

Driver Linux (DRV)

Foundations

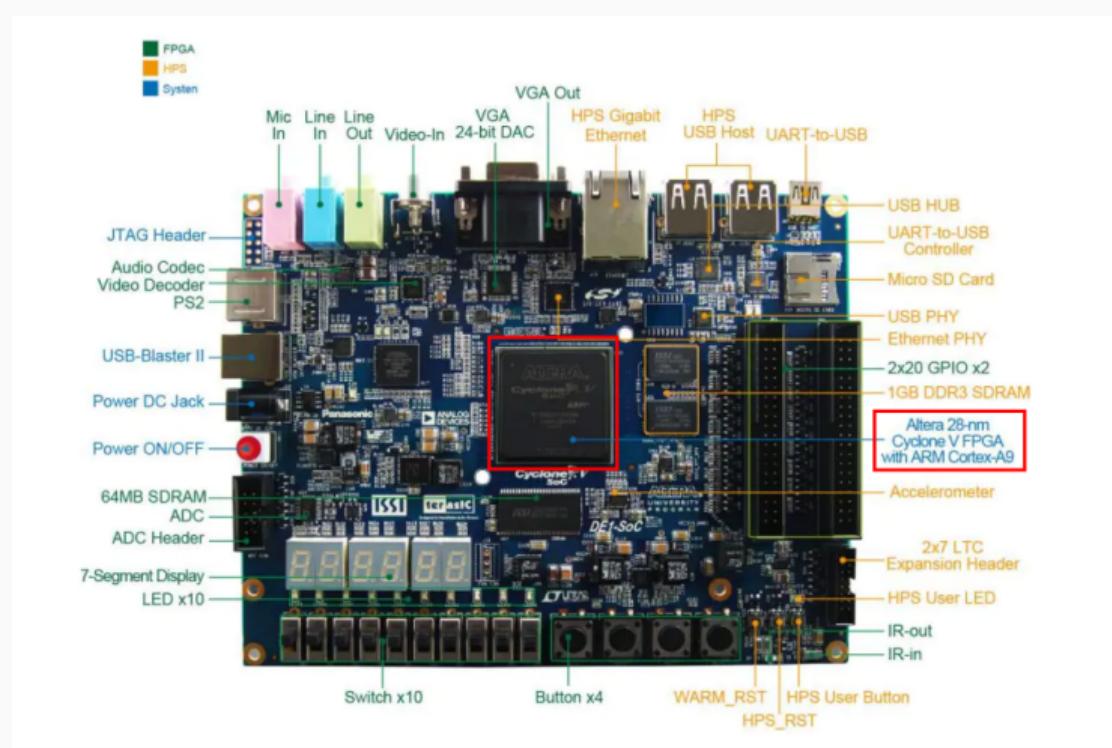
Florian Vaussard, Alberto Dassatti

Février 2024

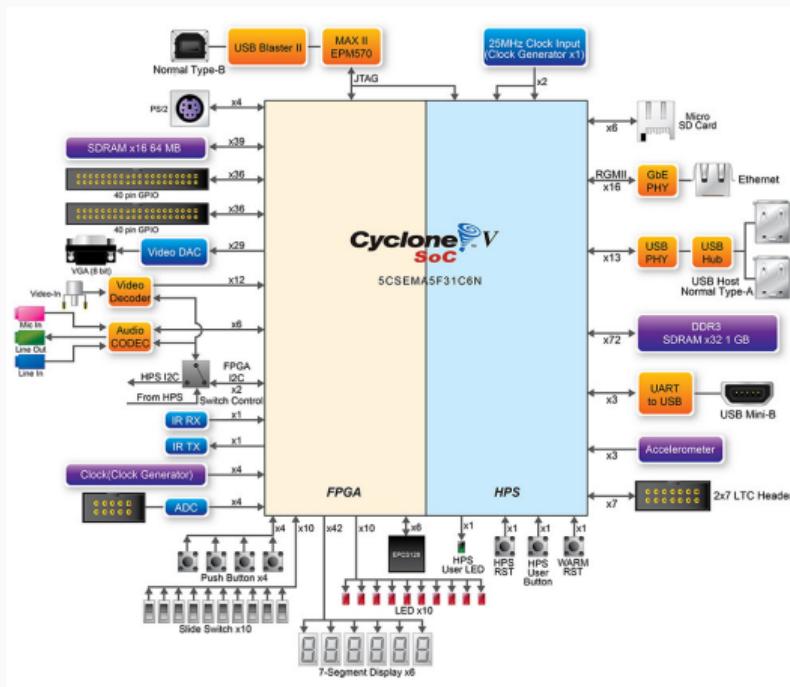
Reconfigurable and Embedded Digital Systems Institute
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

HW CONCEPTS

DE1-SoC

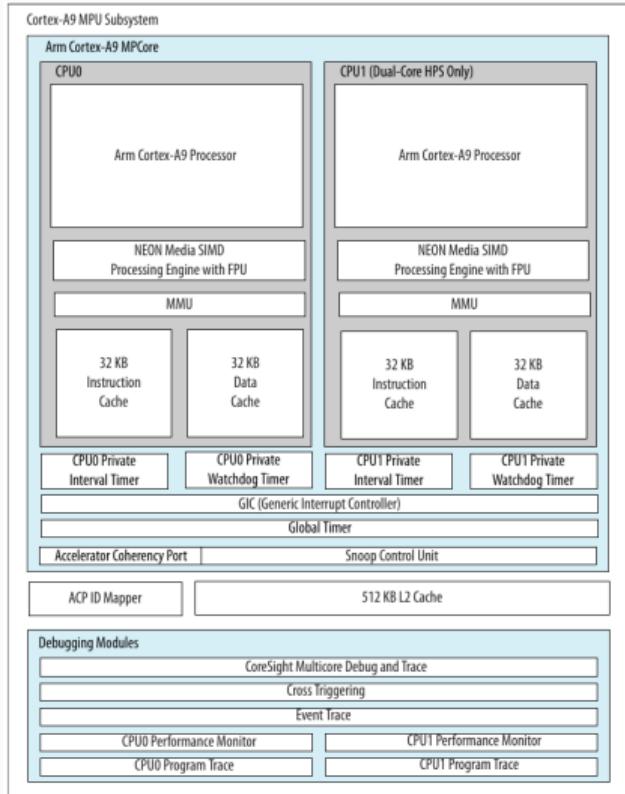
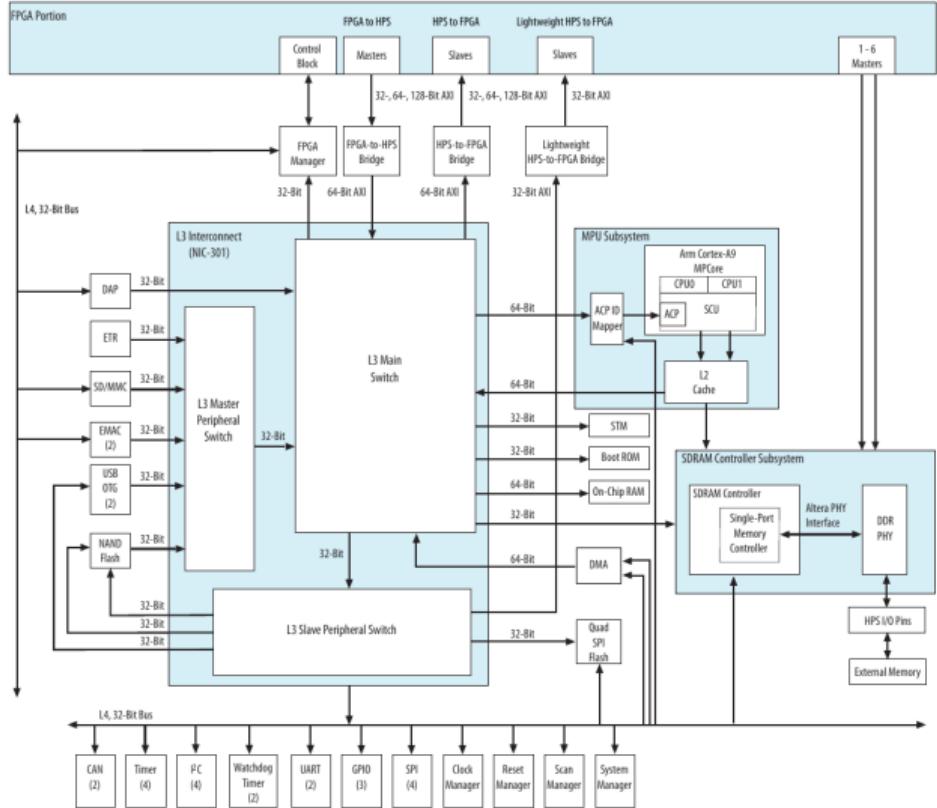


Cyclone V

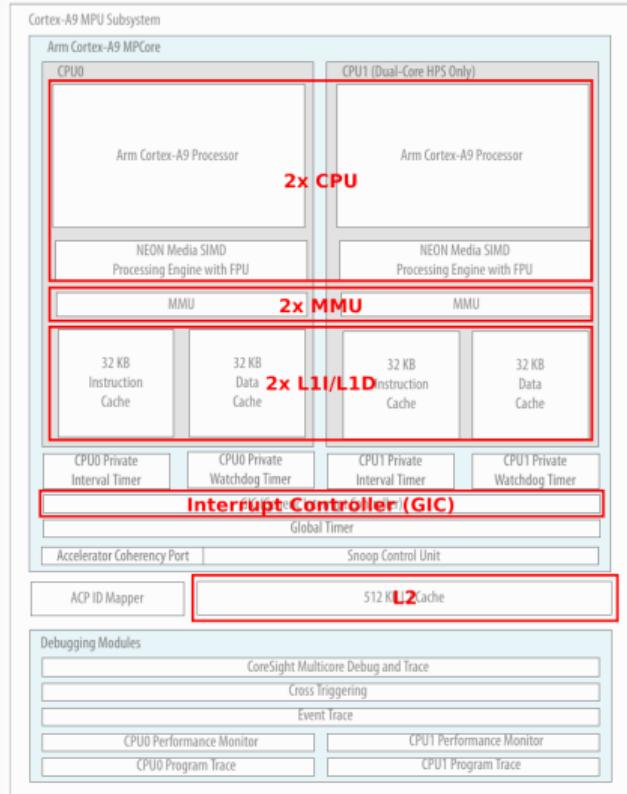
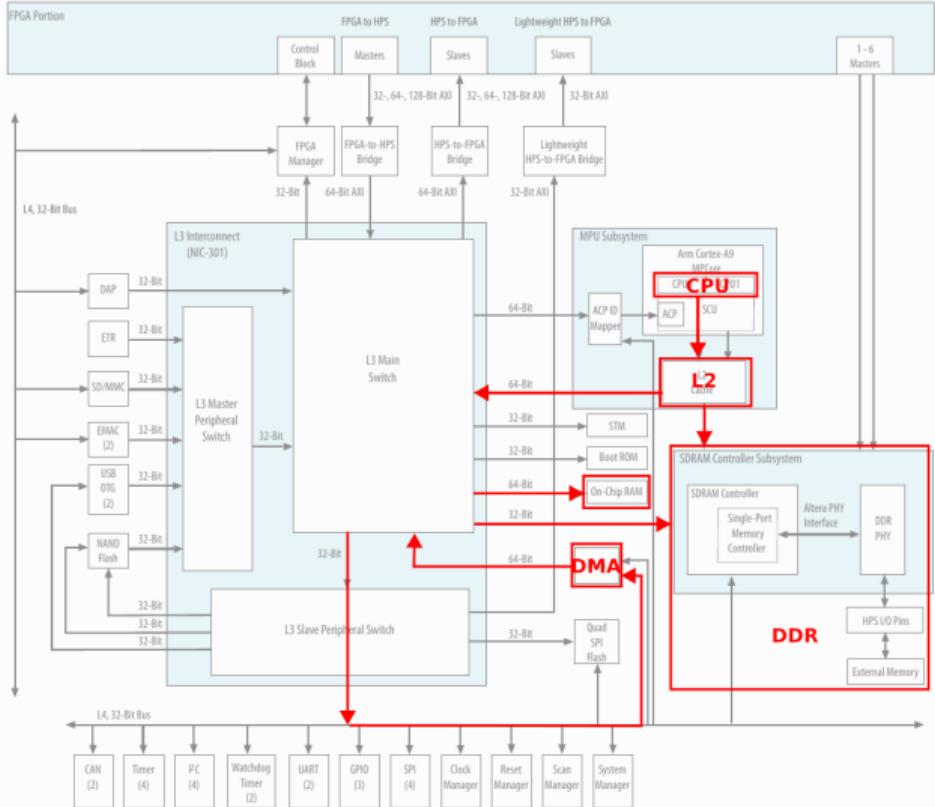


Cyclone® V Hard Processor System Technical Reference Manual (706 pages)

