Name: KEY Id#

COE 205, Term 062

Computer Organization & Assembly Programming

Quiz#5

Date: Saturday, May 12, 2007

Q1. Compare **Macros** and **Procedures** in terms of both memory space and execution time.

Macro invocation is done *at assembly time* by text substitution. However, procedure invocation is done *at run time* by transferring control to the procedure. This leads to the following tradeoff. *Using macros* results in faster execution of the code. However, macros result in increased memory space due to macro expansions. Procedures save space, as only one copy of the procedure is kept. However, procedure invocation overhead (to pass parameters via the stack and for call/ret) increases the execution time. Note that macro invocation causes assembly-time overhead but not run-time overhead. The advantages and disadvantages associated with macros and procedures can be summarized into the following table:

Type of Overhead	Procedure	Macro
memory space	lower	higher
Execution time	higher	lower
Assembly time	lower	higher

Q2. Write a macro than can be used to save <u>any number</u> of registers on the stack by specifying the registers to be saved as arguments. Then, show how the macro can be invoked to save the registers EAX, EBX, and ECX.

SAVE REGS < EAX, EBX, ECX >

Q3. Given the following macro definition:

```
GET_BIG MACRO WORD1, WORD2

LOCAL EXIT

MOV AX, WORD1

CMP AX, WORD2

JG EXIT

MOV AX, WORD2

EXIT:

ENDM
```

Determine the macro expansion resulting from the following macro invocation assuming that FIRST and SECOND are word variables.

GET_BIG FIRST, SECOND

MOV AX, FIRST
CMP AX, SECOND
JG ??0000
MOV AX, SECOND

??0000: