|  |  |
| --- | --- |
| BYTE, SBYTE | 8-bit unsigned integer; 8-bit signed integer |
| WORD, SWORD | 16-bit unsigned & signed integer |
| DWORD, SDWORD | 32-bit unsigned & signed integer |
| QWORD | 64-bit integer |
| TBYTE | 80-bit integer |
| REAL4 | 4-byte IEEE short real |
| REAL8 | 8-byte IEEE long real |
| REAL10 | 10-byte IEEE extended real |

CS

SS

DS

ES

EIP

EFLAGS

**16-bit Segment Registers**

EAX

EBX

ECX

EDX

**32-bit General-Purpose Registers**

FS

GS

EBP

ESP

ESI

EDI

AH

AL

16 bits

8

AX

EAX

8

32 bits

8 bits + 8 bits

### Jumps Based on Specific Flags

|  |  |  |
| --- | --- | --- |
| **Mnemonic** | **Description** | **Flags** |
| JZ | Jump if Zero | ZF == 1 ( set ) |
| JNZ | Jump if Not Zero | ZF == 0 ( not set ) |
| JC | Jump if Carry | CF == 1 ( set ) |
| JNC | Jump if Not Carry | CF == 0 ( not set ) |
| JO | Jump if Overflow | OF == 1 ( set ) |
| JNO | Jump if Not Overflow | OF == 0 ( not set ) |
| JS | Jump if Signed | SF == 1 ( set ) |
| JNS | Jump if Not Signed | SF == 0 ( not set ) |
| JP | Jump if Parity is Even | PF == 1 ( set ) |
| JNP | Jump if Parity is Not Even (i.e. odd) | PF == 0 ( not set ) |

### Jumps Based on Equality

|  |  |
| --- | --- |
| **Mnemonic** | **Description** |
| JE | Jump if Equal ( leftOp == rightOP ) |
| JNE | Jump if Not Equal ( leftOp != rightOP ) |
| JCXZ | Jump if CX == 0 |
| JECXZ | Jump if ECX == 0 |

***Jumps Based on Unsigned Comparisons – All have either an A or a B in the Mnemonic***

* **A(bove)**
* **B(elow)**

|  |  |
| --- | --- |
| **Mnemonic** | **Description** |
| JA | Jump if Above ( leftOp > rightOp ) |
| JNBE | Jump if Not Below or Equal (same as JA) |
| JAE | Jump if Above or Equal ( leftOp >= rightOp ) |
| JNB | Jump if Not Below (same as JAE) |
| JB | Jump if Below ( leftOp < rightOp ) |
| JNAE | Jump if Not Above or Equal ( same as JB ) |
| JBE | Jump if Below or Equal ( leftOp <= right Op ) |
| JNA | Jump if Not Above ( same as JBE ) |

***Jumps Based on Signed Comparisons – All have either a G or a L in the Mnemonic***

* **G(reater)**
* **L(ess)**

|  |  |
| --- | --- |
| **Mnemonic** | **Description** |
| JG | Jump if Greater ( leftOp > rightOp ) |
| JNLE | Jump if Not Less or Equal (same as JG) |
| JGE | Jump if Greater or Equal ( leftOp >= rightOp ) |
| JNL | Jump if Not Less (same as JGE) |
| JL | Jump if Less ( leftOp < rightOp ) |
| JNGE | Jump if Not Greater or Equal ( same as JL ) |
| JLE | Jump if Less or Equal ( leftOp <= right Op ) |
| JNG | Jump if Not Greater ( same as JLE ) |

***Example: Jump to a label if Unsigned EAX is greater than EBX***

* **Solution**: Use CMP, followed by JA

cmp eax,ebx

ja Larger

***Example: Jump to a label if Signed EAX is greater than EBX***

* **Solution**: Use CMP, followed by JG

cmp eax,ebx

jg Larger

***Example: Jump to label L1 if Unsigned EAX is less than or equal to Val1***

* **Solution**: Use CMP, followed by JBE

cmp eax,Val1

jbe L1

***Example: Jump to label L1 if Signed EAX is less than or equal to Val1***

* **Solution**: Use CMP, followed by JLE

cmp eax,Val1

jle L1

### Library Procedures Included with the Irvine Assembler Library

|  |  |
| --- | --- |
| **CloseFile** | Closes an open disk file |
| **Clrscr** | Clears console, locates cursor at upper left corner |
| **CreateOutputFile** | Creates new disk file for writing in output mode |
| **Crlf** | Writes end of line sequence to standard output |
| **Delay** | Pauses program execution for n millisecond interval |
| **DumpMem** | Writes block of memory to standard output in hex |
| **DumpRegs** | Displays general-purpose registers and flags (hex) |
| **GetCommandtail** | Copies command-line args into array of bytes |
| **GetMaxXY** | Gets number of cols, rows in console window buffer |
| **GetMseconds** | Returns milliseconds elapsed since midnight |
| **GetTextColor** | Returns active foreground and background text colors in the console window |
| **Gotoxy** | Locates cursor at row and column on the console |
| **IsDigit** | Sets Zero flag if AL contains ASCII code for decimal digit (0–9) |
| **MsgBox, MsgBoxAsk** | Display popup message boxes |
| **OpenInputFile** | Opens existing file for input |
| **ParseDecimal32** | Converts unsigned integer string to binary |
| **ParseInteger32** | Converts signed integer string to binary |
| **Random32** | Generates 32-bit pseudorandom integer in the range 0 to FFFFFFFFh |
| **Randomize** | Seeds the random number generator |
| **RandomRange** | Generates a pseudorandom integer within a specified range |
| **ReadChar** | Reads a single character from standard input |
| **ReadFromFile** | Reads input disk file into buffer |
| **ReadDec** | Reads 32-bit unsigned decimal integer from keyboard |
| **ReadHex** | Reads 32-bit hexadecimal integer from keyboard |
| **ReadInt** | Reads 32-bit signed decimal integer from keyboard |
| **ReadKey** | Reads character from keyboard input buffer |
| **ReadString** | Reads string from standard input, terminated by [Enter] |
| **SetTextColor** | Sets foreground and background colors of all subsequent console text output |
| **StrLength** | Returns length of a string |
| **WaitMsg** | Displays message, waits for Enter key to be pressed |
| **WriteBin** | Writes unsigned 32-bit integer in ASCII binary format. |
| **WriteBinB** | Writes binary integer in byte, word, or doubleword format |
| **WriteChar** | Writes a single character to standard output |
| **WriteDec** | Writes unsigned 32-bit integer in decimal format |
| **WriteHex** | Writes an unsigned 32-bit integer in hexadecimal format |
| **WriteHexB** | Writes byte, word, or doubleword in hexadecimal format |
| **WriteInt** | Writes signed 32-bit integer in decimal format |
| **WriteString** | Writes null-terminated string to console window |
| **WriteToFile** | Writes buffer to output file |
| **WriteWindowsMsg** | Displays most recent error message generated by MS-Windows |