Cyclone V: HPS



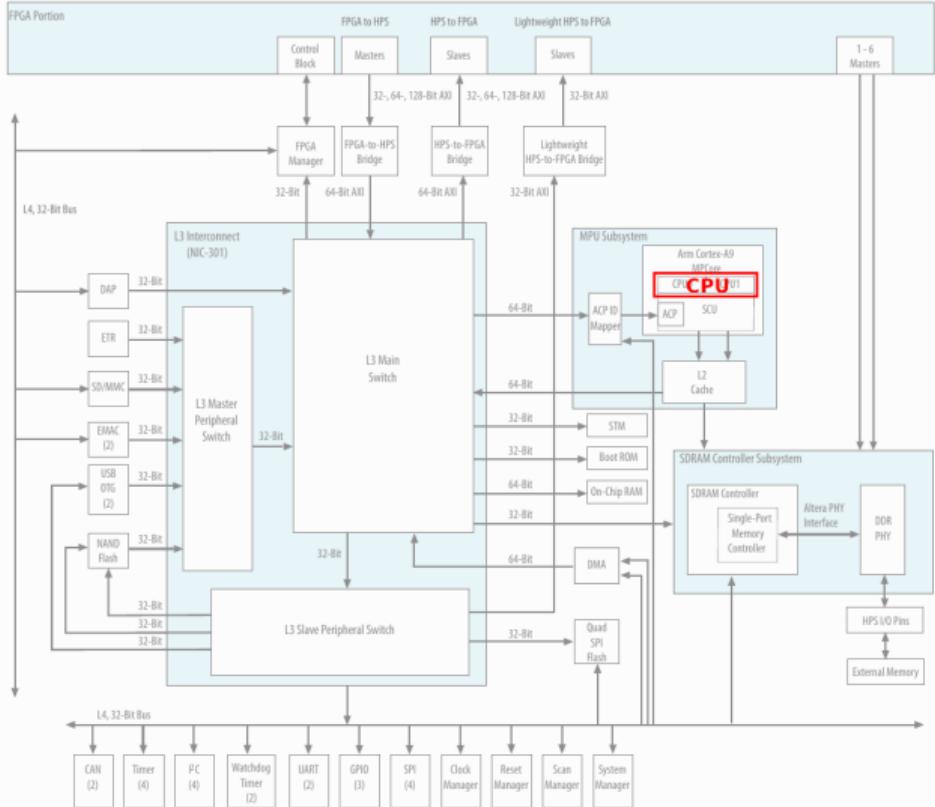
Cyclone V: HPS



HW CONCEPTS

PROCESSOR

Cyclone V: HPS CPU



Cyclone V: HPS CPU

- 2x ARM Cortex-A9 32-bit processor (2007 era)
- ARMv7-A instruction set
- Dual-issue superscalar pipeline with branch prediction
- Out-of-order dispatch and speculative execution
- IEEE-754 floating point unit (FPU)

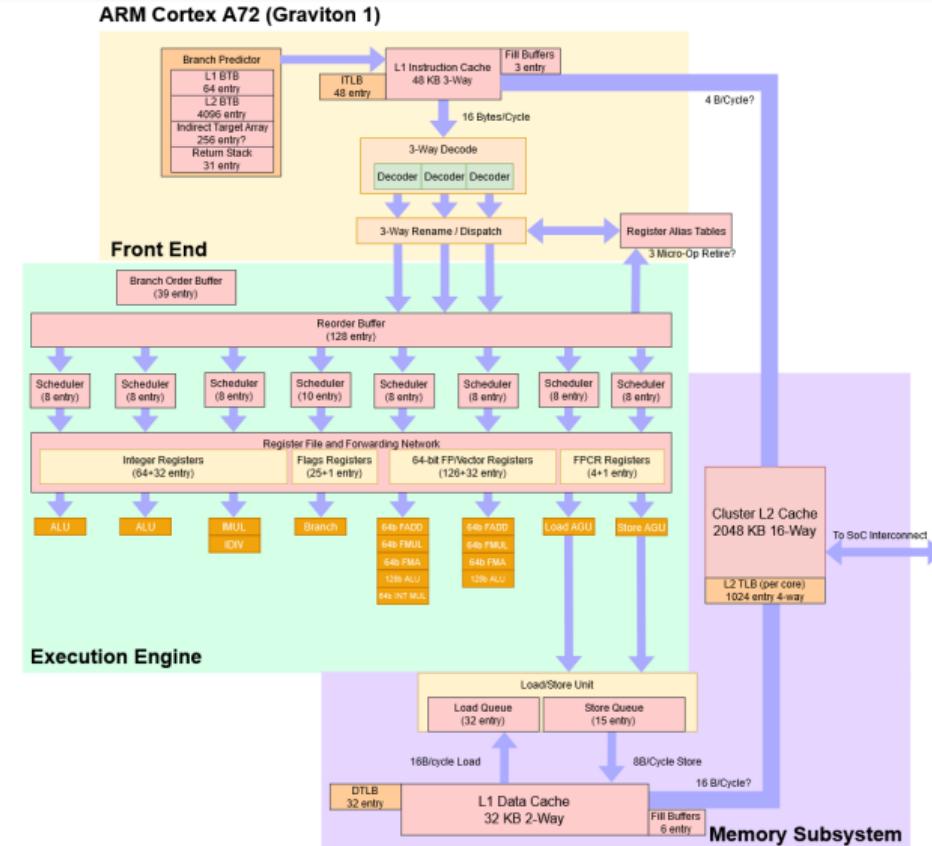
[Wikipedia Cortex-A9](#)

Documentation: [Cortex-A9 Technical Reference Manual](#) (213 pages)

Documentation: [ARM Cortex-A Series Programmer's Guide](#) (421 pages)

Documentation: [ARM Architecture Reference Manual ARMv7-A](#) (2720 pages...)

CPU Microarchitecture (example Cortex-A72)



- Out-of-order execution
 - Compiler can also reorder
- ⇒ Use memory barriers !

Source: Chips and Cheese Cortex-A72

CPU Modes

- Modern CPU has several "Modes" or rings
- Modern OSes uses at least 2 modes: User and Kernel

CPU Modes

- Modern CPU has several "Modes" or rings
- Modern OSes uses at least 2 modes: User and Kernel

Differences between modes

- Instructions

CPU Modes

- Modern CPU has several "Modes" or rings
- Modern OSes uses at least 2 modes: User and Kernel

Differences between modes

- Instructions
- Memory

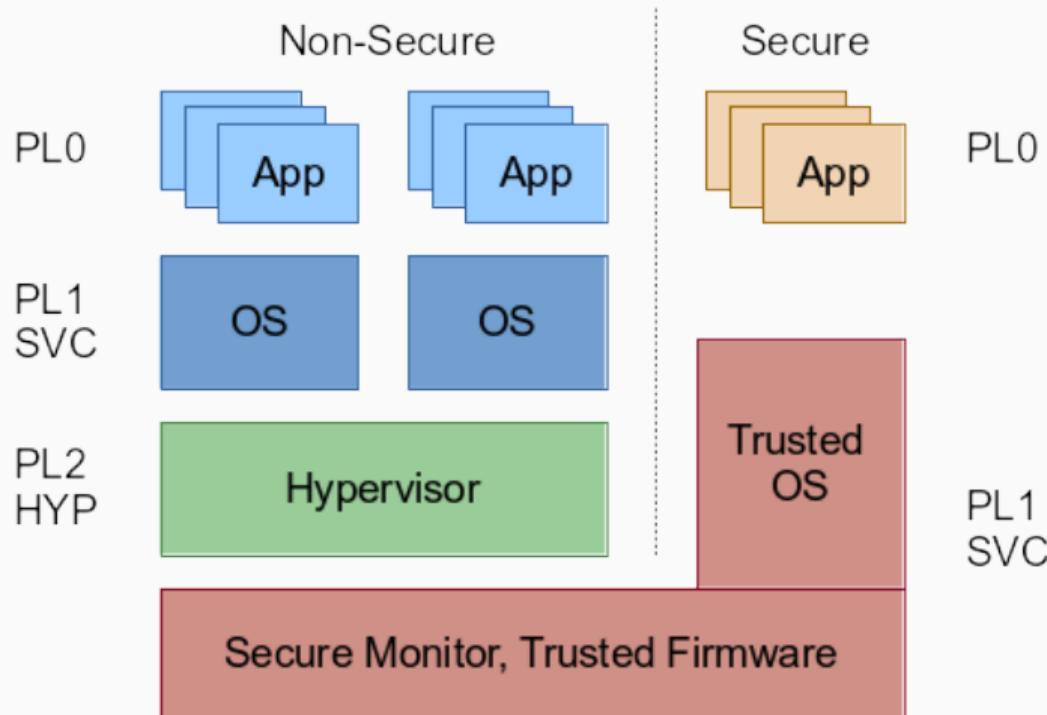
CPU Modes

- Modern CPU has several "Modes" or rings
- Modern OSes uses at least 2 modes: User and Kernel

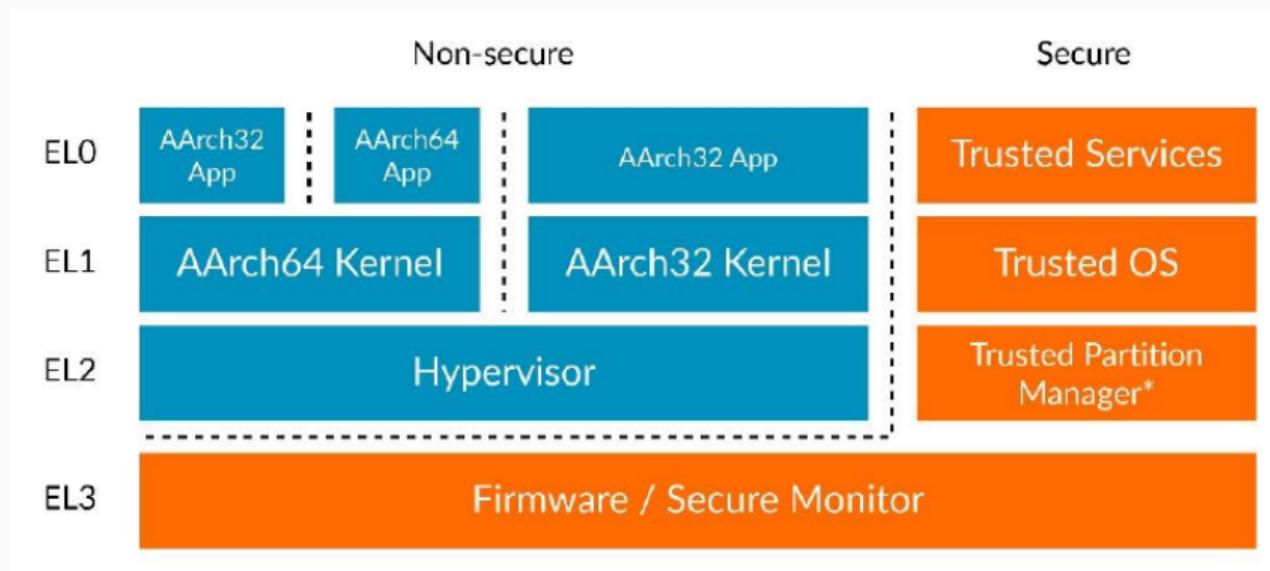
Differences between modes

- Instructions
- Memory
- Registers

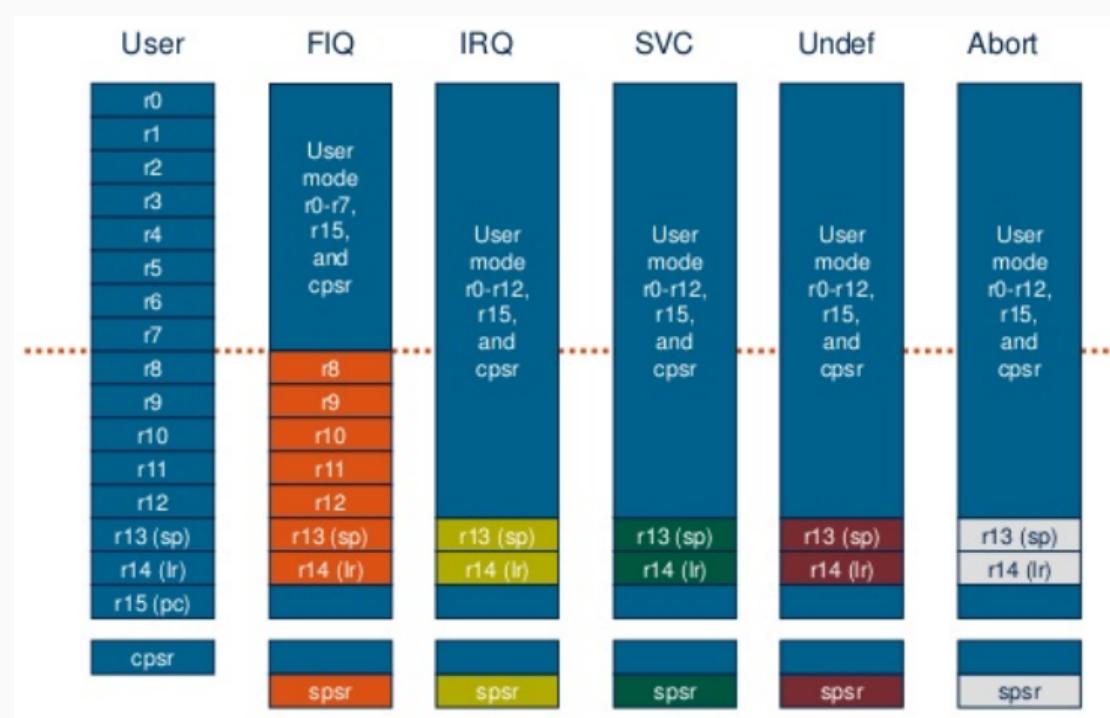
CPU Modes: ARM32 example



CPU Modes: ARM64 example



CPU Modes: ARM registers



Source: Architecture of ARM7 processor

CPU Modes: How to switch

How to change processor mode?

