1 非对齐访存指令

非对齐访存指令共有 4 条：LWL/LWR、SWL/SWR。

一般 LWL/LWR 会配合使用，读取一个跨边界(4 字节对齐处)的字。SWL/SWR 也是配合使用。

(1) 非对齐指令助记符中 Left 和 right：

所谓 left 和 right，都是对 32 位寄存器里的值而言的左侧、右侧。

所以 left 就是指从内存中读数据到寄存器左侧(**数据高位**)，或写寄存器左侧(**数据高位**)到内存中

right 就是指从内存中读数据到寄存器右侧(**数据低位**)，或写寄存器右侧(**数据低位**)到内存中

(2) 读写左侧或右侧的数据，读写几字节呢？

就要用地址偏移去判断了。

left 是读写**数据高位**，所以是访存地址对应的 byte 对应寄存器中**最高字节**，从该 byte 开始向内存中**数据低位**索引直到**下边界**(4 字节对齐处，即地址低两位为 0)，并完成读写

right 是读写**数据低位**，所以是访存地址对应的 byte 对应寄存器中**最低字节**，从该 byte 开始向内存中**数据高位**索引直到**上边界**(4 字节对齐处，即地址低两位为 11) ，并完成读写

(3) 尾端模式

寄存器中，左侧就是数据高位，右侧就是数据低位。但在内存中小地址处为高位还是低位，就是依据具体实现而不同了，有两种方式：

大尾端：**小**地址处为数据**高**位，**大**地址处为数据**低**位。如下表，则是数据 0x12 为数据最高字节。

如果我们大尾端下 load 0xbfc00000，则得到 0x12345678。

小尾端：**小**地址处为数据**低**位，**大**地址处为数据**高**位。如下表，则是数据 0x78 为数据最高字节。

如果我们小尾端下 load 0xbfc00000，则得到 0x78563412。

内存数据：

地址	0xbfc0000	0xbfc0001	0xbfc0002	0xbfc0003
数据	0x12	0x34	0x56	0x78

小尾端：　　低字节　　高字节

现在一般处理器中数据存储和处理实现为小尾端模式，lab5 实验中也是如此。因而我们只用实现小尾端下的非对齐访存。

(4) 小尾端下非对齐访存

left 是读写**高位**，从访存地址指示的 byte 开始向**内存中数据低位(也是小地址处)**索引直到**下边界**(4 字节对齐处，地址低两位为 0)

right 是读写**低位**，从访存地址指示的 byte 开始向**内存中数据高位(也是大地址处)**索引直到**上边界**(4 字节对齐处，地址低两位为 11)

国科大UC2009H课程2016秋季

假设内存中数据存放如下：

内存数据：

地址	0xbfc0000	0xbfc0001	0xbfc0002	0xbfc0003
数据	0x12	0x34	0x56	0x78

小尾端：　　低字节　　高字节

假设寄存器 R 中数据存放如下：

寄存器 R 中原有值

bit 位	[31:24]	[23:16]	[15:8]	[7:0]
数据	0xa0	0xb0	0xc0	0xd0

高字节　　低字节

则执行结果如下：

访存地址低2位		load，只修改寄存器R								store，只修改内存里数据							
		LWL执行结果（寄存器）				LWR执行结果（寄存器）				SWL执行结果（内存）				SWR执行结果（内存）			
2'b00	指示位	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]	2'b00	2'b01	2'b10	2'b11	2'b00	2'b01	2'b10	2'b11
	结果	**0x12**	0xb0	0xc0	0xd0	**0x78**	**0x56**	**0x34**	**0x12**	**0xa0**	0x34	0x56	0x78	**0xd0**	**0xc0**	**0xb0**	**0xa0**
2'b01	指示位	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]	2'b00	2'b01	2'b10	2'b11	2'b00	2'b01	2'b10	2'b11
	结果	**0x34**	**0x12**	0xc0	0xd0	0xa0	**0x78**	**0x56**	**0x34**	**0xb0**	**0xa0**	0x56	0x78	0x12	**0xd0**	**0xc0**	**0xb0**
2'b10	指示位	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]	2'b00	2'b01	2'b10	2'b11	2'b00	2'b01	2'b10	2'b11
	结果	**0x56**	**0x34**	**0x12**	0xd0	0xa0	0xb0	**0x78**	**0x56**	**0xc0**	**0xb0**	**0xa0**	0x78	0x12	0x34	**0xd0**	**0xc0**
2'b11	指示位	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]	2'b00	2'b01	2'b10	2'b11	2'b00	2'b01	2'b10	2'b11
	结果	**0x78**	**0x56**	**0x34**	**0x12**	0xa0	0xb0	0xc0	**0x78**	**0xd0**	**0xc0**	**0xb0**	**0xa0**	0x12	0x34	0x56	**0xd0**

通用表达如下：gpr[]表示读写的通用寄存器，mem[]表示内存：

访存地址低2位	load，只修改寄存器R		store，只修改内存里数据	
	LWL执行结果（寄存器）	LWR执行结果（寄存器）	SWL执行结果（内存）	SWR执行结果（内存）
2'b00	gpr[31:24] = mem[2'b00:00]	gpr[31:0] = mem[2'b00:11]	mem[2'b00:00]=gpr[31:24]	mem[2'b00:11]=gpr[31:0]
2'b01	gpr[31:16] = mem[2'b01:00]	gpr[23:0] = mem[2'b01:11]	mem[2'b01:00]=gpr[31:16]	mem[2'b01:11]=gpr[23:0]
2'b10	gpr[31:8] = mem[2'b10:00]	gpr[15:0] = mem[2'b10:11]	mem[2'b10:00]=gpr[31:8]	mem[2'b10:11]=gpr[15:0]
2'b11	gpr[31:0] = mem[2'b11:00]	gpr[7:0] = mem[2'b11:11]	mem[2'b11:00]=gpr[31:0]	mem[2'b11:11]=gpr[7:0]

国科大体系结构2016秋季