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.

Printed in U.S.A. 1035988-1

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.

Printed in U.S.A. 1035988-1

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.

Printed in U.S.A.

by www.ti99iuc.it

1035988-1

ADDRESSING SUMMARY

Mnemonic	First Operand	Second Operand	Op-code	Format
A	G	G*	A000	I
AB	G	G*	B000	I
ABS	G	–	0740	VI
AI	WR*	I	0220	VIII
ANDI	WR*	I	0240	VIII
B	G	–	0440	VI
BL	G	–	0680	VI
BLWP	G	–	0400	VI
C	G	G	8000	I
CB	G	G	9000	I
CI	WR	I	0280	VIII
CKOF	–	–	03C0	VII
CKON	–	–	03A0	VII
CLR	G	–	04C0	VI
COC	G	WR	2000	III
CZC	G	WR	2400	III
DEC	G	–	0600	VI
DECT	G	–	0640	VI
DIV	G	WR*	3C00	IX
IDLE	–	–	0340	VII
INC	G	–	0580	VI
INCT	G	–	05C0	VI
INV	G	–	0540	VI
JEQ	PC	–	1300	II
JGT	PC	–	1500	II
JH	PC	–	1B00	II
JHE	PC	–	1400	II
JL	PC	–	1A00	II
JLE	PC	–	1200	II
JLT	PC	–	1100	II
JMP	PC	–	1000	II
JNC	PC	–	1700	II
JNE	PC	–	1600	II
JNO	PC	–	1900	II
JOC	PC	–	1800	II
JOP	PC	–	1C00	II
LDCR	G	Note 1	3000	IV
LI	WR*	I	0200	VIII
LIMI	I	–	0300	VIII
LREX	–	–	03E0	VII
LWPI	I	–	02E0	VIII
MOV	G	G*	C000	I
MOVB	G	G*	D000	I
MPY	G	WR*	3800	IX
NEG	G	–	0500	VI
ORI	WR*	I	0260	VIII
RSET	–	–	0360	VII
RTWP	–	–	0380	VII

ADDRESSING SUMMARY (CONTD)

Mnemonic	First Operand	Second Operand	Op-code	Format
S	G	G*	6000	I
SB	G	G*	7000	I
SBO	CRU	–	1D00	II
SBZ	CRU	–	1E00	II
SETO	G	–	0700	VI
SLA	WR*	Note 2	0A00	V
SOC	G	G*	E000	I
SOCB	G	G*	F000	I
SRA	WR*	Note 2	0800	V
SRC	WR*	Note 2	0B00	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	I
SZCB	G	G*	5000	I
TB	CRU	–	1F00	II
X	G	–	0480	VI
XOP	G	Note 3	2C00	IX
XOR	G	WR*	2800	III

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.

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

WR — Workspace Register address
PC — Program counter relative address
CRU — CRU bit address
I — Immediate value
* — The address into which the result is placed when two operands are required

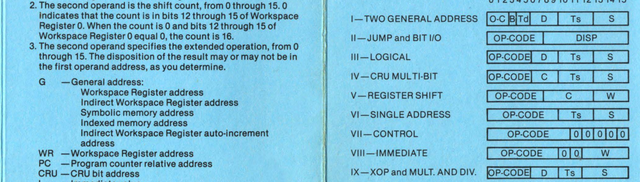
ADDRESSING MODES

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

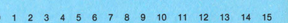
Notes:

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. If it is non-zero, it is an indexed addressing mode, and the non-zero value is the number of the index register. Therefore, Workspace Register 0 cannot be used for indexing.

INSTRUCTION FORMATS



STATUS REGISTER



Name	Bit Number	Meaning
L>	0	Logical greater than
A>	1	Arithmetic greater than
EQ	2	Equal
C	3	Carry
OV	4	Overflow
OP	5	Odd parity
X	6	Extended operation
–	7—11	Reserved
INT. MASK	12—15	Interrupt mask

