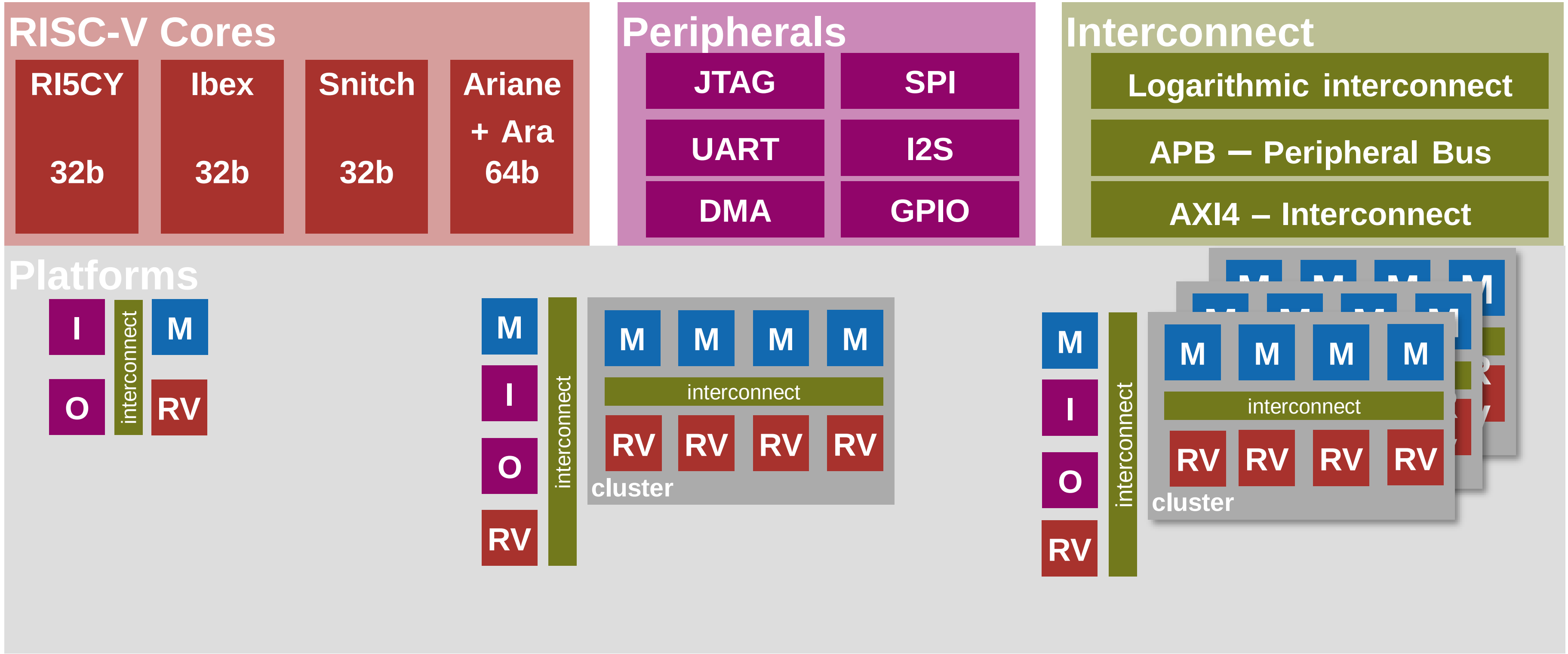
PULP PLATFORM

Open Source Hardware, the way it should be!

# *PULP SDK and PMSIS*

Nazareno Bruschi <nazareno.bruschi@unibo.it>
Germain Haugou <germain.haugou@iis.ee.ethz.ch>

# PULP Architectures



RISC-V Cores

- RI5CY 32b
- Ibex 32b
- Snitch 32b
- Ariane + Ara 64b

Peripherals

- JTAG
- SPI
- UART
- I2S
- DMA
- GPIO

Interconnect

- Logarithmic interconnect
- APB – Peripheral Bus
- AXI4 – Interconnect

Platforms

# Source the target architecture

- $ source configs/pulp-open.sh
- configs/pulp-open.sh
- …
- export BOARD_NAME=pulp
- export PULP_CURRENT_CONFIG=$BOARD_NAME@config_file=config/$BOARD_NAME.json
- …
- export PULPOS_TARGET=pulp
- …
- export GAPY_TARGET=pulp
- …
- source $PULP_SDK_HOME/configs/common.sh

# Configuration files

- **tools/gap-configs/configs/chips/pulp**
    - pulp.json
    - udma.json

# Configuration files

- **tools/gap-configs/configs/chips/pulp**
  - pulp.json
  - udma.json

```
"soc": {

   "base": "0x1A000000",
   "size": "0x06000000",

   "fc": {
     "core"        : "ri5ky_v2_6_sfloat_single_regfile",
     "cluster_id"  : 31,
     "core_id"     : 0,
     "fetch_enable": true,
     "boot_addr"   : "0x1A000080",
…
```

```
"cluster": {
   "base": "0x10000000",
   "alias": "0x00000000",
   "size": "0x00400000",
   "core": "ri5ky_v2_6_sfloat_single_regfile",
   "version": 5,
   "json_file": "cluster_v5_fpu",
   "nb_cluster": 1,
   "nb_pe": 8,
…
"udma": {
       "content"      : "chips/pulp/udma.json",
```

