Model 990 Computer Assembly Language Programming Card

Part No. 2272065-9701 **
1 January 1981

TEXAS INSTRUMENTS
INCORPORATED

DIGITAL SYSTEMS GROUP
POST OFFICE BOX 2909 AUSTIN, TEXAS 78769

990 CPU Instructions

Mnemonic	Operands	Op Code	For-mat	Status Affected	Instruction
A	G,G←	A000	1	0 - 4	Add (word)
AB	G,G←	B000	1	0 - 5	Add (byte)
ABS	G —	0740	6	0 - 4	Absolute Value[8]
AD*	G —	0E40	6	0 - 4	Add Double Precision Real
AI	WR←,I	0220	8	0 - 4	Add Immediate
AM*	G,G←,CNT	002A	11	0 - 4	Add Multiple Precision
ANDI	WR←,I	0240	8	0 - 2	And Immediate
ANDM*	G,G←,CNT	0028	11	0 - 2	And Multiple Precision
AR*	G —	0C40	6	0 - 4	Add Real
ARJ*	PC,CNT,WR	0C0D	17	—	Add to Register and Jump
B	G —	0440	6	—	Branch
BDC*	G,G←,CNT	0023	11	0 - 2,4	Binary to Decimal Conversion
BIND*	G —	0140	6	—	Branch Indirect
BL	G —	0680	6	—	Branch and Link (R11)
BLSK*	WR,I	00B0	6	—	Branch Immediate and Push Link to Stack
BLWP	G —	0400	6	—	Branch; Load Workspace Pointer
C	G,G	8000	1	0 - 2	Compare (Word)
CB	G,G	9000	1	0 - 2,5	Compare (Byte)
CDE*	— —	0C05	7	0 - 4	Convert DBL. Precision Real to Extended Integer
CDI*	— —	0C01	7	0 - 4	Convert DBL. Precision Real to Integer
CED*	— —	0C07	7	0 - 2	Convert Extended Integer to DBL. Precision Real
CER*	— —	0C06	7	0 - 2	Convert Extended Integer to Real
CI	WR,I	0280	8	0 - 2	Compare Immediate
CID*	G —	0E80	6	0 - 2	Convert Integer to DBL. Precision Real
CIR*	G —	0C80	6	0 - 2	Convert Integer to Real
CKOF	— —	03C0	7	—	Clock Off[1,8]
CKON	— —	03A0	7	—	Clock On[1,8]
CLR	G —	04C0	6	—	Clear (Word)
CNTO*	G,G←,CNT	0020	11	2	Count Ones
COC	G,WR	2000	3	2	Compare Ones Corresponding
CRC*	G,G←,CNT,CKPT	0E20	12	2	Cyclic Redundancy Code Calculation
CRE*	— —	0C04	7	0 - 4	Convert Real to Extended Integer
CRI*	— —	0C00	7	0 - 4	Convert Real to Integer
CS*	G,G←,CNT,CKPT	0040	12	0 - 2	Compare Strings
CZC	G,WR	2400	3	2	Compare Zeros Corresponding
DBC*	G,G←,CNT	0024	11	0 - 2,4	Decimal to Binary Conversion
DD*	G —	0F40	6	0 - 4	Divide DBL. Precision Real
DEC	G —	0600	6	0 - 4	Decrement by One
DECT	G —	0640	6	0 - 4	Decrement by Two
DINT*	— —	002F	7	—	Disable Interrupt[1]
DIV	G,WR←	3C00	9	4	Divide
DIVS*	G —	0180	6	0 - 2,4	Divide Signed
DR*	G —	0D40	6	0 - 4	Divide Real
EINT*	— —	002E	7	—	Enable Interrupts[1]
EMD*	—	002D	7	0 - 15	Execute Micro-Diagnostic[1]
EP*	G,G←,CNT,CNT	03F0	21	0 - 2	Extend Precision

990 CPU Instructions (Continued)

Mnemonic	Operands	Op Code	For-mat	Status Affected	Instruction
IDLE	— —	0340	7	—	Computer Idle[1,6]
INC	G —	0580	6	0 - 4	Increment (by One)
INCT	G —	05C0	6	0 - 4	Increment (by Two)
INSF*	G,G←,POS,WID	0C10	16	0 - 2	Insert Field
INV	G —	0540	6	0 - 2	Invert
IOF*	G←,POS,WID	0E00	15	—	Invert Order of Field
JEQ	PC —	1300	2	—	Jump Equal (ST2 = 1)
JGT	PC —	1500	2	—	Jump Greater Than (ST1 = 1)
JH	PC —	1B00	2	—	Jump High (ST0 = 1 and ST2 = 0)
JHE	PC —	1400	2	—	Jump High or Equal (ST0 or ST2 = 1)
JL	PC —	1A00	2	—	Jump Low (ST0 and ST2 = 0)
JLE	PC —	1200	2	—	Jump Low or Equal (ST0 = 0 or ST2 = 1)
JLT	PC —	1100	2	—	Jump Less Than (ST1 and ST2 = 0)
JMP	PC —	1000	2	—	Jump Unconditional
JNC	PC —	1700	2	—	Jump No Carry (ST3 = 0)
JNE	PC —	1600	2	—	Jump Not Equal (ST2 = 0)
JNO	PC —	1900	2	—	Jump No Overflow (ST4 = 0)
JOC	PC —	1600	2	—	Jump on Carry (ST3 = 1)
JOP	PC —	1C00	2	—	Jump Odd Parity (ST5 = 1)
LCS*	WR	00A0	18	—	Load Writable Control Store[1]
LD*	G	0F80	6	0 - 2	Load DBL. Precision Real
LDCR	G[2]	3000	4	0 - 2,5	Load CRU[7]
LDD	G —	07C0	6	—	Long Distance Destination[1]
LDS	G —	0780	6	—	Long Distance Source[1]
LI	WR←,1	0200	8	0 - 2	Load Immediate
LIM*	WR	0070	18	12 - 15	Load Interrupt Mask[1]
LIMI	I —	0300	8	12 - 15	Load Interrupt Mask Immediate[1]
LMF	WR←[3]	0320	10	—	Load Mapfile[1]
LR*	G	0D80	6	0 - 2	Load Real
LREX	— —	03E0	7	12 - 15	Load or Restart Execution[1,6]
LST*	WR	0080	18	0 - 15	Load Status Register[6]
LTO*	G,G←,CNT	001F	11	2	Left Test for Ones
LWP*	WR	0090	16	—	Load Workspace Pointer
LWPI	I —	02E0	8	—	Load Immediate Workspace Pointer
MD*	G	0F00	6	0 - 4	Multiply DBL. Precision Real
MOV	G,G←	C000	1	0 - 2	Move (Word)
MOVA*	C,G←	002B	19	0 - 2	Move Address
MOVB	G,G←	D000	1	0 - 2,5	Move (Byte)
MOVS*	G,G←,CNT,CKPT	0060	12	0 - 2	Move String
MPY	G,WR←	3800	9	—	Multiply
MPYS*	G	01C0	6	0 - 2	Multiply Signed
MR*	G	0D00	6	0 - 4	Multiply Real
MVSK*	G,G←,CNT,CKPT	00D0	12	0 -2	Move String from Stack
MVSR*	G,G←,CNT,CKPT	00C0	12	0 - 2	Move String Reverse
NEG	G —	0500	6	0 - 4	Negate (Two's Complement)
NEGD*		0C03	7	0 - 2	Negate Double Precision Real
NEGR*	—	0C02	7	0 - 2	Negate Real

