

PREVIOUS YEAR QUESTION - SUGGESTION 2021

1. a) What is the utility of the relocation bit?

=> The relocation bit, or R bit, is a bit in the object code of a program that is used to indicate whether or not a particular instruction or data item can be relocated. If *the R bit is set to 1, it indicates that the instruction or data item can be relocated when the program is loaded into memory.* If the R bit is set to 0, it indicates that the instruction or data item cannot be relocated and must remain at its current address. The utility of the R bit is to allow programs to be more flexible and adaptable, and to reduce the size of object code files.

b) Differentiate between Rep and REPT directives with proper examples.

REP and REPT are two different directives that are used in assembly language programming. The REPT directive is used to repeat a block of code a specified number of times, while the REP directive is used to repeat a single instruction a specified number of times.

c) "T0010001E141033" is an example of a Text Record. Write the format of Text Record and explain the above example.

A text record, or T record, is a type of record used in object code files in assembly language programming. It is used to *store the instructions and data that are to be loaded into memory when the program is executed.*

The general format of a text record is as follows -> T<address><length><data>

In this format, "T" indicates that this is a text record, <address> is the starting address of the data in the record, <length> is the length of the data in the record, and <data> is the actual data stored in the record.

For example, the text record "T0010001E141033" can be broken down as follows:

- "T" indicates that this is a text record
- "001000" is the starting address of the data in the record (3 byte hex)
- "1E" is the length of the data in the record (30 bytes)
- "141033" is the data stored in the record (the hexadecimal representation of the data)

In this example, the text record contains 14 bytes of data starting at the address 0x10001E. The data itself is represented in hexadecimal and can be decoded to determine the actual instructions or data stored in the record.

d) What is the function of the following Loader option?

CHANGE RDREC, READ

The CHANGE RDREC, READ option is a loader option that specifies the format of the input records to be loaded by the loader. It is used to indicate that the value of RDREC is to be changed to READ in the external symbol table, for example when the RDREC module is to be replaced by the READ module.

e) Give an example of a one byte instruction available in SIC/XE.

FLOAT , It does the following

$F \leftarrow [A]$, ie value of Accumulator is converted to 48 bit and stored in the F register.

f) How does DLL work on the Windows platform?

A dynamic-link library (DLL) is a file that contains code and data that can be used by multiple programs at the same time. DLLs are an important part of the Windows operating system, as they provide a way for programs to share common code and resources.

When a program needs to use a DLL, it typically calls a function exported by the DLL. The function performs the desired operation and returns a value to the calling program. The DLL can also provide data to the calling program, such as constants and global variables.

To use a DLL, a program must first load it into memory. This is typically done using the LoadLibrary function provided by the operating system. Once the DLL is loaded, the program can call its exported functions using the GetProcAddress function. This function allows the program to obtain the memory address of the exported function, which it can then call system directly.

Overall, DLLs provide a way for programs to share code and resources, improving efficiency and reducing the amount of memory and storage used by the system.

2. What are the advantages and disadvantages of Overlay. Why is an absolute loader is called as a self-destroyed loader? What do you mean by Deferred Linking? 5+4+5=14

- Advantages of Overlay:
 - Allow programs larger than physical memory
 - Reduce memory requirements
 - Reduce Time Requirements
- Disadvantages of Overlay:
 - Overlay map must be specified by the programmers
 - Programmer must know memory requirements

- Overlay module must be completely disjoint.
- Programming design of overlay structure is complex and not possible in all cases.

Absolute Loader is a type of loader in which relocated object files are created, loader accepts those files and places them at specified locations in memory. This is called absolute loader because no relocation information is needed. In this scheme the programmer or the assembler should have knowledge of memory management.

Absolute loader is known as self destroyed loader as when the loader loads the object code in the fixed location, the code of the loader itself is overwritten, when the entire object program is loaded, the loader is destroyed.

In dynamic linking, the compiling and linking code into loadable programs are delayed until run time. This feature is used with system libraries such as languages subroutine libraries. This reduces memory requirements. Due to the delay in linking until runtime, it is also called Deferred Linking.