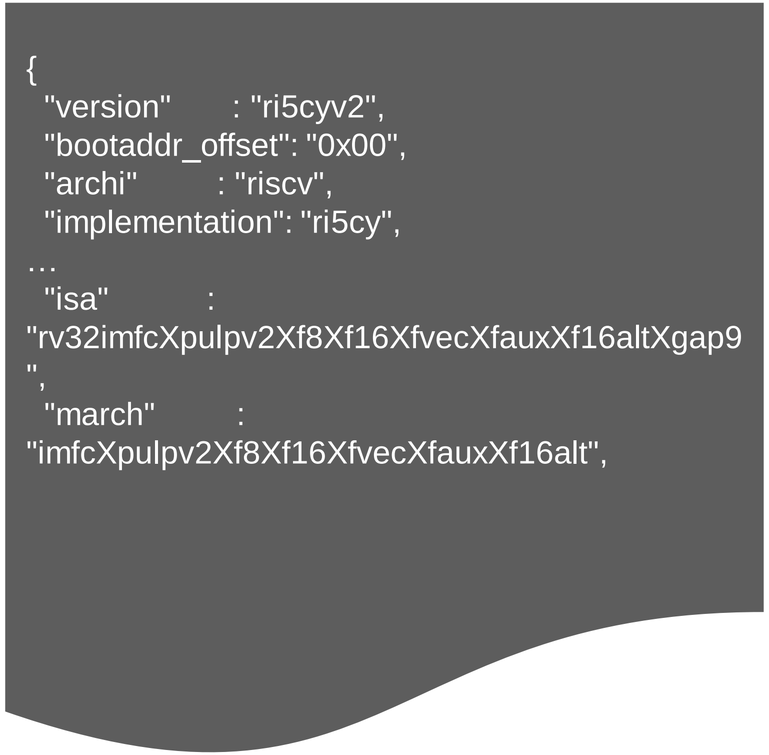
# Configuration files

- tools/gap-configs/configs/chips/pulp
  - pulp.json
  - udma.json

```
 "@includes@": [ "ips/udma/udma_v3.json"],

  "vp_impl": "pulp.udma.udma_v3_pulp_impl",
  "vp_component":
"pulp.udma.udma_v3_pulp_impl",

  "nb_periphs": 8,

  "interfaces" : ["spim", "i2c", "i2s", "uart", "cpi",
"hyper"],

  "properties": {
    "l2_read_fifo_size": 8
  },
```

# Configuration files

- **tools/gap-configs/configs/ips/**
  - riscv/ri5ky_v2_6_sfloat_single_regfile.json



```
{
  "version"       : "ri5cyv2",
  "bootaddr_offset": "0x00",
  "archi"         : "riscv",
  "implementation": "ri5cy",
…
  "isa"           :
"rv32imfcXpulpv2Xf8Xf16XfvecXfauxXf16altXgap9
",
  "march"         :
"imfcXpulpv2Xf8Xf16XfvecXfauxXf16alt",
```

# Makefiles

- tools/rules/pmsis_rules.mk
- …
- rtos/pulpos/pulp/rules/pulpos/targets/pulp.mk

```
CONFIG_NB_CLUSTER_PE ?= 8

PULP_LDFLAGS      +=
PULP_CFLAGS       +=  -D__riscv__
PULP_ARCH_CFLAGS ?=  -march=rv32imcxgap9 -mPE=$(CONFIG_NB_CLUSTER_PE) -mFC=1
PULP_ARCH_LDFLAGS ?=  -march=rv32imcxgap9 -mPE=$(CONFIG_NB_CLUSTER_PE) -mFC=1
PULP_ARCH_OBJDFLAGS ?= -Mmarch=rv32imcxgap9
PULP_CFLAGS    += -fdata-sections -ffunction-sections -include pos/chips/pulp/config.h -I$(PULPOS_PULP_HOME)/include/pos/chips/pulp
PULP_OMP_CFLAGS    += -fopenmp -mnativeomp
PULP_LDFLAGS += -nostartfiles -nostdlib -Wl,--gc-sections -L$(PULPOS_PULP_HOME)/kernel -Tchips/pulp/link.ld -lgcc

PULP_CC = riscv32-unknown-elf-gcc
PULP_AR ?= riscv32-unknown-elf-ar
PULP_LD ?= riscv32-unknown-elf-gcc
PULP_OBJDUMP ?= riscv32-unknown-elf-objdump
```

# GAPY – PULP Runner

- **tools/gap-configs/gapy/targets/**
  - pulp.json

```
{
    "@include@": "common.json",

    "gvsoc": {
        "runner_module": "gv.chips.pulp"
    },

    "rtl": {
        "runner_module": "runner.rtl.chips.pulp"
    },

    "target": {
        "@includes@": [ "chips/pulp/pulp.json" ]
    },

    "runner": {
        "flash_devices": [
          "target/board/devices/flash",
          "target/board/devices/spiflash"
        ],
        "boot": {
            "mode": "flash",
            "device": "target/board/devices/spiflash"
        }
    }
}
```

# Example – New PULP-16 target

- New target
  - configs/pulp-open-16.sh

```
…
export BOARD_NAME=pulp-16
…
export PULPOS_TARGET=pulp-16
…
export GAPY_TARGET=pulp-16
```

# Example – New PULP-16 target

- New target
  - configs/pulp-open-16.sh
- New file
  - tools/gap-configs/configs/chips/pulp/pulp-16.json

```
 "cluster": {
    "base": "0x10000000",
    "alias": "0x00000000",
    "size": "0x00400000",
    "core": "ri5ky_v2_6_sfloat_single_regfile",
    "version": 5,
    "json_file": "cluster_v5_fpu",
    "nb_cluster": 1,
    "nb_pe": 16,
…
 "udma": {
        "content"      : "chips/pulp/udma.json",
```

# Example – New PULP-16 target

- New target
  - configs/pulp-open-16.sh
- New file
  - tools/gap-configs/configs/chips/pulp/pulp-16.json
- New symbolic link
  - $ cd tools/gap-configs/configs/config/
  - $ ln –s pulp-16.json ../chips/pulp/pulp-16.json

