|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REG Name** | **Type** | **Special Uses** | **Comments** | Largest 32b |
| EAX, AX, AH, AL | General | Result of add, dividend in DIV, and quotient= AL, Remainder= AH | MUL:  Mutiplicand, AL, AX, EAX  Product Etx= AX, DX:AX, EDX:EAX | Signed Int  (2^31) - 1 |
| EBX, BX, BH, BL | General | None |  | $-newarray |
| ECX, CX, CH, CL | General | Loop counter |  | Div by type |
| EDX, DX, DH, DL | General | DIV Remainder =DX or EDX | MUL: Product extension = EDX | Size = #elm. |
| EBP, BP | Multi | Extended Frame Pointer | **STATUS FLAGS** | **Instruction=** |
| ESP, SP | Multi | Extended Stack | Carry (CF)- unsigned is too large | Label, Mn, |
| ESI, SI | Multi | Ext Source index | Overflow (OF)- signed is too large | Operand, |
| EDI, DI | Multi | Ext Destination index | Sign (SF)- result = neg | comment |
| EFLAGS | Special | Can’t be directly accessed | Zero(ZF)- result = 0 |  |
| EIP(instruction pointer) | Special | Can’t be directly accessed | Auxiliary Carry (AC)- from bit 3-4 |  |
| CS, ES, SS, FS, DS, GS | Segment (16big) |  | Parity (PF)- LSB = even 1 bits |  |
| (unsigned) JA = lo>ro, JAE | JB = lo<ro, JBE | (Signed JG) = lo > ro, JGE | JL = lo < ro, JLE ;;;;;;; **JE = lo = ro** | **JNE= lo !=ro** |
| **CMP** implied subtraction | D < S CF-1 | D > S CF-0; D=S ZF-1 | (signed) D<S SF!=OF; D>S SF=OF | D=S ZF-1 |

XCHG-REG/REG, MEM/REG, REG/MEM *NO!!! MEM/MEM*

MOVSX- extended sign for sized, MOVZX- extend sign for unsigned

Gotoxy- Move cursor position

DumpMem, displays all registers

GetMseconds- number of system milliseconds elapsed since midnigth

ESP- Modified by PUSH, POP, CALL, RET

WRITEDEC-unsigned (From EAX

RandomRange- produces random number

PROC USES registers (pushes named registers then pops at the end of the PROC)

READSTRING (pre- OFFSET IN EAX)

WriteInt-signed (from EAX)

CALL- moves address to ESP

AND – 1+1 = 1; use it to clear bits where ever the zeros are

OR 1+0=1 0+1=0 1+1 = 0; use it to set bits use 1 to set

XOR 1+1=0

IMUL- product is twice as big, upper register carries the sign

SINGLE FLOAT: S= 1 E=8 Sig= 23

Double Float: s= 1 E=11 Sig = 52

HAMMING CODE- Check 1 Skip 1, Check 2, Skip 2, Check 4, Skip 4, Check 8, Skip 8

ENTER numbytes, nestinglevel – pushes EBP, mov EBP, ESP and adds space for local variables

LEAVE mov ESP, EBP then pop EBP

LOCAL temp:DWORD or wArray[50]:SWORD, push EBP, mov EBP, ESP, adds space for the local variables the to ESP, leave

HOW TO MAKE A STACK FRAME

1. Passed arguments if any are pushed onto the stack
2. The subroutine is called and it’s address is pushed (CALL)
3. As it begins EBP is pushed
4. EBP is set equal to ESP
5. ESP is decremented to leave space for local variables
6. If any registers need to be saved they are pushed onto the stack (PUSHAD)

HOW TO CLEAN THE STACK

C convention – add total of variable space to ESP

STDCALL write RET (size of local variables) before the end

TITLE adds a title to your source file

WATCH FLOATING POINT