RK1108 DSP System Introduction

Sep., 2016
余智超

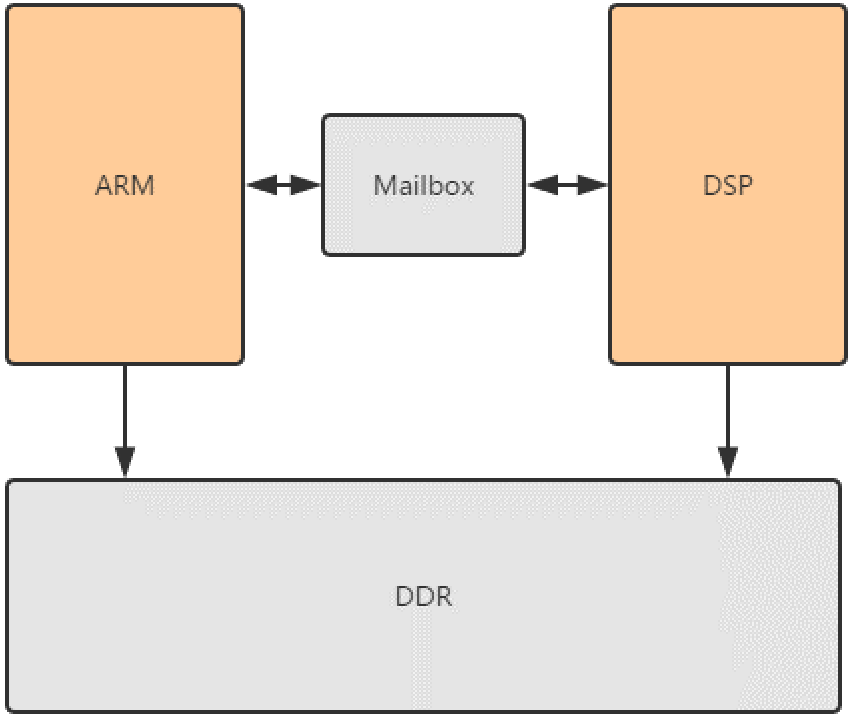
Agenda

- ➢Introduction
- ➢ DSP Software Basic Architecture
- ➢ Framework Layer DPP
- ➢ DSP Driver Introduction
- ➢ DSP System Introduction
- ➢ DSP Development Description
- ➢ Q&A