# Example – New PULP-16 target

- New target
  - configs/pulp-open-16.sh
- New file
  - tools/gap-configs/configs/chips/pulp/pulp-16.json
- New symbolic link
  - $ cd tools/gap-configs/configs/config/
  - $ ln –s pulp-16.json ../chips/pulp/pulp-16.json
- New target-specific rules
  - rtos/pulpos/pulp/rules/pulpos/targets/pulp-16.mk

```
CONFIG_NB_CLUSTER_PE ?= 16

PULP_LDFLAGS      +=
PULP_CFLAGS       +=  -D__riscv__
PULP_ARCH_CFLAGS ?=  -march=rv32imcxgap9 -
mPE=$(CONFIG_NB_CLUSTER_PE) -mFC=1
PULP_ARCH_LDFLAGS ?=  -march=rv32imcxgap9 -
mPE=$(CONFIG_NB_CLUSTER_PE) -mFC=1
PULP_ARCH_OBJDFLAGS ?= -Mmarch=rv32imcxgap9
PULP_CFLAGS     += -fdata-sections -ffunction-sections -include
pos/chips/pulp/config.h -I$(PULPOS_PULP_HOME)/include/pos/chips/pulp
PULP_OMP_CFLAGS    += -fopenmp -mnativeomp
PULP_LDFLAGS += -nostartfiles -nostdlib -Wl,--gc-sections -
L$(PULPOS_PULP_HOME)/kernel -Tchips/pulp/link.ld -lgcc

PULP_CC = riscv32-unknown-elf-gcc
PULP_AR ?= riscv32-unknown-elf-ar
PULP_LD ?= riscv32-unknown-elf-gcc
PULP_OBJDUMP ?= riscv32-unknown-elf-objdump
```

# Example – New PULP-16 target

- New target
  - configs/pulp-open-16.sh
- New file
  - tools/gap-configs/configs/chips/pulp/pulp-16.json
- New symbolic link
  - $ cd tools/gap-configs/configs/config/
  - $ ln –s pulp-16.json ../chips/pulp/pulp-16.json
- New target-specific rules
  - rtos/pulpos/pulp/rules/pulpos/targets/pulp-16.mk
- New file
  - tools/gapy/targets/pulp-16.json

```
{
    "@include@": "common.json",

    "gvsoc": {
        "runner_module": "gv.chips.pulp"
    },

    "rtl": {
        "runner_module": "runner.rtl.chips.pulp"
    },

    "target": {
        "@includes@": [ "chips/pulp/pulp-16.json" ]
    },

    "runner": {
        "flash_devices": [
          "target/board/devices/flash",
          "target/board/devices/spiflash"
        ],
        "boot": {
            "mode": "flash",
            "device": "target/board/devices/spiflash"
        }
    }
}
```

# Example – New PULP-16 target

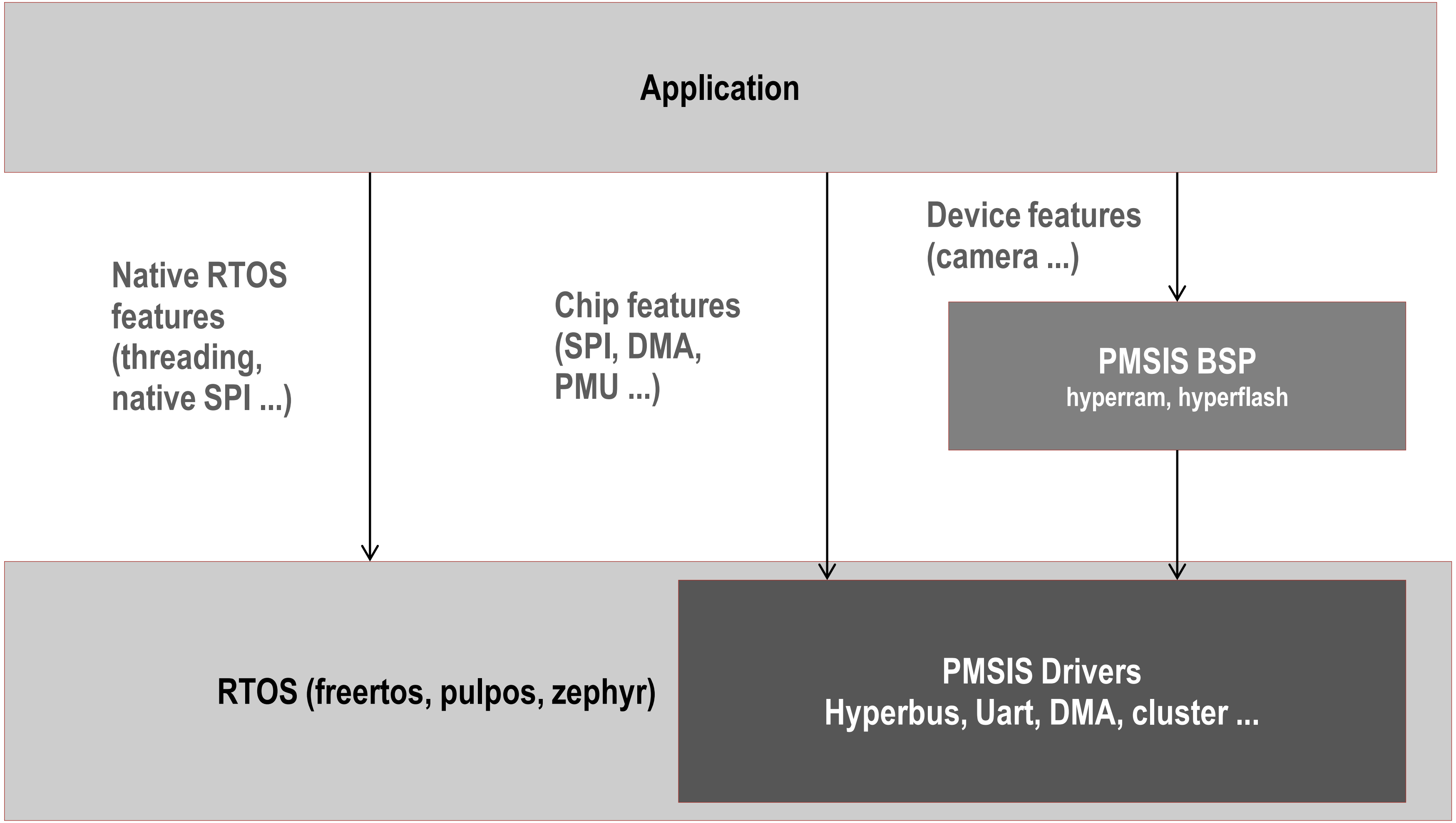
- New target
  - configs/pulp-open-16.sh
- New file
  - tools/gap-configs/configs/chips/pulp/pulp-16.json
- New symbolic link
  - $ cd tools/gap-configs/configs/config/
  - $ ln –s pulp-16.json ../chips/pulp/pulp-16.json
- New target-specific rules
  - rtos/pulpos/pulp/rules/pulpos/targets/pulp-16.mk
- New file
  - tools/gapy/targets/pulp-16.json