CPU Modes: How to switch

How to change processor mode?

HW events

- Interrupt

CPU Modes: How to switch

How to change processor mode?

HW events

- Interrupt
- Exceptions
 - Memory access problems
 - Undefined instructions
 - Division by zero
 - ...

CPU Modes: How to switch

How to change processor mode?

HW events

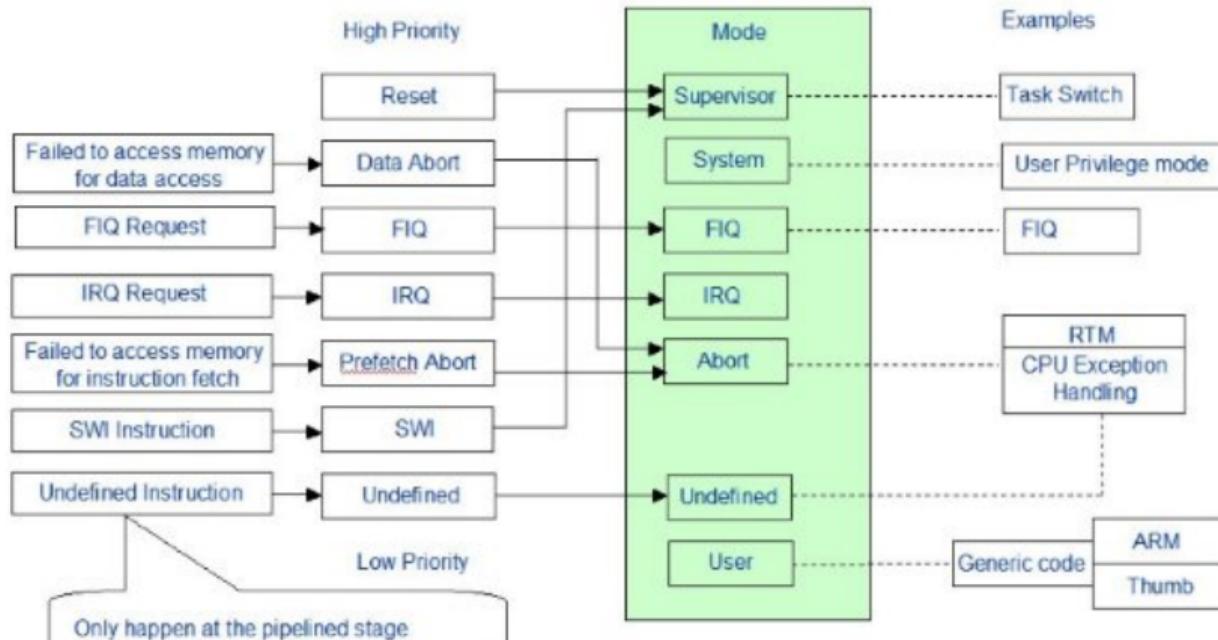
- Interrupt
- Exceptions
 - Memory access problems
 - Undefined instructions
 - Division by zero
 - ...

SW events

- Software interrupt (specific instruction for each CPU)
 - It is the basic for any SysCall

CPU Modes: ARM example

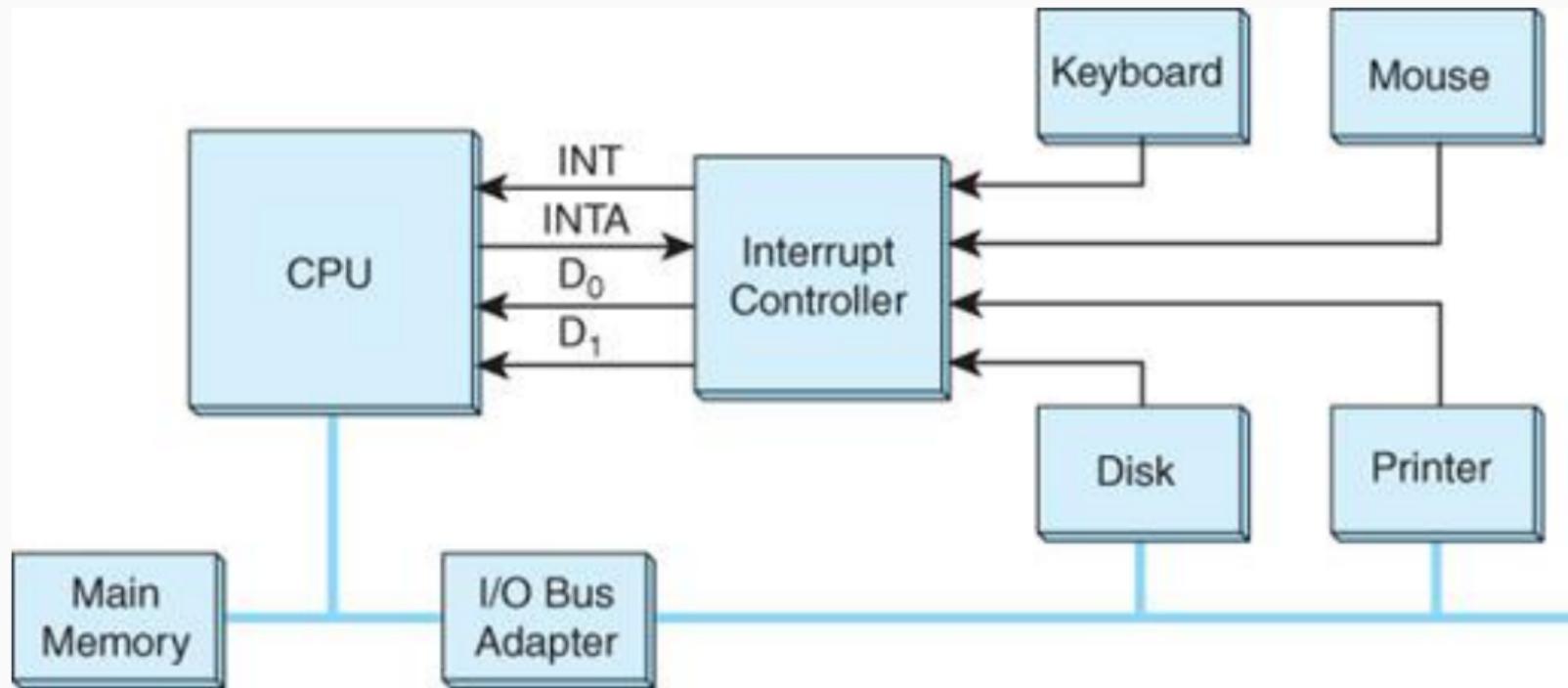
ARM Exceptions and Mode Switch



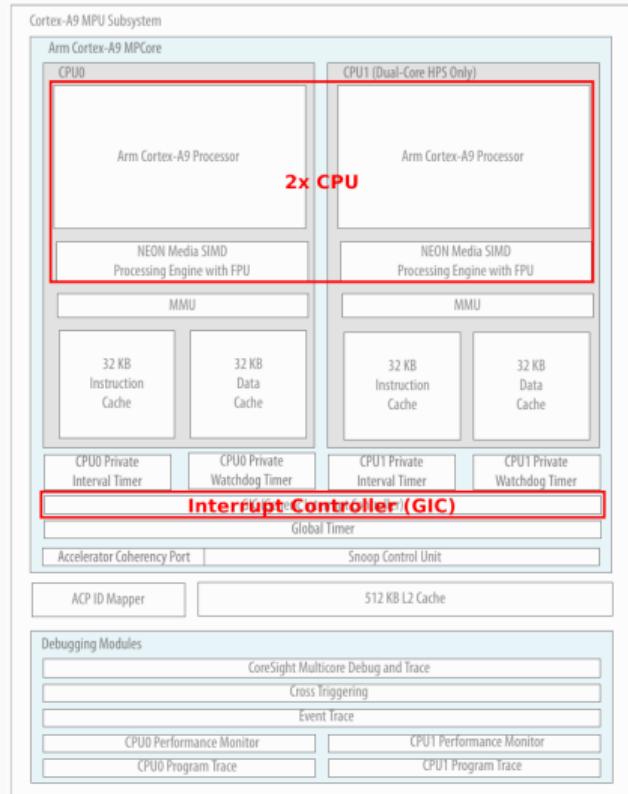
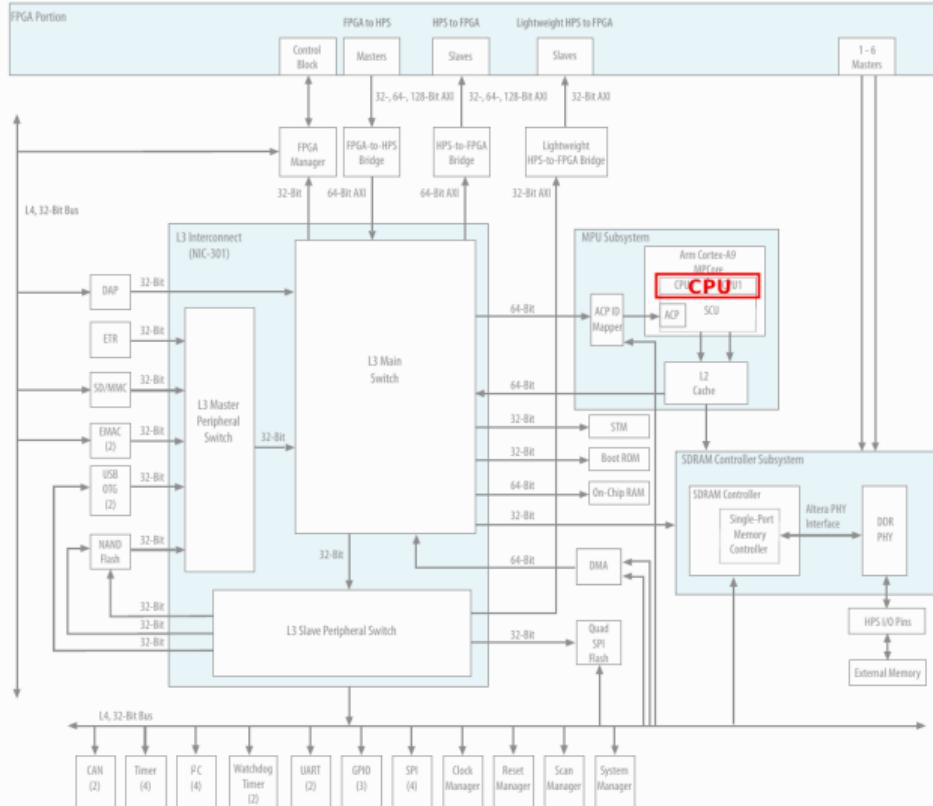
HW CONCEPTS

