EDITOR/ASSEMBLER QUICK REFERENCE CARD

A handy guide to the Instructions, Pseudo-instructions, and Directives of TMS9900 Assembly language and the Utilities for use on the TI-99/4 or TI-99/4A Home Computer. For a full discussion of these and other features, see the Editor/Assembler owner's manual.



Copyright © 1982 Texas Instruments Incorporated

Printedia U.S.A.

1035988-1

ADDRESSING SUMMARY

ADDRES	SING SUMM	RANI	
First Operand	Second Operand	Op-code	Format
G	G*	A000	1
G	G*	B000	- 1
G		0740	VI
			VIII
	1		VBI
Ģ	-		VI
	-		VI
	_		VI
	G		!
			1
WH	1		VIII
_	-		VII
	-		VΙΙ
Ğ	- IND		V)
G			iii
			Ϋ́I
	=		νi
Ğ	we.		iΧ
			νii
	_		νï
			νi
Ğ	_		νi
PG	_		ii ii
PC	_	1500	ii
PC		1B00	If
PC	_	1400	11
	_	1A00	H
	_		H H
	_		31
	-		11
	-		
			H
	-		!!
	_		ü
			II IV
			VIII
	<u>'</u> _		VIII
<u>'</u> _	_		VII
Ē	_		νiii
	G*		ĭ
			i
			ix
	_		νì
WR*	1	0260	VIII
_	_	0360	VII
-	-	0380	VII
	First Operand **	First Operand G G G G G G G G G G G G G G G G G G G	First Operand

ADDRESSING SUMMARY (CONTD)

Mnemonic	First Operand	Second Operand	Op-code	Format
S	G	G*	6000	1
SB	G	G*	7000	i
S80	CRU	_	1D00	ù
SBZ	CRU	-	1E00	ii
SETO	G	_	0700	ŸΙ
SLA	WR*	Note 2	0A00	v
SOC	G	G*	E000	ı
SOCB	G	G*	F000	ı
SRA	WR*	Note 2	0800	٧
SRC	WR*	Note 2	OBGO	V
SRL	WR*	Note 2	0900	V
STCR	G*	Note 1	3400	IV
STST	WR	_	02C0	VIII
STWP	WR	_	02A0	VIII
SWPB	G	-	06C0	VI
SZC	G	G*	4000	1
SZCB	G	G*	5000	1
TB	CRU	-	1F00	ŧI
X	G	_	0480	VI
XOP	Ģ	Note 3	2C00	ŧΧ
XOR	G	WR*	2800	#11

Notes:

- 1. The second operand is the number of bits to be transferred, from 0 through 15, with 0 meaning 16 bits.
 2. The second operand is the shift count, from 0 through 15, 0 indicates that the count is in bits 12 through 15 of Workspace Register 0. When the count is 0 and bits 12 through 15 of Workspace Register 0 equal 0, the count is 16.
 3. The second operand specifies the extended operation, from 0 through 15. The disposition of the result may or may not be in the first operand address, as you determine.

General address:
 Workspace Register address
 Indirect Workspace Register address
 Symbolic memory address
 Indexed memory address
 Indirect Workspace Register auto-increment address

address
WR —Workspace Register address

Workspace Register address
 CRU — Program counter relative address
 CRU — CRU bit address
 I — Immediate value
 The address into which the result is placed when two

operands are required

3

Worksp Worksp Symbol Indexed Worksp

Notes:

1. The laborated the first term of the first ter

2. The form of 10, opera 3. The Tindex follow

I—TWO II-JUN III—LO

IV--GR V-REG

VII-C VIII—IN ix-xo

ADDRESSING MODES

Addressing Mode	T-field Value	Example
Workspace Register Workspace Register Indirect	00 (0) 01 (1)	5 ⁴7
Workspace Register Indirect Symbolic Memory ^{1, 2} Indexed Memory ^{1, 3}	10 (2) 10 (2)	@LABEL @LABEL(5)
Workspace Register Indirect Auto-increment	11 (3)	*7+

Notes:

- 1. The instruction requires an additional word for each T-field binary value of 10. The additional word contains a memory address.