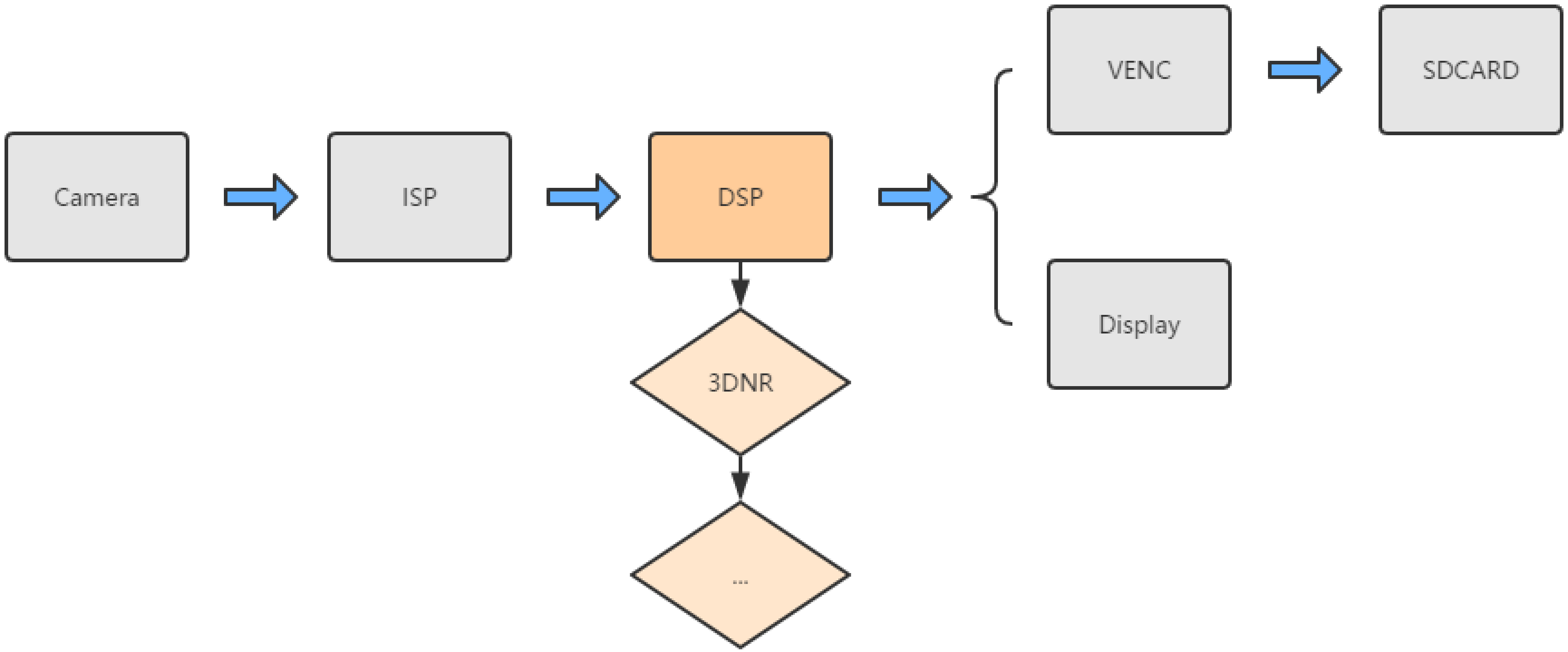
Introduction

RK1108 SoC DSP Architecture

- **DSP is the main arithmetic unit of RK1108**
- **DSP can visit external DDR ，with a wide range of applications**
- **DSP and ARM can communicate via Mailbox**
- **DSP has strong operation ability, 800MS calculating amount can be optimized to 20MS, even lower**
- **Can choose suitable DSP frequency according to algorithm complexity, to achieve optimized consumption**



