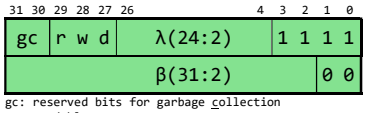
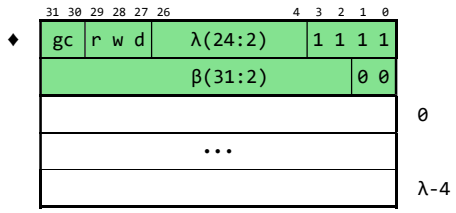
OBJECTS

Generic Header

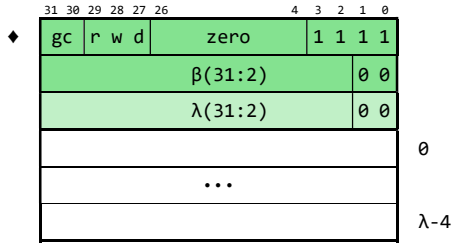


gc: reserved bits for garbage collection
r: readable
w: writable
d: data only (no pointers allowed)
λ: length of this object
β: current access barrier

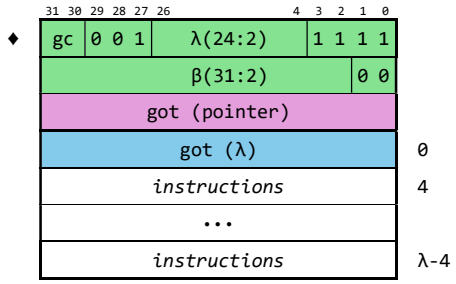
Ordinary



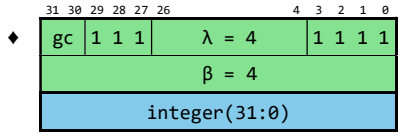
Long



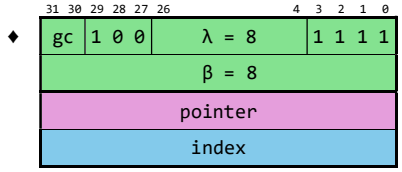
Executable



Immediate (Primitive)



Immediate (Pointer)

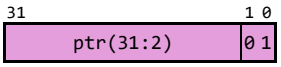


Invariant 1: data-only objects which are not readable and not writable are implicitly executable
Invariant 2: Objects with λ = 0 in their base-header have λ moved to the address following β

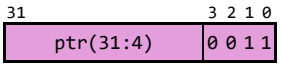
POINTERS & DATA

(in memory)

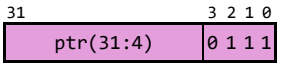
immediate (prim) pointer:



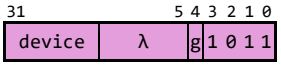
ordinary pointer

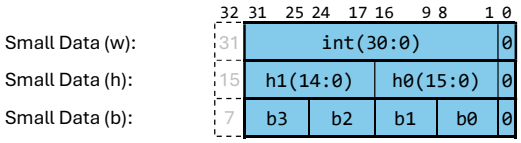


immediate (ptr) pointer:



io pointer:



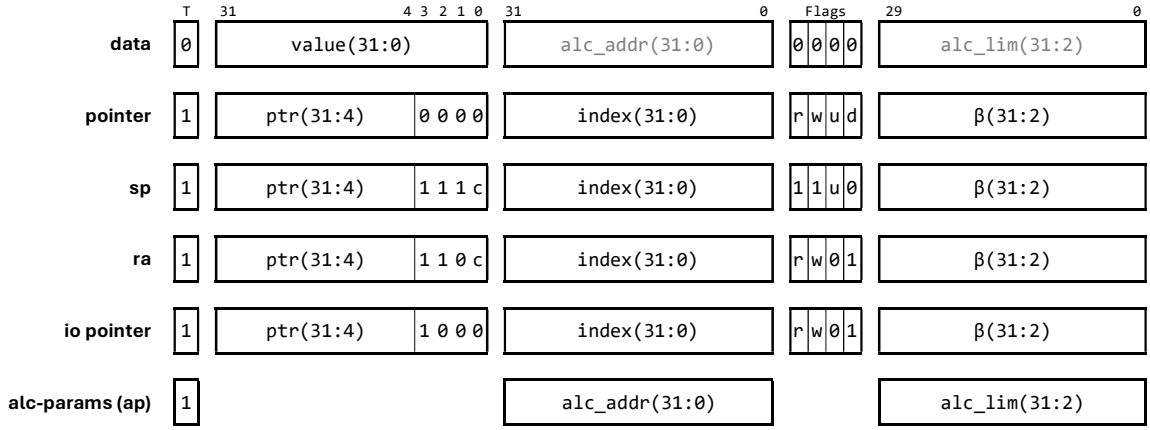


Allocate immediate primitive if:

- sw and rs(30) ≠ rs(31)
- sh at h1 and rs(14) ≠ rs(15)
- sb at b3 and (rs(7) = 1 or rs < 0)

REGISTER FILE & PIPELINE

Architectural Registers (x0-x31, alc-params):



Tags:
r read access, **w** write access, **u** 0 when β = λ else 1, **d** data only

Microarchitectural Registers:

User Mode Instructions (Single Cycle)

Instruction	rd	rs1	rs2	cr	imm	Decision
lui	rd	---	---	-	imm	
auipc	rd	---	---	-	imm	
jal	rd	---	sp	●	imm	
bcc	---	rs1	rs2	-	imm	
arithi	rd	rs1	---	-	imm	
arith	rd	rs1	rs2	-	---	
alc	rd	rs1	ap	-	---	
alci	rd	---	ap	-	imm	
alc.d	rd	rs1	ap	-	---	
alci.d	rd	---	ap	-	imm	
qsz	rd	rs1	---	-	---	
clr	rs1	rs1	---	-	imm	*if imm fits in one cache line*

User Mode Instructions (Multi Cycle)

Instruction	rd	rs1	rs2	cr	imm	Decision
jalr	*rd*	*rs1*	---	-	*imm*	
A jalr	rd	rs1	sp	●	imm	*always*
A lgt	got	rs1	---	-	---	*always (instead of nop)*
lb/bu/h/hu/w	*rd*	*rs1*	---	-	*imm*	
A lb/bu/h/hu/w	rd	rs1	got	●	imm	
a lb/bu/h/hu/w	rd	rd	got	●	imm	*if A loaded an immediate pointer*
sb/h/w	---	*rs1*	*rs2*	-	*imm*	
A alci.d	rs1	rs2	ap	●	4	*if rs2 is data and does not fit in word*
B alci	rs1	rs2	ap	●	8	*if rs2 is pointer with index ≠ 0*
A/B sb/h/w	rs1	rs1	rs2	●	imm	*always*
clr	rs1	rs1	---	●	imm	*if imm spans multiple cache lines*

Supervisor Mode Instructions:

Instruction	rd	rs1	rs2	cr	imm	Notes
sb/h/w.r	---	rs1	rs2	-	imm	*"store raw", allows stores at any point in memory. Uses rs1 as base-ptr*
lb/bu/h/hu/w.r	rd	rs1	---	-	---	*"load raw", same as store raw*
dtp	rd	rs1	---	-	---	*"data to pointer", creates a pointer from data*
ptd	rd	rs1	---	-	---	*"pointer to data", extracts base address of pointer as data*
itd	rd	rs1	---	-	---	*"index to data", extracts index of pointer as data*