 2. The four-bit field immediately following the T-field binary value of 10, called the S (for a source operand) or D (for a destination operand) field, is set to zero by the Assembler.

 3. The T-field binary value of 10 indicates both symbolic and indexed memory addressing modes. If the four-bit field which follows it contains a zero value, it is a symbolic addressing mode, fit is non-zero, it is an indexed addressing mode, and the non-zero value is the number of the index register. Therefore, Workspace Register D cannot be used for indexing.

INSTRUCTION FORMATS

0123456789101112131415

	0123430700	101112	10 14 10
I-TWOGENERALADDRESS	O-C B Td D	Ts	S
II JUMP and BIT I/O	OP-CODE	DIS	Ρ
TII-LOGICAL	OP CODE D	Ts	S
IVCRU MULTI-BIT	OP-CODE C	Ts	S
V-REGISTER SHIFT	OP-CODE	С	W
VI—SINGLE ADDRESS	OP-CODE	Ts	S
VII-CONTROL	OP-CODE	0 0	000
VIII-IMMEDIATE	OP-CODE	00	W
IX—XOP and MULT. AND DIV.	OP-CODE D	Ts	s

STATUS REGISTER

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

L> A> EQ	C OV OP	x	INT MASK
Name	Bit Number	Meaning	
L>	0	Logical greater than	
A>	1	Arithmetic greater than	
EQ	2	Equal	
C	3	Carry	
ΟV	4	Overflow	
OP	5	Odd parity	
Х	6	Extended operation	
_	7-11	Reserved	
INT. MAS	K12—15	Interrupt mask	

INSTRUCTIONS AND PSEUDO-INSTRUCTIONS

Name	Mnemonic	Op-code	Format	Status Bits Affected
Load immediate	Ш	0200	VIII	0-2
Add Immediate	ΑI	0220	VIII	0-4
And immediate	ANDI	0240	VIII	0-2
Orimmediate	ORI	0260	VIII	0-2
Compare Immediate	CI.	0280	VIII	ō—2
Store Workspace pointer	STWP	02A0	νiii	
Store Status	STST	02C0	VIII	_
Load Workspace pointer				
Immediate	LWPI	02E0	VIII	_
Load interrupt mask				
immediate	LIMI	0300	VIII	1215
ldle	IDLE	0340	VII	_
Reset	RSET	0360	VII	_
Return with Workspace				
pointer	RTWP	0380	VII	0-15
Clock on	CKON	03A0	VII	_
Clock off	CKOF	0300	VII	_
Load or restart execution	LREX	03E0	Vil	_
Branch and load				
Workspace pointer	BLWP	0400	VI	**-
Branch	В	0440	VΙ	-
Return	RT	045B	VI	_
Execute	Х	0480	VI	-
Clear	CLR	04C0	VI	_
Negate	NEG	0500	VI	0—2,4
Invert	INV	0540	VΙ	0-2
Increment	INC	0580	VI	0-4
Increment by two	INCT	05C0	VI	0-4
Decrement	DEC	0600	VI	0-4
Decrement by two	DECT	0640	VI	0—4

INSTRUCTIONS AND PSEUDO-INSTRUCTIONS (CONTD)

