Table 6-3. TMS7000 Family Instruction Overview

MNEMONIC		OPCODE	BYTES	CYCLES $T_C(C)$	STATUS C	N	Z	I	OPERATION DESCRIPTION
ADC	B,A	69	1	5	R	R	R	x	(s) + (d) + (C) → (d)
	Rs,A	19	2	8					Add the source, destination, and carry bit
	Rs,B	39	2	8					together. Store at the destination address.
	Rs,Rd	49	3	10					
	%iop,A	29	2	7					
	%iop,B	59	2	7					
	%iop,Rd	79	3	9					
ADD	B,A	68	1	5	R	R	R	x	(s) + (d) → (d)
	Rs,A	18	2	8					Add the source and destination operands at
	Rs,B	38	2	8					the destination address.
	Rs,Rd	48	3	10					
	%iop,A	28	2	7					
	%iop,B	58	2	7					
	%iop,Rd	78	3	9					
AND	B,A	63	1	5	0	R	R	x	(s) .AND. (d) → (d)
	Rs,A	13	2	8					AND the source and destination operands
	Rs,B	33	2	8					together and store at the destination
	Rs,Rd	43	3	10					address.
	%iop,A	23	2	7					
	%iop,B	53	2	7					
	%iop,Rd	73	3	9					
ANDP	A,Pd	83	2	10	0	R	R	x	(s) .AND. (Pn) → (Pn)
	B,Pd	93	2	9					AND the source and destination operands
	%iop,Pd	A3	3	11					together, and store at the destination
									address.
(1) BTJO	B,A,Ofst	66	2	7 (9)	0	R	R	x	
	Rn,A,Ofst	16	3	10 (12)					If (s) .AND. (d) ≠ 0,
	Rn,B,Ofst	36	3	10 (12)					then (PC) + offset → (PC)
	Rn,Rd,Ofst	46	4	12 (14)					If the AND of the source and destination
	%iop,A,Ofst	26	3	9 (11)					operands ≠ 0, the PC will be modified to
	%iop,B,Ofst	56	3	9 (11)					include the offset.
	%iop,Rn,Ofst	76	4	11 (13)					
(1) BTJOP	A,Pn,Ofst	86	3	11 (13)	0	R	R	x	
	B,Pn,Ofst	96	3	10 (12)					If (s) .AND. (Pn) ≠ 0,
	%>iop,Pn,Ofst	A6	4	12 (14)					then (PC) + (offset) → (PC)
									If the AND of the source and destination
									operands ≠ 0, the PC will be modified to
									include the offset.
(1) BTJZ	B,A,Ofst	67	2	7 (9)	0	R	R	x	
	Rn,A,Ofst	17	3	10 (12)					If (s) .AND. NOT(d) ≠ 0,
	Rn,B,Ofst	37	3	10 (12)					then (PC) + (offset) → (PC)
	Rn,Rf,Ofst	47	4	12 (14)					If the AND of the source and NOT(destina-
	%>iop,A,Ofst	27	3	9 (11)					tion operands ≠ 0, the PC will be modi-
	%>iop,B,Ofst	57	3	9 (11)					fied to include the offset.
	%>iop,Rn,Ofst	77	4	11 (13)					

Note: Add two to cycle count if branch is taken.

Legend:
0 Status Bit set always to 0.
1 Status Bit set always to 1.
R Status Bit set to a 1 or a 0 depending on results of operation.
x Status Bit not affected.
b Bit ( ) affected.
Ofst Offset

Table 6-3. TMS7000 Family Instruction Overview (Continued)

