

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	$R_t \leftarrow [C_d(\text{CP0})]$
0100 0000 100t tttt dddd d000 0000 0000	mtc0 R_t, C_d	$C_d(\text{CP0}) \leftarrow [R_t]$
0100 0100 000t tttt dddd d000 0000 0000	mfc1 R_t, F_d	$R_t \leftarrow [F_d(\text{CP1})]$
0100 0100 100t tttt dddd d000 0000 0000	mtc1 R_t, F_d	$F_d(\text{CP1}) \leftarrow [R_t]$

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."