DSP Application on CVR



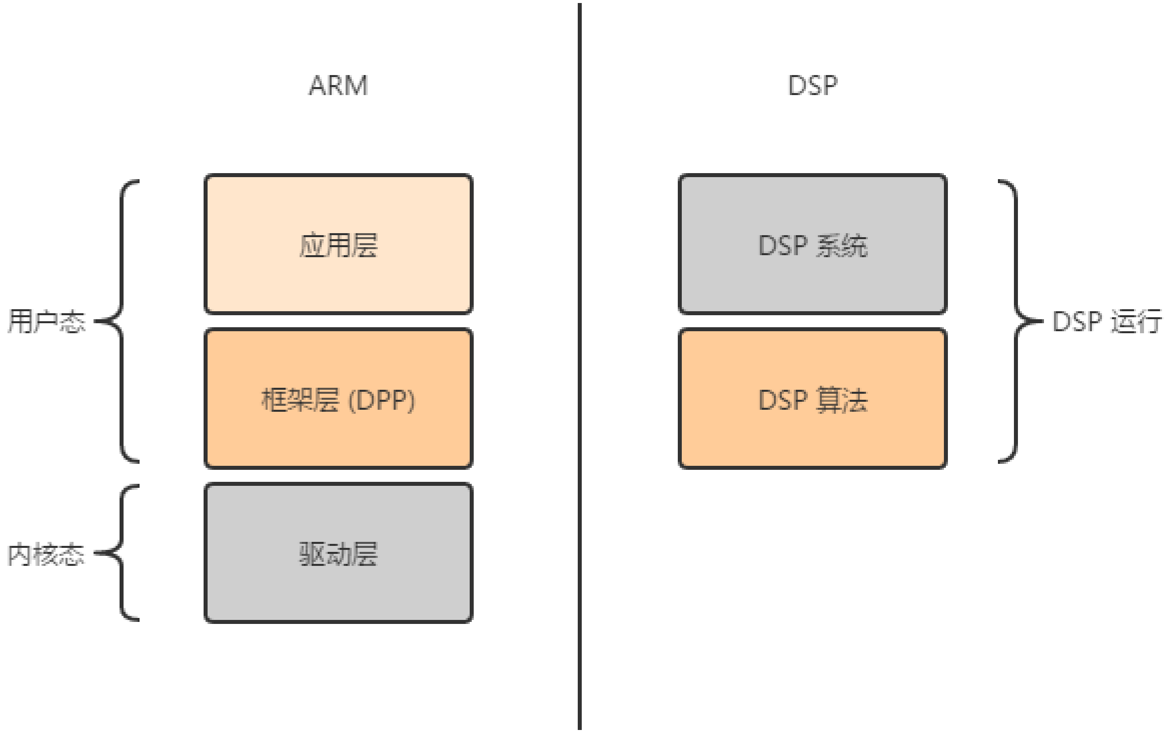
Reduce ISP output image noise via 3DNR

DSP Software Architecture Introduction

RK1108 DSP Software Architecture

Case 1 : If customers don ‘t need algorithm, only need to develop application layer

Case 2 : If customers need their own algorithm, need to develop application layer, framework layer and DSP algorithm.



DSP Software Layer Introduction

1、Application Layer

<SDKRoot>/app/video, realize CVR function，connect each modules, use 3DNR and ADAS algorithm to manage ISP output image data via DPP interface

2、Framework Layer（DPP）

<SDKRoot>/external/dpp，manage input output Buffer, manage work queue, prepare algorithm parameter, call DSP Kernel Driver. Currently, DPP source hasn’t opened yet, but already supports 3DNR and ADAS.

3、DSP Kernel Driver

<SDKRoot>/kernel/arch/arm/mach-rockchip/dsp，responsible for DSP loading, ARM and DSP communication, DSP device power status and managing work request of each process.

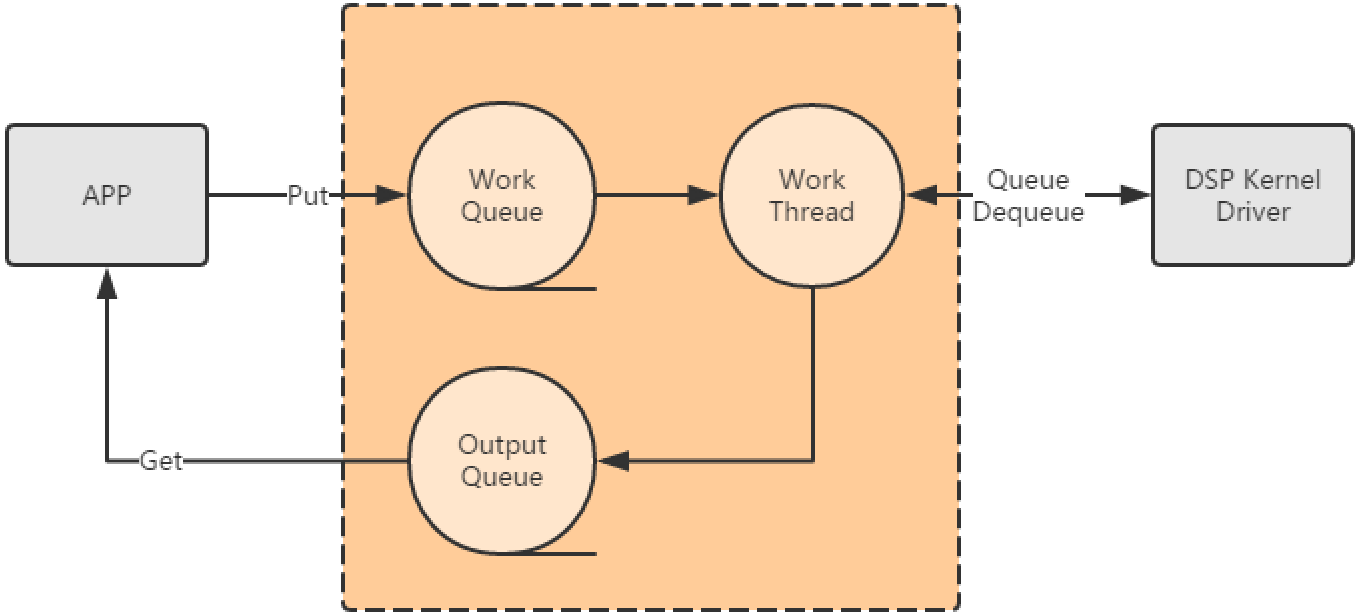
4、DSP System & DSP Algorithm

<SDKRoot>/external/dpp/out/firmware/rkdsp.bin，is the DSP real code，responsible for monitoring ARM work request, calling algorithm and returning processing results.