INSTRUCTIONS AND PSEUDO-INSTRUCTIONS

Name	Mnemonic	Op-code	Format	Status Bits Affected
Load immediate	LI	0200	VIII	0—2
Add immediate	AI	0220	VIII	0—4
And immediate	ANDI	0240	VIII	0—2
Or immediate	ORI	0260	VIII	0—2
Compare immediate	CI	0280	VIII	0—2
Store Workspace pointer	STWP	02A0	VIII	–
Store Status	STST	02C0	VIII	–
Load Workspace pointer immediate	LWPI	02E0	VIII	–
Load interrupt mask immediate	LIMI	0300	VIII	12—15
Idle	IDLE	0340	VII	–
Reset	RSET	0360	VII	–
Return with Workspace pointer	RTWP	0380	VII	0—15
Clock on	CKON	03A0	VII	–
Clock off	CKOF	03C0	VII	–
Load or restart execution	LREX	03E0	VII	–
Branch and load Workspace pointer	BLWP	0400	VI	–
Branch	B	0440	VI	–
Return	RT	045B	VI	–
Execute	X	0480	VI	–
Clear	CLR	04C0	VI	–
Negate	NEG	0500	VI	0—2,4
Invert	INV	0540	VI	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)

Name	Mnemonic	Op-code	Format	Status Bits Affected
Branch and link	BL	0680	VI	–
Swap bytes	SWPB	06C0	VI	–
Set to one	SETO	0700	VI	–
Absolute value	ABS	0740	VI	0—2,4
Shift right arithmetic	SRA	0800	V	0—3
Shift right logical	SRL	0900	V	0—3
Shift left arithmetic	SLA	0A00	V	0—4
Shift right circular	SRC	0B00	V	0—3
Unconditional jump	JMP	1000	II	–
No operation	NOP	1000	II	–
Jump if less than	JLT	1100	II	–
Jump if low or equal	JLE	1200	II	–
Jump if equal	JEQ	1300	II	–
Jump if high or equal	JHE	1400	II	–
Jump if greater than	JGT	1500	II	–
Jump if not equal	JNE	1600	II	–
Jump if no carry	JNC	1700	II	–
Jump on carry	JOC	1800	II	–
Jump if no overflow	JNO	1900	II	–
Jump if logical low	JL	1A00	II	–
Jump if logical high	JH	1B00	II	–
Jump if odd parity	JOP	1C00	II	–
Set CRU bit to one	SBO	1D00	II	–
Set CRU bit to zero	SBZ	1E00	II	–
Test bit	TB	1F00	II	2
Compare ones corresponding	COC	2000	III	2
Compare zeros corresponding	CZC	2400	III	2
Exclusive or	XOR	2800	III	0—2
Extended operation	XOP	2C00	IX	6
Load CRU	LDCR	3000	IV	0—2,5
Store CRU	STCR	3400	IV	0—2,5
Multiply	MPY	3800	IX	–
Divide	DIV	3C00	IX	4
Set zeros corresponding	SZC	4000	I	0—2
Set zeros corresponding, byte	SZCB	5000	I	0—2,5
Subtract words	S	6000	I	0—4
Subtract bytes	SB	7000	I	0—5
Compare words	C	8000	I	0—2
Compare bytes	CB	9000	I	0—2,5
Add words	A	A000	I	0—4
Add bytes	AB	B000	I	0—5
Move word	MOV	C000	I	0—2
Move byte	MOVB	D000	I	0—2,5
Set ones corresponding	SOC	E000	I	0—2
Set ones corresponding, byte	SOCB	F000	I	0—2,5

DIRECTIVES

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

COLORS

Color	Hex Code	Color	Hex Code
Transparent	0	Medium red	8
Black	1	Light red	9
Medium green	2	Dark yellow	A
Light green	3	Light yellow	B
Dark blue	4	Dark green	C
Light blue	5	Magenta	D
Dark red	6	Gray	E
Cyan	7	White	F

ASCII CHARACTER SET

Hex Value	Decimal Value	Character
00	0	NUL
01	1	SOH
02	2	STX
03	3	ETX
04	4	EOT
05	5	ENQ
06	6	ACK
07	7	BEL
08	8	BS
09	9	HT
0A	10	LF
0B	11	VT
0C	12	FF
0D	13	CR
0E	14	SO
0F	15	SI
10	16	DLE
11	17	DC1
12	18	DC2
13	19	DC3
14	20	DC4
15	21	NAK
16	22	SYN
17	23	ETB
18	24	CAN
19	25	EM
1A	26	SUB
1B	27	ESC
1C	28	FS
1D	29	GS
1E	30	RS
1F	31	US
20	32	Space
21	33	!
22	34	"
23	35	#
24	36	$
25	37	%
26	38	&
27	39	'
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	,
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5

