

Mi-CoreX32

**CLI OS**

Specification Manual

*(TOC will be filled once all chapters are finalized)*

Chapter 1

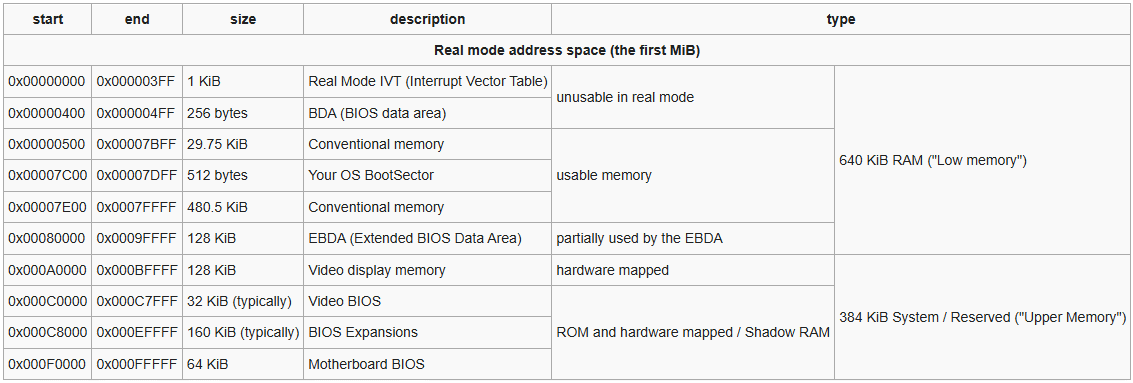
About this Manual

The Micro-CoreX32 (Mi-CoreX32) *Specification Manual* contains the majority of the Operating System - Command Line Interface (CLI) Functions, Routines, Drivers, and Application Programming Interface (API). This was created solely for a hobby and learning how Computer Operating System works. For now, the focus is creating a full working OS (CLI). This also contains multiple memory manipulations, video manipulations, and others. ~~This OS (CLI) uses A20 Line to use the memory above 1MiB of the ram.~~

Chapter 2

Memory Mapping

Memory Mapping contains multiple sections that will be explained below. This is crucial for the OS (CLI) to work flawlessly. It’s sectioned by the system mapping (Fixed Memory Partitions). It contains various data including BIOS, Video Memory, Etc.

System Mapped Partitions (REAL MODE)

← Note2

← Note1

Figure 1.

System Memory Zones — Conventional to Low Memory Address

The Table above should explain every aspect of each Memory Zones. This only affect the first 1MiB of the RAM. Above 1MiB have different allocations.

Note(s)

Note1 : This Conventional Memory region contains the critical API of the OS (CLI) andthese are the Keyboard Functions, Video Functions, and Disk Access Functions.

Note2 : This Conventional Memory region contains the most critical or technically the main program of the whole OS (CLI).

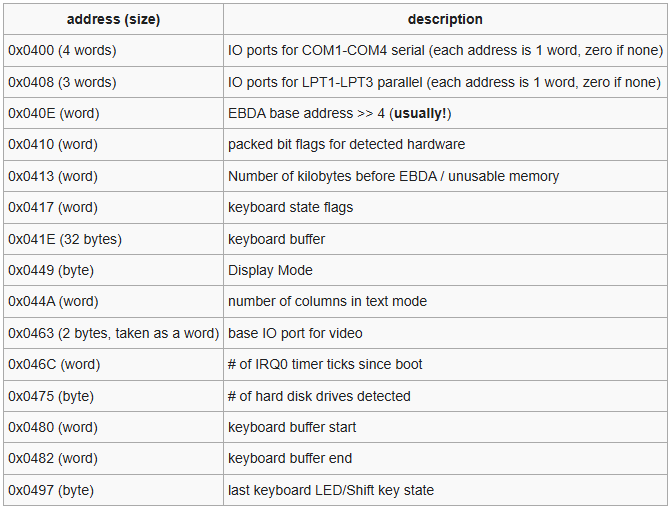
Additionally, there is this important area that is needed to be taken care of, it is the BIOS Data Area (BDA). This contains the information about the motherboard and installed devices. This is supposed to be larger and possibly eat more space when the device is a Laptop. Below is the partial/full information about the BDA. (Note that this may differ vendor to vendor)

Figure 2.

