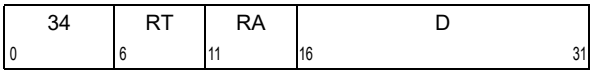
Load Byte and Zero D-form

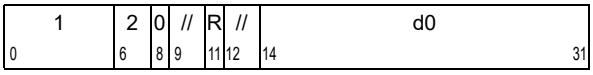
lbz RT,D(RA)



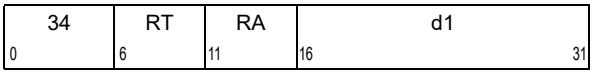
Prefixed Load Byte and Zero MLS:D-form

plbz RT,D(RA),R

Prefix:



Suffix::




```
if “lbz”  then
   EA ← (RA|0) + EXTS64(D)
if “plbz” & R=0 then
   EA ← (RA|0) + EXTS64(d0||d1)
if “plbz” & R=1 then7
   EA ← CIA + EXTS64(d0||d1)

RT ← EXTZ(MEM(EA, 1))
```

For ***lbz***, let the effective address (EA) be the sum (RA|0) + EXTS64(D).

For ***plbz*** with R=0, let EA be the sum of the contents of register RA, or the value 0 if RA=0, and the value d0||d1, sign-extended to 64 bits.

For ***plbz*** with R=1, let EA be the sum of the address of the instruction and the value d0||d1, sign-extended to 64 bits.

The byte in storage addressed by EA is loaded into $RT_{56:63}$. $RT_{0:55}$ are set to 0.

For ***plbz***, if R is equal to 1 and RA is not equal to 0, the instruction form is invalid.

Special Registers Altered:

None

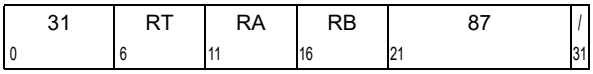
Extended Mnemonics:

Extended mnemonics for *Prefixed Load Byte and Zero*:

Extended mnemonic:		*Equivalent to:*	
plbz	Rx,value(Ry)	plbz	Rx,value(Ry),0
plbz	Rx,value	plbz	Rx,value(0),1

Load Byte and Zero Indexed X-form

lbzx RT,RA,RB




```
if RA = 0 then b ← 0
else          b ← (RA)
EA ← b + (RB)
RT ← 56 0 || MEM(EA, 1)
```

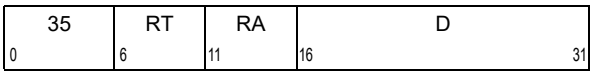
Let the effective address (EA) be the sum (RA|0) + (RB). The byte in storage addressed by EA is loaded into $RT_{56:63}$. $RT_{0:55}$ are set to 0.

Special Registers Altered:

None

Load Byte and Zero with Update D-form

lbzu RT,D(RA)




```
EA ← (RA) + EXTS(D)
RT ← 56 0 || MEM(EA, 1)
RA ← EA
```

Let the effective address (EA) be the sum (RA) + D. The byte in storage addressed by EA is loaded into $RT_{56:63}$. $RT_{0:55}$ are set to 0.

EA is placed into register RA.

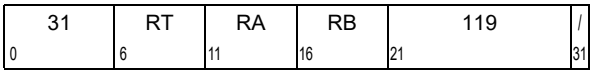
If RA=0 or RA=RT, the instruction form is invalid.

Special Registers Altered:

None

Load Byte and Zero with Update Indexed X-form

lbzux RT,RA,RB




```
EA ← (RA) + (RB)
RT ← 56 0 || MEM(EA, 1)
RA ← EA
```

Let the effective address (EA) be the sum (RA) + (RB). The byte in storage addressed by EA is loaded into $RT_{56:63}$. $RT_{0:55}$ are set to 0.

EA is placed into register RA.

