Basic I/O on SIC

Instructor: Quincy Wu

TA: Changyu Wu

Outline

- Install SIC
- How to use the SIC Assembler
- How to use the SIC Simulator

Install SIC

- %sicinstall
  - SIC
    - DEV00
    - DEVF1
    - DEVF2
    - LOG
    - INTFILE <intermediate working file for the assembler>
    - LISFILE <the assembly listing>
    - SRCFILE <the source program to be assembled>
    - OBJFILE <the object program generated by the assembler>

How to use the SIC Assembler

- %vi SRCFILE
- Source format

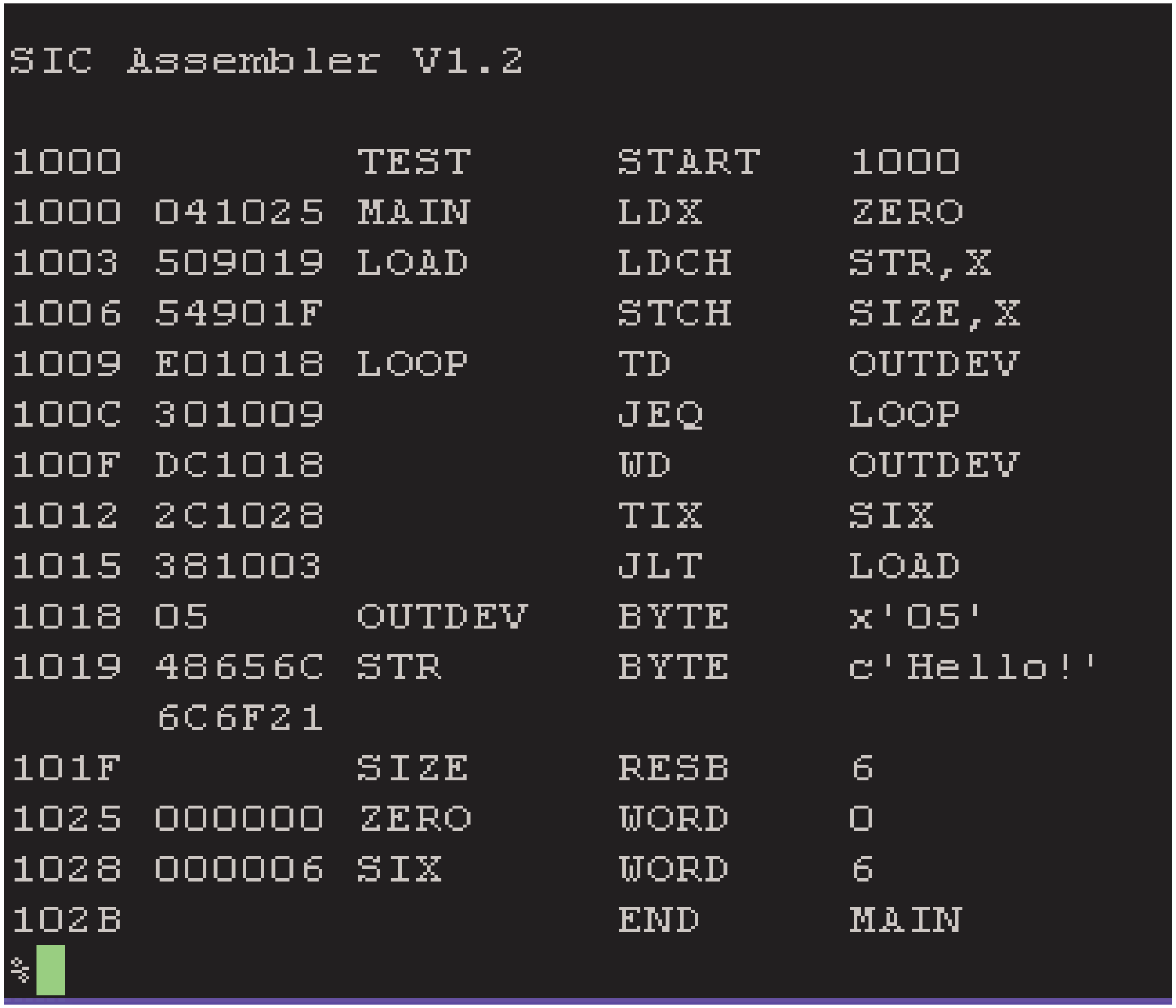
❑ Bytes	1-8	Label
	9	Blank
	10-15	Operation code (or Assembler directive)
	16-17	Blank
	18-35	Operand
	36-66	Comment