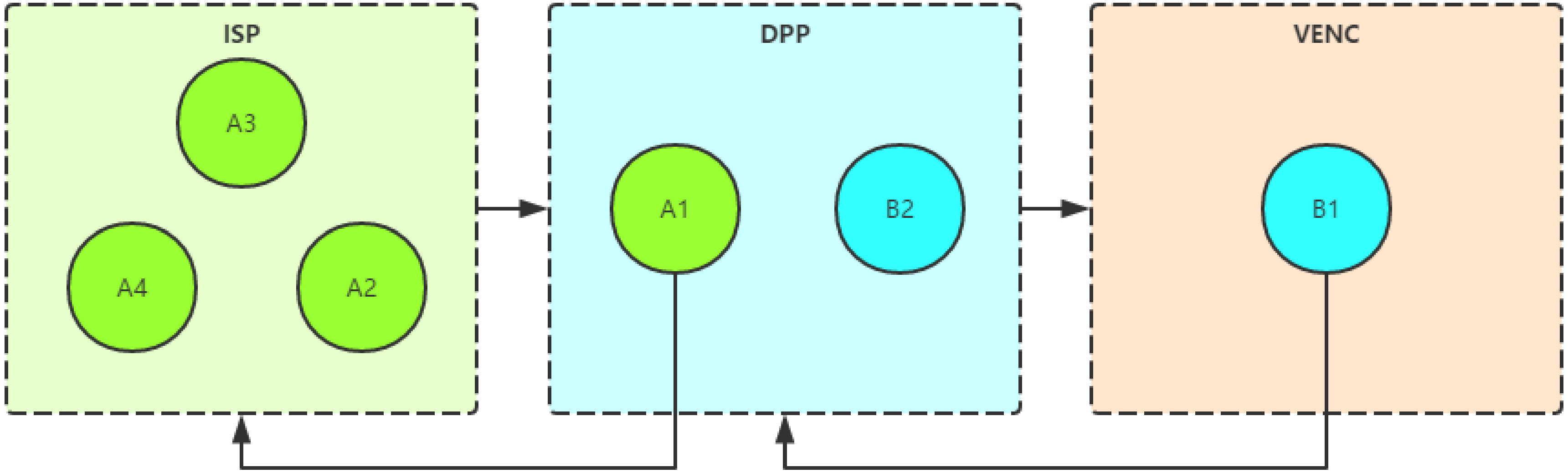
Framework Layer DPP

DSP Process Platform Introduction

- Manage Input and Output Buffer
- Manage Work Queue
- Prepare DSP Work Parameter
- Return DSP Processing Results to Application Layer



DPP Buffer Management



- ➢ **Need reduce Buffer application and copy**
- ➢ **Must handle Buffer reference relationship**
- ➢ **DSP Buffer must be physical continuous and allocated from CMA**

DPP Interface Description

DPP framework interface head file is under directory <SDKRoot>/external/dpp/inc

1、dpp_create() – Create dpp context，a dpp context is abstract of an algorithm.

2、dpp_start() – Start dpp work thread，dpp start processing work queue

3、dpp_packet_init() – Create dpp packet，packet includes all required algorithm parameters, configured by application layer

4、dpp_put_packet() – Add work request packet to dpp work queue

5、dpp_get_frame() – Get processing results from dpp

6、dpp_stop() – Stop dpp work thread

7、dpp_destroy() – Destroy dpp context

DSP Driver Introduction

DSP Kernel Driver Introduction

DSP Driver Main Functions :

- **Guide and Start DSP**
- **Manage different DSP progress work request**
- **Manage communication between ARM and DSP**
- **Manage DSP power status**
- **DSP algorithm parameters are always available for DSP driver, it is universal**
- **DSP driver is open resource, customer doesn’t need to change driver**

