



## Console ROMs

The TI-99/4A console contains two 4K ROMs that correspond to addresses >0000 to >1FFF. One of them holds the even-address bytes, the other the odd-address bytes. Together they allow for full 16-bit access from the TMS9900 CPU.

[Pinout](#)  
[Content](#)

### Pinout

TMS 4732 / 2352			
A7	1	o	T
A6	2	M	23
A5	3	S	22
A4	4	21	A9
A3	5	7	20
A2	6	3	19
A1	7	2	18
A0	8	/	17
D0	9	2	16
D1	10	3	15
D2	11	5	14
Vss	12	2	13
			D3
+			

[Power supply](#)  
Vcc +5V  
Vss Ground

### CPU interface

**A0-A11** Address bus. Connected to the address bus, lines A3 to A14 in the console.

**D0-D7** Data bus. In the TI-99/4A, one chip is connected to lines D0 to D7, the second chip to lines D8-D15.

**CS1\*** Chip select. In the TI-99/4A, active (low) when memory in the range >0000-1FFF is accessed. The selection is performed by a 74LS138 decoder, with A0, A1 and A2 as inputs, and MEMEN\* as enabling input G2A\*. Output Y0\* is connected to CS1\* of both ROM chips via a 1K resistor.

**CS2\*** Chip select. Ditto. Hardwired to ground in the TI-99/4A..

## Contents

The TMS9900 microprocessor uses the beginning of its address space for special purposes. The console ROMs contain the adequate values for (some of) these features.

Address >0000: vector for power-up (WS = >83E0, PC = >0024)

Addresses >0000 to >003F contain vectors for interrupts levels 0 to 15. In the TI-99/4A, all interrupts are hard-wired as level 1, so only this vector is relevant. Nevertheless, a vector for level 2 is also present (and level 0 is the same as power-up).

Address >0004: Interrupt vector 1 (WS = >83C0, PC = >0900)

Address >0008: Interrupt vector 2 (WS = >83C0, PC = >0A92)

Addresses >0040 to >007F contain vectors for the 16 XOP opcodes. Only the first two are implemented for sure, although some consoles carry a third one. The remaining addresses contain assembly language, which may or may not result in meaningful vectors (see discussion of the [XOP instruction](#) in my TMS9900 page).

Address >0040: XOP0 (WS = >280A, PC = >0C1C) Calls a never-released extended GPL card (i.e. crashes)

Address >0044: XOP1 (WS = >FFD8, PC = >FFF8) User defined operation.

Address >0048: XOP2 (WS = >83A0, PC = >8300). User-defined operation. Not on all consoles!.

After these special addresses, the console ROMs mainly contain:

The [GPL interpreter](#),  
The [Keyboard scanning routine](#),  
The [Interrupt service routine](#).

Routines that scan the GROM and card ROM headers for subprograms, DSR, etc.

Some [mathematical](#) subroutines (many others are in GROM),

Low-level routines for [cassette](#) operations (DSRs are in GROM).

Here is a more detailed description of the ROM contents. For a commented listing, see Heiner Martin's book: TI-99/4A Intern ([pdf 560k](#)).

Address	Contents
>0000	Power up vector (>83E0, >0024).
>0004	Level 1 interrupt vector (>83C0, >0900)
>0008	Level 2 interrupt vector (>83C0, >0A92. Not used)
>000C	CPU clock speed (>28 or >30) and header mark (>AA).
>000E	Address of the keyboard scanning routine (>02B2).
>0016	Alternate entry point into the GPL interpreter (with an opcode in R9).
>001C	Alternate entry point into the GPL interpreter (with no interrupts).
>0020	Address of the routine that tests the <Clear> key (>04B2).
>0024	Entry point into the GPL interpreter at power-up time.
>0036	Return from the (unreleased) extended-GPL board.
>0040	Vectors for XOP0 (>280A, >0C1C).
>0044	Vectors for XOP1 (>FFD8, >FFF8).
>0048	Vectors for XOP2 (>83A0, >8300). Absent on some consoles.
>004E	Interpretation of GPL opcode SWGR.
>0060	Alternate entry point into the GPL interpreter (with an address in R6).
>0070	Main entry point into the GPL interpreter.
>0086	Dispatch of the standard GPL opcodes (>00 to >FF).
>00CC	Interpretation of GPL opcodes CGE, CH, CHE, CGT, CLOG, CZ, CEQ, CARRY, OVF, H, GT, B, BS, BR, ABS, NEG, CLR, INV, FETCH, CASE, PUSH, DECT, INCT, INC, SUB, DEC, ADD, AND, OR, XOR, ST, EX, SRA, SLL, SRL, SRC, MPY, and DIV (see <a href="#">table</a> below).
>0270	Dispatch of the special GPL opcodes (>00 to >1F).
>027A	Interpretation of GPL opcodes RAND, BACK, SCAN.
>02B2	Keyboard scanning routine.
>04B2	Test <Clear> key (Fctn-4).
>04DE	Subinterpreter for GPL opcodes FMT, VCHA, HCHA, VTEX, HTEX, IROW, ICOL, RPTB, FEND, LOOP, ROW, COL, SCRO, HTEX (see <a href="#">table</a> below).
>05A2	Interpretation of GPL opcodes ALL, I/O, XML, MOVE, COINC, RTGR, RTN, RTNC, CALL, PUSH, and various subroutines used by the interpreter (see <a href="#">table</a> below).
>0900	Interrupt service routine. >0918: Peripheral cards interrupts. >094A: VDP interrupts.
>0AC0	SROM (XML >19). Search card ROMs for subprograms, DSRs, or power-up routines (= DSRLNK).
>0B24	SGROM (XML >1A). Search GROMs for subprograms, DSRs, or power-up routines.