Example1

Label	Operation code	Operand
TEST	START	1000
MAIN	LDX	ZERO
LOAD	LDCH	STR,X
	STCH	SIZE,X
LOOP	TD	OUTDEV
	JEQ	LOOP
	WD	OUTDEV
	TIX	SIX
	JLT	LOAD
OUTDEV	BYTE	X'05'
STR	BYTE	c'Hello!'
SIZE	RESB	6
ZERO	WORD	0
SIX	WORD	6
	END	MAIN

Source program assembling

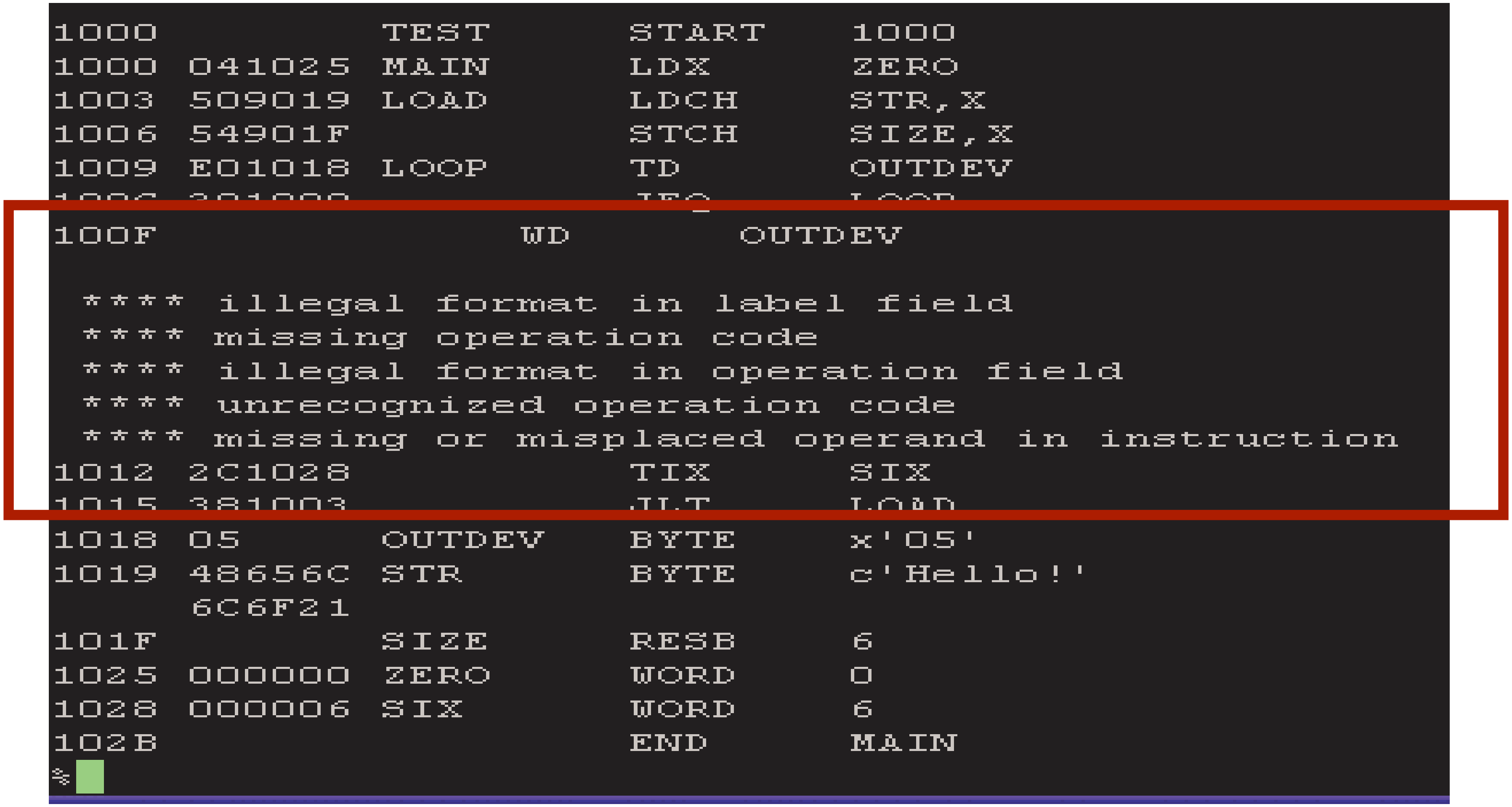
- %sicasm
- %more LISFILE



```
SIC Assembler V1.2

1000        TEST     START   1000
1000 041025 MAIN     LDX     ZERO
1003 509019 LOAD     LDCH    STR,X
1006 54901F          STCH    SIZE,X
1009 E01018 LOOP     TD      OUTDEV
100C 301009          JEQ     LOOP
100F DC1018          WD      OUTDEV
1012 2C1028          TIX     SIX
1015 381003          JLT     LOAD
1018 05     OUTDEV   BYTE    x'05'
1019 48656C STR      BYTE    c'Hello!'
     6C6F21
101F        SIZE     RESB    6
1025 000000 ZERO     WORD    0
1028 000006 SIX      WORD    6
102B                 END     MAIN
%
```