- **$ source configs/pulp-open-16.sh**
- **$ cd tests/hello**
- **$ make clean all run USE_CLUSTER=1 NB_CORES=16**

# Example – New PULP-16 target (FASTER)

- **Just add in Application Makefile**
  - CONFIG_NB_CLUSTER_PE=16
- **and modify**
  - tools/gap-configs/configs/chips/pulp.json with nb_pe: 16
- **$ cd tests/hello**
- **$ make clean all run USE_CLUSTER=1 NB_CORES=16**

# PMSIS – PULP Runtime



Application
Native RTOS features (threading, native SPI ...)
Chip features (SPI, DMA, PMU ...)
Device features (camera ...)
PMSIS BSP
hyperram, hyperflash
RTOS (freertos, pulpos, zephyr)
PMSIS Drivers
Hyperbus, Uart, DMA, cluster ...

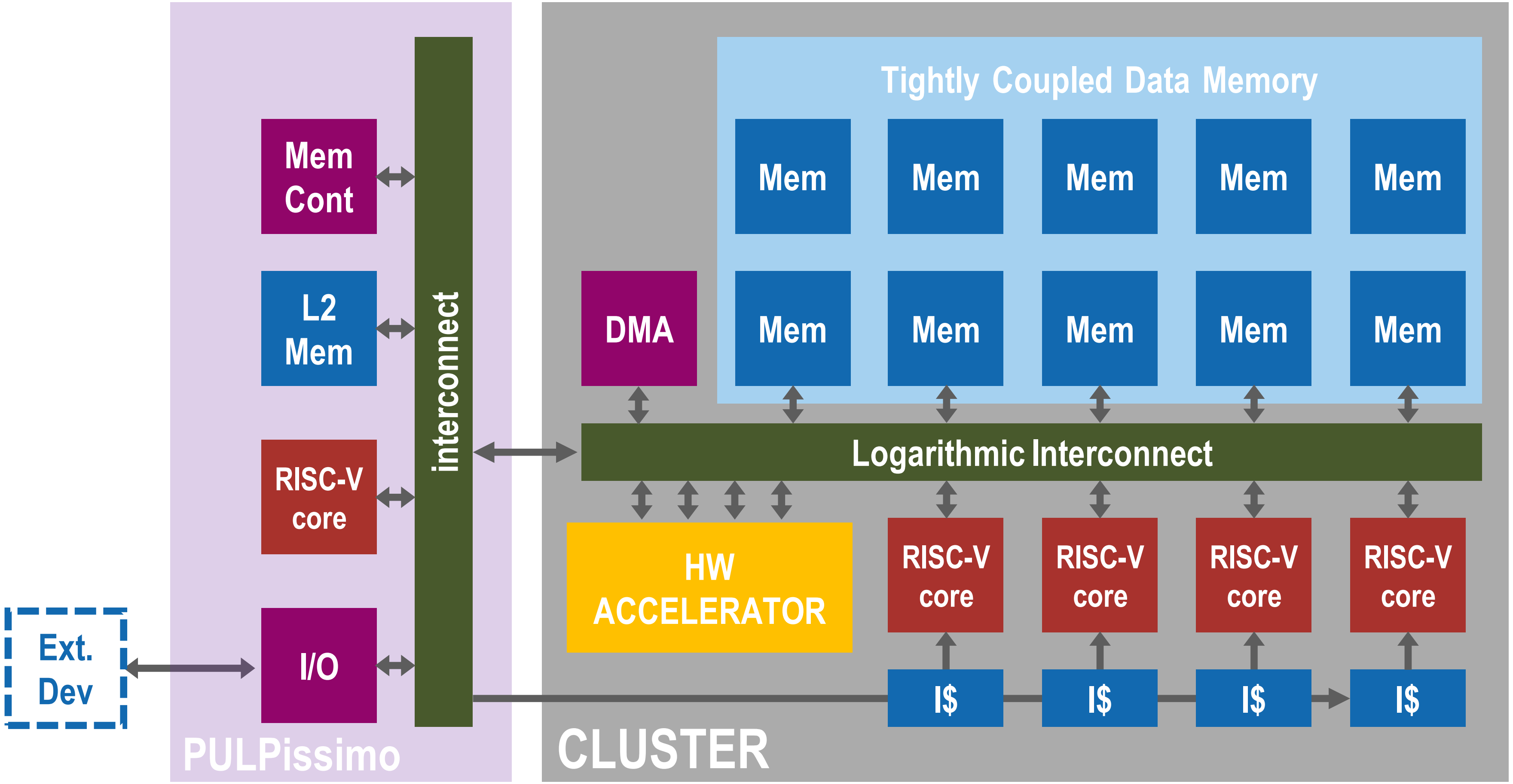
# PULP Devices

```
typedef struct pi_device {
        struct pi_device_api *api;
        void *config;
        void *data;
} pi_device_t;
```