DSP Kernel Driver User Interface

- **Device Files/dev/dsp**

  User mode process operates DSP driver through open() interface to open DSP device file

- **IOCTL**

  Use ioctl() interface to call DSP driver IOCTL :

  1) DSP_IOC_QUEUE_WORK :

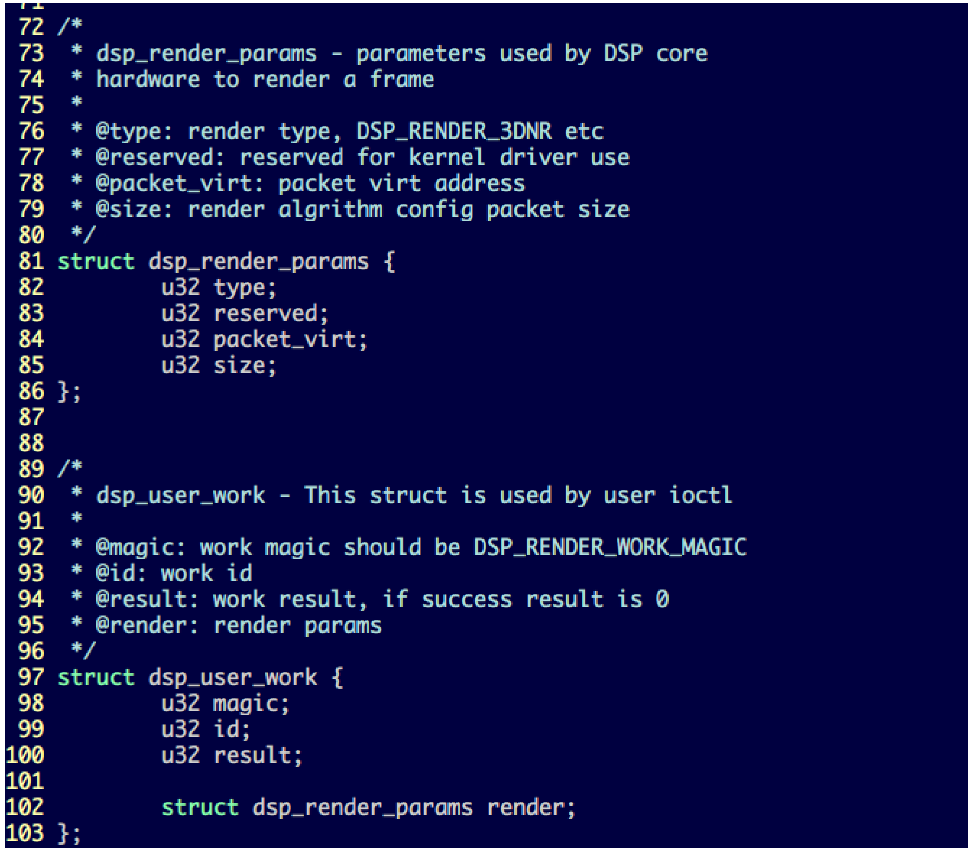
  Transmit work parameter struct dsp_user_work to DSP , to request DSP to call suitable algorithm

  2) DSP_IOC_DEQUEUE_WORK

  Get DSP operation results, this interface is block interface, it will wait for DSP processing results in kernel.

DSP Kernel Driver Work Parameter

- Packet is a container, customer can pass its own structure packet to DSP driver, driver will pass parameters to DSP
- Operation Buffer must be physically continuous
- DSP will select algorithm according to type value



```
/*
 * dsp_render_params - parameters used by DSP core
 * hardware to render a frame
 *
 * @type: render type, DSP_RENDER_3DNR etc
 * @reserved: reserved for kernel driver use
 * @packet_virt: packet virt address
 * @size: render algrithm config packet size
 */
struct dsp_render_params {
        u32 type;
        u32 reserved;
        u32 packet_virt;
        u32 size;
};


/*
 * dsp_user_work - This struct is used by user ioctl
 *
 * @magic: work magic should be DSP_RENDER_WORK_MAGIC
 * @id: work id
 * @result: work result, if success result is 0
 * @render: render params
 */
struct dsp_user_work {
        u32 magic;
        u32 id;
        u32 result;

        struct dsp_render_params render;
};
```