3. Write a macro to find the maximum of 10 numbers using SIC/8086. What are the differences between positional parameter passing and key word parameter passing in macro? 10+4=14

MaxOfTen Macro A1, N1

MOV CL,N1

MOV CH,00

MOV SI, A1

MOV AL, [SI]

DEC CL

INC SI

J2: CMP AL, [SI]

 JNC J1

 MOV AL, [SI]

J1: INC SI

 DEC CL

 JNZ J2

ENDM

Positional Parameter Passing	Key Word Passing
It must be specified in a particular order every time the macro is called	It can be specified in any order when the macro is called
It has no default values	We can specify default values
All values must be supplied by the caller (if some needs to be omitted/null then put consecutive commas).	If the caller exists, supply any value. If not then pass the default value. If no default value is specified pass Null string is assigned
No naming restrictions	It has equal sign as the last character of parameter name

4. What are advantages and disadvantages of holding symbolic operation codes in a separate symbol table? Write an Assembly language Program to find the number of vowels. 4+10=14

—

Advantages : fast

Disadvantages : Chance of collision if two label has same hash value

—

;8086 PROGRAM: COUNT NUMBER OF VOWELS IN GIVEN LINE OF A TEXT/SENTENCE

.DOSSEG

.MODEL SMALL

.STACK 100H

.DATA

STRING DB 10,13,"The quick brown fox jumped over lazy sleeping dog\$"

VOWEL DB ?

MSG1 DB 10,13,"Number of vowels are: \$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV SI, OFFSET STRING

MOV BL, 00

BACK: MOV AL, [SI]

CMP AL,'\$'

JZ FINAL

CMP AL,'A'

JZ COUNT

CMP AL,'E'

JZ COUNT

CMP AL,'I'

JZ COUNT

CMP AL,'O'

JZ COUNT

CMP AL,'U'

JZ COUNT

CMP AL,'a'

JZ COUNT

CMP AL,'e'

JZ COUNT

CMP AL,'i'

JZ COUNT

CMP AL,'o'

JZ COUNT

CMP AL,'u'

JZ COUNT

INC SI

JMP BACK

COUNT: INC BL

INC SI

JMP BACK

FINAL: MOV AH,2H

MOV DL,BL

INT 21H ;print number of vowels

MOV AH, 4CH

INT 21H

MAIN ENDP

END

5. What are the differences between line editor and screen editor? Explain the differences in user mode and kernel mode in device driver architecture. How UART device driver works in EXINU. 4+5+5=14

Line editor: In a line editor the unit of editing consists of an entire line, ie, to modify even a single character the user has to re-enter the entire line.

Screen Editor: Screen oriented editor allows the use of cursor keys/mouse to point to a position at which the next editing is to be carried out. All modern systems came up with one or more such Screen Oriented Editors such as VIM, Emacs, etc.

Differences between user mode and Kernel mode: Running the device driver in user mode can result in increased stability as a poorly written device driver cannot crash the system by overriding kernel memory. Whereas running the device driver in kernel mode increases the chances of system crash by overriding kernel memory. In order to talk to the kernel, the device drivers registers with subsystems to correspond to the result. The user interface talks to these device drivers using device files. Device file is a mechanism provided by the kernel, explicitly for this direct user-driver interface.

UART : https://xinu.cs.mu.edu/index.php/UART_Driver

6. Consider the following two modules written in SIC/XE assembly language.

ONE	CSECT	TWO	CSECT
	EXTREF B1,B2		EXTREF A1,A2
	EXTDEF A1,A2		EXTDEF B1,B2
	LDA STORE		+LDA A1
	STA VAL		+LDB A2
	+STA B1		STA B1
	LDA A1		STB B2
	STA A2	B1	RESW 1
	+STA B2	B2	RESW 1
VAL	RESW 1		END TWO

A1	RESW 1
A2	RESW 1
STORE	RESW 1
	END ONE

(i) Show the M-records produced by the assembler for the modules ONE and TWO. Each M record should have an address (in decimal), the number of bytes, the flag ('+' or '-') and an appropriate symbol.