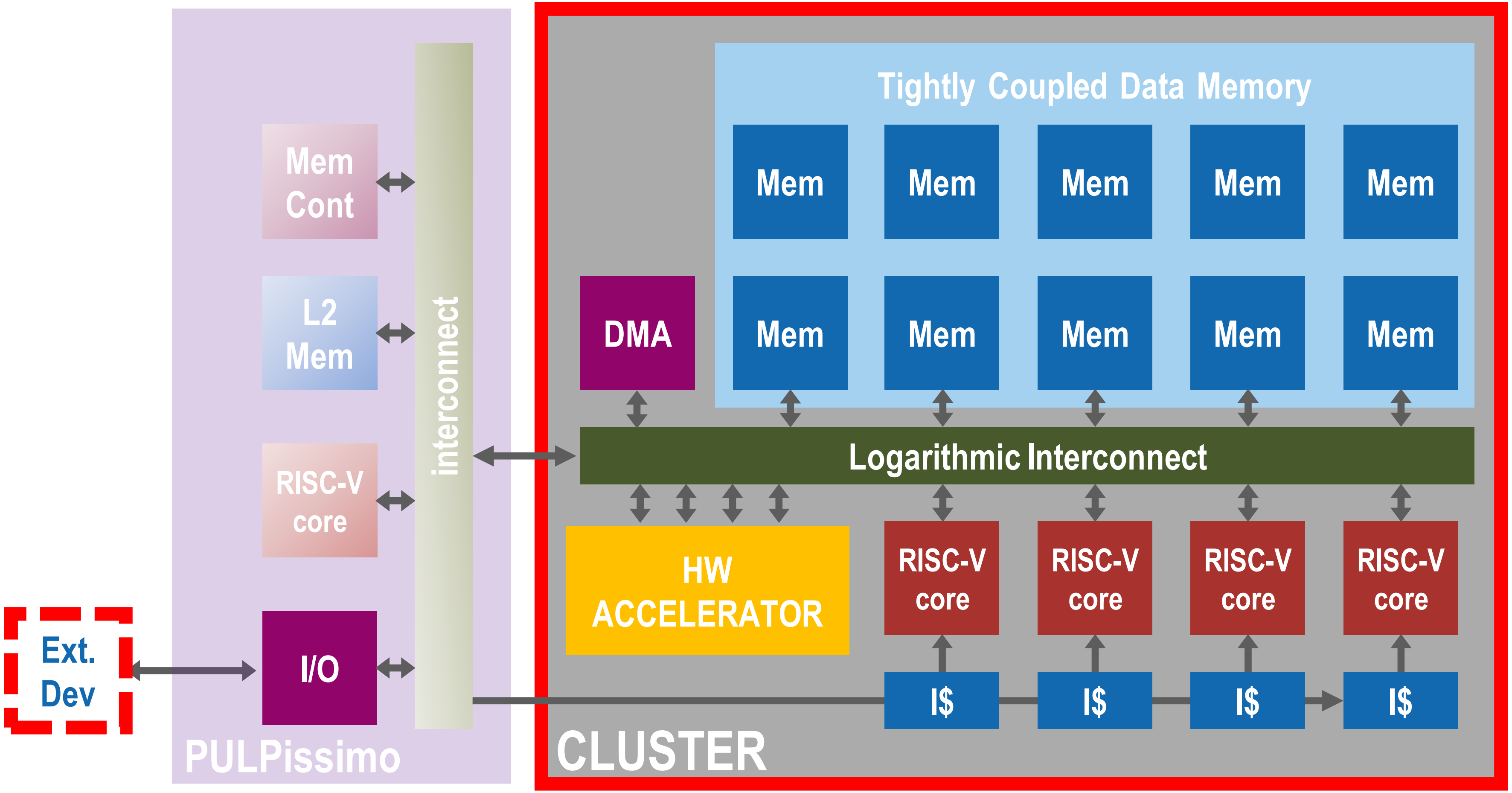
- Basic structure that contains details about the specific device
- Generic functions extract specific information from device description

- rtos/pmsis/pmsis_api/include/pmsis/pmsis_types.h

# PULP Devices



# PULP Devices



# PULP Devices: How to manage them

- conf_init()
- open_from_conf()
- open()
- .. does something ..
- close()

- Device specific (cluster/ram/flash)
- Initialize the pi_device to be used in the program
- Release the resources

# Example – Cluster

- struct pi_device cluster;
- struct pi_cluster_conf cluster_conf;
- pi_cluster_conf_init(&cluster_conf);
- pi_open_from_conf(&cluster, &cluster_conf);
- pi_cluster_open(&cluster);
- rtos/pulpos/pulp/drivers/cluster/cluster.c

# Example – HYPERRAM

- Requires BSP: #include <bsp/bsp.h>
- struct pi_device ram;
- struct pi_hyperram_conf ram_conf;
- pi_hyperram_conf_init(&ram_conf);
- pi_open_from_conf(&ram, &ram_conf);
- pi_ram_open(&ram);
- rtos/pmsis/pmsis_bsp/ram/ram.c