Source program assembling(2)

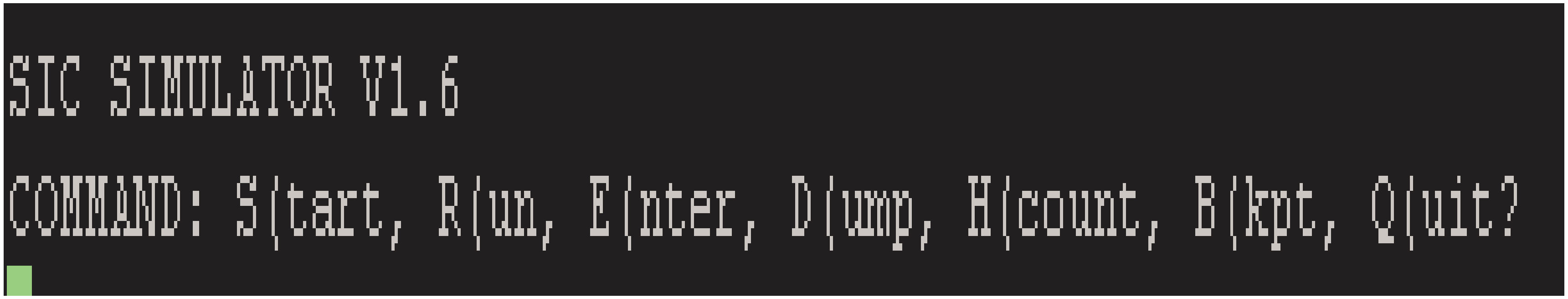


```
1000         TEST     START   1000
1000  041025 MAIN     LDX     ZERO
1003  509019 LOAD     LDCH    STR,X
1006  54901F          STCH    SIZE,X
1009  E01018 LOOP     TD      OUTDEV
100C  301009          JEQ     LOOP
100F                       WD       OUTDEV

 **** illegal format in label field
 **** missing operation code
 **** illegal format in operation field
 **** unrecognized operation code
 **** missing or misplaced operand in instruction
1012  2C1028          TIX     SIX
1015  381003          JLT     LOAD
1018  05     OUTDEV   BYTE    x'05'
1019  48656C STR      BYTE    c'Hello!'
      6C6F21
101F         SIZE     RESB    6
1025  000000 ZERO     WORD    0
1028  000006 SIX      WORD    6
102B                  END     MAIN
%
```

