Chapter 1: Instructions

# Instruction Word Format

Almost all instructions follow the same 2-byte opcode and addressing mode format. This does not account for displacements. The only exceptions are instructions that require no registers, displacements, and only have 1 addressing mode, NOP, BRK, and HLT.

|  |  |  |  |
| --- | --- | --- | --- |
| 16 8 | 7 6 | 5 3 | 2 0 |
| Op Code | Mod | R/m | R/m |
| 8 | 2 | 3 | 3 |

The first byte is reserved for the opcode. This is to reserve space for at least 256 instructions. The next 2 bits are the Mod, or mode. It has 4 possibilities. The rest of the byte is split between 2 R/m’s. These can me registers, or memory. There are 8 registers, and 8 displacement modes.

Modes:

|  |  |
| --- | --- |
| ID | Summary |
| 00 | Register and Register |
| 01 | Register and Displacement |
| 10 | Displacement and Displacement |
| 11 | Displacement and Displacement 8-bit Mode |

There are 8 general purpose registers, and the EDS (displacement shift) register as a program portability register, for the assembler. There are also other registers for holding values, operations, etc., but you cannot modify them with machine code, so they are not documented. For a quick reference of what they are, there is an Instruction Register, Mod Register, Address Register, and Value Register.

Register Table:

|  |  |  |
| --- | --- | --- |
| ID | Name | Type |
| 000 | EAX | 32 General |
| 001 | EBX | 32 General |
| 010 | ECX | 32 General and Loop |
| 011 | EDX | 32 General |
| 100 | ESI | 32 General |
| 101 | ESP | 32 General |
| 110 | EDI | 32 General |
| 111 | EBP | 32 General |

Displacement Modes: (Note that the AddrX modes only fetch a byte at the address if 8-bit mode)

|  |  |  |
| --- | --- | --- |
| ID | Name | Summary |
| 000 | Disp32 | Uses next 4 bytes |
| 001 | Disp32 + EDS | Uses next 4 bytes plus EDS |
| 010 | Partial | Uses 4 bytes, half for each displacement |
| 011 | Addr32 | Uses 4 bytes at 32-bit address |
| 100 | Addr32 + EDS | Uses 4 bytes for the address, the value plus EDS |
| 101 | Addr32, EDS | Uses 4 bytes added with EDS as the address |
| 110 | Indirect | Indirect (Jump Table) address to address to value |
| 111 | Indirect, EDS + EDS | Indirect with EDS added to both addresses |

The EDS register is used here when adding to the displacement. This is so the assembler can create code with jump statements that can be placed anywhere. At the beginning of the program the EDS register would be set to the current Program Counter address, and not be changed until the program is executed.

# Internals and Memory

## Memory

The CPU gets connected to a Virtual RAM, with 32-bit addresses. The memory can be written by loading a portable file into memory starting at an address. The CPU handles memory mapping automatically.

## Internals

The CPU is the biggest part, but it communicated with other components such as an LCDE. It is a instance based system with a central stack so you can have multiple cores. The central component manages the distribution of RAM and disk space.

# Instruction Set Reference

With the current architecture there are 256 possible instructions, but this uses a RISC similar system where most instructions use an extra byte to get more information on addressing modes etc. The addressing modes are listed in section 1.1, so only the main job of each instruction will be listed, unless there is an exception on how the instruction interacts.