990 CPU Instructions (Continued)

Mnemonic	Operands	Op Code	For-mat	Status Affected	Instruction
NRM*	G,G←,CNT	0C08	11	0 - 2	Normalize
ORI	WR←,I	0260	8	0 - 2	OR Immediate
ORM*	G,G←,CNT	0027	11	0 - 2	OR Multiple Precision
POPS*	G,G←,CNT,CKPT	00E0	12	0 - 2	Pop String from Stack
PSHS*	G,G←,CNT,CKPT	00F0	12	0 - 2	Push String to Stack
RSET	— —	0360	7	12 - 15	Reset AU[1,6]
RTO*	G,G←,CNT	001E	11	2	Right Test for Ones
RTWP	— —	0380	7	0 - 15	Return from Subroutine[6]
S	G,G←	6000	1	0 - 4	Subtract (Word)
SB	G,G←	7000	1	0 - 5	Subtract (Byte)
SBO	CRU —	1D00	2	—	Set CRU Bit to One
SBZ	CRU —	1E00	2	—	Set CRU Bit to Zero
SD*	G	0EC0	6	0 - 4	Subtract DBL. Precision Real
SEQB*	G,G,CNT, CKPT	005O	12	0 - 2	Search String for Equal Byte
SETO	G, —	0700	6	—	Set Ones
SLA	WR[4]	0A00	5	0 - 4	Shift Left Arithmetic
SLAM*	G,G←,CNT	001D	13	0 - 4	Shift Left Arithmetic Multiple Precision
SLSL*	COND,G,G←	0021	20	2	Search List Logical Address
SLSP*	COND,G,G←	0022	20	2	Search List Physical Address[1]
SM*	G,G←,CNT	0029	11	0 - 4	Subtract Multiple Precision
SNEB*	G,G,CNT,CKPT	0E10	12	0 - 2	Search String for Not Equal Byte
SOC	G,G←	E000	1	0 - 2	Set Ones Corresponding (Word)
SOCB	G,G←	F000	1	0 - 2,5	Set Ones Corresponding (Byte)
SR*	G	0CC0	6	0 - 4	Subtract Real
SRA	WR←[4]	0800	5	0 - 3	Shift Right Arithmetic (MSB Extended)
SRAM*	G,G←,CNT	001C	13	0 - 3	Shift Right Arithmetic Multiple Precision
SRC	WR←[4]	0B00	5	0 - 3	Shift Right Circular
SRJ*	PC,CNT,WR←	0C0C	17	—	Subtract from Register and Jump
SRL	WR←[4]	0900	5	0 - 3	Shift Right Logical
STCR	G←[2]	3400	4	0 - 2,5	Store from CRU[7]
STD*	G	0FC0	6	0 - 2	Store Double Precision Real
STPC*	WR,—	0030	18	—	Store Program Counter
STR*	G	0DC0	6	0 - 2	Store Real
STST	WR, —	02C0	18	—	Store Status Register
STWP	WR, —	02A0	18	—	Store Workspace Pointer
SWPB	G, —	06C0	6	—	Swap Bytes
SWPM*	G,G,CNT	0025	11	0 - 2	Swap Multiple Precision
SZC	G,G←	4000	1	0 - 2	Set Zeros Corresponding (Word)
SZCB	G,G←	5000	1	0 - 2,5	Set Zeros Corresponding (Byte)
TB	CRU —	1F00	2	2	Test CRU Bit
TCMB*	G,POS	0C0A	14	2	Test and Clear Memory Bit
TMB*	G,POS	0C09	14	2	Test Memory Bit
TS*	G,G←,CNT,CKPT	0E30	12	0 - 2	Translate Strings
TSMB*	G,POS	0C0B	14	2	Test and Set Memory Bit
X	G —	0480	6	—	Execute
XF*	G,G←,POS,WID	0C30	16	0 - 2	Extract Field
XIT*	—	0C0E/F	7	—	Exit from Floating Point Interpreter