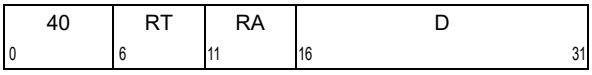
If RA=0 or RA=RT, the instruction form is invalid.

Special Registers Altered:

None

Load Halfword and Zero D-form

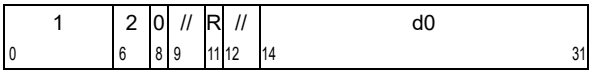
lhz RT,D(RA)



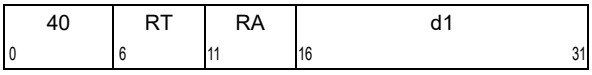
Prefixed Load Halfword and Zero MLS:D-form

plhz RT,D(RA),R

Prefix:



Suffix::




```
if "lhz"  then
   EA ← (RA|0) + EXTS64(D)
if "plhz" & R=0 then
   EA ← (RA|0) + EXTS64(d0||d1)
if "plhz" & R=1 then
   EA ← CIA + EXTS64(d0||d1)

RT ← EXTZ(MEM(EA, 2))
```

For ***lhz***, let the effective address (EA) be the sum (RA|0) + EXTS64(D).

For ***plhz*** with R=0, let the effective address (EA) be the sum of the contents of register RA, or the value 0 if RA=0, and the value d0||d1, sign-extended to 64 bits.

For ***plhz*** with R=1, let the effective address (EA) be the sum of the address of the instruction and the value d0||d1, sign-extended to 64 bits.

The halfword in storage addressed by EA is loaded into $RT_{48:63}$. $RT_{0:47}$ are set to 0.

For ***plhz***, if R is equal to 1 and RA is not equal to 0, the instruction form is invalid.

Special Registers Altered:

None

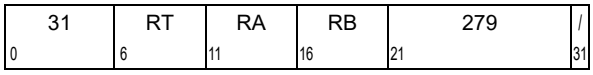
Extended Mnemonics:

Extended mnemonics for *Prefixed Load Halfword and Zero*:

Extended mnemonic:		*Equivalent to:*	
plhz	Rx,value(Ry)	plhz	Rx,value(Ry),0
plhz	Rx,value	plhz	Rx,value(0),1

Load Halfword and Zero Indexed X-form

lhzx RT,RA,RB




```
if RA = 0 then b ← 0
else          b ← (RA)
EA ← b + (RB)
RT ← 48 0 || MEM(EA, 2)
```

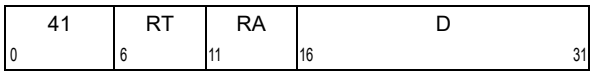
Let the effective address (EA) be the sum (RA|0) + (RB). The halfword in storage addressed by EA is loaded into $RT_{48:63}$. $RT_{0:47}$ are set to 0.

Special Registers Altered:

None

Load Halfword and Zero with Update D-form

lhzu RT,D(RA)




```
EA ← (RA) + EXTS(D)
RT ← 48 0 || MEM(EA, 2)
RA ← EA
```

Let the effective address (EA) be the sum (RA) + D. The halfword in storage addressed by EA is loaded into $RT_{48:63}$. $RT_{0:47}$ are set to 0.

EA is placed into register RA.

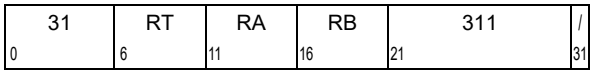
If RA=0 or RA=RT, the instruction form is invalid.

Special Registers Altered:

None

Load Halfword and Zero with Update Indexed X-form

lhzux RT,RA,RB




```
EA ← (RA) + (RB)
RT ← 48 0 || MEM(EA, 2)
RA ← EA
```

Let the effective address (EA) be the sum (RA) + (RB). The halfword in storage addressed by EA is loaded into $RT_{48:63}$. $RT_{0:47}$ are set to 0.

EA is placed into register RA.

If RA=0 or RA=RT, the instruction form is invalid.

Special Registers Altered:

None