Interrupts

Interrupts: HW

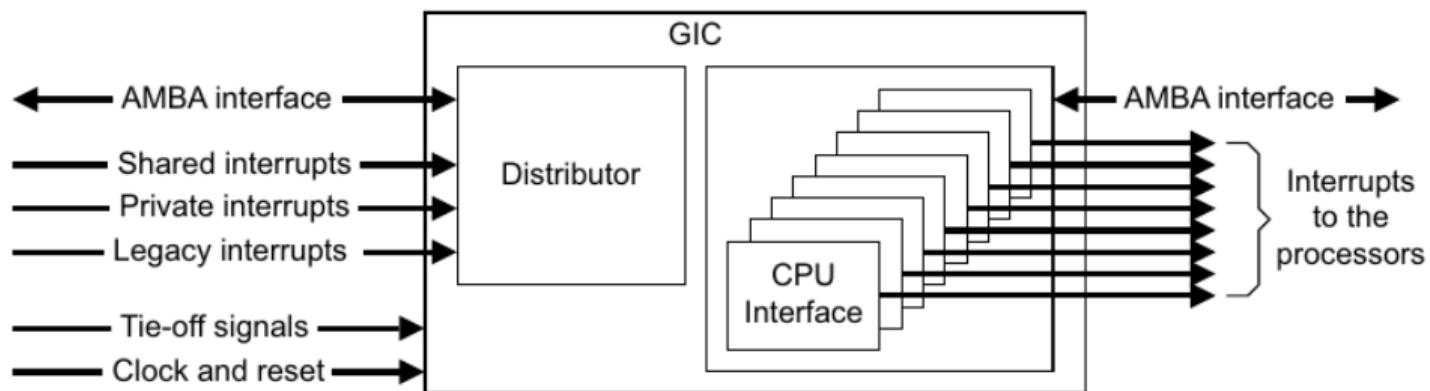


Cyclone V: HPS Interrupt



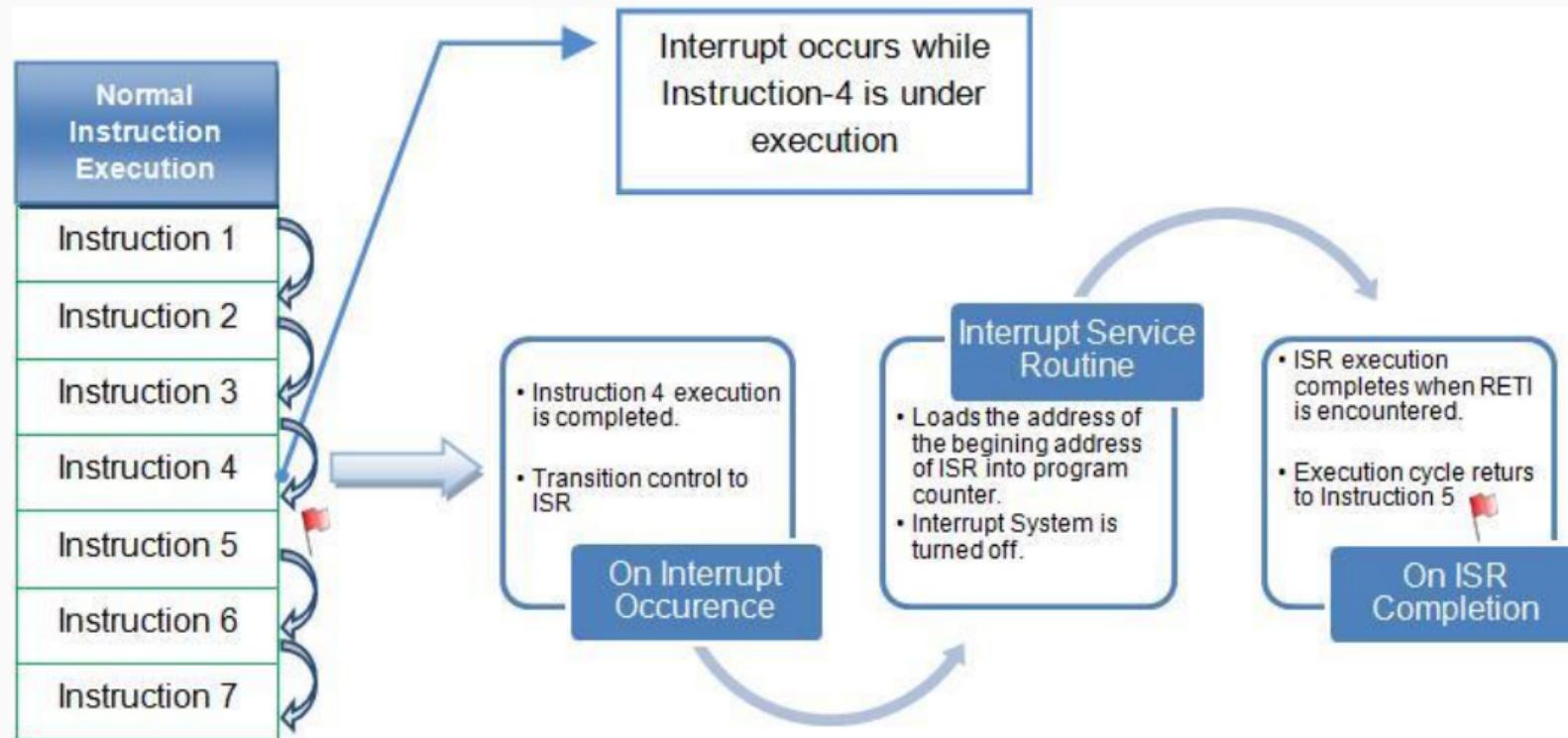
Cyclone V: HPS Interrupt

Interrupt controller for multi-core (ARM GIC)

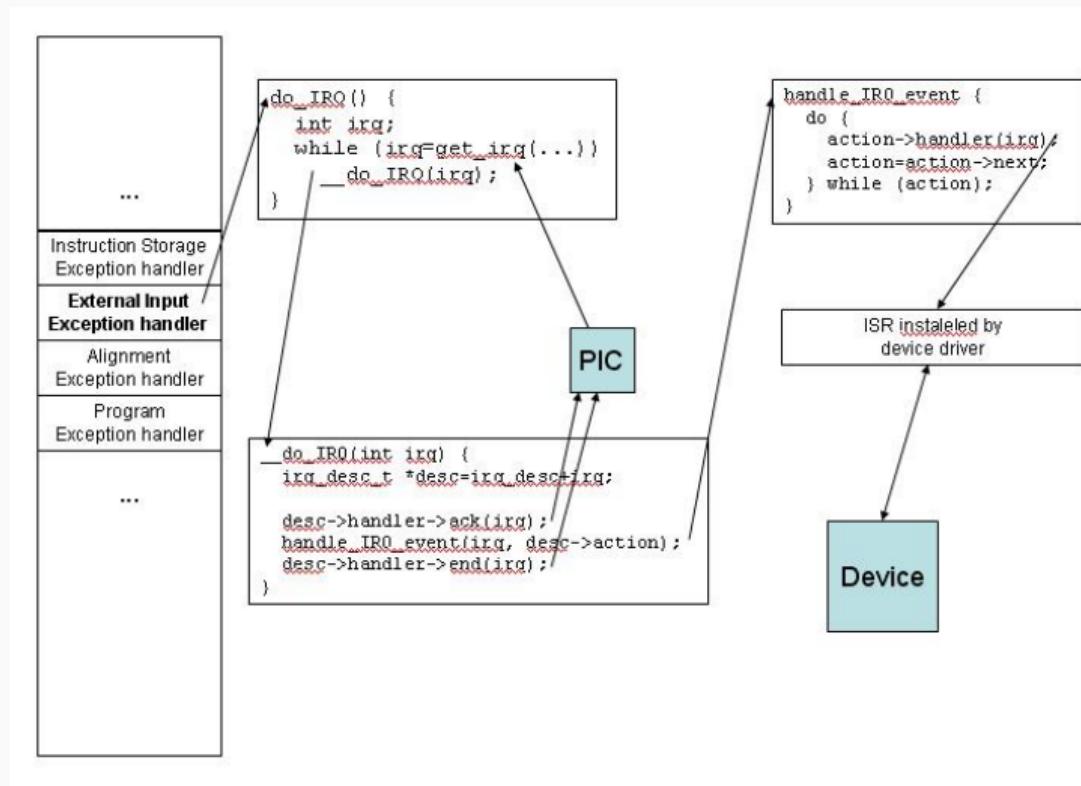


Documentation: [PrimeCell Generic Interrupt Controller](#) (only 105 pages...)