MNEMONIC		OPCODE	BYTES	CYCLES $T_c(C)$	STATUS C	N	Z	I	OPERATION DESCRIPTION
(1) BTJZP	A,Pn,Ofst	87	3	11 (13)	0	R	R	x	If (s) .AND. NOT(Pn) ≠ 0, then (PC) + offset → (PC) If the AND of the source and NOT(destination) operands ≠ 0, the PC will be modified to include the offset.
	B,Pn,Ofst	97	3	10 (12)					
	%>iop,Pn,Ofst	A7	4	12 (14)					
BR	@Label	8C	3	10	x	x	x	x	(d) → (PC) The PC will be replaced with the contents of the destination operand.
	@Label(B)	AC	3	12					
	*Rn	9C	2	9					
CALL	@Label	8E	3	14	x	x	x	x	(SP) + 1 → (SP) (PC MSB) → (Stack) (SP) + 1 → (SP) (PC LSB) → (Stack) Operand Address → (PC)
	@Label(B)	AE	3	16					
	*Rn	9E	2	13					
CLR	A	B5	1	5	0	0	1	x	0 → (d) Clear the destination operand.
	B	C5	1	5					
	Rd	D5	2	7					
CLRC		B0	1	6	0	R	R	x	0 → (C) Clears the carry bit.
CMP	B,A	6D	1	5	R	R	R	x	(d) - (s) computed Set flags on the result of the source operand subtracted from the destination operand.
	Rn,A	1D	2	8					
	Rn,B	3D	2	8					
	Rn,Rn	4D	3	10					
	%iop,A	2D	2	7					
	%iop,B	5D	2	7					
	%iop,Rn	7D	3	9					
CMPA	@Label	8D	3	12	R	R	R	x	(A) - (s) computed Set flags on result of the source operand subtracted from A.
	@Label(B)	AD	3	14					
	*Rn	9D	2	11					
DAC	B,A	6E	1	7	R	R	R	x	(s) + (d) + (C) → (d) (BCD) The source, destination, and the carry bit are added, and the BCD sum is stored at the destination address.
	Rs,A	1E	2	10					
	Rs,B	3E	2	10					
	Rs,Rd	4E	3	12					
	%>iop,A	2E	2	9					
	%>iop,B	5E	2	9					
	%>iop,Rd	7E	3	11					
DEC	A	B2	1	5	R	R	R	x	(d) - 1 → (d) Decrement destination operand by 1.
	B	C2	1	5					
	Rd	D2	2	7					
DECD	A	BB	1	9	R	R	R	x	(rp) - 1 → (rp) Decrement register pair by 1. C = 0 on 0 - FFFF transition.
	B	CB	1	9					
	Rp	DB	2	11					
DINT		06	1	5	0	0	0	0	0 → (global interrupt enable bit) Clear the I bit.

Note: Add two to cycle count if branch is taken.

Legend:

0	Status Bit set always to 0.
1	Status Bit set always to 1.
R	Status Bit set to a 1 or a 0 depending on results of operation.
x	Status Bit not affected.
b	Bit ( ) affected.
Ofst	Offset

Table 6-3. TMS7000 Family Instruction Overview (Continued)

MNEMONIC		OPCODE	BYTES	CYCLES $T_c(C)$	STATUS C	N	Z	I	OPERATION DESCRIPTION
(1) DJNZ	A,Ofst	BA	2	7 (9)	x	x	x	x	(d) - 1 → (d);
	B,Ofst	CA	2	7 (9)					If (d) ≠ 0, (PC) + (offset) → (PC)
	Rd,Ofst	DA	3	9 (11)					
DSB	B,A	6F	1	7	R	R	R	x	(d) - (s) - 1 + (C) → (d) (BCD)
	Rs,A	1F	2	10					The source operand is subtracted from the
	Rs,B	3F	2	10					destination; this sum is then reduced by 1
	Rs,Rd	4F	3	12					and the carry bit is then added to it. The
	%>iop,A	2F	2	9					result is stored as a BCD number.
	%>iop,B	5F	2	9					
	%>iop,Rd	7F	3	11					
EINT		05	1	5	1	1	1	1	1 → (global interrupt enable bit)
									Set the I bit.
IDLE		01	1	6	x	x	x	x	(PC) → (PC) until interrupt
									(PC) + 1 → (PC) after return from interrupt
									Stops µC execution until an interrupt.
INC	A	B3	1	5	R	R	R	x	(d) + 1 → (d)
	B	C3	1	5					Increase the destination operand by 1.
	Rd	D3	2	7					
INV	A	B4	1	5	0	R	R	x	NOT(d) → (d)
	B	C4	1	5					1's complement the destination operand.
	Rd	D4	2	7					
JMP	Ofst	E0	2	7	x	x	x	x	(PC) + (offset) → (PC)
									The PC is modified by an offset to create a
									new PC value.
(1)					x	x	x	x	
JC	Ofst	E3	2	5 (7)					If conditions are met,
JEQ	Ofst	E2	2	5 (7)					then (PC) + offset → (PC)
JGE	Ofst	E5	2	5 (7)					If the needed conditions are met, the PC
JGT	Ofst	E4	2	5 (7)					is modified by the offset to form a new
JHS	Ofst	E3	2	5 (7)					PC value.
JL	Ofst	E7	2	5 (7)					
JNC	Ofst	E7	2	5 (7)					
JNE	Ofst	E6	2	5 (7)					
JNZ	Ofst	E6	2	5 (7)					
JP	Ofst	E4	2	5 (7)					
JPZ	Ofst	E5	2	5 (7)					
JZ	Ofst	E2	2	5 (7)					
LDA	@Label	8A	3	11	0	R	R	x	(s) → (A)
	@Label(B)	AA	3	13					Move the source operand to A.
	*Rn	9A	2	10					

Note: Add two to cycle count if branch is taken.

