

# MIPS

## Memory Map and the MIPS Privileged Resource Architecture

[www.mips.com](http://www.mips.com)

This is the Memory Map section of the MIPS software training course.

## Kernel Versus User Mode

### ▪ Kernel Mode

- Set at startup and on exception
- All address are accessible
- All instructions are permitted (processor configuration and status)

### ▪ User Mode

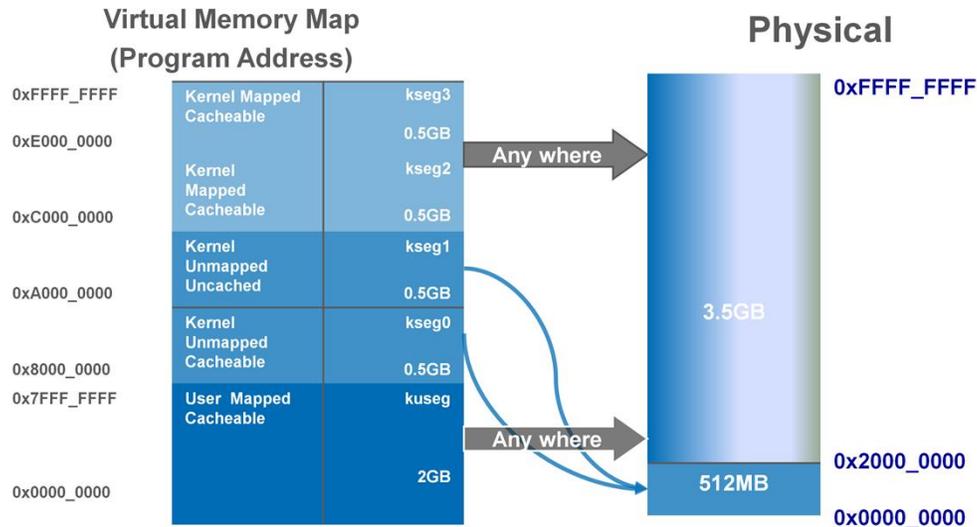
- Set by KSU field in the Status Register
- In 32 bit mode Program Addresses above 2GB (top bit set) are illegal and will cause a exception.
- In 64 bit mode Program Addresses above 0x0000 FFFF FFFF FFFF are illegal and will cause a exception.
- CPU control instructions are illegal



The CPU cold boots in Kernel Mode and is in Kernel mode when exception processing starts. In kernel mode everything is allowed, all addresses are accessible and all instructions are permitted.

The CPU is in User mode when the KSU field in the CP0 status register is set for User Mode. This is usually done by an OS that is running in kernel mode when it starts a user process running. In user mode programs can access program address between 0 and 2GB only. These address can be mapped to any physical address through the TLB. Instructions the read or write TLB entries and Co processor control registers or control the Caches are illegal.

# Memory Map With TLB MIPS32



MIPS

5

There are different memory mappings depending on the state of the system and if you have a TLB or Fixed mapping MMU. I will cover all of the different mappings in this section. We going to start with the most common map, the map of Virtual memory after the boot process has completed.

The virtual memory map is broken down into segments. These segments differ in three characteristics:

Whether access to an address is mapped; that is, the address is virtual and is passed through the translation lookaside buffer (TLB) to translate the virtual address into a physical address

Whether an address can be accessed when the CPU is operating in user mode or in kernel mode

Whether access to an address is cached; that is, looked up in the primary and secondary caches before it is sent to main memory

+ Here is a picture of the Virtual Memory map.

The first thing to note is all program address for both Kernel or user modes are virtual address. Also all address are in hexadecimal.

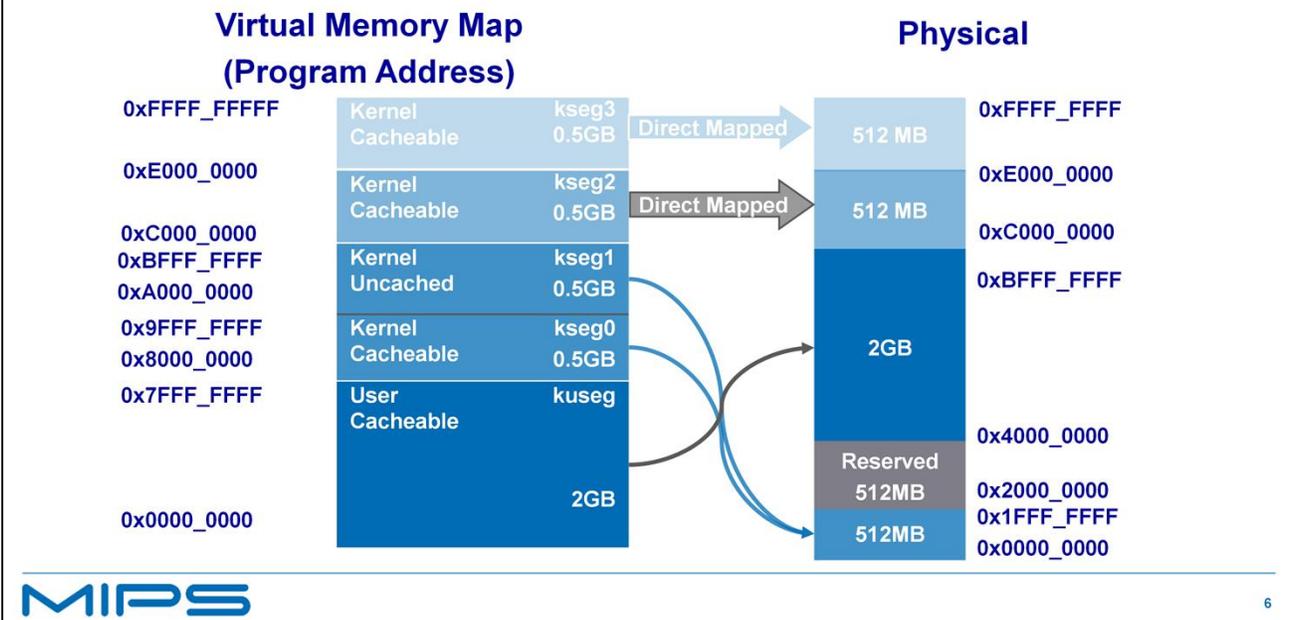
+On a MIPS32 Core the memory map covers a 4 gigabyte range of Physical memory.