>0C0C	Common entry point for extended-GPL opcodes.
>0C1C	Entry points to the never-released extended-GPL board (XOP 0).
>0C36	Jump tables for the GPL interpreter.
>0CFA	XML master table (i.e. address of other tables).
>0D1A	FLTAB: jump table for XML >0x.
>0D3A	FCOMP (XML >0A). Floating point comparison.
>0D46	SCOMP (XML >0F). Comparison in stack.
>0D74	SSUB (XML >0C). Subtraction in stack.
>0D7C	FSUB (XML >07). Floating point subtraction.
>0D80	FADD (XML >06). Floating point addition.
>0D84	SADD (XML >0B). Addition in stack.
>0E88	FMUL (XML >08). Floating point multiplication.
>0E8C	SMUL (XML >0D). Multiplication in stack.
>0F54	ROUND1 (XML >01). Rounds real number to 14 digits.
>0F4A	STST (XML >03). Store status after comparing a real number to zero.
>0FB2	ROUND (XML >02). Round a real number with specified number of bytes in >8354.
>0FC2	OVEXP (XML >04). React to overflow/underflow (according to sign of >8376).
>0FCC	OV (XML >05). React to overflow.
>0FF4	FDIV (XML >09). Floating point division.
>0FF8	SDIV (XML >0E). Division in stack.
>11A2	CSNGR(XML >11) CSN from a string in GROM or VDP.
>11AE	CSN (XML >10). Convert a string to a real number (and round it if needed).
>12A0	Jump table for XML >1x.
>12B8	CFI (XML >12). Convert a floating point number to an integer.
>1346	Cassette write (GPL opcode I/O 4).
>1404	Interrupt service routine used during cassette operations (meant for TMS9901 timer interrupts only).
>1426	Cassette verify (GPL opcode I/O 6).
>142E	Cassette read (GPL opcode I/O 5).
>15D6	SCHSYM (XML >16). Search a Basic variable by name .
>163C	Entry points for XML >17 (>163C), XML >15 (>1642), XML >13 (>1648) and XML >14 (>164E).
>1670	SMB (XML >14). Get the value of a Basic variable.
>176A	SYM (XML >13). Search the symbol table for a variable named in a Basic statement.
>1788	ASSIGNV (XML >15). Assigns value to a Basic variable.
>1868	PGMCH (XML >1B). Get next Basic token.
>18C8	Interpretation of GPL opcode PARSE.
>1920	Interpretation of GPL opcode CONT.
>1968	Interpretation of GPL opcode EXEC.
>19E6	EXEC entry points for DEF, DIM, DATA, REM, OPTION.
>19F0	Interpretation of GPL opcode RTNB.
>1A2C	EXEC entry points for "error 0", END, STOP, unquoted string (from PARSE), GO, ON, GOSUB, GOTO, RETURN, IF, LET, NEXT.
>1C9C	Jump <a href="#">table</a> for EXEC.
>1CE2	Jump <a href="#">table</a> for PARSE.
>1DE2	Jump <a href="#">table</a> for CONT.
>1D3E	CONT entry points for: () = + - * / ^
>1E4A	Various subroutines for use by the Basic interpreter.
>1EAA	VPUSHG (XML >17). Pushes data (symbol description) on Basic value stack. Alternate entry at >1E9C, returns with CONT.
>1F2E	VPOP (XML >18). Pops data (symbol description) from the Basic value stack
>1F7E	Various subroutines used by the Basic interpreter.
>1FFC	ROM checksum (>2A61, >A38A).

Detailed table of GPL interpreter routines

Address	Opcode	Address	Opcode	Address	Opcode
>0024	EXIT	>019A	XOR	>05C8	I/O
>004E	SWGR	>019E	ST	>05D6	I/O sound
>00CC	CGTE	>01A2	EX	>05E8	I/O cru in
>00D6	CH	>01B0	SRA	>05EA	I/O cru out
>00DA	CHE	>01B4	SLL	>0608	XML
>00DE	CGT	>01B8	SRL	>061E	MOVE
>00E2	CLOG	>01C2	SRC	>06D2	COIN
>00EA	CZ	>01CE	MPY	>07AA	Arguments decoding
>00EC	CEQ	>01EA	DIV	>082C	RTGR
>00F4	CAR, OVF, HIGH, GT	>017A	RND	>0838	RTN
>0104	B	>029E	BACK	>083E	RTNC
>010E	BS	>02AE	SCAN	>0842	Pop address from substack
>011A	BR	>04DE	FMT Subinterpreter	>085A	CALL
>0136	ABS	>0502	Fmt VCHA	>0864	Push address on stack
>013A	NEG	>0504	Fmt HCHA	>0880	Set screen ptr to write
>013E	CLR	>0508	Fmt VTEX	>0884	Set screen ptr to read
>0140	INV	>050A	Fmt HTEX	>08AA	Set address in multicolor mode
>0144	FETC	>0532	Fmt IROW	>0C0C	Extended-GPL opcodes
>0162	CASE	>0534	Fmt ICOL	>0C1A	Extended-GPL opcode >1F
>016E	PUSH	>053A	Fmt RPTB	>1346	I/O cassette write
>0182	DECT	>0550	Fmt LOOP, FEND	>1426	I/O cassette verify
>0184	INCT	>0584	Fmt ROW, COL	>142E	I/O cassette read
>0186	INC, SUB	>058E	Fmt SCRO	>18C8	PARS
>0188	DEC, ADD	>0596	Fmt HSTR	>1920	CONT
>0190	AND	>05A2	ALL	>1968	EXEC
>0196	OR			>19F0	RTB

Revision 1. 3/2/00 Preliminary.  
Revision 2. 3/4/00 OK to release.