Legend:

0 Status Bit set always to 0.
1 Status Bit set always to 1.
R Status Bit set to a 1 or a 0 depending on results of operation.
x Status Bit not affected.
b Bit ( ) affected.
Ofst Offset

Table 6-3. TMS7000 Family Instruction Overview (Continued)

MNEMONIC		OPCODE	BYTES	CYCLES $T_C(C)$	STATUS C	N	Z	I	OPERATION DESCRIPTION
LDSP		0D	1	5	x	x	x	x	(B) → (SP) Load SP with Register B's contents.
MOV	A,B	C0	1	6	0	R	R	x	(s) → (d) Replace the destination operand with the source operand.
	A,Rd	D0	2	8					
	B,A	62	1	5					
	B,Rd	D1	2	7					
	Rs,A	12	2	8					
	Rs,B	32	2	8					
	Rs,Rd	42	3	10					
	%>iop,A	22	2	7					
	%>iop,B	52	2	7					
	%>iop,Rd	72	3	9					
MOVD	%>iop,Rp	88	4	15	0	R	R	x	(rp) → (rp) Copy the source register pair to the destination register pair.
	%>iop(B),Rp	A8	4	17					
	Rp,Rp	98	3	14					
MOVP	A,Pd	82	2	10	0	R	R	x	(s) → (d) Copy the source operand into the destination operand.
	B,Pd	92	2	9					
	%>iop,Pd	A2	3	11					
	Ps,A	80	2	9					
	Ps,B	91	2	8					
MPY	B,A	6C	1	44	0	R	R	x	(s) × (d) → (A,B) Multiply the source and destination operands, store the result in Registers A (MSB) and B (LSB).
	Rs,A	1C	2	47					
	Rs,B	3C	2	47					
	Rn,Rn	4C	3	49					
	%>iop,A	2C	2	46					
	%>iop,B	5C	2	46					
	%>iop,Rn	7C	3	48					
NOP		00	1	5	x	x	x	x	(PC) + 1 → (PC) Add 1 to the PC.
OR	B,A	64	1	5	0	R	R	x	(s) .OR. (d) → (d) Logically OR the source and destination operands, and store the results at the destination address.
	Rs,A	14	2	8					
	Rs,B	34	2	8					
	Rs,Rd	44	3	10					
	%>iop,A	24	2	7					
	%>iop,B	54	2	7					
	%>iop,Rd	74	3	9					
ORP	A,Pd	84	2	10	0	R	R	x	(s) .OR. (d) → (d) Logically OR the source and destination operands, and store the results at the destination address.
	B,Pd	94	2	9					
	%>iop,Pd	A4	3	11					
POP	A	B9	1	6	0	R	R	x	(Stack Top) → (d) (SP) -1 → (SP) Copy the last byte on the stack into the destination address.
	B	C9	1	6					
	Rd	D9	2	8					

Note: Add two to cycle count if branch is taken.

Legend:
0 Status Bit set always to 0.
1 Status Bit set always to 1.
R Status Bit set to a 1 or a 0 depending on results of operation.
x Status Bit not affected.
b Bit ( ) affected.

Table 6-3. TMS7000 Family Instruction Overview (Continued)

