汇编作业讲解3

杨乐 2021.9

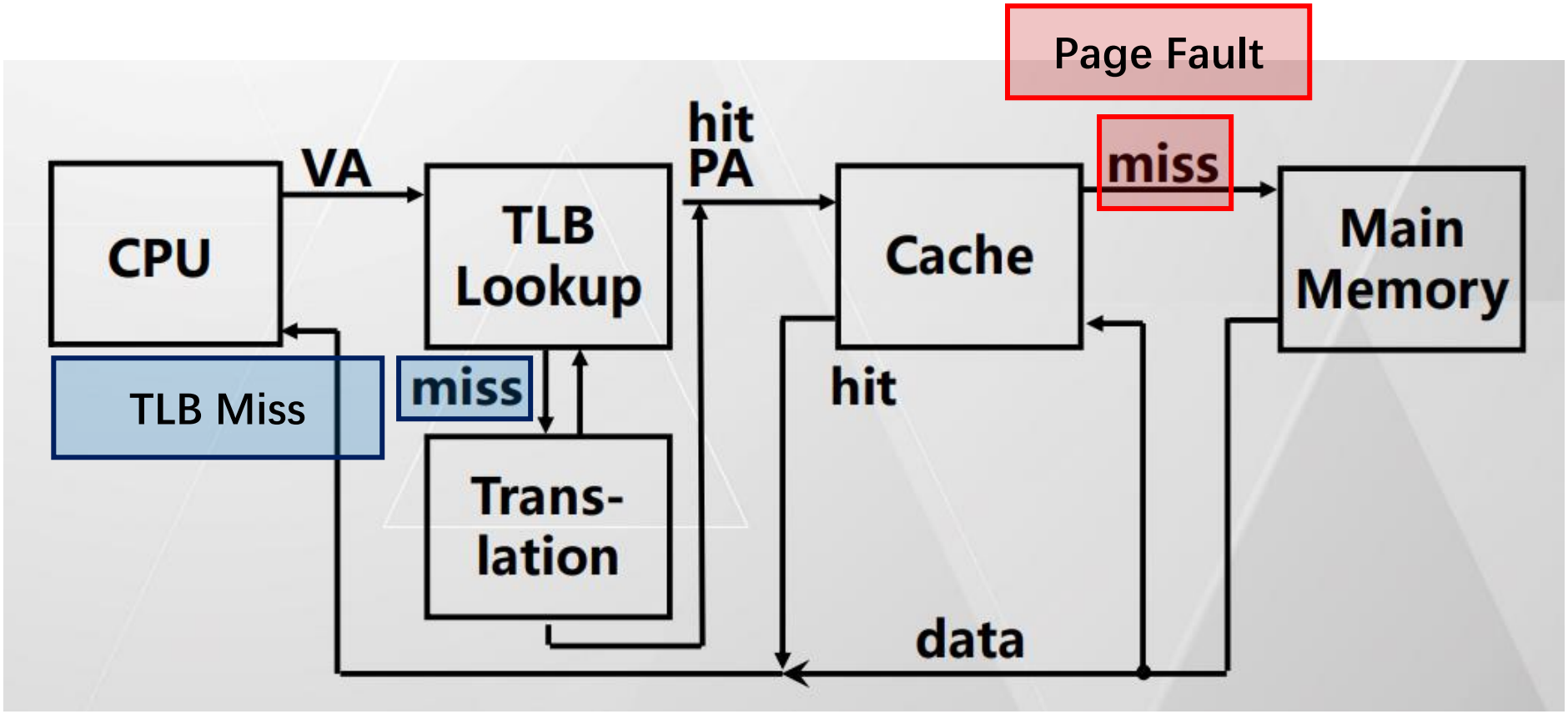
1 TLB工作原理

- MMU使用**页表**将**虚拟地址**转换为**物理地址**。
- TLB则通过**缓存**的方式来加速这一转换的过程。

- Page Fault异常：该页不在内存当中
- 如何解决：将该页存放到内存中

- TLB Miss异常：某虚拟地址在TLB中不存在匹配项
- 如何解决：通过页表查询，随后将该项填入TLB

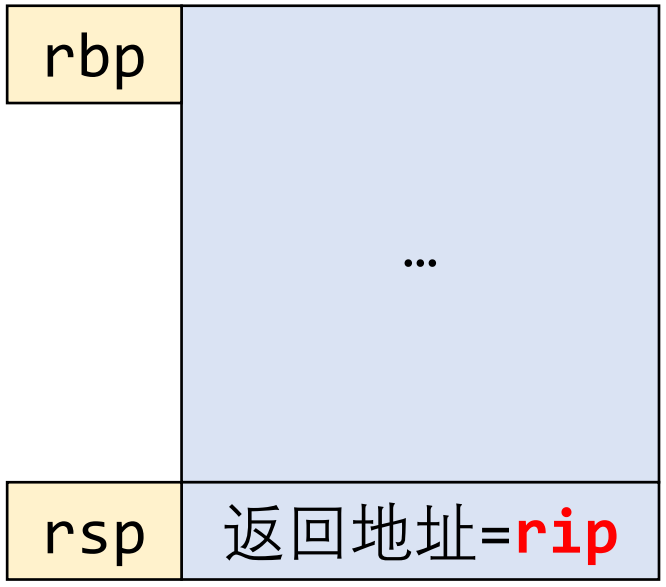
1 TLB工作原理 cont



2 SwapContext上下文切换

使用空间：14*8=112B
r10/r11/rax调用者保存

- （1）保存当前上下文：普通寄存器，rbp，rsp，rip
- （2）恢复至新的上下文：普通寄存器，rbp，rsp，rip



保存

```
movq   (%rsp), %rcx      ①
movq   %rcx, oRIP(%rdi)
leaq   8(%rsp), %rcx     ②
movq   %rcx, oRSP(%rdi)
```

保存返回地址

保存rsp

恢复

```
movq   oRSP(%rsi), %rsp
movq   oRIP(%rsi), %rcx  ③
pushq  %rcx
```

Push
返回地址
Rsp-=8

3、 有如下的 C 代码及其对应的 X86-64 汇编代码，请问

（1）局部变量 result 如何存储？ 存在rdx中

（2）i 如何存储？ 存在edi中

（3）EXPR1、EXPR2、EXPR3、EXPR4、EXPR5 分别是？请用常数或者 C 程序中的变量表示。

```
long int puzzle(int a, int b)
{
int i;                    i=a; i>0; i-=b
long int result = EXPR1;  result=rdx=b