# Init – Cluster

```
void pi_cluster_conf_init(struct pi_cluster_conf *conf) {
    conf->id = 0;
    conf->flags = PI_CLUSTER_FLAGS_FORK_BASED;
}
```

- rtos/pulpos/pulp/drivers/cluster/cluster.c

# Init – HYPERRAM

-
  ```
  void pi_hyperram_conf_init(struct pi_hyperram_conf *conf) {
      conf->ram.api = &hyperram_api;
      conf->baudrate = 0;
      conf->xip_en = 0;
      conf->reserve_addr_0 = 1;
      bsp_hyperram_conf_init(conf);
  }
  ```
- rtos/pmsis/pmsis_bsp/ram/hyperram/hyperram.c

# BSP Init – HYPERRAM

```
void bsp_hyperram_conf_init(struct pi_hyperram_conf *conf) {
        conf->ram_start = CONFIG_HYPERRAM_START;
        conf->ram_size = CONFIG_HYPERRAM_SIZE;
        conf->skip_pads_config = 0;
        conf->hyper_itf = CONFIG_HYPERRAM_HYPER_ITF;
        conf->hyper_cs = CONFIG_HYPERRAM_HYPER_CS;
}
```

- rtos/pmsis/pmsis_bsp/bsp/pulp.c

# Open from Conf

- rtos/pulpos/common/kernel/device.c

```
void pi_open_from_conf(struct pi_device *device, void *conf)
{
    device->config = conf;
}
```

# Doing something with PULP Devices

- **Chip-specific drivers: rtos/pulpos/pulp/drivers/**
  - cluster/cluster.c
  - hyperbus/hyperbus-v3.c
  - uart/uart-v1.c

# Doing something with PULP Devices

- Chip-specific drivers: rtos/pulpos/pulp/drivers/
  - cluster/cluster.c
  - hyperbus/hyperbus-v3.c
  - uart/uart-v1.c

```
static int pos_cluster_init()
{
  pos_irq_set_handler(POS_EVENT_FC_ENQUEUE, pos_task_remote_enqueue);

  pos_irq_mask_set(1<<POS_EVENT_FC_ENQUEUE);

  return 0;
}
…
int pi_cluster_send_task_to_cl(struct pi_device *device, struct pi_cluster_task
*task)
{
    pi_task_t fc_task;
    pi_task_block(&fc_task);

    if (pi_cluster_send_task_to_cl_async(device, task, &fc_task))
    {
        return -1;
    }

    pi_task_wait_on(&fc_task);
    return 0;
}
```

# Doing something with PULP Devices

- Chip-specific drivers: rtos/pulpos/pulp/drivers/
  - cluster/cluster.c
  - hyperbus/hyperbus-v3.c
  - uart/uart-v1.c

```
void pi_hyper_conf_init(struct pi_hyper_conf *conf)
{
  conf->id = 0;
}
…
void pi_hyper_read(struct pi_device *device, uint32_t hyper_addr, void *addr,
uint32_t size)
{
  pi_task_t task;
  pi_hyper_read_async(device, hyper_addr, addr, size, pi_task_block(&task));
  pi_task_wait_on(&task);
}
…
int pi_hyper_id_alloc(struct pi_device *device)
{
  pos_hyper_t *hyper = (pos_hyper_t *)device->data;
  return plp_hyper_id_alloc(hyper->hyper_id);
}
```

# Doing something with PULP Devices

- Chip-specific drivers: rtos/pulpos/pulp/drivers/
  - cluster/cluster.c
  - hyperbus/hyperbus-v3.c
  - uart/uart-v1.c

```
int pi_uart_write_async(struct pi_device *device, void *buffer, uint32_t size,
pi_task_t *task)
{
  pos_uart_t *uart = (pos_uart_t *)device->data;
  pos_udma_enqueue(&uart->tx_channel, task, (uint32_t)buffer, size,
UDMA_CHANNEL_CFG_SIZE_8);
  return 0;
}
…
int pi_uart_read_async(struct pi_device *device, void *buffer, uint32_t size,
pi_task_t *task)
{
  pos_uart_t *uart = (pos_uart_t *)device->data;
  pos_udma_enqueue(&uart->rx_channel, task, (uint32_t)buffer, size,
UDMA_CHANNEL_CFG_SIZE_8);
  return 0;
}
…
int pi_uart_write_byte(pi_device_t *device, uint8_t *byte)
{
  int ret = pi_uart_write(device, byte, 1);
  return ret;
}
```

# Doing something with PULP Devices

- Chip-specific drivers: rtos/pulpos/pulp/drivers/
  - cluster/cluster.c
  - hyperbus/hyperbus-v3.c
  - uart/uart-v1.c
- Drivers map to basic functionalities (HAL) using architecture information (ARCHI)
- ARCHI: rtos/pulpos/pulp_archi/include/archi/
- HAL: rtos/pulpos/pulp_hal/include/hal

# PULP ARCHI

- **rtos/pulpos/pulp_archi/include/archi/**
- **Chip-specific: ./chips/pulp/**
  - pulp.h
  - properties.h
  - memory_map.h

