An Updated Emulated Architecture to Support the Study of Operating Systems

µMPS3

Mattia Biondi

Alma Mater Studiorum · Università di Bologna
Scuola di Scienze
Corso di Laurea in Informatica

27 Maggio 2020

What

An educational computer system architecture based around the MIPS R2/3000 microprocessor featuring

- a front-end emulator
- a set of I/O devices

When

- 1999 MPS
- 2004 µMPS
- 2011 µMPS2
- **2020 µMPS3**

Where

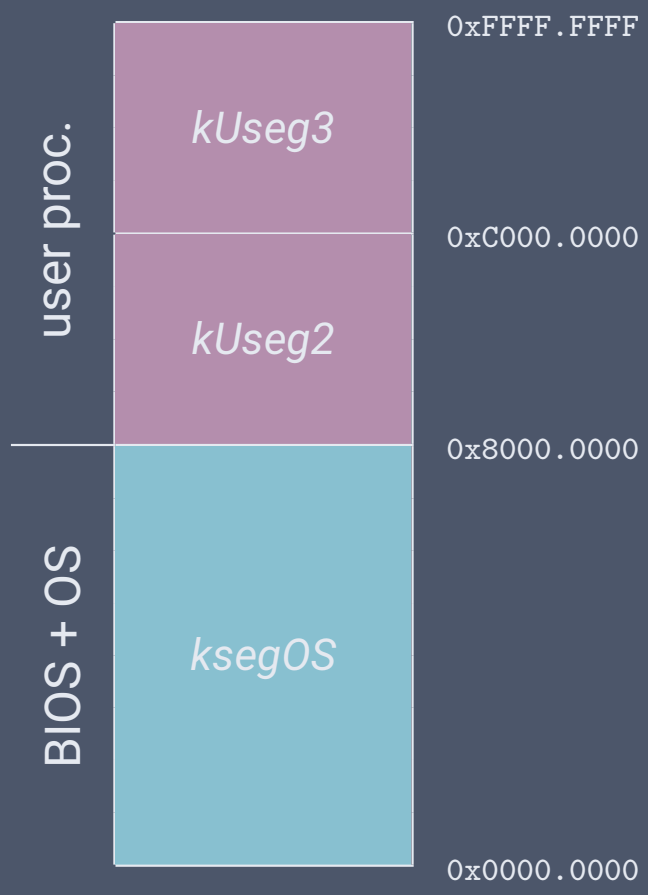
Created

- University of Bologna
- Xavier University

Used

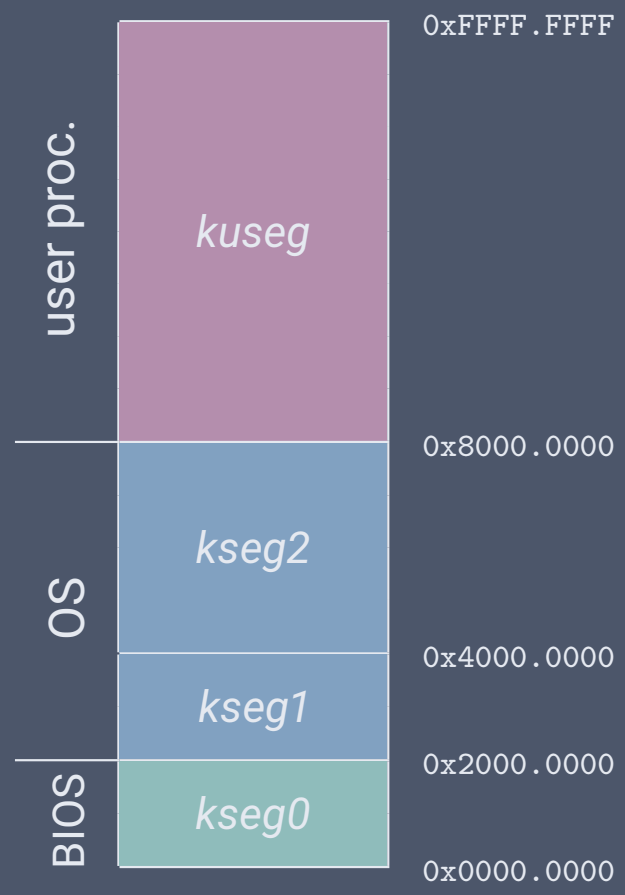
- all around the world

µMPS2



- Segmentation

µMPS3



- Fixed partitions

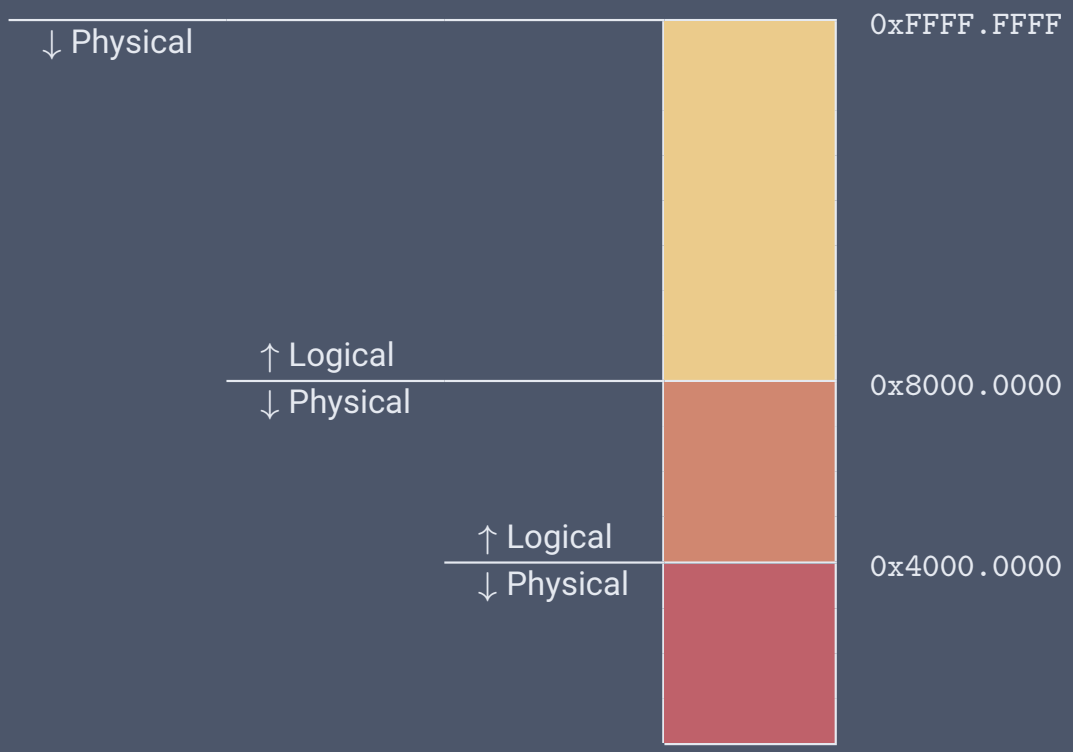
TLB Floor Address

Possible options

- `0x4000.0000`
- `0x8000.0000`
- `VM OFF` (`0xFFFF.FFFF`)