for (i = EXPR2; i > EXPR3; i -= EXPR4)
result *= EXPR5;          result*=i
return result;            result=rdx
}
```

```
puzzle:
movslq %esi,%rdx          rdx = b
jmp .L60
.L61:
movslq %edi,%rax          rax = a
subl %esi, %edi           a -= b
imulq %rax, %rdx          rdx *= rax(a)
.L60:
testl %edi, %edi
jg .L61                   if a>0 goto L61
movq %rdx, %rax
ret                       return rdx
```

4 MIPS指令

假设存在如下的完成计数任务的 mips32 汇编代码（左侧框图内），被两个**同时运行**的任务调用，且这两个任务代码中的地址 65540($4)指向同一个物理内存地址，为确保代码能够正确的实现程序语义，需要替换原始代码中的两条指令，如何替换？此外，汇编器将现有的左侧代码转换为了右侧框图内的等价指令，请填空。

(1) 替换指令

将lw替换为ll

将sw替换为sc

在多线程程序中，为了实现对共享变量的互斥访问，一般需要一个TestAndSet的**原子操作**。在MIPS中，是通过特殊的Load/Store指令：LL（Load Linked，链接加载）以及SC（Store Conditional，条件存储）这一指令对完成的。

(2) 转换指令

lw $2, 65540($4)

R[2] = R[R[4]+65540]

lui $1, 1

addu $1, $1 , $4

lw $2, 4 ($1)

R[1] = 65536* 1

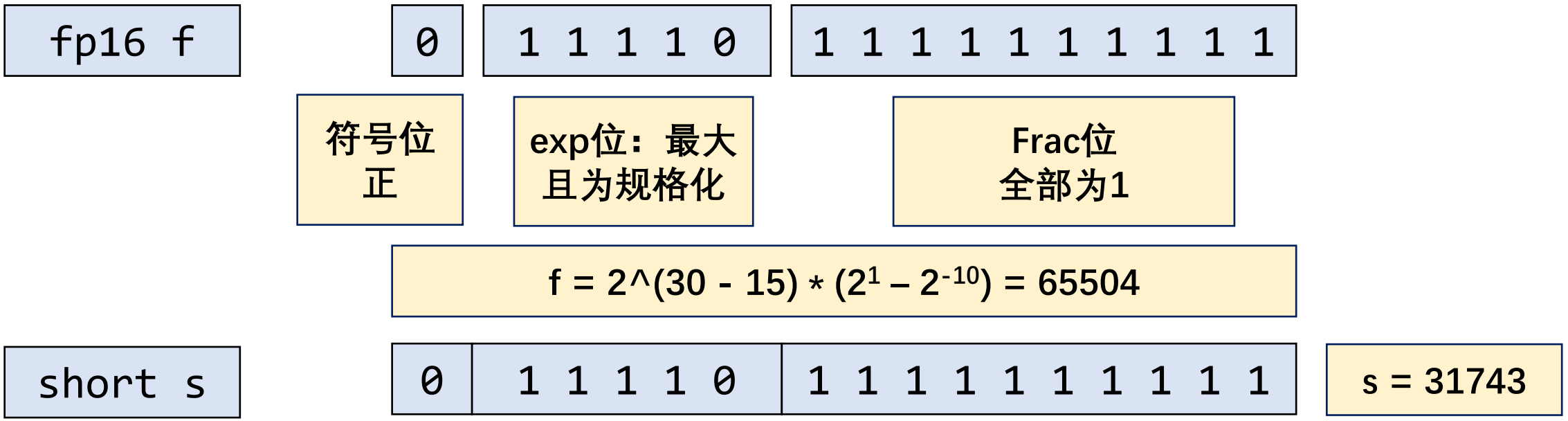
R[1] = R[1] + R[4]