How to use the SIC Simulator

- %sicsim



```
SIC SIMULATOR V1.6
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
```

- You may now enter any of the commands described below; each command may be abbreviated by entering only its first letter.

Commands

- START
  - Entering S causes the simulator to read 128 bytes of data from device 00 into memory, starting at address 0000
- RUN
  - This command causes the simulator to begin executing SIC machine language instructions from a program in memory.

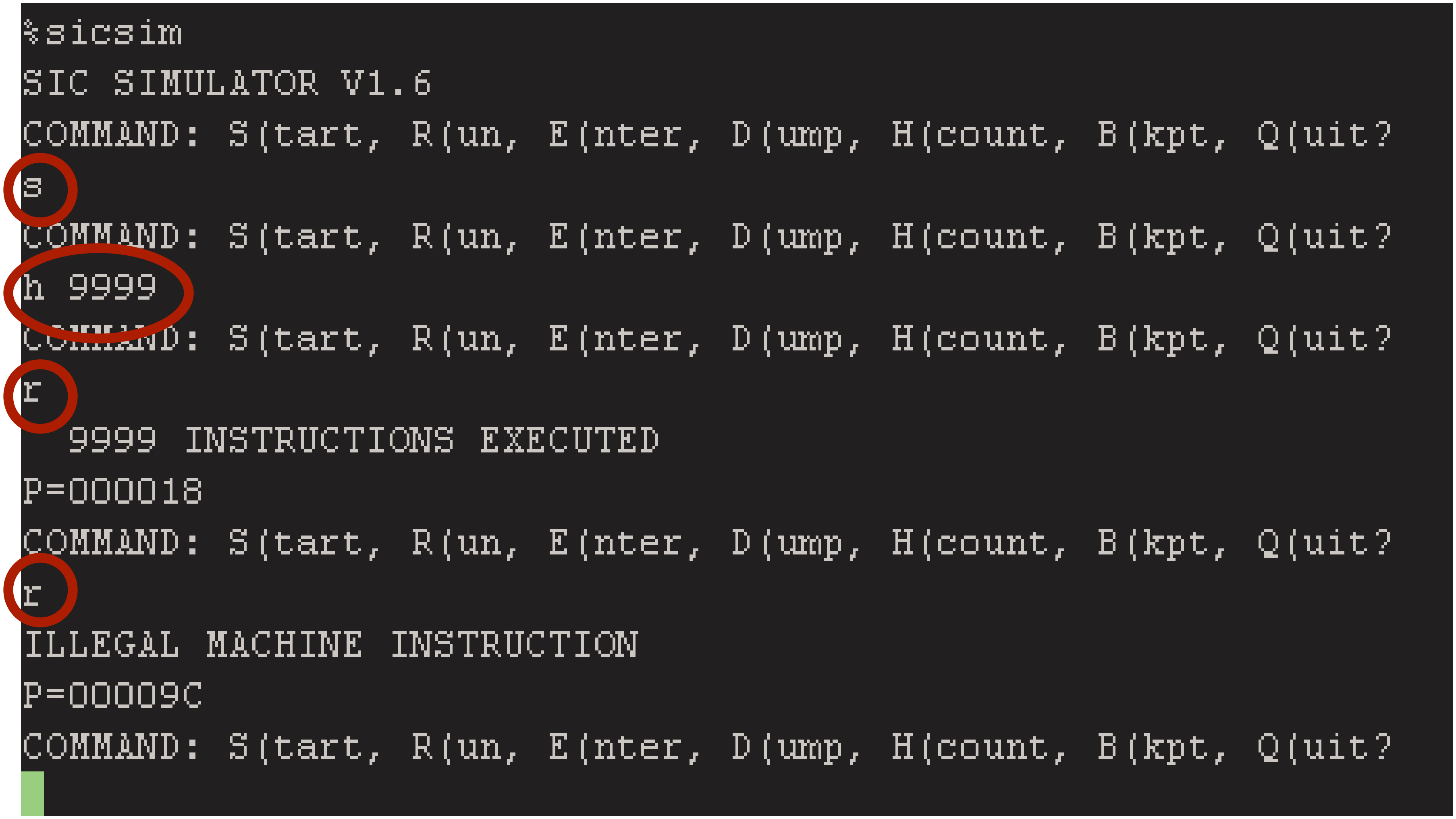
Commands(2)

