

Simplified Mnemonics for PowerPC 555 Assembly

Instruction	Description	Other Registers Altered	Explanation of Operation
add rD, rA, rB	Add	CR0 (LT, GT, EQ, SO)	$rD \leftarrow rA + rB$
addi rD, rA, value	Add immediate	None	$rD \leftarrow rA + \text{value}$
addis rD, rA, value	Add immediate shifted left by 16 bits	None	$rD \leftarrow rA + (\text{value} \ll 16)$
and rA, rS, rB	AND	None	$rA \leftarrow rS \& rB$
andi rA, rS, value	AND Immediate	CR0 (LT, GT, EQ, SO)	$rA \leftarrow rS \& \text{value}$
andis rA, rS, value	AND Immediate shifted left by 16 bits	CR0 (LT, GT, EQ, SO)	$rA \leftarrow rS \& (\text{value} \ll 16)$
b target_addr	Branch Always	None	Branch to target_addr
ble target_addr	Branch if less than or equal to (LT or EQ flags of CR0 set)	None	Branch to target_addr if LT = 1 or EQ = 1
blt target_addr	Branch if less than (LT of CR0 set)	None	Branch to target_addr if LT = 1
beq target_addr	Branch if equal (EQ of CR0 set)	None	Branch to target_addr if EQ = 1
bge target_addr	Branch if greater than or equal to (GT or EQ of CR0 set)	None	Branch to target_addr if GT = 1 or EQ = 1
bgt target_addr	Branch if greater than (GT of CR0 set)	None	Branch to target_addr if GT = 1
bl target_addr	Branch and link to target_addr	None	Branch to target_addr and place return address in LR
blr	Branch to LR (Link Register)	None	Branch to address in link register
bne target_addr	Branch if not equal (EQ of CR0 not set)	None	Branch to target_addr if EQ = 0
cmpw rA, rB	Compare Word	CR0 (LT, GT, EQ, SO)	$rA - rB$
cmpwi rA, value	Compare Word Immediate	CR0 (LT, GT, EQ, SO)	$rA - \text{value}$
la rD, label	Load Address based upon offset value	None	$rD \leftarrow \text{label}$
lbz rD, d(rA)	Load Byte and Zero	None	$rD \leftarrow m[rA + d]$
lbzx rD, rA, rB	Load Byte and Zero Indexed	None	$rD \leftarrow m[rA + rB]$
lhz rD, d(rA)	Load Half Word and Zero	None	$rD \leftarrow M[rA + d]_{15..0}$
lhzx rD, rA, rB	Load Half Word and Zero Indexed	None	$rD \leftarrow M[rA + rB]_{15..0}$
li rA, value	Load immediate	None	$rA \leftarrow \text{value}$
lis rA, value	Load immediate shifted left by 16 bits	None	$rA \leftarrow (\text{value} \ll 16)$
lwz rD, d(rA)	Load Word and Zero	None	$rD \leftarrow M[rA + d]$
lwzx rD, rA, rB	Load Word and Zero Indexed	None	$rD \leftarrow M[rA + rB]$
mr rA, rS	Move Register	None	$rA \leftarrow rS$
not rA, rS	Complement Register (invert)	None	$rA \leftarrow \sim rS$
ori rA, rS, value	OR Immediate	None	$rA \leftarrow rS \text{value}$
oris rA, rS, value	OR Immediate shifted left by 16 bits	None	$rA \leftarrow rS (\text{value} \ll 16)$
slwi rA, rS, value	Shift Left Immediate	None	$rA \leftarrow (rS \ll \text{value})$
srwi rA, rS, value	Shift Right Immediate	None	$rA \leftarrow (rS \gg \text{value})$
stb rS, d(rA)	Store Byte	None	$m[rA + d] \leftarrow rS_{7..0}$
stbx rS, rA, rB	Store Byte Indexed	None	$m[rA + rB] \leftarrow rS_{7..0}$
sth rS, d(rA)	Store Half Word	None	$M[rA + d]_{15..0} \leftarrow rS_{15..0}$
sthx rS, rA, rB	Store Half Word Indexed	None	$M[rA + rB]_{15..0} \leftarrow rS_{15..0}$
stw rS, d(rA)	Store Word	None	$M[rA + d] \leftarrow rS$
stwx rS, rA, rB	Store Word Indexed	None	$M[rA + rB] \leftarrow rS$
sub rD, rA, rB	Subtract	None	$rD \leftarrow rA - rB$
subi rD, rA, value	Subtract Immediate	None	$rD \leftarrow rA - \text{value}$
subis rD, rA, value	Subtract Immediate shifted left by 16 bits	None	$rD \leftarrow rA - (\text{value} \ll 16)$