BIOS Data Area (BDA)

Chapter 3

Essential I/O Subsystems

The Essential I/O Subsystem (EOIS) is similar to BIOS but with better compatibility on the Mi-Core OS (CLI), These drivers are placed with fixed address just right above the Bootloader segment (0x7C00).

The following drivers and routines form the backbone of early hardware control, ensuring that the OS can function independently of BIOS after boot (minimal explanation):

* Disk Function Protocol (DFP)

Handles sector-based disk reads/writes through LBA or CHS addressing, replacing BIOS Int 13h.

* Video Services Routines (VSR10)

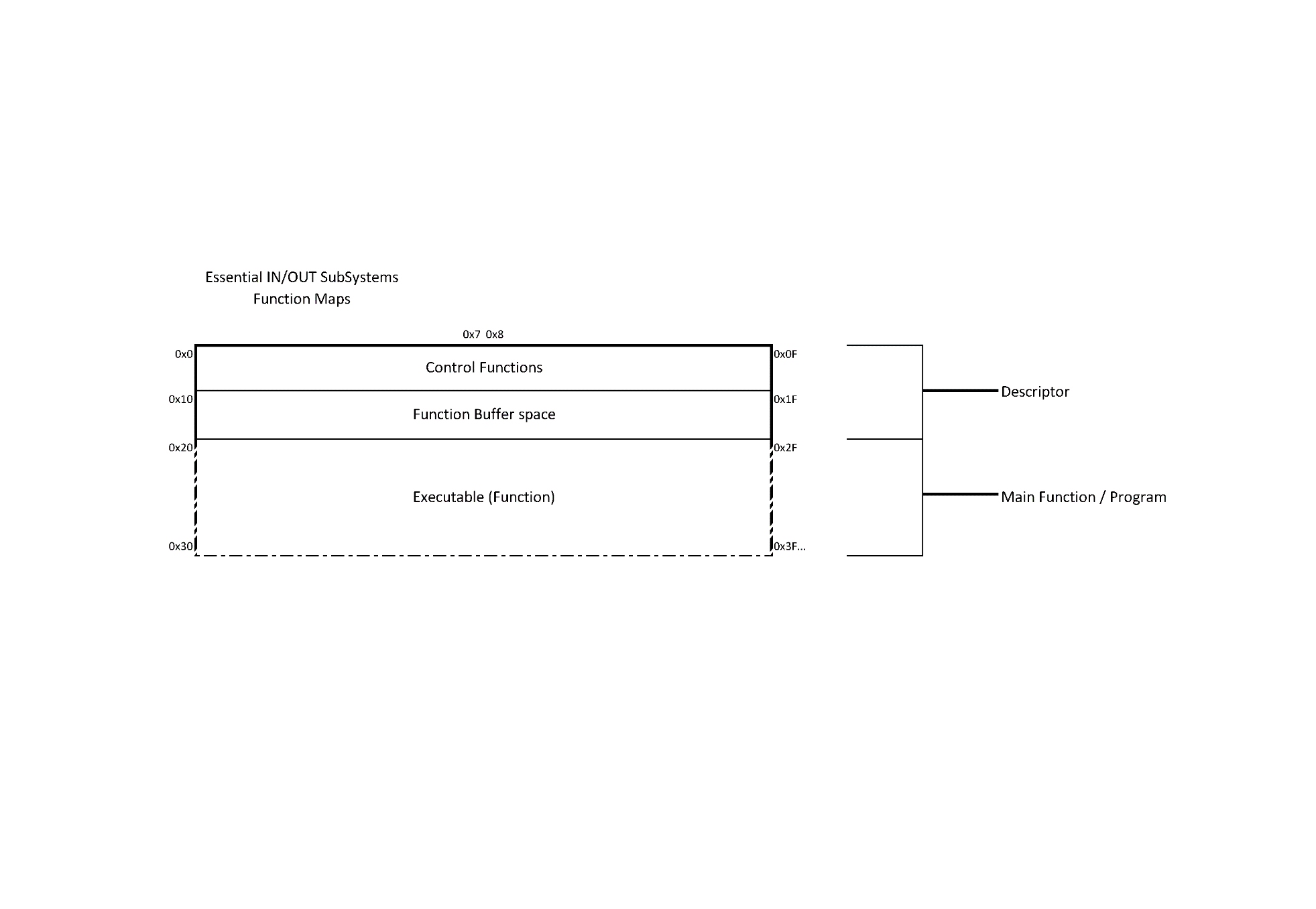
Handles the Video Services, similar functions to INT 0x10 thus the name VSR10. Although its similar. The core functionality of the settings is different.

