

The TAL Instruction Set (cont.)		
Machine Code	Format	Effect
0010 00ss ssst tttt iiii iiii iiii iiii	addi R _t , R _s , I	R _t ← [R _s] + ([I ₁₅] ¹⁶ I _{15.0})
0010 01ss ssst tttt iiii iiii iiii iiii	addiu R _t , R _s , I	R _t ← [R _s] + ([I ₁₅] ¹⁶ I _{15.0}) (overflow ignored)
0011 00ss ssst tttt iiii iiii iiii iiii	andi R _t , R _s , I	R _t ← 0 ¹⁶ ([R _s] _{15.0} AND I _{15.0})
0011 1100 000t tttt iiii iiii iiii iiii	lui R _t , I	R _t ← I _{15.0} 0 ¹⁶
0011 01ss ssst tttt iiii iiii iiii iiii	ori R _t , R _s , I	R _t ← [R _s] _{31..16} ([R _s] _{15.0} OR I _{15.0})
0011 10ss ssst tttt iiii iiii iiii iiii	xori R _t , R _s , I	R _t ← [R _s] _{31..16} ([R _s] _{15.0} XOR I _{15.0})
0000 0000 0000 0000 dddd d000 0001 0000	mfhi R _d	R _d ← [HI]
0000 00ss sss0 0000 0000 0000 0001 0001	mthi R _s	HI ← [R _s]
0000 0000 0000 0000 dddd d000 0001 0010	mflo R _d	R _d ← [LO]
0000 00ss sss0 0000 0000 0000 0001 0011	mtlo R _s	LO ← [R _s]
0000 0000 000t tttt dddd diii ii00 0000	sll R _d , R _t , I	R _d ← [R _t] _{31-(I_{10.6})..0} 0 ^(I_{10.6})
0000 0000 000t tttt dddd diii ii00 0010	srl R _d , R _t , I	R _d ← 0 ^(I_{10.6}) [R _t] _{31..(I_{10.6})}
0000 0000 000t tttt dddd diii ii00 0011	sra R _d , R _t , I	R _d ← ([R _t] ₃₁) ^(I_{10.6}) [R _t] _{31..(I_{10.6})}
0000 00ss ssst tttt dddd d000 0000 0100	sllv R _d , R _t , R _s	R _d ← [R _t] _{(31-[R_s]_{14.0})..0} 0 ^{[R_s]_{14.0}}
0000 00ss ssst tttt dddd d000 0000 0110	srlv R _d , R _t , R _s	R _d ← 0 ^{[R_s]_{14.0}} [R _t] _{31..([R_s]_{14.0})}
0000 00ss ssst tttt dddd d000 0000 0111	srav R _d , R _t , R _s	R _d ← ([R _t] ₃₁) ^{[R_s]_{14.0}} [R _t] _{31..([R_s]_{14.0})}
1100 01bb bbbt tttt iiii iiii iiii iiii	lwcl F _t , I(R _b)	F _t ← M[[R _b] + ([I ₁₅] ¹⁶ [I] _{15.0})]
1110 01bb bbbt tttt iiii iiii iiii iiii	swcl F _t , I(R _b)	F _t → M[[R _b] + ([I ₁₅] ¹⁶ [I] _{15.0})]
0100 0110 000t tttt ssss sddd dd00 0000	add.s F _d , F _s , F _t	F _d ← [F _s] + [F _t]
0100 0110 000t tttt ssss sddd dd00 0001	sub.s F _d , F _s , F _t	F _d ← [F _s] - [F _t]
0100 0110 000t tttt ssss sddd dd00 0010	mul.s F _d , F _s , F _t	F _d ← [F _s] * [F _t]
0100 0110 000t tttt ssss sddd dd00 0011	div.s F _d , F _s , F _t	F _d ← [F _s]/[F _t]
0100 0110 1000 0000 ssss sddd dd10 0000	cvt.s.w G _d , W _s	G _d ← [W _s]
0100 0110 0000 0000 ssss sddd dd10 0100	cvt.w.s W _d , G _s	W _d ← [G _s]
0000 01ss sss0 0000 iiii iiii iiii iiii	bltz R _s , I	if([R _s] < 0), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)
0000 01ss sss0 0001 iiii iiii iiii iiii	bgez R _s , I	if([R _s] ≥ 0), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)
0001 10ss sss0 0000 iiii iiii iiii iiii	blez R _s , I	if([R _s] ≤ 0), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)
0001 11ss sss0 0000 iiii iiii iiii iiii	bgtz R _s , I	if([R _s] > 0), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)
0001 00ss ssst tttt iiii iiii iiii iiii	beq R _s , R _t , I	if([R _s] = [R _t]), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)
0001 01ss ssst tttt iiii iiii iiii iiii	bne R _s , R _t , I	if([R _s] ≠ [R _t]), then PC ← [PC] + 4 + ([I ₁₅] ¹⁴ [I] _{15.0} 0 ²)