B

N P L F

P

R

	Name	Мпетопіс	On-code	Format	Status Bits Affected
L			•	VI	
1	Branch and link	BŁ SWPB	0680	VI	_
1	Swap bytes		06C0 0700	Vi	-
1	Set to one	SETÓ ABS	0740	VI	_ 02,4
ļ	Absolute value		0800	V	0-2,4
ı	Shift right arithmetic	SRA SRL	0900	v	0-3
ı	Shift right logical	SH.A	0A00	v	04
1	Shift left arithmetic	SRC	0800	v	0-3
ļ	Shift right circular	JMP	1000	Ĭi.	
7	Unconditional jump	NOP	1000	ii	_
ı	No operation Jump if less than	JLT	1100	ii	_
		JLE	1200	ii	
:	Jump If low or equal	JEQ	1300	ii	Ξ
i	Jump if equal Jump if high or equal	JHE	1400	ü	_
ı	Jump if greater than	JGT	1500	ii	_
ı	Jump if not equal	JNE	1600	ii	_
1	Jump if no carry	JNC	1700	ä	_
	Jump on carry	JOC	1800	ii	_
ı	Jump if no overflow	JNO	1900	ii	_
ŝ	Jump if logical low	JL	1A00	ii	_
ì	Jump if logical high	JH JL	1B00	ii .	Ξ
1	Jump if odd parity	JOP	1000	ii	_
l	Set CRU bit to one	SBO	1000	ii	
ı	Set CRU bit to zero	SBZ	1E00	ii	Ξ
•	Test bit	TB	1F00	ii	2
ĺ	Compare ones	10	11 00	•••	2
•	corresponding	COC	2000	101	2
	Compare zeros	000		,,,,	-
ţ	corresponding	CZC	2400	111	2
i	Exclusive or	XOR	2800	iii	0-2
į	Extended operation	XOP	2C00	ix	6
	Load CRU	LDCR	3000	ίν	0-2,5
İ	Store CRU	STCR	3400	ίΫ	0-2,5
1	Multiply	MPY	3800	ΪX	
ł	Divide	DIV	3C00	iX	4
	Set zeros corresponding	SZC	4000	ï	02
:	Set zeros corresponding,				
1	byte	SZCB	5000	ı	0-2.5
ì	Subtract words	S	6000	j	0-4
	Subtract bytes	SB	7000	- 1	0-5
l	Compare words	C	8000	1	0-2
١.	Compare bytes	CB	9000	ì	0-2,5
Î	Add words	Ä	A000	i	0-4
İ	Add bytes	AB	8000	1	0-5
l	Move word	MOV	C000	į	0-2
	Move byte	MOVB	D000	- 1	02,5
	Set ones corresponding	SOC	E000	1	0-2
i	Set ones corresponding,				
Ĺ	byte	SOCE	F000	- 1	02,5

5

DIRECTIVES

Name	Mnemonic	: Syntax Definition
Absolute Origin Block Ending with	AORG	AORG < wd-exp>
Symbol	BES	BES < wd-exp>
Block Starting with Symbol	BSS	BSS < wd-exp>
Initialize Byte Common Segment	BYTE CEND	BYTE < exp>[, < exp>] CEND
Common Segment End	CSEG	CSEG
Copy File	COPY	COPY "< file name>"
Initialize Word	DATA	DATA <exp>[.exp>]</exp>
External Definition	DEF	DEF <symbol>[,<symbol>]</symbol></symbol>
Data Segment End	DEND	DEND
Dummy Origin Data Segment	DORG DSEG	DORG <exp> DSEG</exp>
Define Extended		Patricia de la companya della companya della companya de la companya de la companya della compan
Operation Program End	DXOP END	DXOP < symbol > , < term > END[< symbol >]
Define Assembly-Time		al fall FOU amon
Constant Word Boundary	EQU EVEN	<label> EQU <exp> EVEN</exp></label>
Program Identifier	IDT	IDT' <string>'</string>
List Source Force Load	LIST LOAD	LIST LOAD
		<symbol>[,<symbol>]</symbol></symbol>
Page Eject Program Segment End	PAGE PEND	PAGE PEND
Program Segment	PSEG	PSEG
External Reference	REF	REF <symbol>[,<symbol>]</symbol></symbol>
Relocatable Origin	RORG	RORG[<exp>]</exp>
Secondary External Reference	SREF	SREF
Initialize Text	TEXT	<symbol>,[<symbol>] TEXT [-]' < string > '</symbol></symbol>
Page Title	TITL	TITL' <string>"</string>
No Source List	UNL	UNL