Interrupts: SW



Interrupts: Linux Example

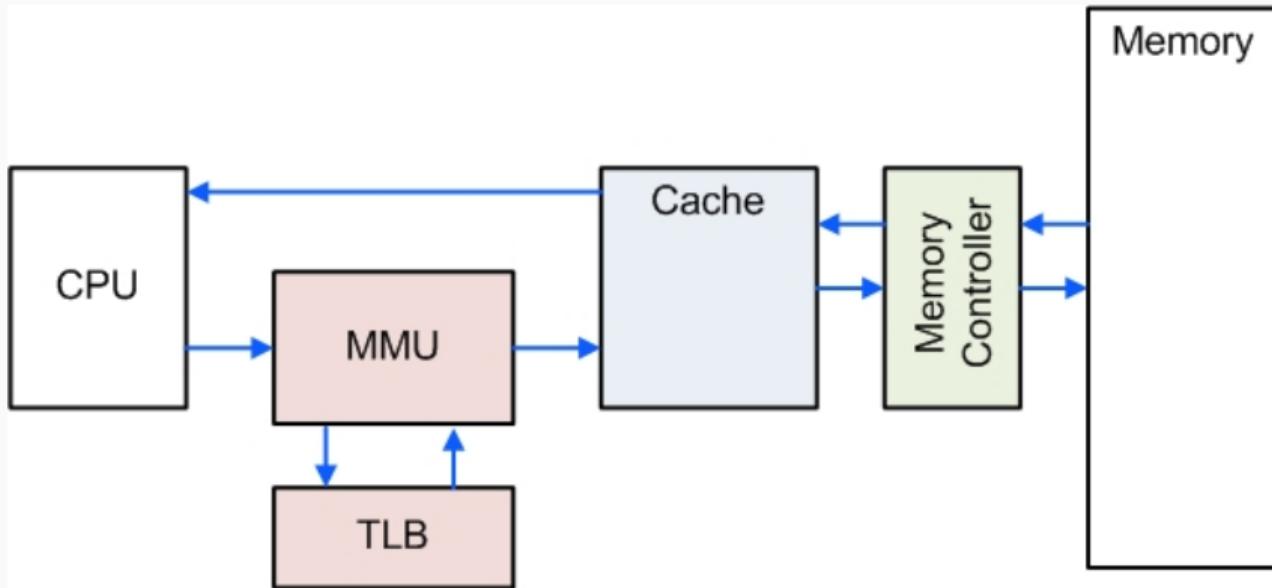


⚠ slightly out-dated

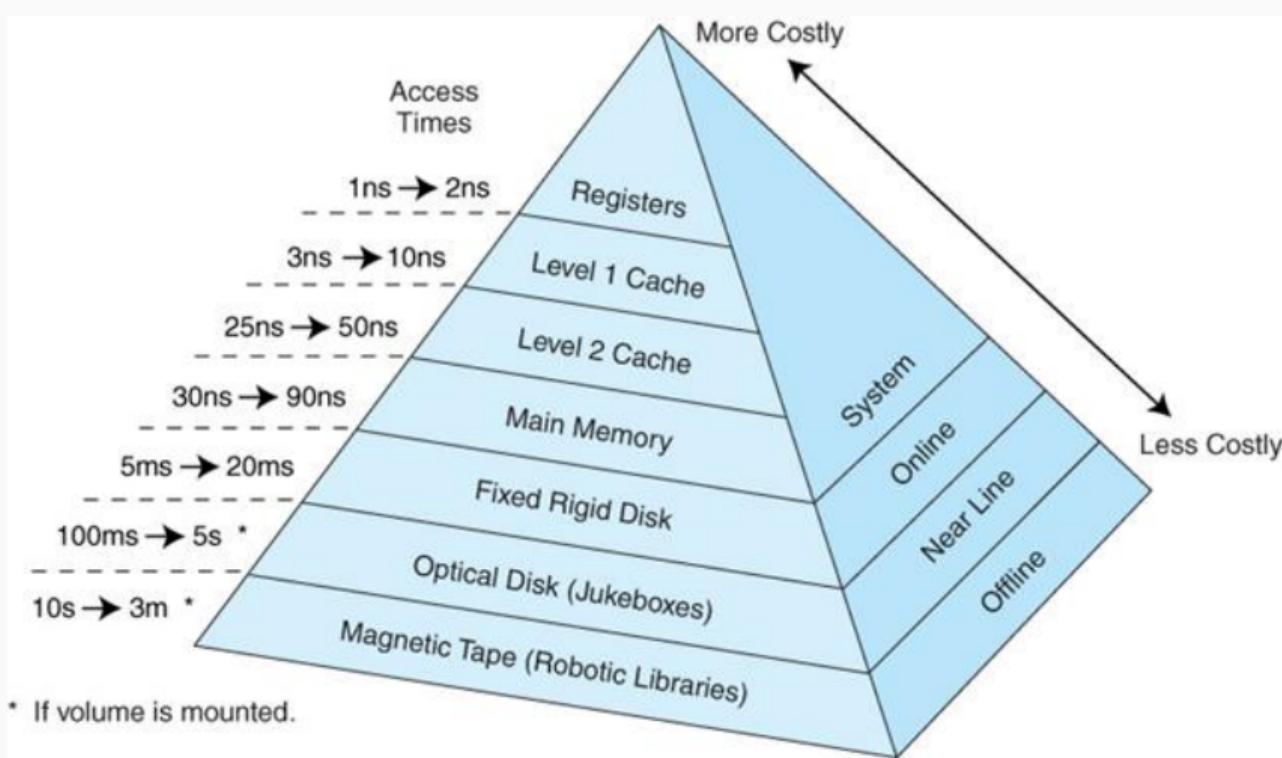
HW CONCEPTS

Memories

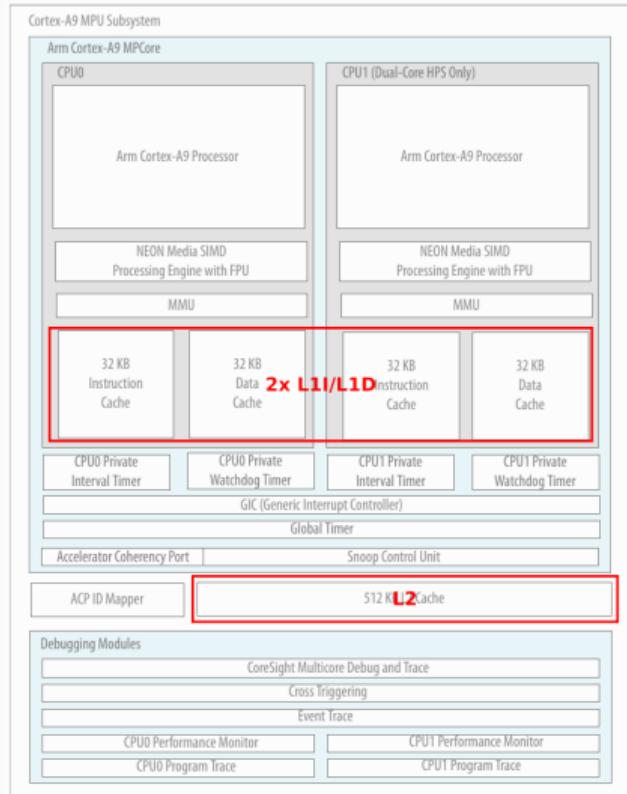
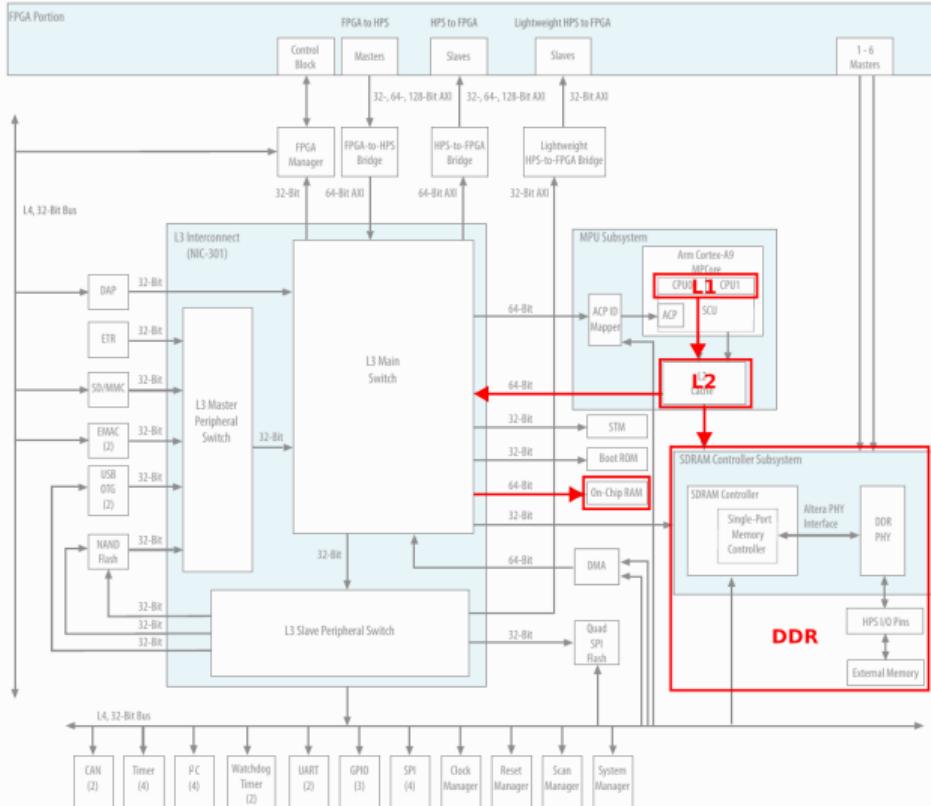
Memories



Memories



Cyclone V: HPS memories



Cyclone V: HPS caches

L1 cache:

- 32 kB 4-way associative instruction cache (L1I)
- 32 kB 4-way associative data cache (L1D)
- 32 bytes per line
- Latency: 1 clock

L2 cache:

- 512 kB 8-way associative
- 32 bytes per line
- Latency: 6 clocks (best case)

Cyclone V: HPS memories

On-Chip RAM:

- 64 kB RAM

SDRAM controller:

- Supports DDR2, LPDDR2 and DDR3
- Up to 2x 512 MB
- DE1-SoC: 1 GB

Complexity

RAM attributes

- Read-Only, Write, Execute
- Cachability
- Write policy (write back, write through)

⇒ Stored in the page table (MMU)

Cache maintenance (multi-threading)

- Cache invalidate
- Cache flush

Complexity

Memory coherency (multi-core / DMA)

- Hardware (bus snooping)
 - ARM: Snooping Control Unit (SCU)
- Software (cache maintenance)

Memory ordering (single/multi core)

- Know the memory model
 - ARM: normal, device, strongly-ordered
- Use memory barrier !

For hardcore kernel hackers: [Linux kernel memory barriers](#)

Cache

`./02-caches.pdf`

Source: [UNSW Advanced Operating Systems](#)

HW CONCEPTS

Addresses

Addresses

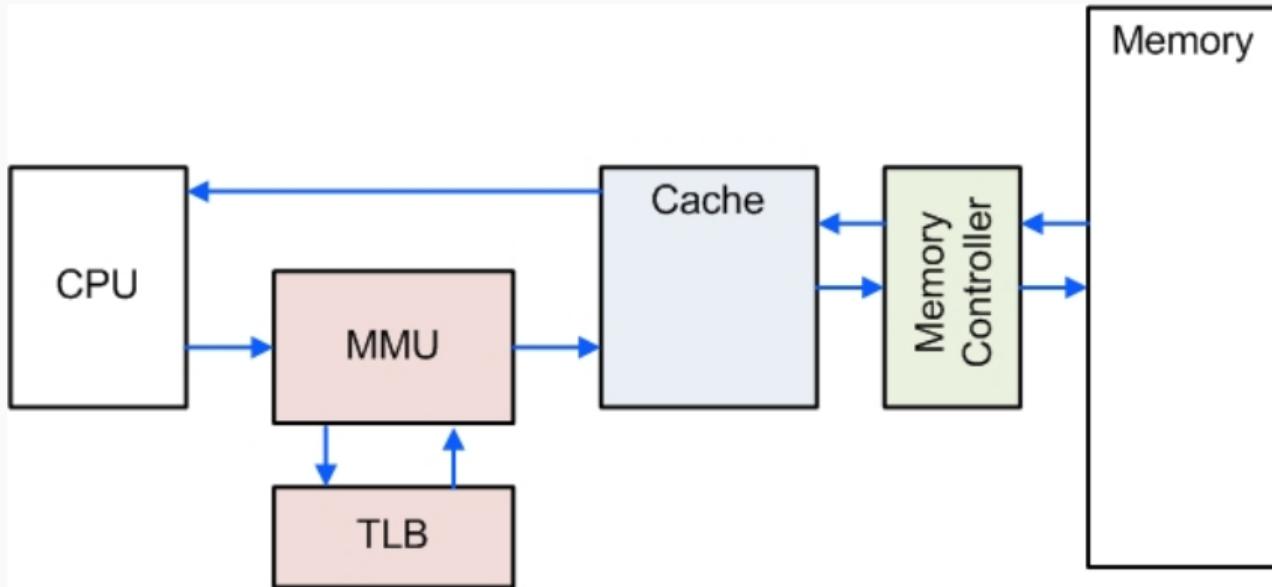
Address types

For instruction and data:

- Virtual address (used by the CPU)
- Physical address (used by the memory)

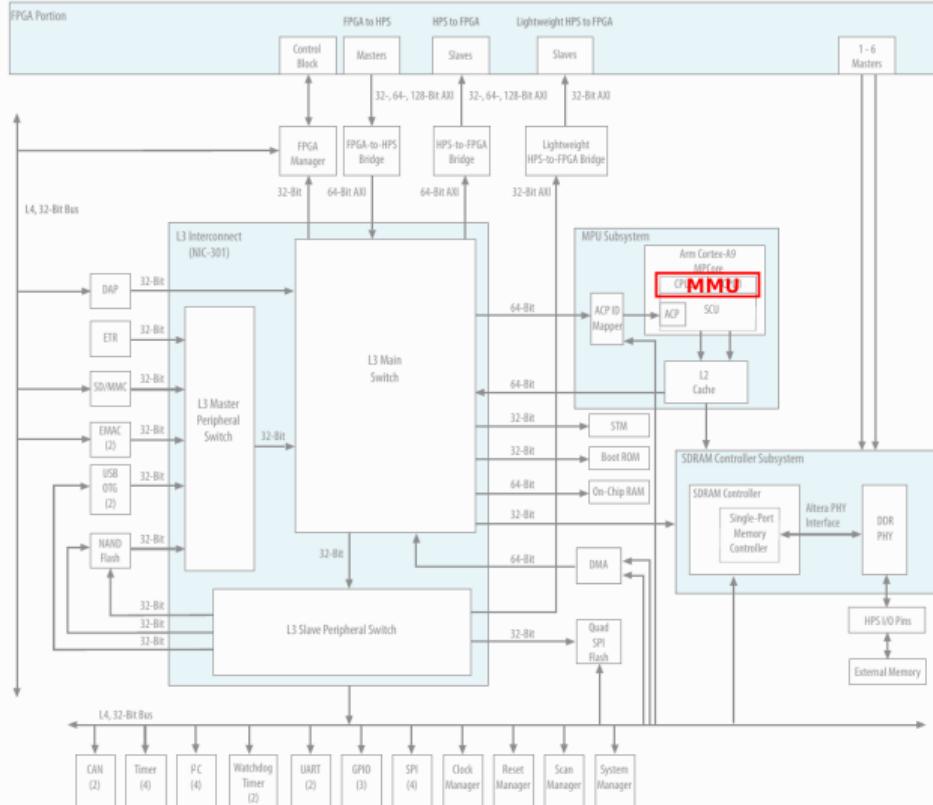
⇒ Translate using the MMU/TLB and page tables

MMU

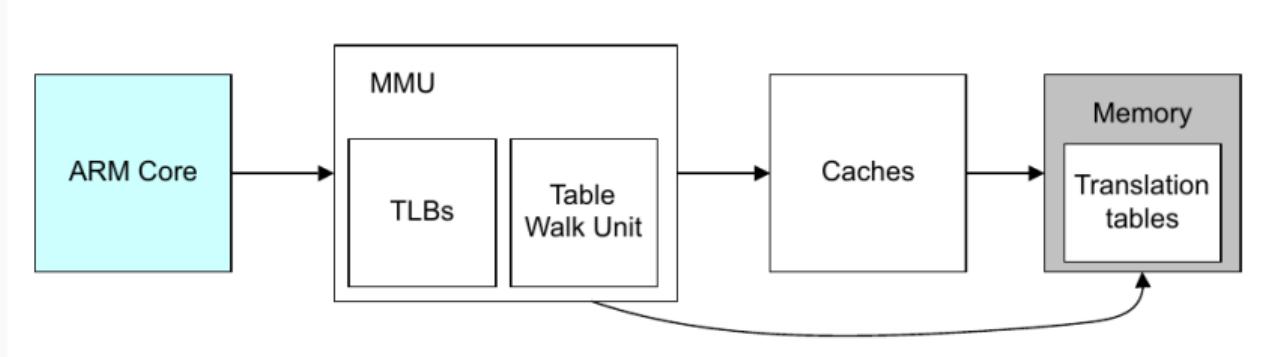


Virtual $\leftarrow \rightarrow$ Physical

Cyclone V: HPS MMU



Cyclone V: HPS MMU



Cyclone V: HPS MMU

MMU:

- Page size: 4 kB, 64 kB, 1 MB
- Access permission: read / read+write (+ execute), privileged / unprivileged
- Memory type: normal, device, strongly-ordered
- Supports 2 page tables (TTBRO / TTBR1)