(ii) Assume that the starting addresses of modules ONE and TWO are 400 (decimal) and 600 (decimal) respectively. Show the External Symbol Table (EST). All the addresses specified in EST must be in decimal. 4+10=14

Ans :

00	ONE	CSECT
00		EXTREF B1,B2
00		EXTDEF A1,A2
00		LDA STORE
03		STA VAL
06		+STA B1
10		LDA A1
13		STA A2
16		+STA B2

20	VAL	RESW 1
23	A1	RESW 1
26	A2	RESW 1
29	STORE	RESW 1
32		END ONE

Only the instructions with external reference need to be modified
M^000007^05+B1 (Address is 20 bits i.e 5 Half-bytes)
M^000017^05+B2

00	TWO	CSECT	
00		EXTREF A1,A2	
00		EXTDEF B1,B2	
00		+LDA A1	
04		+LDB A2	
08		STA B1	
11		STB B2	
14	B1	RESW 1	
17	B2	RESW 1	
20		END TWO	

M^000001^05+A1
M^000005^05+A2

ESTAB

PROGADDR (ONE) = 400, CSADDR = 400 (ONLY ONE CONTROL SECTION), CSLTH = 32

PROGADDR (TWO) = 600, CSADDR = 400 (ONLY ONE CONTROL SECTION), CSLTH = 20

CONTROL SECTION	SYMBOL	ADDRESS	LENGTH
ONE		0400	32
	A1	0423	
	A2	0426	
TWO		0600	20
	B1	0614	
	B2	0617	

7. Instructions of 8086 can execute parallel. – Justify it. Design a flowchart for one-pass assembler mentioning all data structure and steps clearly. 4+10=14

Ans :

8086 employs parallel processing i.e both BIU(Bus Interface Unit) and EU(Executionion unit) work at the same time. This is unlike 8085 in which sequential fetch and execute operations take place.

The Execution unit gives instructions to BIU starting from where to fetch the data and then decode and execute those instructions. Its function is to control operations on data using the instruction decoder & ALU. EU has no direct connection with system buses as shown in the above figure, it performs operations over data through BIU.

BIU takes care of all data and address transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU has no direction connection with System Buses so this is possible with the BIU. EU and BIU are connected with the Internal Bus

Check flowchart :

 Single-Pass-Assembler.pdf

CLASS TEST OF SYSTEM PROGRAMMING

1. WHAT DO YOU MEAN BY FORWARD REFERENCE AND CROSS REFERENCE.

Forward Reference: Reference to a label/variable that is defined later in an assembly language program. We can use two pass assembler to deal with this problem.

Cross Reference: Also known as external reference? symbols that are used in this control section and are defined elsewhere

→ List of all the places in a program where a particular label is used is Cross Reference.

2. What Alternative Data Structures Can Be For MOT ?

Array [?]

//elaborate ??

3. Write a set of instructions which gets a string response from the keyboard and displays it at row 17 and column 20.

```
.model small
.stack 100h
.data
var1 db 100 dup("$")
.code
main proc

; input string
mov ax, @data
mov ds, ax
mov si, offset var1
11:
mov ah, 1
int 21h
```

```

cmp al, 13
je programend

mov [si], al
inc si
jmp 11

programend:

; Clear screen
mov ax, 0600h
mov bh, 07
mov cx, 0
mov dx, 184fh
int 10h

; Set position
mov ah, 2 ;<■■ SERVICE TO SET CURSOR POSITION.
mov bh, 0 ;<■■ VIDEO PAGE.
mov dl, 20 ;<■■ SCREEN COLUMN 20 (X) .
mov dh, 17 ;<■■ SCREEN ROW 17 (Y) .

; Print string
mov dx, offset var1
mov ah, 9
int 21h
mov ah, 4ch
int 21h

main endp
end main

```

4. How does assembler deal with relocatable address.

Ans: Assembler uses Modification records to deal with relocatable address. Assembler keeps modification records which will denote where in the object program fields will be modified.

Modification record format:

$M^{2-7}(\text{Address where to modify})^{8-9}(\text{length of field to be modified})(+/-)\text{LABEL}$

5. Immediate operands and literals are both ways of specifying an operand value in a source statement. When might each be preferable?

Immediate operands are specified directly in the machine instruction, while literal values are stored in memory and the address of this value is provided in the instruction. For every time this constant is used the same address is used. This address mapping is stored in the LITTAB.

In some cases the immediate operand might be large in size and we might need to use a type 4 instruction. However in case of literal operand, we need only use the address which might need fewer bits and consequently a shorter instruction size.

