3.为quard-star添加flash | TimerのBlog

yanglianoo.github.io/2023/06/15/QEMU中自定义开发板-3-为quard-star添加flash

2023年6月15日

1.quard_star.h 修改

```
You, 1小时前 | 1 author (You)
struct QuardStarState {
    /*< private >*/
    MachineState parent;

    /*< public >*/
    RISCVHartArrayState soc[QUARD_STAR_SOCKETS_MAX];
    PFlashCFI01 *flash;
};

enum {
    QUARD_STAR_MROM,
    QUARD_STAR_SRAM,
    QUARD_STAR_UART0,
    QUARD_STAR_FLASH,
    QUARD_STAR_DRAM,
};
```

2.quard_star.c 修改

定义flash起始地址和映射大小

```
static const MemMapEntry quard_star_memmap[] = {
    [QUARD_STAR_MROM]  = {        0x0,        0x8000 },
    [QUARD_STAR_SRAM]  = {     0x8000,        0x8000 },
    [QUARD_STAR_UART0] = { 0x10000000,         0x100 },
    [QUARD_STAR_FLASH] = { 0x20000000,     0x2000000 },
    [QUARD_STAR_DRAM]  = { 0x80000000,          0x80 },
};
```

创建flash并映射

C

```c
/* 创建flash并映射 */
static void quard_star_flash_create(MachineState *machine)
{
    #define QUARD_STAR_FLASH_SECTOR_SIZE (256 * KiB)  //0x40000
    QuardStarState *s = RISCV_VIRT_MACHINE(machine);
    MemoryRegion *system_memory = get_system_memory();
    DeviceState *dev = qdev_new(TYPE_PFLASH_CFI01);

    qdev_prop_set_uint64(dev, "sector-length", QUARD_STAR_FLASH_SECTOR_SIZE);
    qdev_prop_set_uint8(dev, "width", 4);
    qdev_prop_set_uint8(dev, "device-width", 2);
    qdev_prop_set_bit(dev, "big-endian", false);
    qdev_prop_set_uint16(dev, "id0", 0x89);
    qdev_prop_set_uint16(dev, "id1", 0x18);
    qdev_prop_set_uint16(dev, "id2", 0x00);
    qdev_prop_set_uint16(dev, "id3", 0x00);
    qdev_prop_set_string(dev, "name","quard-star.flash0");

    object_property_add_child(OBJECT(s), "quard-star.flash0", OBJECT(dev));
    object_property_add_alias(OBJECT(s), "pflash0",
                              OBJECT(dev), "drive");

    s->flash = PFLASH_CFI01(dev);
    pflash_cfi01_legacy_drive(s->flash,drive_get(IF_PFLASH, 0, 0));

    hwaddr flashsize = quard_star_memmap[QUARD_STAR_FLASH].size;
    hwaddr flashbase = quard_star_memmap[QUARD_STAR_FLASH].base;

    assert(QEMU_IS_ALIGNED(flashsize, QUARD_STAR_FLASH_SECTOR_SIZE));
    assert(flashsize / QUARD_STAR_FLASH_SECTOR_SIZE <= UINT32_MAX);
    qdev_prop_set_uint32(dev, "num-blocks", flashsize /
QUARD_STAR_FLASH_SECTOR_SIZE);
    sysbus_realize_and_unref(SYS_BUS_DEVICE(dev), &error_fatal);

    memory_region_add_subregion(system_memory, flashbase,
                                sysbus_mmio_get_region(SYS_BUS_DEVICE(dev),
                                                       0));
}
```

C

```c
static void quard_star_machine_init(MachineState
*machine)
{
   // 创建主存
   quard_star_memory_create(machine);
   //创建flash
   quard_star_flash_create(machine);
}
```

还需要修改一下主存的代码，在第一篇博客中提到如果板子支持pflash需要修改`riscv_setup_rom_reset_vec`的入参。

```
riscv_setup_rom_reset_vec(machine, &s->soc[0],
                          quard_star_memmap[QUARD_STAR_FLASH].base,
                          quard_star_memmap[QUARD_STAR_MROM].base,
                          quard_star_memmap[QUARD_STAR_MROM].size,
                          0x0, 0x0);
```