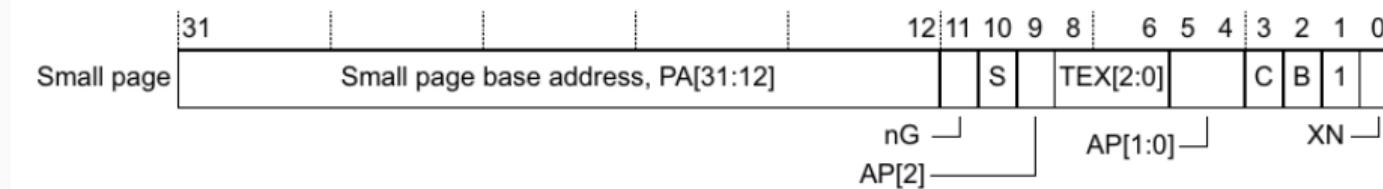
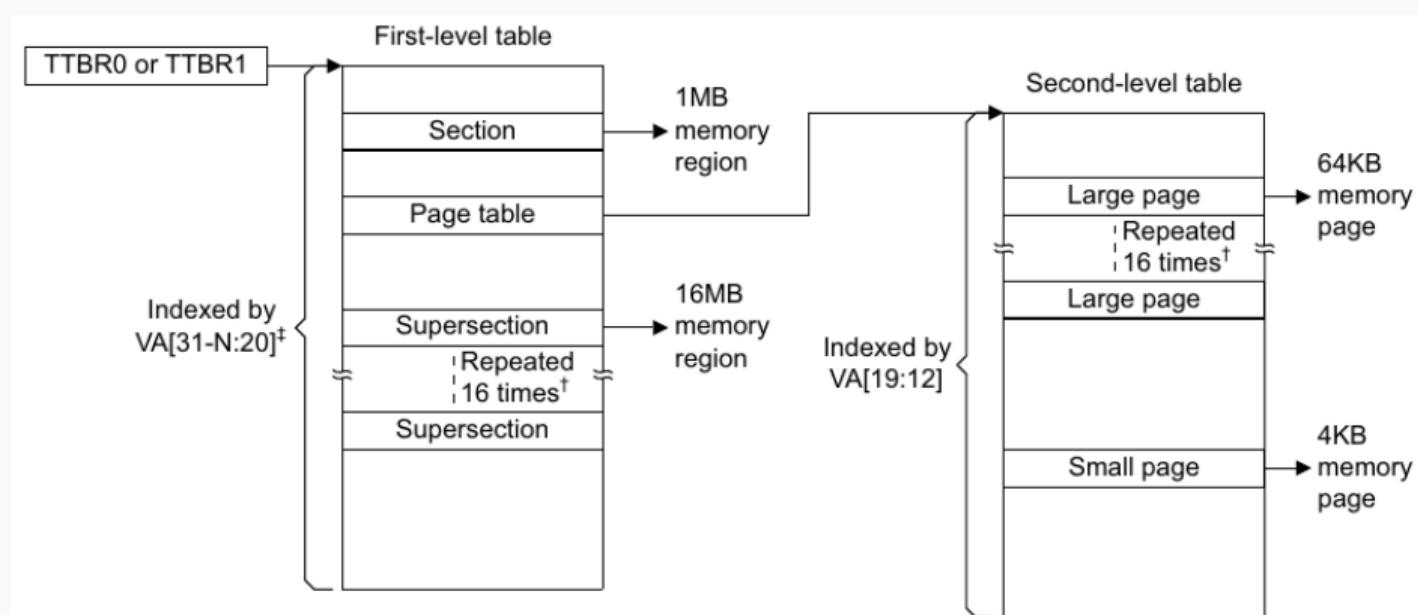
TLB:

- L1I: 32 entries fully associative
- L1D: 32 entries fully associative
- L2: 128 entries 2-way associative
- Maintenance (context-switch): flush all / flush range

Documentation: [ARM Cortex-A Series Programmer's Guide](#)

Documentation: [ARM Architecture Reference Manual ARMv7-A](#)

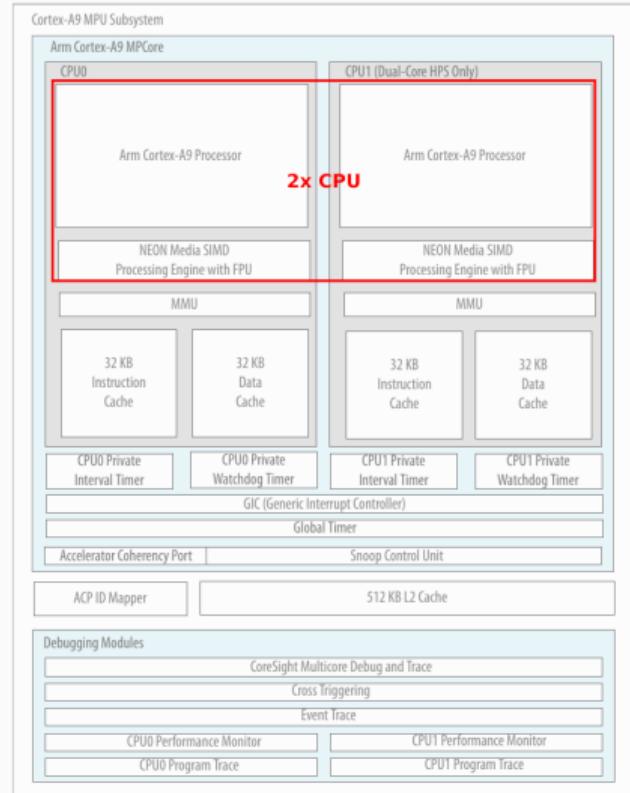
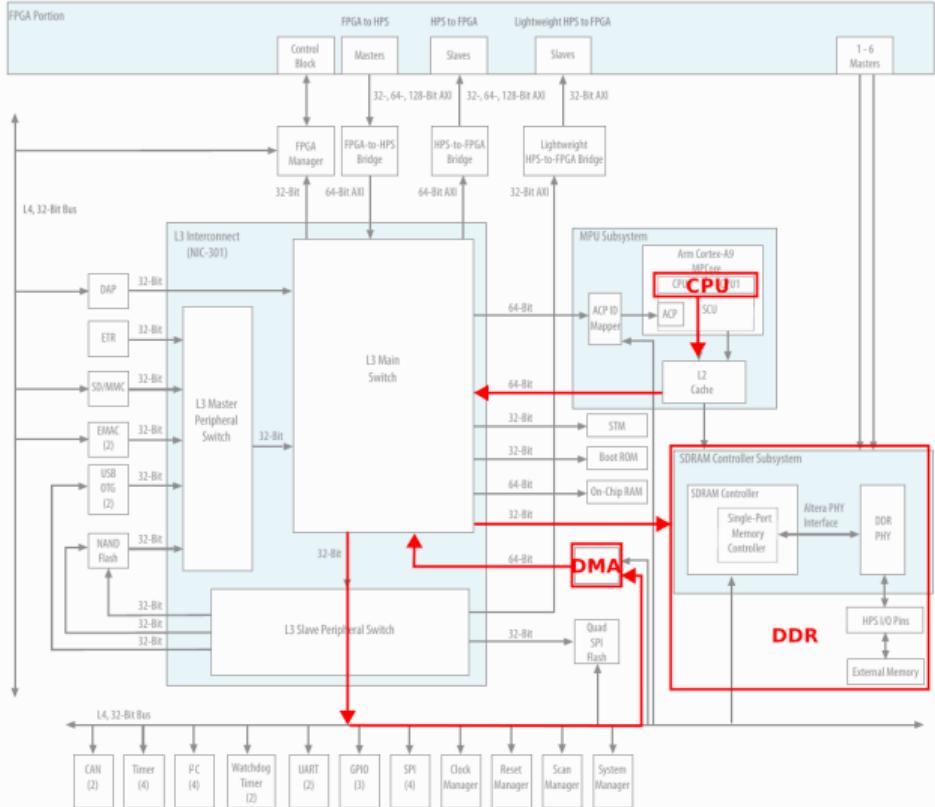
Cyclone V: HPS MMU



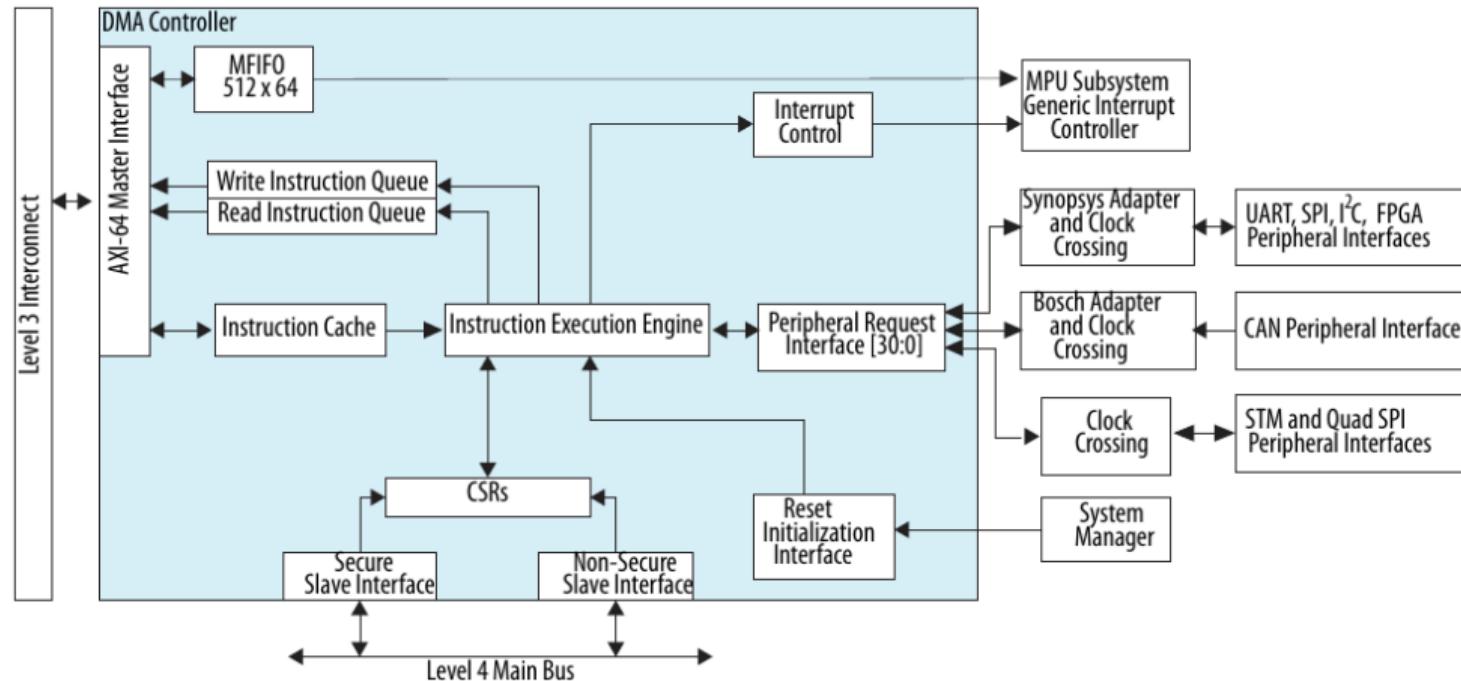
HW CONCEPTS

DMA

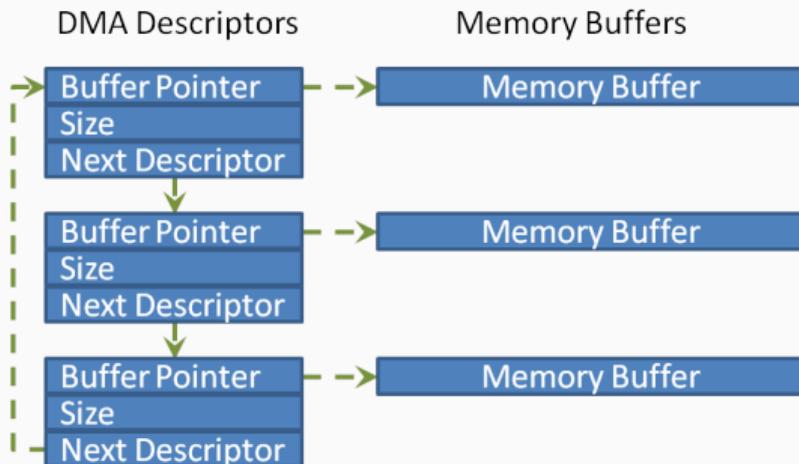
Cyclone V: HPS DMA



Cyclone V: HPS DMA



DMA: SW



Used for...

- Networking
- Audio processing
- Anything with a decent data rate...

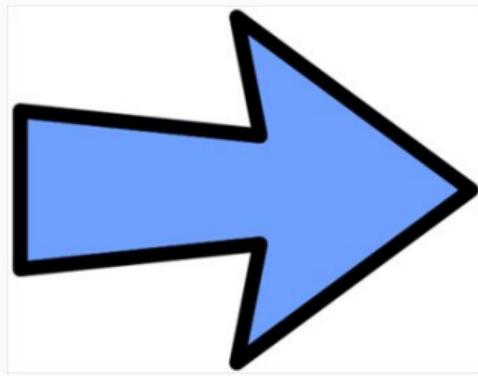
Challenges...

