0000000111000000 base 10: 448 <<<<==== third RSHIFT 7 of -8192
InstructionReg : 111111111000000 base 10: 65472
TempInstr : 1100000000000000 base 10: 49152
StackPointer : 0000011111001100 base 10: 1996
ARegister : 0000000000000111 base 10: 15
BRegister : 1111111111111111 base 10: 65535
CRegister : 0000110101111011 base 10: 3451
DRegister : 0000000000000000 base 10: 0
ERegister : 0000000000000000 base 10: 0
FRegister : 0000000000000000 base 10: 0

Total cycles : 302

Type decimal address to view memory, q to quit or c to continue: c <<<<=== continue to next halt

The new PC is : 000000000010111

ProgramCounter	: 000000000011010	base 10:	26	
Accumulator	: 000000000000001	base 10:	1	<<<<=== fourth RSHIFT 15 of -32768
InstructionReg	: 111111111000000	base 10:	65472	
TempInstr	: 110000000000000	base 10:	49152	
StackPointer	: 0000011111001100	base 10:	1996	
ARegister	: 0000000000001111	base 10:	15	
BRegister	: 111111111111111	base 10:	65535	
CRegister	: 0000110101111011	base 10:	3451	
DRegister	: 000000000000000	base 10:	0	
ERegister	: 000000000000000	base 10:	0	
FRegister	: 000000000000000	base 10:	0	

Total cycles : 372

Type decimal address to view memory, q to quit or c to continue: c <<<<=== continue to next halt

The new PC is : 000000000011010

ProgramCounter	: 000000000011101	base 10:	29	
Accumulator	: 000000000111011	base 10:	59	<<<<=== last RSHIFT 4 of 944
InstructionReg	: 111111111000000	base 10:	65472	
TempInstr	: 110000000000000	base 10:	49152	
StackPointer	: 0000011111001100	base 10:	1996	
ARegister	: 0000000000001111	base 10:	15	
BRegister	: 111111111111111	base 10:	65535	
CRegister	: 0000110101111011	base 10:	3451	
DRegister	: 000000000000000	base 10:	0	
ERegister	: 000000000000000	base 10:	0	
FRegister	: 000000000000000	base 10:	0	

Total cycles : 420

Type decimal address to view memory, q to quit or c to continue: q
MIC-1 emulator finishing, goodbye

<<<<=== all procedures and results shown above for the NAND, RSHIFT example