COLORS

Color	Hex Code	Color	Hex Code
Transparent	Ō	Medium red	8
Black	1	Light red	9
Medium green	2	Dark yellow	A
Light green	3	Light yellow	₽
Dark blue	4	Dark green	С
Light blue	5	Magenta	D
Dark red	6	Gray	E
Cyan	7	White	F

	ASCII CHARACTER SE	т	ASCII	CHARACTER SET (C	CONTD)	ASCII	CHARACTER SET (C	ONTD)
Hex Value	Decimal Value	Character	Hex Value	Decimal Value	Character	Hex Value	Decimal Value	Character
00 01 02 03 04 05 06 07 08 09 08 09 00 00 00 00 00 00 00 00 00 00 00 00	0 1 2 3 4 4 5 6 7 8 9 10 11 12 11 14 11 15 11 14 15 16 17 18 19 20 1 22 22 24 25 6 27 22 9 30 31 32 33 34 36 36 36 36 36 36 36 36 36 36 36 36 36	NULH STXX TEOTO EACH EACH STXX TEOTO EACH EACH EACH EACH EACH EACH EACH EACH	38 37 38 38 38 38 38 38 38 38 38 39 40 41 42 44 44 44 45 46 47 48 48 48 49 48 49 48 48 48 48 48 48 48 48 48 48 48 48 48	54 55 56 57 58 59 60 61 62 63 64 66 67 68 69 71 72 73 74 75 76 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 100 100 100 100 100 100	6789:<=>?@ABCDEFGH-JKLMNOPQRSTUVWXYZ[\]< .abcdefgh-	6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 78 79 7A 7B 7D 7E 7D 7E 1nput/Output Error Co Code 0 Bad device 1 Device is v 2 Bad open a incorrect records in records in A Out of table 5 Attempt to occurs, the records in Device error parity and 7 File error s 7 File error s	107 108 109 110 1111 112 113 114 115 116 117 118 119 120 121 122 123 125 126 127 ERROR MESSAGES des Meaning mame. write protected, attribute such as inco- scord length, incorrect a relative record file. ration; i.e., an operative er or buffer space on tread past the end of if file is closed. Also gia a relative record file. or. Covers all hard devead medium errors, uch as program/data e opened in INPUT modeling O errors. erflow.	k m n n o p q r s t u v w X y z [i
35	63	5 g			10			11

<te

< ir

< 8:

] [] < b:

< rl

< dı < uj < re

< qı

	SPE	CIAL KEY	rs	
Name	TI-99/4	TI-99/4A	Hex Code	Action
<del character="">	SHIFT F	FCTN 1	03	Deletes a character in the Editor.
<ins character=""></ins>	SHIFTG	FCTN 2	04	Inserts a character in the Editor.
<delete line=""></delete>	SHIFTT	FCTN 3	07	Deletes a line from the screen.
<roll-up></roll-up>	SHIFT C	FCTN 4	02	Displays the next 24 lines of the file.
<next-window></next-window>	SHIFTW	FCTN 5	0 E	Moves the display to the next window.
<roll-down></roll-down>	SHIFTV	FCTN 6	0C	Displays the previous 24 lines of the file.
<tab></tab>	SHIFTA	FCTN 7	01	Moves the cursor to the next tab position.
<insert line=""> <esc></esc></insert>	SHIFT R SHIFT Z		06 0F	Inserts a line. Returns to the previously displayed screen. In the Editor, enters the command mode.
<esc></esc>	SHIFTX	FCTNX	0A	Used as the <escape> key in the Debugger.</escape>
£.		FCTN F	7B	Types the left brace
1		FCTN G	7D	Types the right brace).
I		FCTNR	58	Types the left bracket [.
1		FCTNT	5D	Types the right bracket].
<left-arrow> or <backspace></backspace></left-arrow>	SHIFTS	FCTNS	08	Moves the cursor to the left one character.
<right-arrow></right-arrow>	SHIFTD	FCTND	09	Moves the cursor to the right one character.
<down-arrow></down-arrow>	SHIFTX	FCTN X	0A	Moves the cursor down one line.
<up-arrow></up-arrow>	SHIFTE	FCTN E	0B	Moves the cursor up one line.
<return></return>	ENTER	ENTER	0D	Tells the computer to accept the information that
<quit></quit>	SHIFTQ	FCTN=	05	you type. Leaves the Editor/Assembler.