Consequences

- *VM bit* removed



Mode of operation

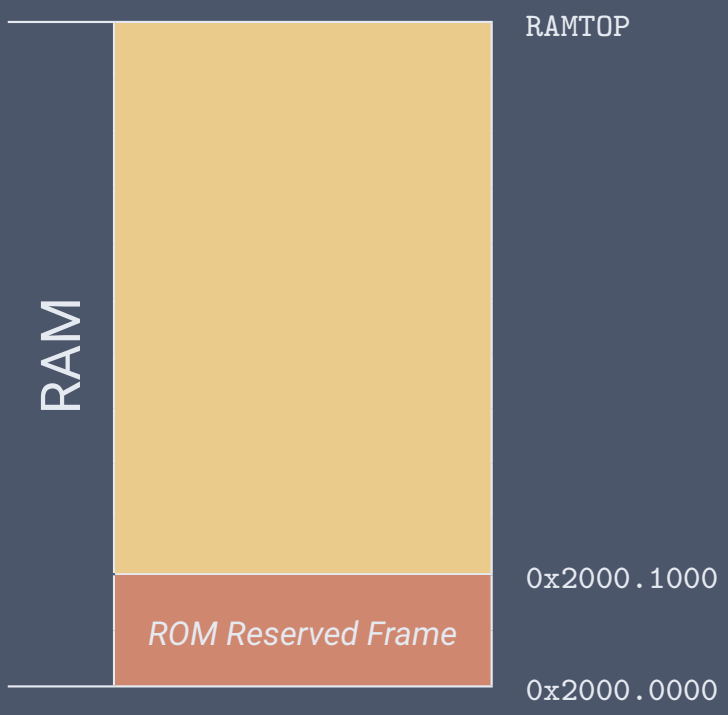
Addresses below *TLB Floor Address* ↓

- considered physical
- exempt from address translation

Addresses above *TLB Floor Address* ↑

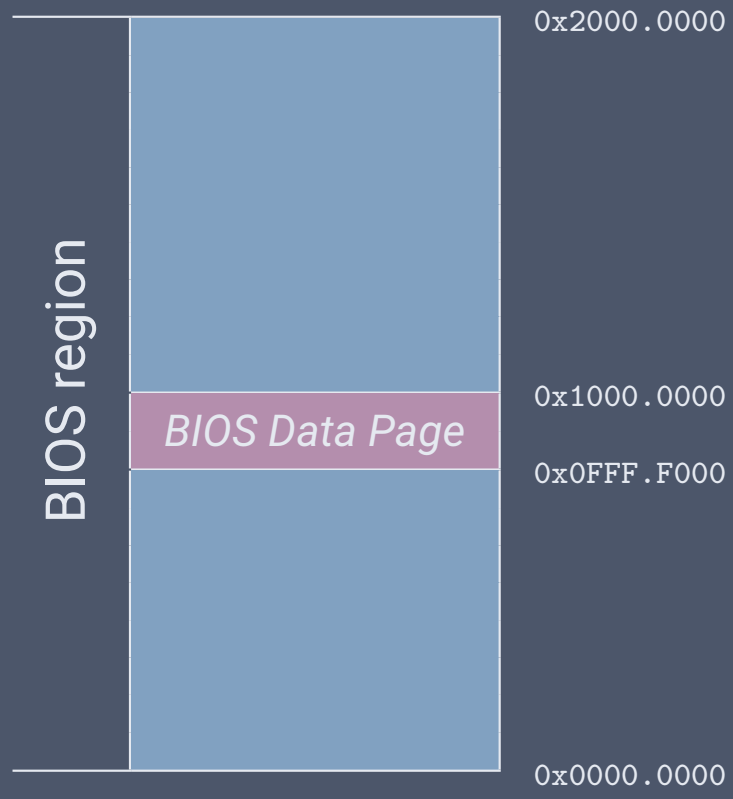
- considered logical
- subject to address translation

µMPS2 - ROM Reserved Frame



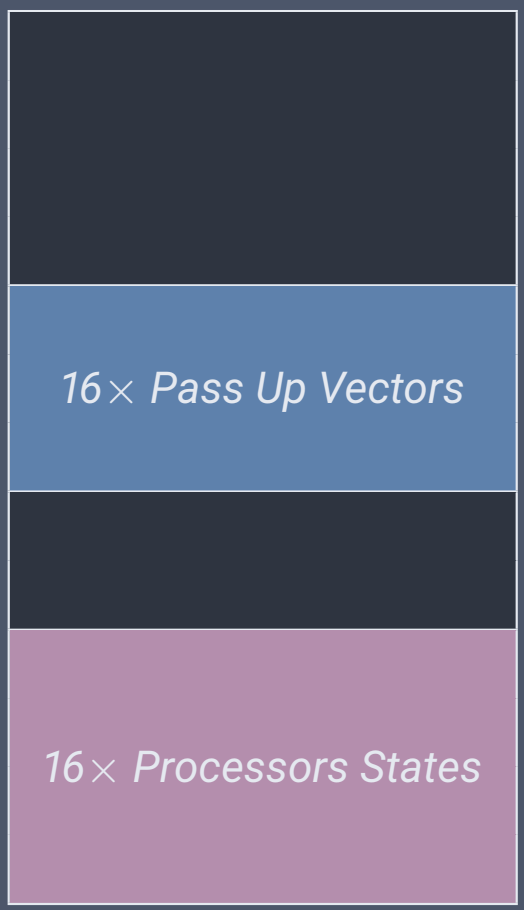
- space for just one processor state
- segment table
- space for multiprocessor purpose

