

mmu设置

一. 设置步骤

- 初始化一个页表条目数组PageTable，大小为16kb，16kb对齐。
- 根据相应的虚拟地址和物理地址的映射来填充对应页表条目。
- 设置页表基地址，即将PageTable对应地址写入TTBR0寄存器。

二. 设置代码

```
/* 初始化一个页表条目数组PageTable，大小为16kb，16kb对齐 */
u32 PageTable[4096] __attribute__((aligned (16*1024)));

/* 根据相应的虚拟地址和物理地址的映射来填充对应页表条目 */
u32 build_PGTL (void)
{
    u32 vaddr;
    u32 paddr;
    u32 i;
    u32 *pt_phys = (u32*)VIRT_TO_PHYS(PageTable);

    for(i = (sizeof(PageTable) / sizeof(u32)); i; i -= sizeof(u32))
    {
        pt_phys[i]=0;
    }

    /* 将0x1F000000到0x1FFFFFFF的16m虚拟地址映射到0x1F000000到0x1FFFFFFF的16m物理地址,写回写分配,
    所有权级可执行和读写,共享,全局所有进程可用 */
    for (paddr = SDRAM_PHYS_BASE, vaddr = SDRAM_VIRT_BASE; vaddr < (SDRAM_VIRT_BASE +
SDRAM_SIZE/2); vaddr += PGTL_L1_SECT_SIZE, paddr += PGTL_L1_SECT_SIZE)
        pt_phys[VADDR_TO_L1_INDEX(vaddr)] = PGTL_L1_SECT_WBWA_ENTRY(paddr , 0, 0, 0,
AP_PRW_URW, 1, 0); //wbwa

    /* 将0x00000000到0x00FFFFFF的16m虚拟地址映射到0x20000000到0x20FFFFFF的16m物理地址,写回写分配,
    所有权级可执行,共享,全局所有进程可用 */
    for (paddr = 0x0, vaddr = 0x20000000; vaddr < 0x21000000; vaddr += PGTL_L1_SECT_SIZE,
paddr += PGTL_L1_SECT_SIZE )
        pt_phys[VADDR_TO_L1_INDEX(vaddr)] = PGTL_L1_SECT_WBWA_ENTRY(paddr , 0, 0, 0,
AP_PRW_URW, 1, 0); //wbwa

    /* 将0x20000000到0x20FFFFFF的16m虚拟地址映射到0x20000000到0x20FFFFFF的16m物理地址,不缓存,所有
    权级可执行,共享,全局所有进程可用 */
    for (paddr = 0x20000000, vaddr = 0x20000000; vaddr < 0x21000000; vaddr +=
PGTL_L1_SECT_SIZE, paddr += PGTL_L1_SECT_SIZE )
```

```

    pt_phys[VADDR_TO_L1_INDEX(vaddr)] = PGTL_L1_SECT_ENTRY(paddr,
MEM_CACHE_TYPE(MEM_NC, MEM_NC) , 0, 0, 0, AP_PRW_URW, 1, 0); //nc

    //io device mapping
    /* 将0xfc000000到0xffffffff的64m虚拟地址映射到0xfc000000到0xffffffff的64m物理地址，共享的设备区域 */
    for (paddr = 0xfff00000, vaddr = 0xfff00000; vaddr >= 0xfc000000; vaddr -=
PGTL_L1_SECT_SIZE, paddr -= PGTL_L1_SECT_SIZE)
        pt_phys[VADDR_TO_L1_INDEX(vaddr)] = PGTL_L1_SECT_ENTRY(paddr, MEM_SHARE_DEVICE , 0,
0, 0, AP_PRW_URW, 1, 0); //sd

    return (u32)pt_phys;
}

/* 设置页表基地址，即将PageTable对应地址写入TTBR0寄存器 */
@@ 设置页表的基地址,See ARMv7-A,R, section B4.1.154
    orr    r0, r0, #TTB_FLAGS_SMP        @ 打开TTBR0的第1, 3, 5和6位
    mcr    p15, 0, r0, c2, c0, 0

```