MNEMONIC	OPCODE	BYTES	CYCLES $T_c(C)$	STATUS C N Z I	OPERATION DESCRIPTION
POP ST	08	1	6	0 R R x	(Stack Top) → (Status Register) (SP) - 1 → (SP) Replace the Status Register with the last byte of the stack.
PUSH A B Rs	B8 C8 D8	1 1 2	6 6 8	0 R R x	(s) → (Stack) (SP) + 1 → (SP) Copy the operand onto the stack.
PUSH ST	0E	1	6	0 R R x	(Status Register) → (Stack) (SP) + 1 → (SP) Copy the Status Register onto the stack.
RETI	0B	1	9	Loaded from the stack	Stack → (PC) LSByte (SP) - 1 → (SP) Stack → (PC) MSByte (SP) - 1 → (SP) Stack → Status Register (SP) - 1 → (SP)
RETS	0A	1	7	x x x x	(Stack) → (PC LSB) (SP) - 1 → (SP) (Stack) → (PC MSB) (SP) - 1 → (SP)
RL A B Rd	BE CE DE	1 1 2	5 5 7	b7 R R x	Bit(n) → Bit(n + 1) Bit(7) → Bit(0) and Carry
RLC A B Rd	BF CF DF	1 1 2	5 5 7	b7 R R x	Bit(n) → Bit(n + 1) Carry → Bit(0) Bit(7) → Carry
RR A B Rd	BC CC DC	1 1 2	5 5 7	b0 R R x	Bit(n + 1) → Bit(n) Bit(0) → Bit(7) and Carry
RRC A B Rd	BD CD DD	1 1 2	5 5 7	b0 R R x	Bit(n + 1) → Bit(n) Carry → Bit(7) Bit(0) → Carry
SBB B,A Rs,A Rs,B Rs,Rd %>iop,A %>iop,B %>iop,Rd	6B 1B 3B 4B 2B 5B 7B	1 2 2 3 2 2 3	5 8 8 10 7 7 9	R R R x	(d) - (s) - 1 + (C) → (d) Destination minus source minus 1 plus carry; stored at the destination address.
SETC	07	1	5	1 0 1 x	1 → (C) Set the carry bit.

Note: Add two to cycle count if branch is taken.

Legend:
0 Status Bit set always to 0.
1 Status Bit set always to 1.
R Status Bit set to a 1 or a 0 depending on results of operation.
x Status Bit not affected.
b Bit ( ) affected.

Table 6-3. TMS7000 Family Instruction Overview (Concluded)

MNEMONIC	OPCODE	BYTES	CYCLES $T_c(C)$	STATUS C	N	Z	I	OPERATION DESCRIPTION
STA @Label	8B	3	11	0	R	R	x	(A) → (d)
@Label(B)	AB	3	13					Store A at the destination.
*Rd	9B	2	10					
STSP	09	1	6	x	x	x	x	(SP) → (B)
								Copy the SP into Register B.
SUB B,A	6A	1	5	R	R	R	x	(d) - (s) → (d)
Rs,A	1A	2	8					Store the destination operand minus the
Rs,B	3A	2	8					source operand into the destination.
Rs,Rd	4A	3	10					
%>iop,A	2A	2	7					
%>iop,B	5A	2	7					
%>iop,Rd	7A	3	9					
SWAP A	B7	1	8	R	R	R	x	d(Hn,Ln) → d(Ln,Hn)
B	C7	1	8					Swap the operand's hi and lo nibbles.
Rn	D7	2	10					
TRAP 0–23	E8–FF	1	14	x	x	x	x	(SP) + 1 → (SP)
								(PC MSB) → (Stack)
								(SP) + 1 → (SP)
								(PC LSB) → (Stack)
								(Entry Vector) → (PC)
TSTA	B0	1	6	0	R	R	x	0 → (C)
								Set carry bit; set sign and zero flags on the value of Register A.
TSTB	C1	1	6	0	R	R	x	0 → (C)
								Set carry bit; set sign and zero flags on the value in Register B.
XCHB A	B6	1	6	0	R	R	x	(B) ←→ (d)
Rn	D6	2	8					Swap the contents of Register B with (d).
XOR B,A	65	1	5	0	R	R	x	(s) .XOR. (d) → (d)
Rs,A	15	2	8					Logically exclusive OR the source and
Rs,B	35	2	8					destination operands, store at the
Rs,Rd	45	3	10					destination address.
%>iop,A	25	2	7					
%>iop,B	55	2	7					
%>iop,Rd	75	3	9					
XORP A,Pd	85	2	10	0	R	R	x	(s) .XOR. (Pn) → (Pn)
B,Pd	95	2	9					Logically exclusive OR the source and
%>iop,Pd	A5	3	11					destination operands, store at the destination.

Note: Add two to cycle count if branch is taken.

Legend:
0 Status Bit set always to 0.
1 Status Bit set always to 1.
R Status Bit set to a 1 or a 0 depending on results of operation.
x Status Bit not affected.
b Bit ( ) affected.