µMPS3 - BIOS Data Page



- space for 16× processors states
- space for 16× Pass Up Vectors

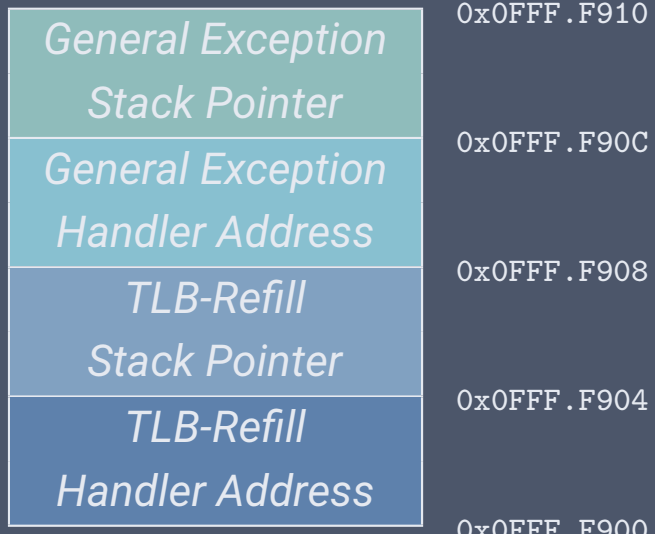
BIOS Data Page



Exceptions Types

- *general* exceptions
- *TLB-Refill* exceptions

Pass Up Vector



Processor Interface Registers

Each processor has its own private instance of

- `BIOS Reserved 1` (address of the processor exception state area)
- `BIOS Reserved 2` (address of the processor Pass Up Vector)

Processor 0: *Bootstrap BIOS Services*

- `BIOS Reserved 1` → `0x0FFF.F000`
- `BIOS Reserved 2` → `0x0FFF.F900`

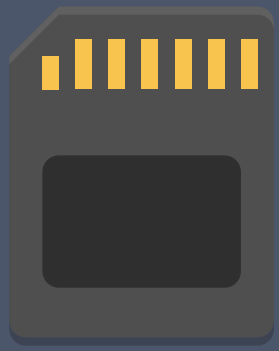
Processors 1-15: *Library Services*

```
void INITCPU (uint32_t cpuid, state_t *start_state);
```

- `BIOS Reserved 1` → `(cpuid × 35 words) + 0x0FFF.F000`
- `BIOS Reserved 2` → `(cpuid ×  4 words) + 0x0FFF.F900`

Characteristics

- read/writable DMA device
- single contiguous block addressable space
- maximum size of 64GB
- replaces tapes



Creation

```
$ umps3-mkdev -f <flashfile>.umps <file> [blocks [wt]]
```

- `<flashfile>` : flash device image file name
- `<file>` : file to be written → `/dev/null` for an empty device
- `blocks` : number of blocks
- `wt` : average write time (microsecs.) → slower than disks by default

Building Process

Migrated from Autotools to CMake

- simpler 2-step compilation → `$ cmake && make`
- faster building time
- lighter codebase → **-11339** physical source lines of code
- easier to learn for novices
- immediate integration with *Qt5*

Graphical User Interface

- updated from *Qt4* to *Qt5*
- replaced old icon theme
- designed new logo with *Inkscape*
- various other GUI improvements

Power ON

Power OFF

Debian

- soon available in official repositories

Arch Linux

- already available in the *AUR* (Arch User Repository)

Source Code

- `github.com/virtualsquare/umps3`

License

- updated from **GPL-2.0** to **GPL-3.0**