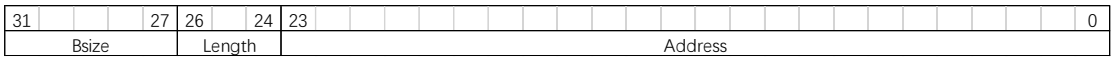
RISCV-Y 指令扩展集手册（精简描述）

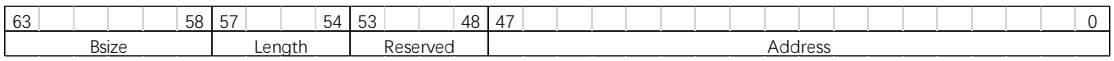
1. 内存模型和指针表示

RISCV-Y 指令增加采用标签指针类型，在 RV32-Y 的指针中定义 24 位有效地址和 8 位元数据，在 RV64-Y 的指针中定义最高 48 位有效地址和 16 位元数据，具体格式如下：

RV32-Y 的标签指针格式



RV64-Y 的标签指针格式



采用 RISCV-Y 指令扩展的处理器在进行访存地址计算时，截取有效地址 Address 进行计算（Memory Tagging），而元数据用于计算数组的上界和下届。

Bsize 定义基准内存块大小为，Length 定义数组中基准块的数目，数组总大小为 Length*(2^BSize)，Length!=0。在 RV32-Y 中数组的首地址 2^(BSize+3)对齐，在 RV64-Y 中数组的首地址 2^(BSize+4)对齐。

Length 为 0 时，表示标量内存访问。此时 ADDPI/load/store 中的 imm12 以及 ADDP/SUBP 中的 rs2 值必须为 0，否则如果大于 0 则上溢，小于 0 则下溢，结果为 rs1。

2. 增加指令

增加 ADDP, SUBP, ADDPI 算数指令，rs1 是标签指针，rs2/imm12 是 index，首先根据标签指针计算出上界和下届，然后比较 rs1+index 与上界和下届的大小。如果大于上界，计算结果为数组最后一项，发出上溢例外；如果小于下界，计算结果为数组第一项，发出下溢例外。如果 length 为 0，功能同上，计算结果为 rs1。

更改所有的 load/store 指令语义，地址运算功能和 ADDIP 相同，如果 length 为 0，功能同上，有效地址为 rs1。

ADDP/SUBP 指令编码，OP=0110011，OP1=000，修改 func 位域中 instr[29]为 1 表示指针运算。（在下图 ADD/SUB 基础上改 instr[29]位为 1）

0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB

ADDIP 指令编码，增加 OP=0010011，OP1=001。（在下图 ADDI/SLTI 中间加入 ADDPI）

imm[11:0]	rs1	000	rd	0010011	ADDI
imm[11:0]	rs1	010	rd	0010011	SLTI