DSP Kernel Driver Debug Tips

- Trace DSP driver Log :

```
echo 0xffffffff > /sys/kernel/debug/dsp/debug
```

- Check DSP Staus :

```
io -4 -l 64 0x620bfc00
```

- Check whether DSP hang up or not :

```
io -4 -l 8 0x33400680
```

- Check Mailbox Message :

```
io -4 -l 128 0x33810000
```

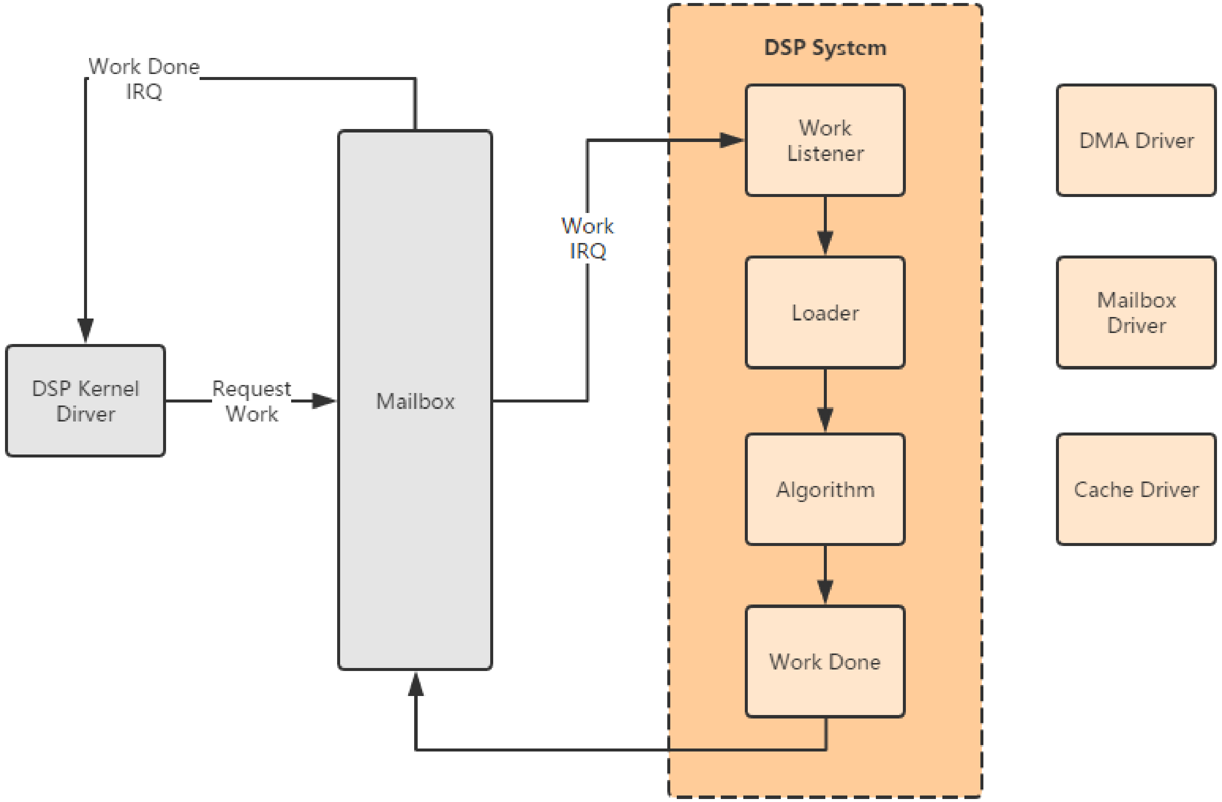
DSP System Introduction

DSP System Introduction

DSP System operates in PTCM inside DSP, its main features are as follow :