- Memory coherency
- Memory allocation
- Programming model

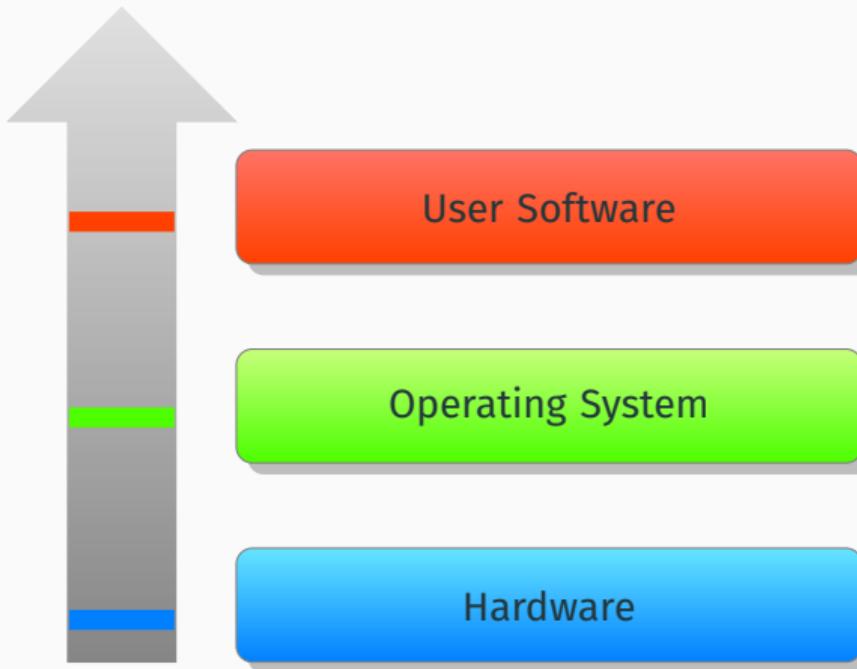
Using Direct Memory Access effectively in media-based embedded applications

OS CONCEPTS

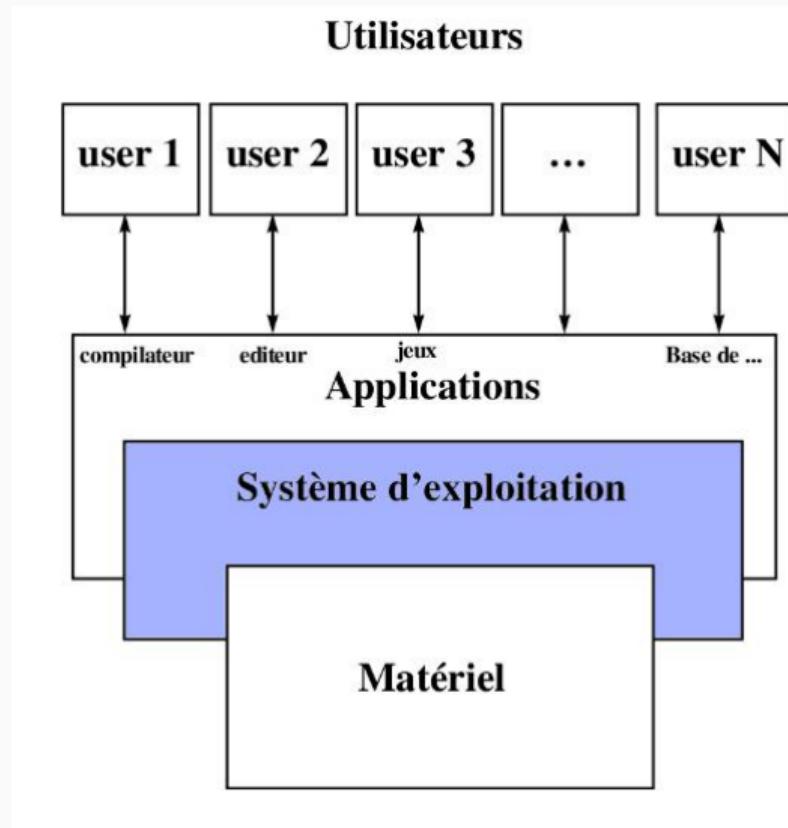
What



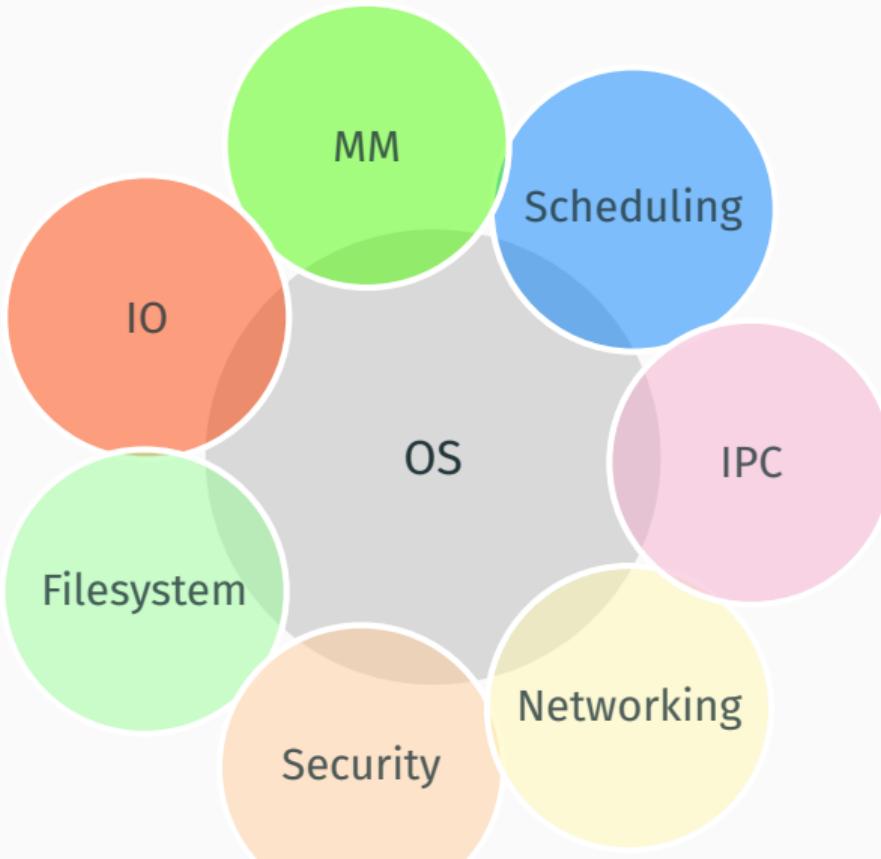
What



Abstraction

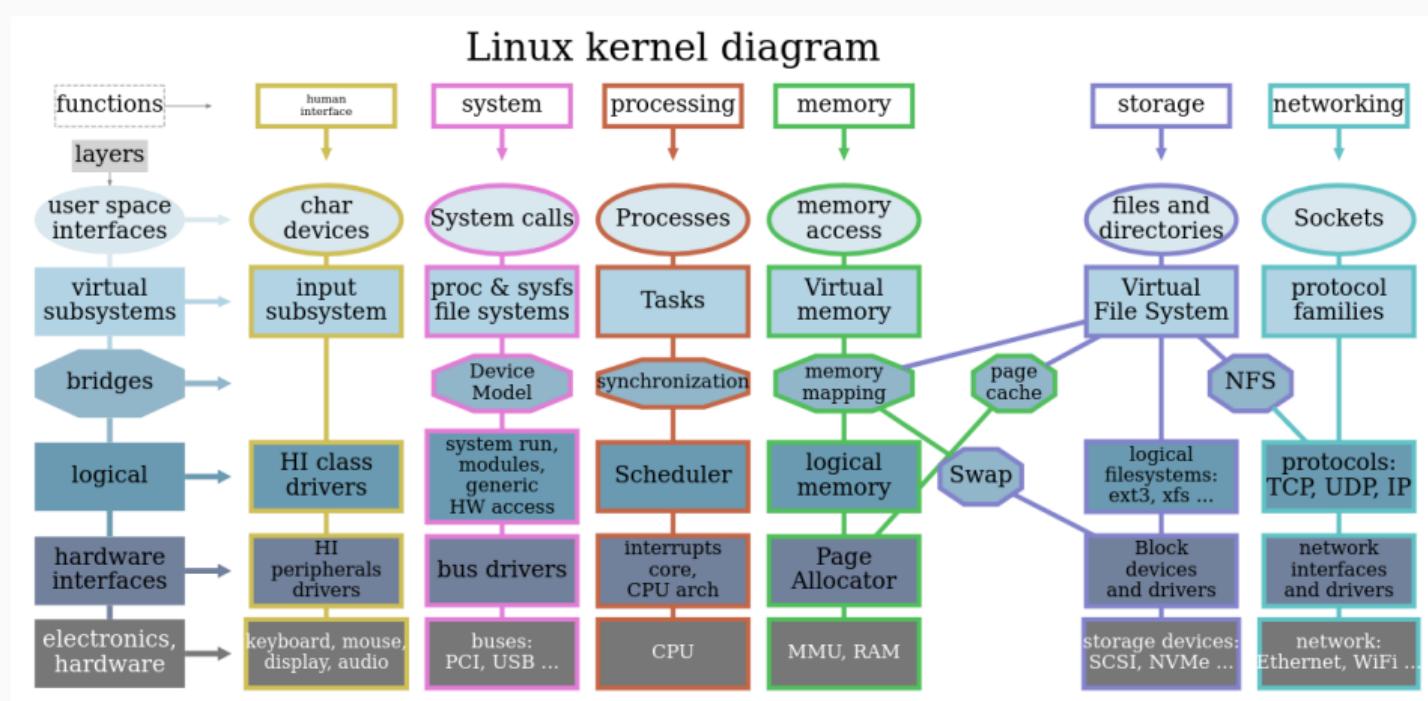


Main services



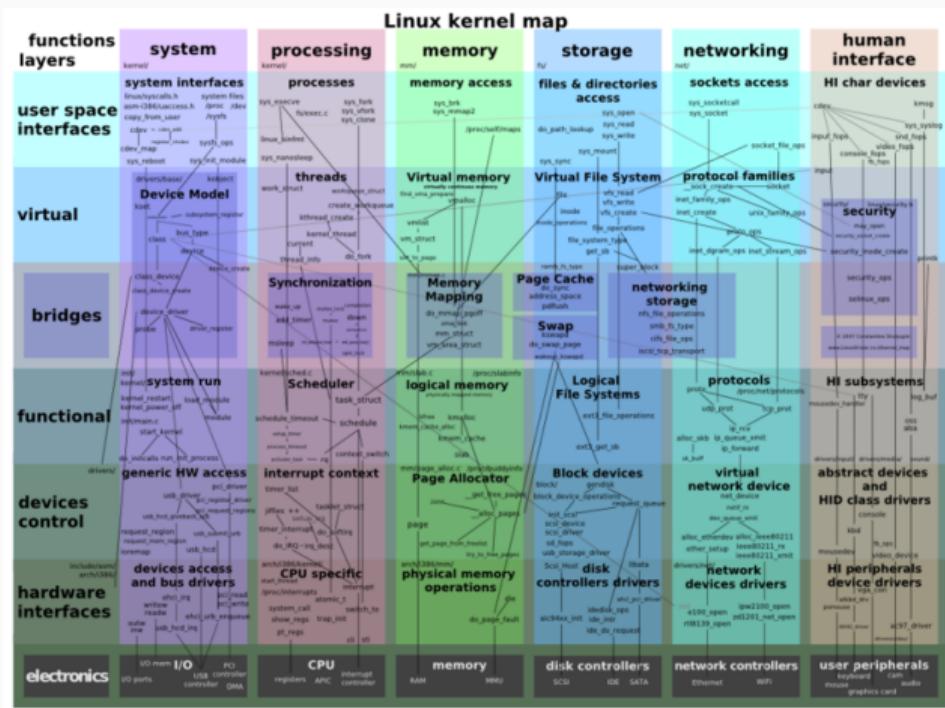
Linux kernel

Linux kernel diagram



Linux kernel diagram

Linux kernel



Interactive map of Linux kernel

SW CONCEPTS

Compilation process

Source Code

```
#include <stdio.h>

int main(){
    printf("Hello world\n");
    return 0;
}
```

```
$ hexdump hello_world
00000000 457f 464c 0102 0001 0000 0000 0000 0000
00000010 0002 003e 0001 0000 1040 0040 0000 0000
00000020 0040 0000 0000 0000 3928 0000 0000 0000
00000030 0000 0000 0040 0038 000d 0040 0020 001f
...
$ readelf --all hello_world
En-tête ELF:
    Magique: 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
    Classe: ELF64
    Version: 1 (actuelle)
    OS/ABI: UNIX - System V
...
$ ./hello_world
Hello world
```

Process Setup

Few steps...

- Load text...
- Load BSS, Data
- Setup heap/stack pointers
- Move pc to ... main()??

Process Setup

Few steps...

- Load text...
- Load BSS, Data
- Setup heap/stack pointers
- Move pc to ... main()?? No!

[http://jvns.ca/blog/2013/11/29/
what-happens-when-you-run-a-unix-program/](http://jvns.ca/blog/2013/11/29/what-happens-when-you-run-a-unix-program/)

What about an OS?

Few steps...

- Load text...
- Load BSS, Data
- Setup heap/stack pointers
- Move pc to ... main()??No!