Command	Letter
Load Memory with ASCII	Α
Breakpoint Set/Clear	В
CRU Inspect/Change	C E F
Execute	Ē
Find Word or Byte	F
GROM Base Change	G
Inspect Screen Location	ı
Find Data Not Equal	K
Memory Inspect/Change	М
Move Block	N
Compare Memory Blocks	P
Quit Debugger	Q
Inspect or Change WP, PC, and SR	R
Execute in Step Mode	S
Trade Screen	Т
Toggle Offset to and from TI BASIC	U
VDP Base Change	V
Inspect or Change Registers	Ŵ
Change Bias	X, Y, or Z
Hexadecimal to Decimal Conversion	>
Decimal to Hexadecimal Conversion	
Hexadecimal Arithmetic	н

TI BASIC SUBPROGRAMS

CALL CHARPAT(character-number,string-variable[....])
CALL LINK("program-name"[.parameter-list])
CALL LOAD("object-filename"[."object-filename"....]) or (address,value[.value......])
CALL PEEK(address,variable-list[,"",...])
CALL PEEK(address,variable-list[,"",...])
CALL PEEK(address,variable-list[,"",...])
CALL POKEV(address,value-list[,"",...])

TI BASIC UTILITIES

Name	Use
ERR NUMASG NUMREF STRASG STRREF	Reports errors. Makes a numeric assignment. Gets a numeric parameter. Makes a string assignment. Gets a string parameter.

UTILITIES

Name	Use
DSRLNK	Links your program to Device Service Routines.
GPLLNK	Links your program to Graphics Programming Language routines.
KSCAN	Scans the keyboard.
LOADER	Links your program to the Loader to load TMS9900 tagged object code.
VMBR	Reads multiple bytes from VDP RAM.
VMBW	Writes multiple bytes to VDP RAM.
VSBR	Reads a single byte from VDP RAM.
VSBW	Writes a single byte to VDP RAM,
VWTR	Writes a single byte to a VDP Register.
XMLLNK	Links your program to the assembly language routines in the console ROM or in RAM.

OBJECT TAG SUMMARY

Tag	Vse	Fleid 1	Field 2
0	Program Identification Entry Point Definition	Program Length Absolute Address	Program ID
2	Entry Point Definition	Relocatable Address	
3	External References	Relocatable Address of Chain	Symbol
4	External References	Absolute Address of Chain	Symbol
5	External Definitions	Relocatable Address	Symbol
6	External Definitions	Absolute Address	Symbol
7	Checksum Indicator	Checksum	•
8	Checksum Ignore	Any Value	
9	Load Address	Absolute Value	
Α	Load Address	Relocatable Address	
В	Data	Absolute Value	
С	Data	Relocatable Address	
F	End of Record		

	INFD		

Name	Address	Data Contained
GPLWS	>83E0	GPL Interpreter Workspace.
GRMRA	>9802	GROM/GRAM read address.
GRMRD	> 9800	GROM/GRAM read data.
GRMWA	>9C02	GROM/GRAM write address.
GRMWD	>9000	GROM/GRAM write data.
PAD	>8300	The scratch pad used by TI BASIC,
		GPL, TI BASIC, and other programs.
		You may use some areas. See the
		Appendix for a detailed description of
		this area.
SCAN	> 000E	Entry address of the keyboard scan
		utility.
SOUND	> 8400	Sound chip.
SPCHRD	> 9000	Speech read.
SPCHWT	>9400	Speech write.
UTLTAB	> 2022	Utility variable table.
VOPRO	> 8800	VDP RAM read data.
VDPSTA	>8802	VDP RAM status.
VDPWA	>8C02	VDP RAM write address.
VDPWD	>8C00	VDP RAM write data.