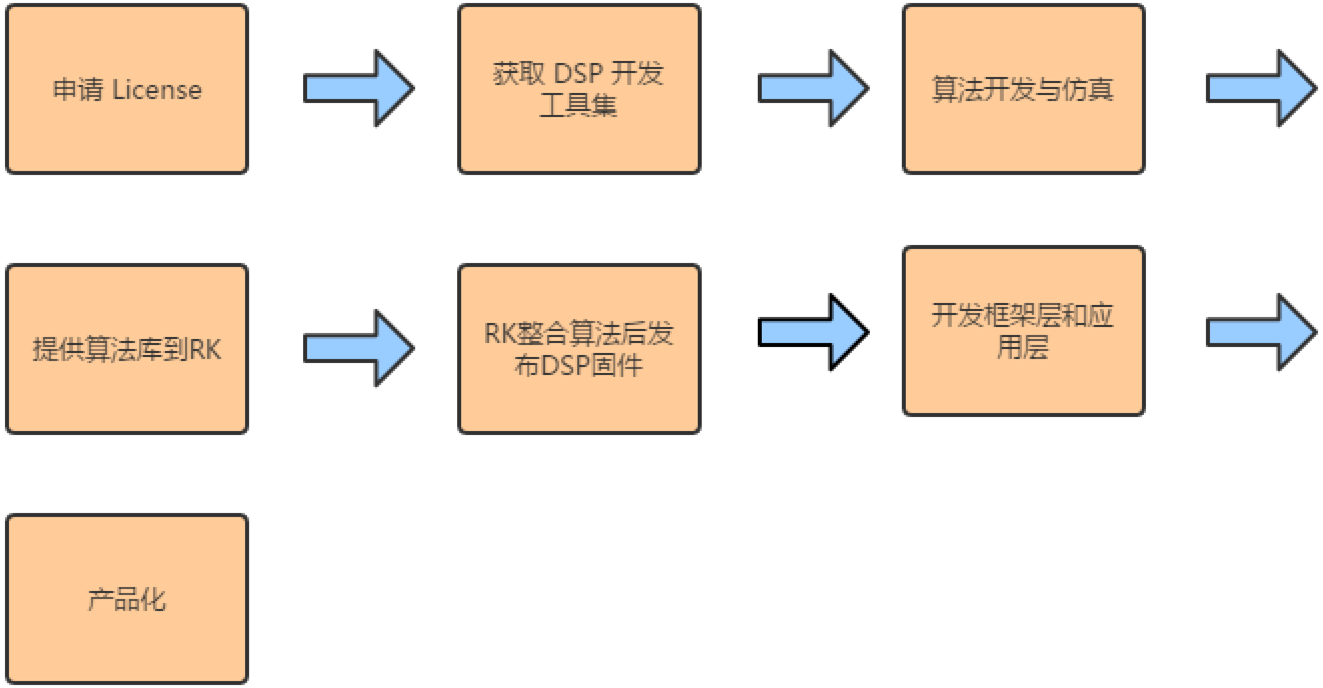
- **Monitor work request from Mailbox**
- **Responsible for loading code and data of different algorithm**
- **Call executing algorithm**
- **Return processing results to ARM via Mailbox**

DSP System Architecture



DSP Development Introduction

DSP Algorithm Development Workflow



申请 License

获取 DSP 开发工具集

算法开发与仿真

提供算法库到RK

RK整合算法后发布DSP固件

开发框架层和应用层

产品化

DSP Algorithm Development Tips

1、All code segment and data segment in algorithm must be named by RK defined segment name.

2、DSP has limited internal resource, so only put code and data occupied most calculating amount in internal DSP TCM, put other code and data in external DDR, PTCM is limited in 28K，DTCM is limited in 118K.

3、After get the License, RK will provide a basis algorithm development project, which contains necessary libraries and drivers. Suggest to develop algorithms based on RK foundation project.

4、After algorithm debugging is completed, use RK script to generate algorithm Library file.

5、Library interface unified defined format：int do_your_algorithm(void *params);

6、DSP firmware is packaged by RK,   in particular cases, customer can use firmware packing tool provide by RK to generate DSP firmware

7、Algorithm framework layer is optional, customer can also directly call DSP driver to realize DSP algorithm in application layer.

How to Develop DSP Algorithm?

Our colleague in Core Algorithm Dept. will introduce in anther course, mainly contains:

- **CEVA-XM4 Framework Introduction**
- **Algorithm Realization**
- **How to Optimize Algorithm in DSP**

Q&A

Thank you!