

Assignment No: Group 13: - 2 (Pass & Macropercessol)

- Title: Design suitable data structure and implement pass-1 Macroprocessor of a two pass Macroprocessor for pseudo-machine. The output of Pass-I (MNT, MDT and intermediate code file without any macro-clefinations) Uhowed be input for Pass-2 Hacroproces.
- Software / Hardware Requirement:

 + Software requirement:

 i) Java Deulopment kit.
 - - 2) IDE 3) OR Notepad.
 - + Hordware requirement

 - 1) Computer System. Puocessor: 15 9th Gen Ram: 8GB.
 - 2) I/O peripherals like keyboard & Mouse 3) Monitor: 720p/1080p fHD/IPS.
- Learning Objective.
 - 1) To understand what is a Macro and Ma du processor.
 - 2) Identify and unders tand two pass haveoprocessor.



4) Learning Outcome.

- i) One should be able to unclerstand what is a Macres and Mario processor.
- 2) one should be able to implement pass 1.

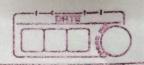
5) concept Related to Theory:

A macus instruction is the potational convenience for the programmer. For every occurrence of macro the whole macro body or macro block of statement gets explained expanded in the main source code. Thus Macro instructions make writing code more convenient.

Salient features of Macro Processor:

- i) It represents a group of commonly used state.

 men's in the source programming language
- ii) Movis processor suplaces each macro instructions with the corresponding group of source longuage statements. This is known as the expansion of Macros.
- the mechanical details to be handled by the macro processor.



iv) races processor involves defination, invocation. and expansion

* Macro reginition and Expansion:

Line Label Opcode
5 COPY START RDBUFF MACRO

90 95

MEND

Here

L'ine 10

MACIO O'L NAME OF MOCIO definition of the Macio. EINDEV & BUFADR one the parameters present in the operand part. Each parameter regin with character E.

From Line 15 to line 90

From Line 15 to Line 90 Macro Body us

prejent. Macro directives are the statements that make up the body of the macuo definition.

L'ne 95:

MEND is the assembler directive that means the end of the macro definition.



Marcho	11	110	MY	1119	1	9
		VV	600	* 1 67		

Line Label opcode

180 first STL

190 Cloop RDBUFF

15

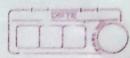
255

END

Here,

L'ine 190

RDBUFF is the Macro invocation or nacro call that gives the name of the macro instruction being invoked and FI, Buffer one the organients to be used in expanding the macro. The statement that form the expansion of a macro are generated each time the macro is invoked.



	The state of the s					
6) +	Algorithm for Possf-Macroprocessor.					
1)	Initialization of counters ofor MDT & MNT.					
2)	Read Next 3 n spuictions (and clivide it into it's					
	various field as Labl, mnemonic (opcode arguments).					
3)	theck for mocro definition start					
	if apcode = MACRO goto Step 5 else to step 4.					
4)	a) white copy of instruction to output of Pass-I					
	b) check whether opcode = END or not!					
	c) if OPCODE "* END goto 8 tep 2					
	0) if OPCODE = END goto POSS-2 i.e End of this algorithm					
	for Pass-1.					
5)	Start of Macro appinition is identified. Now Pass-1					
	will process content of Macro objinition ofter					
	Aseudo of MACRO to MEND *1					
6)	Process other instructions in macro definition					
	including MEND.					
7) *	The idota structures associated with Pass-I Mouropro-					
	aros.					
1)	MDT (Macus Definition Table)					
	Inclex MDT-Instruction					
2)	MNT (Macro Name Table)					
	Inclex Macro-Name MDT-Index					

~	3) Augument List Arriay (ALA)
	x Index Macro-Name MDT-Index.
	Index Dummy Argument
8)	Conclusion:
	understood the working of Poss I Mours. processor.
9)	Repevences:
1 - 2 - 2	1) Geeks for Geeks. 2) SCRIBD 3) Youtube/ Gonesk Kakele.
	s) pearage, genes, rescue
0.0002004.204.2	Literary Arity Apropries are a series where we are
	Copper State of the Copper