R[2] = R[R[1] + 4]

5 union

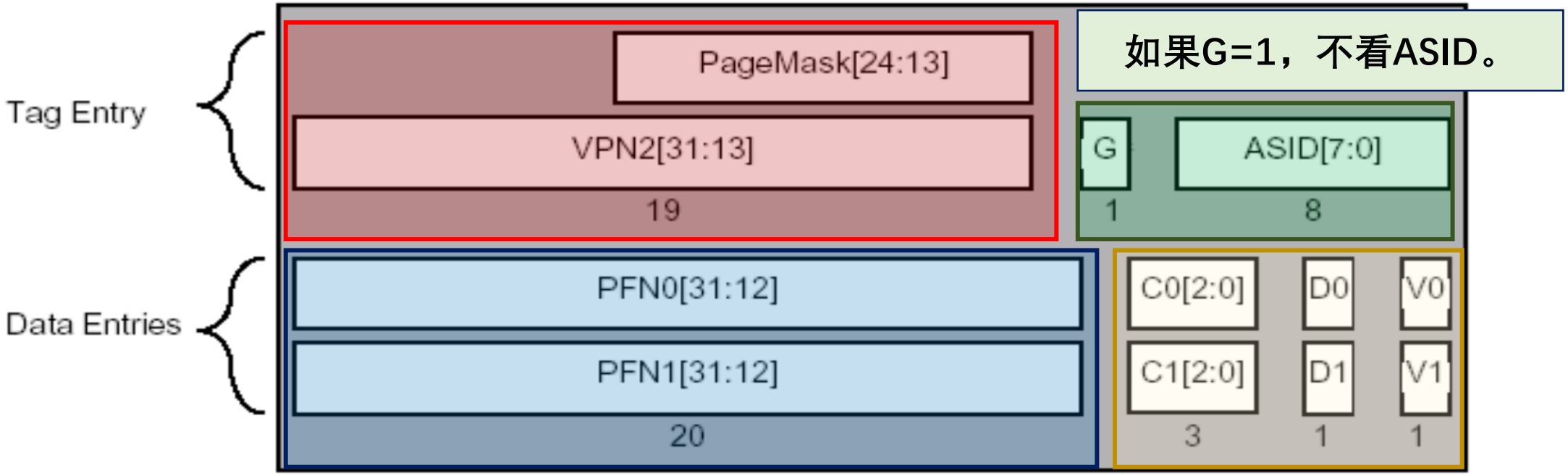
```
union {
  fp16 f;
  short s;
}
```

浮点数：求最大规格化数（符号1位，exp5位，frac10位）
整数：求对应整数



6 虚拟地址到物理地址

看page size。如果page size是4K（12位），则看31-13；
如果page size是16M（24位），则看31-25。



如果G=1，不看ASID。

看tag的最低位来选择哪一路。如果page size是4K，则看第12位；
如果page size是16M，则看第24位。

标志位

6 虚拟地址到物理地址 cont

0x 8E2AE 320	0x12	写	Page size = 4K，offset=12位

0x8E2AE = 1000 1110 0010 1010 1110

VPN = 1000 1110 0010 1010 111 = 0x47157; Way = 0

	VPN2	G	ASID	PFN0	PFN1	D0/D1	V0/V1
1	0x47157（二进制为 100 0111 0001 0101 0111)	0	0x12	0x12345	0x12340	0/0	1/1
2	0x47157（二进制为 100 0111 0001 0101 0111)	0	0x13	0x22346	0x22340	1/1	1/1

看ASID选JTLB[1], Way=0 : PFN0 = 0x12345, D0 = 0, V0 = 1

物理地址 = 0x12345 320；但需要看标志位

V=0直接无效；D=0写无效；其余有效