* Keyboard Services (KBS)

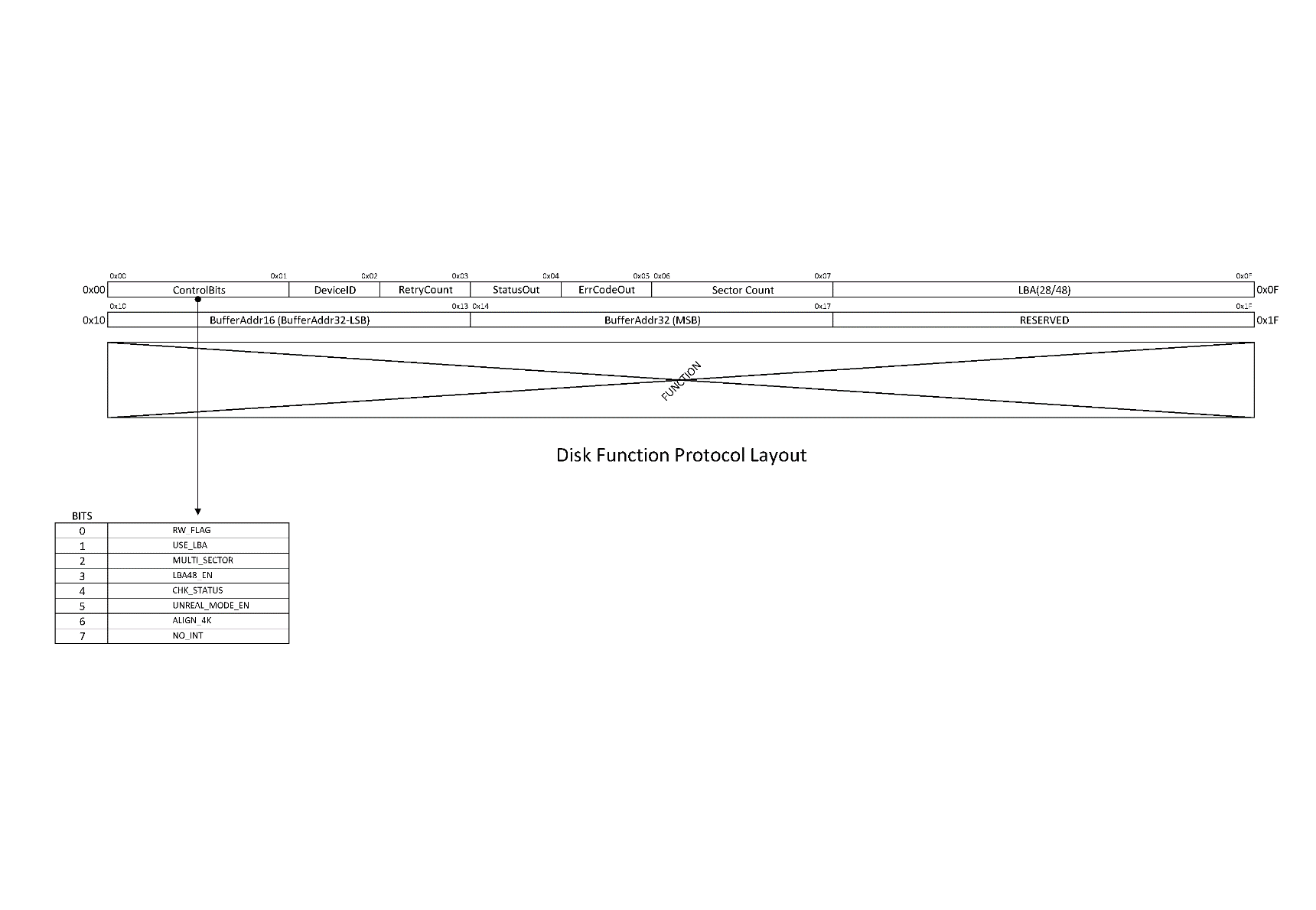
This is very similar to BIOS functions; this might be 1:2 similarity. It GPR instead of Shadowed Buffered GPR.

First of all, this explain how and what are the reserved space and how to use the functions. Each entry has a starting segment or address, the map below will show what are and how it is structured:

Figure 3.

EOIS-Function Map

Disk Function Protocol (DFP)

Base Address in Memory (FLAT) : 0x0000\_0500