6. Compare relative advantages and disadvantages for one pass and two pass assembler.

Pros of one pass:

1. It is faster than two pass as it generates executable in a single pass.
2. Easy to write and implement due to its simple design as compared to two pass.

Cons of one pass:

1. Forward reference problem exists. In this problem operands are utilized before declaration, this assembler has no info about the operand address this is to be present in the final executable code.
2. All errors are not detected

Pros of two pass:

1. Solved forward reference problem by using first pass to generate a system table.
2. All errors are detected. Simple errors are solved in the first pass, complex errors are solved in the second pass.

Cons of two pass:

1. Slower due to the second pass.
2. Design and implementation are more sophisticated as compared to single pass.

7. What are different addressing modes of 8086 ?

Immediate addressing mode:

The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.

Example

```
MOV CX, 4929 H,  
ADD AX, 2387 H,  
MOV AL, FFH
```

Register addressing mode

It means that the register is the source of an operand for an instruction.

Example

```
MOV CX, AX ; copies the contents of the 16-bit AX register into  
           ; the 16-bit CX register),  
ADD BX, AX
```

Direct addressing mode

The addressing mode in which the effective address of the memory location is written directly in the instruction.

Example

```
MOV AX, [1592H],  
MOV AL, [0300H]
```

Register indirect addressing mode

This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI.

Example

```
MOV AX, [BX] ; Suppose the register BX contains 4895H, then the contents  
             ; 4895H are moved to AX  
ADD CX, {BX}
```

Based addressing mode

In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.

Example

```
MOV DX, [BX+04],  
ADD CL, [BX+08]
```

Indexed addressing mode

In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.

Example

```
MOV BX, [SI+16],  
ADD AL, [DI+16]
```

Based-index addressing mode

In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.

Example

```
ADD CX, [AX+SI],  
MOV AX, [AX+DI]
```

Based indexed with displacement mode

In this addressing mode, the operands offset is computed by adding the base register contents. An Index registers contents and 8 or 16-bit displacement.

Example

```
MOV AX, [BX+DI+08],  
ADD CX, [BX+SI+16]
```

8. What is a bootstrap loader ? How does it work?

When a computer is started up or restarted or simply booted, a special absolute loader called bootstrap loader is executed. The bootstrap loads the first program to be run by the program, usually OS. The bootstrap begins at address 0. It loads the OS in starting address 0x80 from the F1 device. No header record or control information, the object code is allocated consecutive bytes of memory. After loading

the OS, it jumps to location 0x80 to run the operating system.
(from slide):

3.4.3 Bootstrap Loaders

- An absolute loader program is permanently resident in a read-only memory (ROM)
 - Hardware signal occurs
- The program is executed directly in the ROM
- The program is copied from ROM to main memory and executed there.

71

3.4.3 Bootstrap Loaders


- Bootstrap and bootstrap loader
 - Reads a fixed-length record from some device into memory at a fixed location.
 - After the read operation is complete, control is automatically transferred to the address in memory.
 - If the loading process requires more instructions than can be read in a single record, this first record causes the reading of others, and these in turn can cause the reading of more records.

9. Compare similarities and dissimilarities between linkage editor and loader-linker.

1. Linkage editor needs relocating loader to load the machine code into memory. But loader-linker loads the code into memory itself.
2. Both linkage editor and loader-linker are used to link object code after assembling.
3. **Linkage editor produces a linked object file. Loader-linker produces no such file.**

Linking Loader	Linkage Editor
Performs all linking and relocation operations, including automatic library search, and loads the linked program into memory for execution.	Produces a Linked version of the program, which is normally written to a file or library for later execution.
Suitable when a program is reassembled for nearly every execution.	Suitable when a program is executed many times without being reassembled.
Resolution of external reference and library searching is performed more than once.	Resolution of external reference and library searching is performed only once.
Linking loaders perform linking operations at load time.	Linkage editors perform linking operations before the program is loaded for execution
There is no need of relocating loader.	The relocating loader loads the load module into the memory
The Loading may requires two passes.	The Loading can be accomplished in one pass.
When program is in development stage then at that time the linking loader can be used.	When the program development is finished or when the library is built then linkage editor can be used.

Some resources:

 Page495-499-Appendix-A.pdf