3. Kconfig修改

新增PFLASH_CFI01 设备

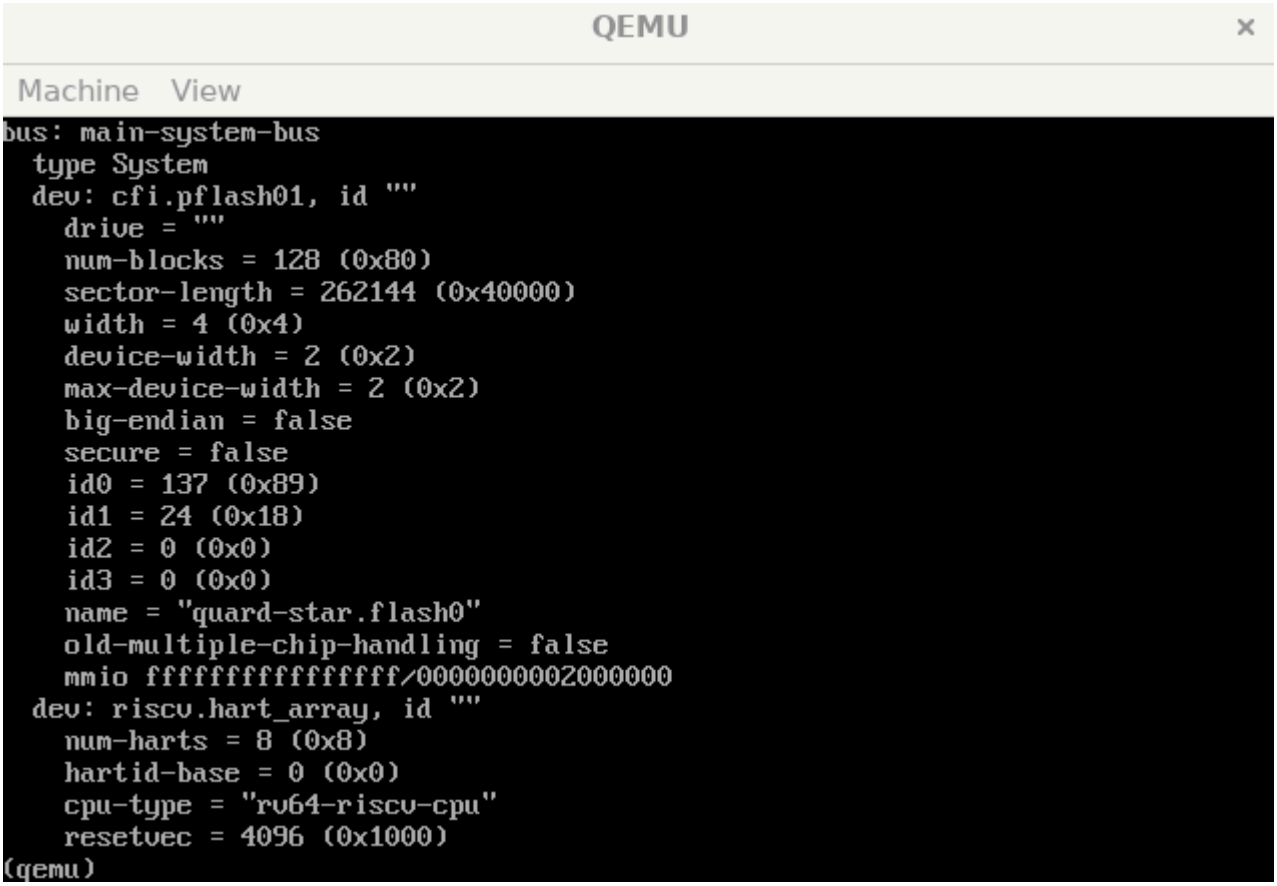
plaintext

```
config QUARD_STAR
    bool
    select SERIAL
    select PFLASH_CFI01  //选
中FLASH
```

4. 测试

sh

```
timer@DESKTOP-JI9EVEH:~/quard-star$
./build.sh
timer@DESKTOP-JI9EVEH:~/quard-star$
./run.sh
```



可以看到pflash设备成功被添加，挂载在系统总线上。

> 代码地址：yanglianoo/quard-star: 从零基于qemu创建riscv嵌入式开发板，并移植操作系统 (github.com)
>
> 有问题请与我联系：wechat：13699648817