# Core Driver Infrastructure

## GPU Hardware Structure

Each ASIC is a collection of hardware blocks. We refer to them as "IPs" (Intellectual Property blocks). Each IP encapsulates certain functionality. IPs are versioned and can also be mixed and matched. E.g., you might have two different ASICs that both have System DMA (SDMA) 5.x IPs. The driver is arranged by IPs. There are driver components to handle the initialization and operation of each IP. There are also a bunch of smaller IPs that don't really need much if any driver interaction. Those end up getting lumped into the common stuff in the soc files. The soc files (e.g., vi.c, soc15.c nv.c) contain code for aspects of the SoC itself rather than specific IPs. E.g., things like GPU resets and register access functions are SoC dependent.

An APU contains more than just CPU and GPU, it also contains all of the platform stuff (audio, usb, gpio, etc.). Also, a lot of components are shared between the CPU, platform, and the GPU (e.g., SMU, PSP, etc.). Specific components (CPU, GPU, etc.) usually have their interface to interact with those common components. For things like S0i3 there is a ton of coordination required across all the components, but that is probably a bit beyond the scope of this section.

With respect to the GPU, we have the following major IPs:

GMC (Graphics Memory Controller)

This was a dedicated IP on older pre-vega chips, but has since become somewhat decentralized on vega and newer chips. They now have dedicated memory hubs for specific IPs or groups of IPs. We still treat it as a single component in the driver however since the programming model is still pretty similar. This is how the different IPs on the GPU get the memory (VRAM or system memory). It also provides the support for per process GPU virtual address spaces.

IH (Interrupt Handler)

This is the interrupt controller on the GPU. All of the IPs feed their interrupts into this IP and it aggregates them into a set of ring buffers that the driver can parse to handle interrupts from different IPs.

PSP (Platform Security Processor)

This handles security policy for the SoC and executes trusted applications, and validates and loads firmwares for other blocks.

SMU (System Management Unit)

This is the power management microcontroller. It manages the entire SoC. The driver interacts with it to control power management features like clocks, voltages, power rails, etc.

DCN (Display Controller Next)

This is the display controller. It handles the display hardware. It is described in more details in :ref:`Display Core <amdgpu-display-core>`.

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, **line 57**); ***backlink***

Unknown interpreted text role "ref".

SDMA (System DMA)

This is a multi-purpose DMA engine. The kernel driver uses it for various things including paging and GPU page table updates. It's also exposed to userspace for use by user mode drivers (OpenGL, Vulkan, etc.)

GC (Graphics and Compute)

This is the graphics and compute engine, i.e., the block that encompasses the 3D pipeline and and shader blocks. This is by far the largest block on the GPU. The 3D pipeline has tons of sub-blocks. In addition to that, it also contains the CP microcontrollers (ME, PFP, CE, MEC) and the RLC microcontroller. It's exposed to userspace for user mode drivers (OpenGL, Vulkan, OpenCL, etc.)

VCN (Video Core Next)

This is the multi-media engine. It handles video and image encode and decode. It's exposed to userspace for user mode drivers (VA-API, OpenMAX, etc.)

## Graphics and Compute Microcontrollers

CP (Command Processor)

The name for the hardware block that encompasses the front end of the GFX/Compute pipeline. Consists mainly of a bunch of microcontrollers (PFP, ME, CE, MEC). The firmware that runs on these microcontrollers provides the driver interface to interact with the GFX/Compute engine.

MEC (MicroEngine Compute)
    This is the microcontroller that controls the compute queues on the GFX/compute engine.

MES (MicroEngine Scheduler)
    This is a new engine for managing queues. This is currently unused.

RLC (RunList Controller)

This is another microcontroller in the GFX/Compute engine. It handles power management related functionality within the GFX/Compute engine. The name is a vestige of old hardware where it was originally added and doesn't really have much relation to what the engine does now.

## Driver Structure

In general, the driver has a list of all of the IPs on a particular SoC and for things like init/fini/suspend/resume, more or less just walks the list and handles each IP.

Some useful constructs:

KIQ (Kernel Interface Queue)
    This is a control queue used by the kernel driver to manage other gfx and compute queues on the GFX/compute engine. You can use it to map/unmap additional queues, etc.

IB (Indirect Buffer)
    A command buffer for a particular engine. Rather than writing commands directly to the queue, you can write the commands into a piece of memory and then put a pointer to the memory into the queue. The hardware will then follow the pointer and execute the commands in the memory, then returning to the rest of the commands in the ring.

## Memory Domains

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 127)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/uapi/drm/amdgpu_drm.h
   :doc: memory domains
```

## Buffer Objects

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 133)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_object.c
   :doc: amdgpu_object
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 136)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_object.c
   :internal:
```

## PRIME Buffer Sharing

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 142)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_dma_buf.c
   :doc: PRIME Buffer Sharing
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 145)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_dma_buf.c
   :internal:
```

## MMU Notifier

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 151)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_mn.c
   :doc: MMU Notifier
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 154)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_mn.c
   :internal:
```

## AMDGPU Virtual Memory

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 160)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_vm.c
   :doc: GPUVM
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 163)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_vm.c
   :internal:
```

## Interrupt Handling

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master] [Documentation] [gpu] [amdgpu]driver-core.rst, line 169)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_irq.c
   :doc: Interrupt Handling
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 172)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/amdgpu/amdgpu_irq.c
   :internal:
```

## IP Blocks

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 178)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/include/amd_shared.h
   :doc: IP Blocks
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\gpu\amdgpu\[linux-master][Documentation][gpu][amdgpu]driver-core.rst`, line 181)**

Unknown directive type "kernel-doc".

```
.. kernel-doc:: drivers/gpu/drm/amd/include/amd_shared.h
   :identifiers: amd_ip_block_type amd_ip_funcs
```