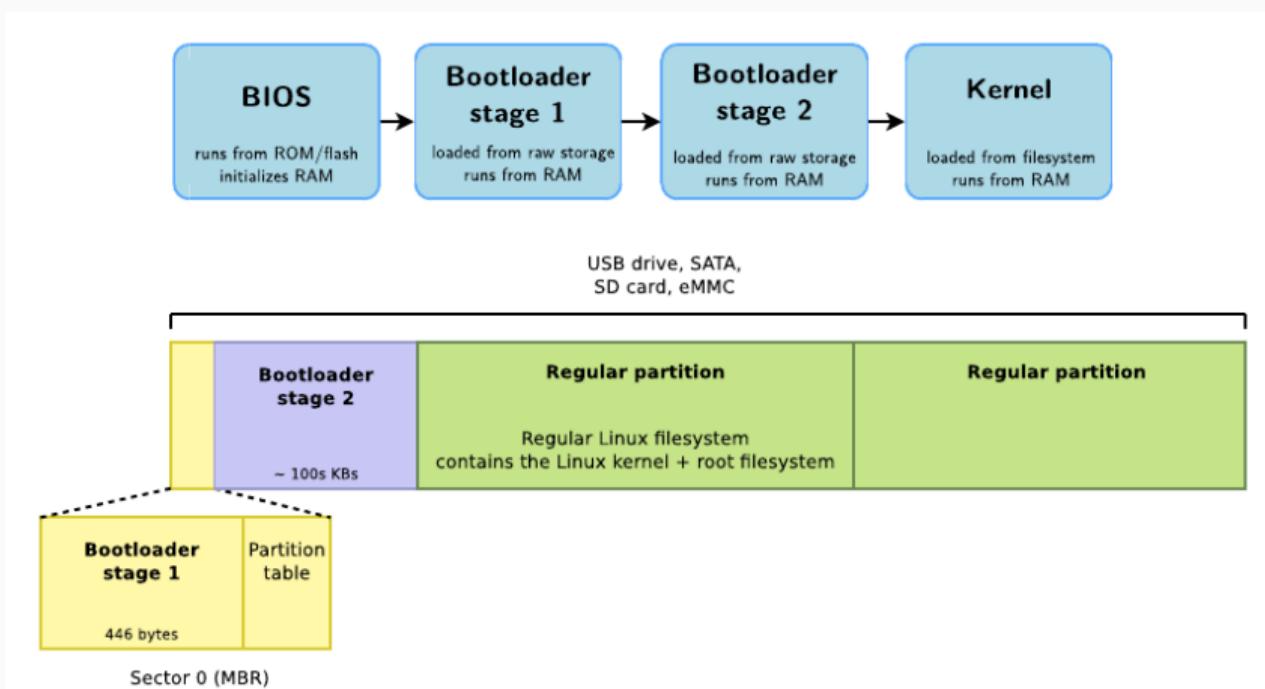
What about an OS?

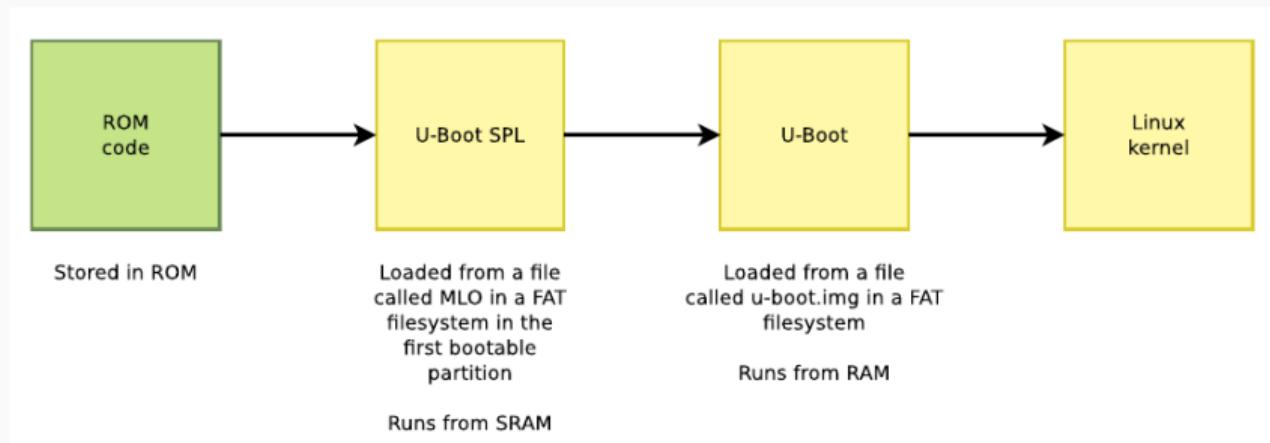
Few steps...

- Load text...
- Load BSS, Data
- Setup heap/stack pointers
- Move pc to ... main()??No!

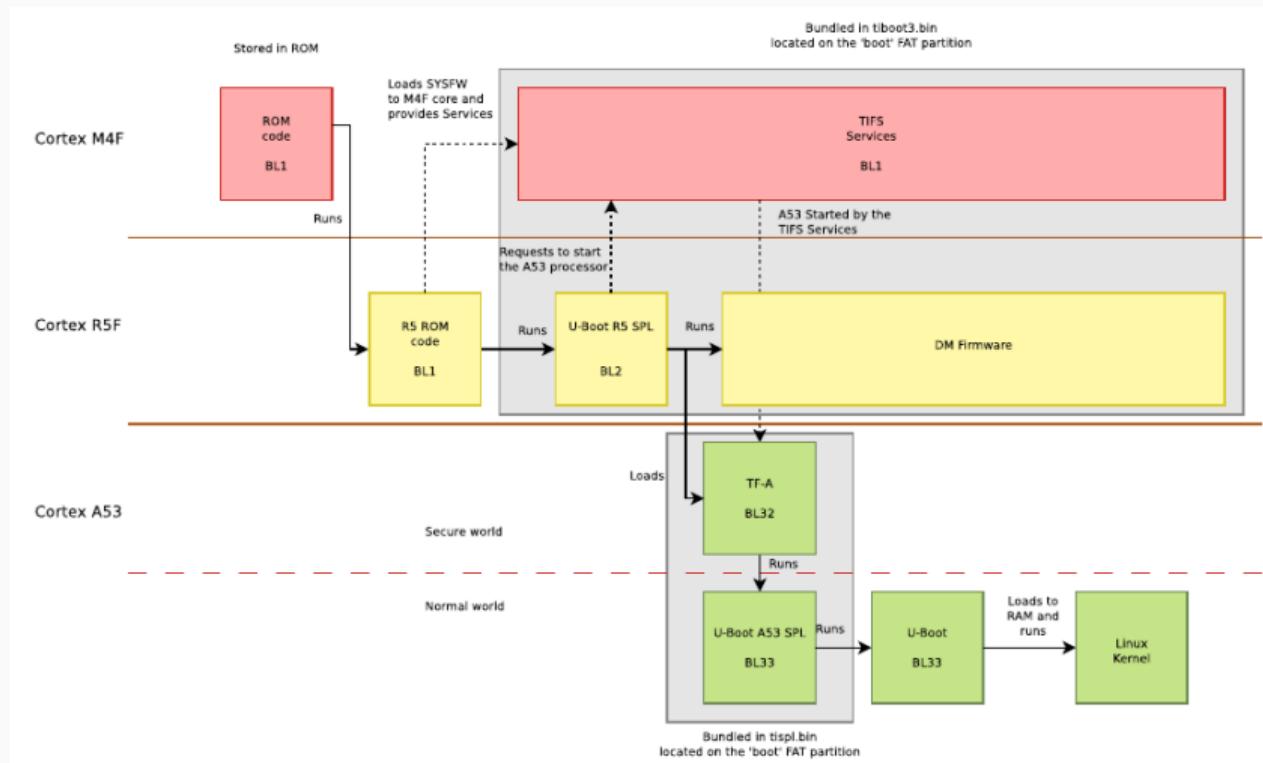
Wait a minute: Who does it???

x86 BIOS





Source



Debug

General

- JTAG emulator
- Bit banging (... protocols)
- UART / serial
- Printing
- gdb /debugger
- dump analysis
- Emulation (QEMU)
- curiosity, experience, luck
- ...

abstraction
↓
usefulness

OS specific

Linux:

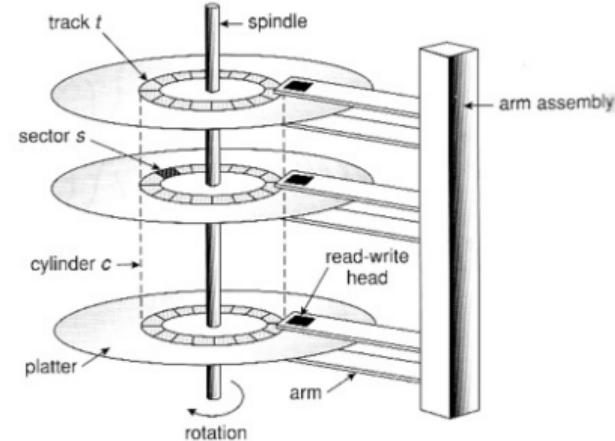
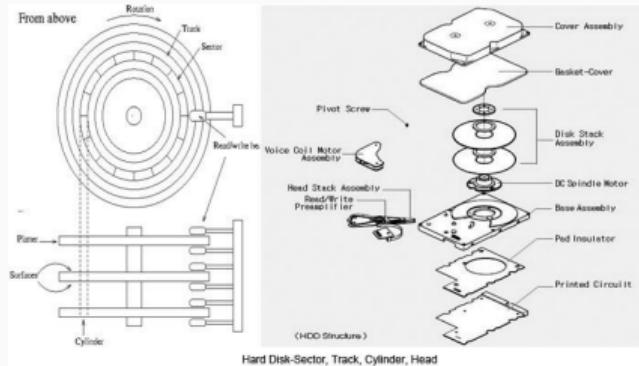
- printk
- [Linux Device Driver CH4](#)
- KGDB
- perf, ftrace, ...

Windows:

- ...

PUTTING IT TOGETHER

Example

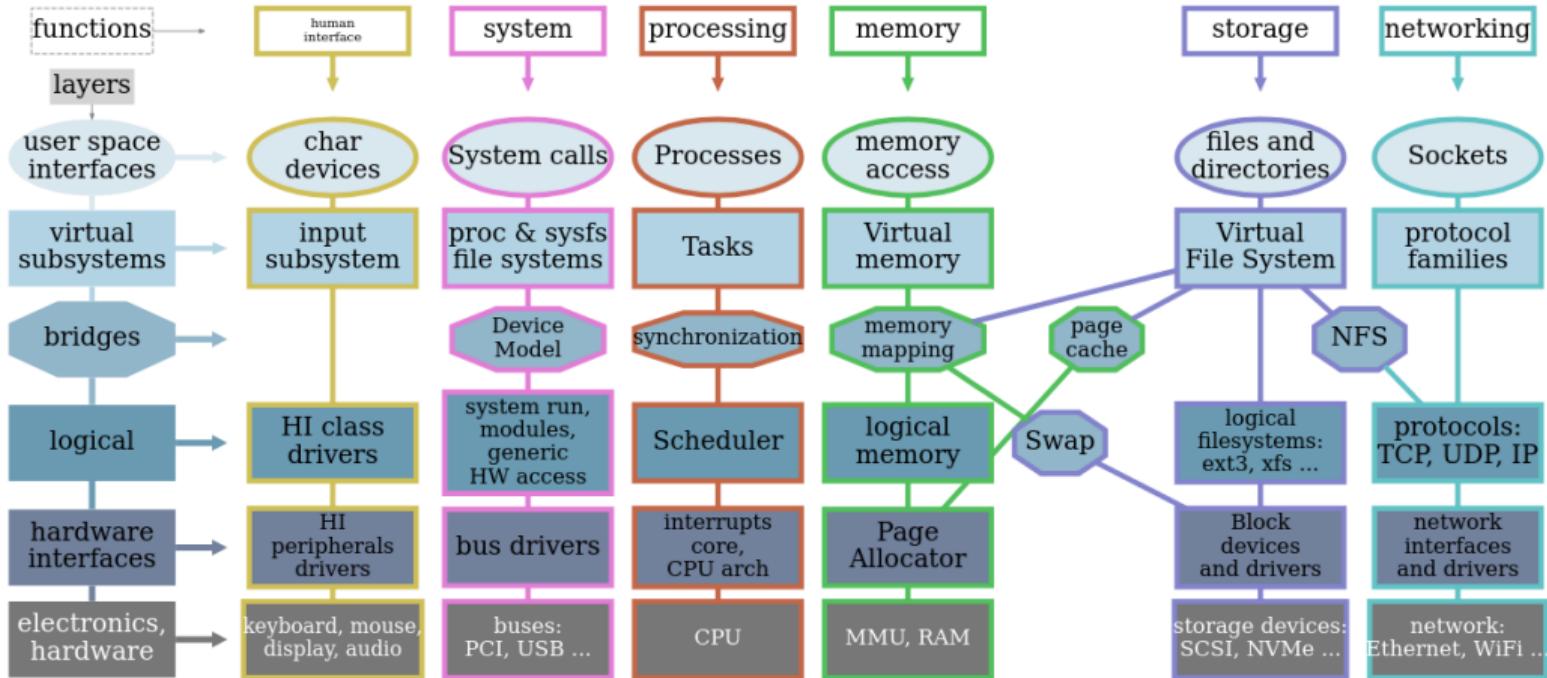


Interface POSIX

```
FILE *fp;  
char buff[1];  
  
fp = fopen("./test.txt", "a+");  
fread(buff, 1, 1, fp);  
fwrite(buff, 1, 1, fp);  
fclose(fp);
```

Linux kernel

Linux kernel diagram



REFERENCES

References

- google.com
- Linux Device Driver 3rd edition:
 - PDF: <http://lwn.net/Kernel/LDD3/>
 - HTML: <http://www.makelinux.net/ldd3/>
- <https://www.kernel.org/doc/html/latest/>
- <https://elixir.bootlin.com/linux/latest/source>
- <https://kernelnewbies.org/>