+The lower segment of memory, kuseg, covers a 2 gigabyte virtual address range starting at 0. It can be cached and mapped using the TLB to anywhere in physical memory. This segment can be accessed in Kernel or user mode.

+The next two sections kseg0 and kseg1 are designed to be use for the OS code and data. These segments can only be accessed in Kernel mode. If the processor is in User Mode any accesses to the kernel segment it will cause an address error exception. Both Kseg0 and Kseg1 sections are directly translated to the same lower 512 megabytes of physical memory. For example address 80 million and A0 million both are directly mapped to physical address 0. The difference between the two is, Kseg0 addresses are cacheable and can be used once the cache has been initialized. Kseg1 address are not cached and are used at boot time and for memory mapped I/O.

+The next two segments cover the virtual address range that runs contiguously from C0 million to FF FF FF FF hex and is accessible only in Kernel mode. The region is divided into two halves called kseg2 and kseg3, kseg2 was designed to be accessible to programs running in supervisor mode but supervisor mode is seldom, if ever, used. An OS such as Linux concatenates these segments together and refers to it as kernel high memory.

# Memory Map With Fixed MMU MIPS32



You may choose to use a Fixed Memory Translation. For FMT the memory translation is not dynamic but static. Instead of just Kseg 0 and 1 being mapped to a specific physical location all virtual address are mapped to specific physical locations.

+The Kernel and User Segment is fixed mapped to

+Physical memory area starting at 40 million.

+ The two Kernel segments 0 and 1 are always Fixed mapped to

+The lower 512 megabytes of Physical Memory

+The 2 high memory kernel segments are

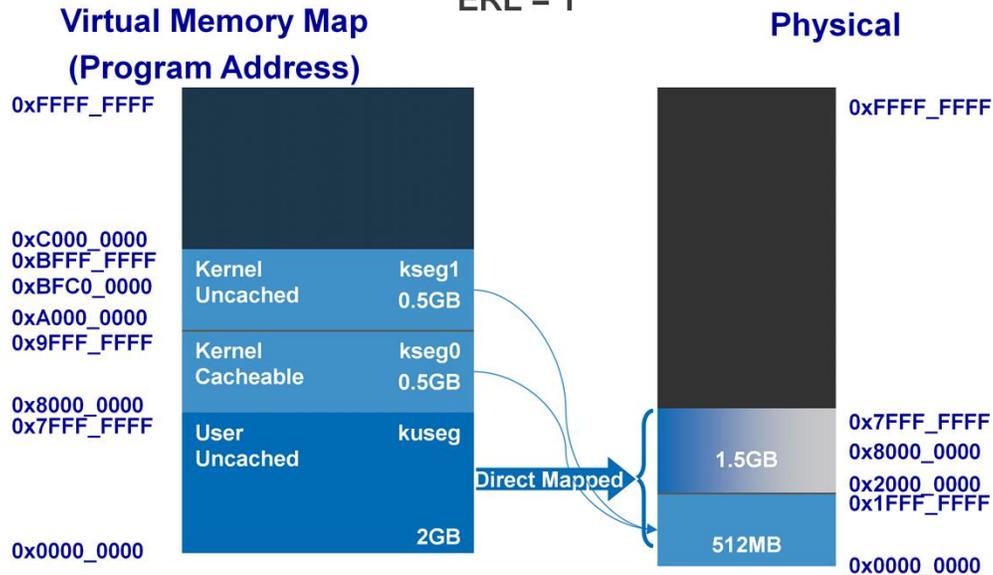
+ directly mapped to the Physical memory

+There is an inaccessible 512 megabyte Physical memory address area

starting at 20 million

# Memory Map Reset, NMI or Cache Error MIPS32

ERL = 1



MIPS

7

This is the memory map at boot time, before the Cache or TLB is initialized or when a NMI or Cache error exception occurs.

+The boot code must be in Kseg1 which as always is

+ fixed mapped to the lower 512 megabytes of physical memory. This segment contains the default boot exception vector BF C0 00 00 which is where the CPU will start fetching instructions at power up or reset.

+Once the Cache has been initialized the code can execute from Kseg0 which will access the same lower 512 megabytes of memory through the cache.

+The kernel can also access the lower 2 gigabytes of virtual Memory which directly maps

+ to the lower 2 gigabytes of physical memory this segment is also uncached.

KSEG2/3 will be mapped through the TLB. Note: On NMI all exception vectors to revert to their cold boot locations within your boot ROM so TLB refills will use page tables setup for booting and not the page tables that

the OS was using!

At this point you can see the virtual addresses 0, 80 million and A zero million all point to physical address 0.