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h

```
#include "archi/chips/pulp/properties.h"
#include "archi/chips/pulp/apb_soc_ctrl.h"

#include "archi/gpio/gpio_v3.h"
#include "archi/riscv/priv_1_10.h"
#include "archi/riscv/pcer_v2.h"
#include "archi/itc/itc_v1.h"

#include "archi/chips/pulp/memory_map.h"
#include "archi/chips/pulp/apb_soc_ctrl/apb_soc_ctrl.h"
#include "archi/chips/pulp/cluster_ctrl_unit/cluster_ctrl_unit.h"
#include "archi/chips/pulp/cluster_icache_ctrl/cluster_icache_ctrl.h"
#include "archi/stdout/stdout_v3.h"
#include "archi/eu/eu_v3.h"
#include "archi/dma/mchan_v7.h"

#include "archi/udma/cpi/udma_cpi_v1.h"
#include "archi/udma/i2c/udma_i2c_v2.h"
#include "archi/udma/i2s/udma_i2s_v2.h"
#include "archi/udma/spim/udma_spim_v3.h"
#include "archi/udma/uart/udma_uart_v1.h"
#include "archi/udma/hyper/udma_hyper_v3.h"
#include "archi/udma/udma_v3.h"
```

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h

```
/*
 * MEMORIES
 */
#define ARCHI_HAS_L2                1
#define ARCHI_HAS_L2_MULTI          1
#define ARCHI_HAS_L1                1
#define ARCHI_L2_PRIV0_ADDR  0x1c000000
#define ARCHI_L2_PRIV0_SIZE  0x00008000
…
/*
 * IP VERSIONS
 */
#define UDMA_VERSION        3
#define PERIPH_VERSION      2
#define TIMER_VERSION       2
…
/*
 * UDMA
 */
#define ARCHI_UDMA_HAS_SPIM     1
#define ARCHI_UDMA_HAS_UART     1
#define ARCHI_UDMA_HAS_SDIO     1
…
```

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h

```
/*
 * SOC PERIPHERALS
 */
#define ARCHI_SOC_PERIPHERALS_ADDR    0x1A100000
#define ARCHI_FC_TIMER_SIZE           0x00000800
#define ARCHI_FLL_OFFSET              0x00000000
#define ARCHI_GPIO_OFFSET             0x00001000
#define ARCHI_UDMA_OFFSET             0x00002000
…
/*
 * FC
 */
#define ARCHI_FC_ADDR                 0x00000000
#define ARCHI_FC_GLOBAL_ADDR          0x1B000000
/*
 * CLUSTER
 */
#define ARCHI_CLUSTER_ADDR            0x00000000
#define ARCHI_CLUSTER_SIZE            0x00400000
```

# PULP ARCHI

- **rtos/pulpos/pulp_archi/include/archi/**
- **Chip-specific: ./chips/pulp/**
  - pulp.h
  - properties.h
  - memory_map.h
- **Module description: ./**
  - riscv/priv_1_10.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h
- Module description: ./
  - riscv/priv_1_10.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

```
#define RV_CSR_MSTATUS  0x300
#define RV_CSR_MEPC     0x341
#define RV_CSR_MCAUSE   0x342
#define RV_CSR_MTVAL    0x343
#define RV_CSR_MESTATUS 0x7C0
#define RV_CSR_MISA     0xF10
#define RV_CSR_MIMPID   0xF13
#define RV_CSR_MHARTID  0xF14
#define CSR_PCCR(N)       (0x780 + (N))
#define CSR_PCER          0xCC0
#define CSR_PCMR          0xCC1
#define CSR_STACK_CONF   0x7D0
#define CSR_STACK_START  0x7D1
#define CSR_STACK_END    0x7D2
#define CSR_MESTATUS_INTEN_BIT 0
#define CSR_MESTATUS_PRV_BIT   1
#define CSR_MESTATUS_PRV_MACH  3
#define CSR_HWLOOP0_START   0x7C0
#define CSR_HWLOOP0_END     0x7C1
#define CSR_HWLOOP0_COUNTER 0x7C2
#define CSR_HWLOOP1_START   0x7C4
#define CSR_HWLOOP1_END     0x7C5
#define CSR_HWLOOP1_COUNTER 0x7C6
```

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h
- Module description: ./
  - riscv/priv_1_10.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

```
//
// REGISTERS FIELDS
//
// Format is operation dependent. See below. (access: R/W)
#define MCHAN_CMD_CMD_BIT                            0
#define MCHAN_CMD_CMD_WIDTH                          32
#define MCHAN_CMD_CMD_MASK                           0xffffffff

// Format is operation dependent. See below. (access: R/W)
#define MCHAN_STATUS_STATUS_BIT                      0
#define MCHAN_STATUS_STATUS_WIDTH                    32
#define MCHAN_STATUS_STATUS_MASK                     0xffffffff

// Transfer length in bytes configuration bitfield. (access: W)
#define MCHAN_CMD_CMD_LEN_BIT                        0
#define MCHAN_CMD_CMD_LEN_WIDTH                      17
#define MCHAN_CMD_CMD_LEN_MASK                       0x1ffff
```

# PULP ARCHI

- rtos/pulpos/pulp_archi/include/archi/
- Chip-specific: ./chips/pulp/
  - pulp.h
  - properties.h
  - memory_map.h
- Module description: ./
  - riscv/priv_1_10.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

```
/* How many devices in the same hyperchip: cs=0 HyperRam, cs=1 HyperFlash
*/
#define HYPER_NB_CS 2
/* Architectural configuration
*/
#define HYPER_FIFO_DEPTH 8
#define HYPER_NB_CHANNELS 4 /* 4 tran_id and 1 common space */
#define HYPER_CHANNELS_OFFSET 0x80
#define HYPER_CHANNELS_ID_OFFSET(id) HYPER_CHANNELS_OFFSET*(id)
#define HYPER_COMMON_REGS_OFFSET
HYPER_CHANNELS_OFFSET*HYPER_NB_CHANNELS

#define HYPER_NB_COMMON_REGS 10
#define HYPER_NB_CONFIG_REGS 16
#define HYPER_NB_REGS HYPER_NB_COMMON_REGS +
HYPER_NB_CONFIG_REGS*HYPER_NB_CHANNELS

#define HYPER_NB_TWD_REGS 6
#define HYPER_NB_CTL_REGS 2
```