ASCII CHARACTER SET (CONTD)

Hex Value	Decimal Value	Character
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_
60	96	`
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j

ASCII CHARACTER SET (CONTD)

Hex Value	Decimal Value	Character
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7D	125	}
7E	126	~
7F	127	DEL

ERROR MESSAGES

Input/Output Error Codes

Code	Meaning
0	Bad device name.
1	Device is write protected.
2	Bad open attribute such as incorrect file type, incorrect record length, incorrect I/O mode, or no records in a relative record file.
3	Illegal operation; i.e., an operation not supported on the peripheral or a conflict with the OPEN attributes.
4	Out of table or buffer space on the device.
5	Attempt to read past the end of file. When this error occurs, the file is closed. Also given for non-existing records in a relative record file.
6	Device error. Covers all hard device errors such as parity and bad medium errors.
7	File error such as program/data file mismatch, non-existing file opened in INPUT mode, etc.

Loader Error Codes

Code	Meaning
0—7	Standard I/O errors.
8	Memory overflow.
9	Not used.
A	Illegal tag.
B	Checksum error.
C	Duplicate definition.
D	Unresolved reference.

SPECIAL KEYS

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>	SHIFT G	FCTN 2	04	Inserts a character in the Editor.
<delete line>	SHIFT T	FCTN 3	07	Deletes a line from the screen.
<roll-up>	SHIFT C	FCTN 4	02	Displays the next 24 lines of the file.
<next-window>	SHIFT W	FCTN 5	0E	Moves the display to the next window.
<roll-down>	SHIFT V	FCTN 6	0C	Displays the previous 24 lines of the file.
<tab>	SHIFT A	FCTN 7	01	Moves the cursor to the next tab position.
<insert line>	SHIFT R	FCTN 8	06	Inserts a line.
<esc>	SHIFT Z	FCTN 9	0F	Returns to the previously displayed screen. In the Editor, enters the command mode.
<esc>	SHIFT X	FCTN X	0A	Used as the <escape> key in the Debugger.
|		FCTN F	7B	Types the left brace {.
~		FCTN G	7D	Types the right brace }.
[		FCTN R	5B	Types the left bracket [.
]		FCTN T	5D	Types the right bracket].
<left-arrow> or <backspace>	SHIFT S	FCTN S	08	Moves the cursor to the left one character.
<right-arrow>	SHIFT D	FCTN D	09	Moves the cursor to the right one character.
<down-arrow>	SHIFT X	FCTN X	0A	Moves the cursor down one line.
<up-arrow>	SHIFT E	FCTN E	0B	Moves the cursor up one line.
<return>	ENTER	ENTER	0D	Tells the computer to accept the information that you type.
<quit>	SHIFT Q	FCTN =	05	Leaves the Editor/Assembler.

DEBUGGER COMMANDS

Command	Letter
Load Memory with ASCII	A
Breakpoint Set/Clear	B
CRU Inspect/Change	C
Execute	E
Find Word or Byte	F
GROM Base Change	G
Inspect Screen Location	I
Find Data Not Equal	K
Memory Inspect/Change	M
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	T
Toggle Offset to and from TI BASIC	U
VDP Base Change	V
Inspect or Change Registers	W
Change Bias	X, Y, or Z
Hexadecimal to Decimal Conversion	>
Decimal to Hexadecimal Conversion	.
Hexadecimal Arithmetic	H

TI BASIC SUBPROGRAMS

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

TI BASIC UTILITIES

Name	Use
ERR	Reports errors.
NUMASG	Makes a numeric assignment.
NUMREF	Gets a numeric parameter.
STRASG	Makes a string assignment.
STRREF	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	Use	Field 1	Field 2
0	Program Identification	Program Length	Program ID
1	Entry Point Definition	Absolute Address	
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	
A	Load Address	Relocatable Address	
B	Data	Absolute Value	
C	Data	Relocatable Address	
F	End of Record		

PREDEFINED ADDRESSES

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	>9C00	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.
VDPRD	>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 BASIC, 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.
Bit 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 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 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 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 Editor/Assembler and >07 in TI BASIC and TI Extended BASIC.
Bits 0—3 The color code of the foreground color in text mode.
Bits 4—7 The color code for the background color in all modes.

- Revisited by TI99 Italian User Club (info@ti99iuc.it) in January, 2014