- Hcount
  - This command is used to specify the maximum number of SIC instructions to be executed in response to a RUN command.
  - The maximum value is 9999

```
h n
```

- Quit
  - This command is used to terminate the simulation

Commands(4)



```
%sicsim
SIC SIMULATOR V1.6
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
s
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
h 9999
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
r
  9999 INSTRUCTIONS EXECUTED
P=000018
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
r
ILLEGAL MACHINE INSTRUCTION
P=00009C
COMMAND: S(tart, R(un, E(nter, D(ump, H(count, B(kpt, Q(uit?
```

Example2-1

```
ADD     START   1000      ADD FILE FROM INPUT TO OUTPUT
MAIN    LDA     THREE     LOAD CONSTANT 3 INTO REGISTER A
        STA     SAVE1     STORE IN SAVE1
        STA     SAVE      STORE IN SAVE
        JSUB    PRINT     JUMPS TO THE SUBROUTINE
        LDCH    CADD      LOAD CHARACTER “+” INTO REGISTER A
        JSUB    PRINTC    JUMPS TO THE SUBROUTINE
        LDA     TWO       LOAD CONSTANT 2 INTO REGISTER A
        STA     SAVE2     STORE IN SAVE2
        STA     SAVE      STORE IN SAVE
        JSUB    PRINT     JUMPS TO THE SUBROUTINE
        LDCH    CEQU      LOAD CHARACTER “=” INTO REGISTER A
        JSUB    PRINTC    JUMPS TO THE SUBROUTINE
        LDA     SAVE1     LOAD CONSTANT 3 INTO REGISTER A
        ADD     SAVE2     ADD 2 INTO REGISTER A
        STA     SAVE      STORE IN SAVE
        JSUB    PRINT     JUMPS TO THE SUBROUTINE
        J       EXIT      LEAVE
```

Example2-2

```
PRINT    LDCH    CZERO     LOAD CHARACTER “+” INTO REGISTER A
         ADD     SAVE      ADD 2 INTO REGISTER A
LOOP     TD      OUTDEV    Test output device
         JEQ     LOOP      Loop until device is ready
         WD      OUTDEV    Write one byte to output device
         RSUB              LEAVE THE SUBROUTINE
PRINTC   TD      OUTDEV    Test output device
         JEQ     PRINTC    Loop until device is ready
         WD      OUTDEV    Write one byte to output device
         RSUB              LEAVE THE SUBROUTINE
```

Example2-3

```
ONE        WORD   1
TWO        WORD   2
THREE      WORD   3
SAVE       RESW   1
SAVE1      RESW   1
SAVE2      RESW   1
OUTDEV     BYTE   x'05'
CZERO      BYTE   c'0'
CADD       BYTE   c'+'
CEQU       BYTE   c'='
EXIT       END     MAIN
```

Program Result

-
  ```
  %more DEV05
    =>  3+2=5
  ```