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/

```
#ifndef __ARCHI_PULP_H__
#define __ARCHI_PULP_H__

#include "archi/pulp_defs.h"
#include "archi/utils.h"

#define  __A_PULP_CHIP_INC(x) #x
#define  _A_PULP_CHIP_INC(x) __A_PULP_CHIP_INC(archi/chips/x/pulp.h)
#define  A_PULP_CHIP_INC(x) _A_PULP_CHIP_INC(x)

#if defined(PULP_CHIP_FAMILY)
#include A_PULP_CHIP_INC(PULP_CHIP_FAMILY_STR)
#else
#include A_PULP_CHIP_INC(PULP_CHIP_STR)
#endif

#endif
```

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/
- Chip-specific: ./chips/pulp/
  - pulp.h

```
#ifndef __HAL_CHIPS_PULP_PULP_H__
#define __HAL_CHIPS_PULP_PULP_H__

#include "hal/riscv/riscv_v5.h"
#include "hal/eu/eu_v3.h"
#include "hal/itc/itc_v1.h"
#include "hal/dma/mchan_v7.h"
#include "hal/timer/timer_v2.h"
#include "hal/soc_eu/soc_eu_v2.h"
#include "hal/cluster_ctrl/cluster_ctrl_v2.h"
#include "hal/icache/icache_ctrl_v2.h"
#include "hal/fll/fll_v1.h"
#include "hal/gpio/gpio_v3.h"
#include "hal/rom/rom_v2.h"

#include "hal/udma/udma_v3.h"
#include "hal/udma/cpi/udma_cpi_v1.h"
#include "hal/udma/i2c/udma_i2c_v2.h"
#include "hal/udma/spim/udma_spim_v3.h"
#include "hal/udma/uart/udma_uart_v1.h"
#include "hal/udma/hyper/udma_hyper_v3.h"

#endif
```

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/
- Chip-specific: ./chips/pulp/
  - pulp.h
- Module bare-metal drivers: ./
  - riscv/risc_v5.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/
- Chip-specific: ./chips/pulp/
  - pulp.h
- Module bare-metal drivers: ./
  - riscv/risc_v5.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

```
static inline unsigned int hal_spr_read_then_clr(unsigned int reg, unsigned int val)
{
  return __builtin_pulp_read_then_spr_bit_clr(reg, val);
}

static inline unsigned int hal_spr_read_then_set(unsigned int reg, unsigned int val)
{
  return __builtin_pulp_read_then_spr_bit_set(reg, val);
}

static inline void hal_spr_write(unsigned int reg, unsigned int val)
{
  __builtin_pulp_spr_write(reg, val);
}

static inline unsigned int hal_spr_read(unsigned int reg)
{
  return __builtin_pulp_spr_read(reg);
}
```

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/
- Chip-specific: ./chips/pulp/
  - pulp.h
- Module bare-metal drivers: ./
  - riscv/risc_v5.h
  - dma/mchan_v7.h
  - udma/hyper/udma_hyper_v3.h

```
static inline int plp_dma_memcpy(mchan_ext_t ext, unsigned int loc, unsigned
short size, int ext2loc) {
  unsigned int counter = plp_dma_counter_alloc();
  unsigned int cmd = plp_dma_getCmd(ext2loc, size, PLP_DMA_1D,
PLP_DMA_TRIG_EVT, PLP_DMA_NO_TRIG_IRQ, PLP_DMA_SHARED);
  plp_dma_cmd_push(cmd, loc, ext);
  return counter;
}
…
static inline void plp_dma_barrier() {
  while(DMA_READ(MCHAN_STATUS_OFFSET) & 0xFFFF) {
    eu_evt_maskWaitAndClr(1<<ARCHI_CL_EVT_DMA0);
  }
  DMA_WRITE(-1, MCHAN_STATUS_OFFSET);
}
static inline void plp_dma_wait(unsigned int counter) {
  while(DMA_READ(MCHAN_STATUS_OFFSET) & (1 << counter)) {
    eu_evt_maskWaitAndClr(1<<ARCHI_CL_EVT_DMA0);
  }
  plp_dma_counter_free(counter);
}
static inline unsigned int plp_dma_status() {
  return DMA_READ(MCHAN_STATUS_OFFSET);
}
```

# PULP HAL

- <archi/pulp.h> is included everywhere
- rtos/pulpos/pulp_hal/include/hal/
- Chip-specific: ./chips/pulp/
    - pulp.h
- Module bare-metal drivers: ./
    - riscv/risc_v5.h
    - dma/mchan_v7.h
    - udma/hyper/udma_hyper_v3.h

```
static inline int plp_hyper_get_reg(unsigned int offset){
  return pulp_read32(offset);
}

static inline int plp_hyper_check_memsel(unsigned char hyper_id){
  return plp_hyper_get_reg(UDMA_HYPER_BASE_ADDR(hyper_id) +
MEM_SEL);
}

static inline int plp_hyper_nb_tran(unsigned char hyper_id, unsigned int tran_id){
  return plp_hyper_get_reg(UDMA_HYPER_BASE_ADDR(hyper_id) +
STATUS(tran_id)) >> 1;
}

static inline int plp_hyper_is_busy(unsigned char hyper_id, unsigned int tran_id){
  return plp_hyper_get_reg(UDMA_HYPER_BASE_ADDR(hyper_id) +
STATUS(tran_id)) & 0x00000001;
}

static inline void plp_hyper_wait(unsigned char hyper_id, unsigned int tran_id){
   while(plp_hyper_is_busy(hyper_id, tran_id)){
   }
}
```