The TAL Instruction Set		
Machine Code	Assembly Language Format	Effect
1000 11bb bbbt tttt iiii iiii iiii iiii	lw R _t , I(R _b)	R _t ← M[[R _b] + (I ₁₅) ¹⁶ I _{15.0}]
1000 00bb bbbt tttt iiii iiii iiii iiii	lb R _t , I(R _b)	R _t ← m[[R _b] + (I ₁₅) ¹⁶ I _{15.0}] ₇ ²⁴ m[[R _b] + (I ₁₅) ¹⁶ I _{15.0}]
1001 00bb bbbt tttt iiii iiii iiii iiii	lbu R _t , I(R _b)	R _t ← 0 ²⁴ m[[R _b] + (I ₁₅) ¹⁶ I _{15.0}]
1010 11bb bbbt tttt iiii iiii iiii iiii	sw R _t , I(R _b)	R _t → M[[R _b] + (I ₁₅) ¹⁶ I _{15.0}]
1010 00bb bbbt tttt iiii iiii iiii iiii	sb R _t , I(R _b)	[R _t] _{7..0} → m[[R _b] + (I ₁₅) ¹⁶ I _{15.0}]
0000 00ss ssst tttt dddd d000 0010 0000	add R _d , R _s , R _t	R _d ← [R _s] + [R _t]
0000 00ss ssst tttt dddd d000 0010 0010	sub R _d , R _s , R _t	R _d ← [R _s] - [R _t]
0000 00ss ssst tttt 0000 0000 0001 1000	mult R _s , R _t	HI LO ← [R _s] * [R _t]
0000 00ss ssst tttt 0000 0000 0001 1010	div R _s , R _t	LO ← [R _s]div[R _t]; HI ← [R _s]mod[R _t]
0000 00ss ssst tttt dddd d000 0010 0001	addu R _d , R _s , R _t	R _d ← [R _s] + [R _t], (overflow ignored)
0000 00ss ssst tttt dddd d000 0010 0011	subu R _d , R _s , R _t	R _d ← [R _s] - [R _t], (overflow ignored)
0000 00ss ssst tttt 0000 0000 0001 1001	multu R _s , R _t	HI LO ← [R _s] * [R _t], (overflow ignored)
0000 00ss ssst tttt 0000 0000 0001 1011	divu R _s , R _t	LO ← [R _s]div[R _t]; HI ← [R _s]mod[R _t] (overflow ignored)
0000 00ss ssst tttt dddd d000 0010 0100	and R _d , R _s , R _t	R _d ← [R _s] AND [R _t]
0000 00ss ssst tttt dddd d000 0010 0111	nor R _d , R _s , R _t	R _d ← [R _s] NOR [R _t]
0000 00ss ssst tttt dddd d000 0010 0101	or R _d , R _s , R _t	R _d ← [R _s] OR [R _t]
0000 00ss ssst tttt dddd d000 0010 0110	xor R _d , R _s , R _t	R _d ← [R _s] XOR [R _t]

The TAL Instruction Set (cont.)		
Machine Code	Format	Effect
0000 00ss ssst tttt dddd d000 0010 1010	slt R_d, R_s, R_t	if($[R_s] < [R_t]$), then $R_d \leftarrow 0^{31} \parallel 1$ else $R_d \leftarrow 0^{32}$
0010 10ss ssst tttt iiii iiii iiii iiii	slti R_t, R_s, I	if($[R_s] < ([I_{15}]^{16} \parallel [I]_{15..0})$), then $R_t \leftarrow 0^{31} \parallel 1$ else $R_t \leftarrow 0^{32}$
0000 10ii iiii iiii iiii iiii iiii iiii	j I	$PC \leftarrow [PC]_{31..28} \parallel [I]_{25..0} \parallel 0^2$
0000 00ss sss0 0000 0000 0000 0000 1000	jr R_s	$PC \leftarrow [R_s]$
0000 11ii iiii iiii iiii iiii iiii iiii	jal I	$R_{31} \leftarrow [PC] + 4$; $PC \leftarrow [PC]_{31..28} \parallel [I]_{25..0} \parallel 0^2$
0000 00ss sss0 0000 dddd d000 0000 1001	jalr R_d, R_s	$R_d \leftarrow [PC] + 4$; $PC \leftarrow [R_s]$
0000 0000 0000 0000 0000 0000 0000 1100	syscall	$PC \leftarrow \text{ExceptionHandler}$
0000 00xx xxxx xxxx xxxx 0xxx xx00 1101	break	$PC \leftarrow \text{ExceptionHandler}$
0100 0010 0000 0000 0000 0000 0001 0000	rfe	restore state information
0100 0000 000t tttt dddd d000 0000 0000	mfc0 R_t, C_d	$R_t \leftarrow [C_{d(CP0)}]$
0100 0000 100t tttt dddd d000 0000 0000	mtc0 R_t, C_d	$C_{d(CP0)} \leftarrow [R_t]$
0100 0100 000t tttt dddd d000 0000 0000	mfc1 R_t, F_d	$R_t \leftarrow [F_{d(CP1)}]$
0100 0100 100t tttt dddd d000 0000 0000	mtc1 R_t, F_d	$F_{d(CP1)} \leftarrow [R_t]$

General Notes

- (1) I specifies part of the instruction.
- (2) $M[i]$ is the value of the (aligned) word of memory beginning at location i.
- (3) $m[i]$ is the value of the byte of memory at location i.
- (4) $R_b, R_d, R_s,$ and R_t specify general registers. R_{31} specifies register 31.
- (5) C_d specifies a control register (co-processor 0).
- (6) F_d specifies a floating point register (co-processor 1). G_d and G_s specify a floating point register in single-precision floating point format. W_d and W_s specify a floating point register in two's complement format.
- (7) \parallel indicates concatenation of bit fields.
- (8) Superscripts indicate repetitions of a binary value.
- (9) Subscripts indicate bit positions (Little-Endian) of sub-field.
- (10) Square brackets ([]) indicate "the contents of."