Reference	Address	Name	Information
UTLTAB	> 2022	UTLTAB	Entry address.
UTLTAB+>2	> 2024	FSTHI	First free address in
			high memory.
UTLTAB + > 4	> 2026	LSTHI	Last free address in
			high memory.
UTLTAB $+ > 6$	> 2028	FSTLOW	First free address in
			low memory.
UTLTAB+>8	>202A	LSTLOW	Last free address in
			low memory.
UTLTAB + > A	> 202C	CHKSAV	Checksum.
UTLTAB+>>C	>202E	FLGPTR	Pointer to the flag in
			the PAB.
UTLTAB + > E	>2030	SVGPRT	GPL return address.
UTLTAB + > 10	>2032	SAVCRU	CRU address of the
		_	peripheral.
UTLTAB + > 12	> 2034	SAVENT	Entry address of the
			DSR or subprogram.
UTLTAB + > 14	>2036	SAVLEN	Device or
			subprogram name
			length.
UTLTAB + > 16	> 2038	SAVPAB	Pointer to the device
			or subprogram name
			in the PAB.
UTLTAB + > 18	>203A	SAVVER	Version number of
			the DSR.

VDP WRITE-ONLY REGISTERS

VDP Register 0	The default for Register 0 is > 00 for the Editor/Assembler, TI BASIC, and TI Extended
	BASIC.
	Bits 0—5 Reserved.
	Bit 6 Mode bit 3.
	Bit 7 External video enable/disable.
VDP Register 1	The default for Register 1 is > E0 in the
	Editor/Assembler, TI BAS&C, and TI Extended BASIC. Note: Before changing this Register,
	put a copy of the new value you want it to have
	at address > 83D4,
	Bit 0 4/16K selection.
	Bit 1 Screen display enable/disable.
	Bit 2 Interrupt enable/disable.
	Bit 3 Mode bit 1.
	Bit 4 Mode bit 2.
	Bit 5 Reserved.
	Bit 6 Sprite size selection.
	Blt 7 Sprite magnification selection.
VDP Register 2	The default for Register 2 is > 00 in the
	Editor/Assembler, Ti BASIC, and Ti Extended
	BASIC.
	When multiplied by > 400, defines the base address of the Screen Image Table.
VDP Register 3	The default for Register 3 is > 0E in the
Apr (Jedisto) 2	Editor/Assembler, > 0C in TI BASIC, and > 20
	in TI Extended BASIC.
	When multiplied by > 40, defines the base
	address of the Color Table.
VDP Register 4	The default for Register 4 is >01 in the
	Editor/Assembler and >00 in Ti BASIC and Ti
	Extended BASIC.
	When multiplied by > 800, defines the base
MDD m. 1	address of the Pattern Descriptor Table.
VDP Register 5	The default for Register 5 is > 06 in the
	Editor/Assembler, TI BASIC, and TI Extended BASIC.
	When multiplied by > 80, defines the base
Minn on allers in	address of the Sprite Attribute List.
VDP Register 6	The default for Register 6 is > 00 in the Editor/Assembler, TI BASIC, and TI Extended
	BASIC.
	When multiplied by > 800, defines the base
	address of the Sprite Descriptor Table.
VDP Register 7	The default for Register 7 is > F5 in the
2	Editor/Assembler and > 07 In TI BASIC and TI
	Extended BASIC.
	Bits 0-3 Trie color code of the foreground
	color in text mode.
	Bits 4—7 The color code for the background
	color in all modes.

ng