TI-99/4a TMS9900 Cheat Sheet - V 1.1 - 2024-01-27				Address-Mode / Cycles							Status Flags							
	OpCode	Description	Fmt	R	*R	*R+	@L	@L(R)	Im	Rel	High	Gt	Equ	Carry	Ovf	Par	Xop	Int
Arithmetic	A s,d	Add s + d → d	1	14	+4	+8	+8	+8			x	x	x	x	x			
	AB s,d	Add byte s + d → d (MSB)	1	14	+4	+6	+8	+8			x	x	x	x	x	x		
	ABS d	Absolute value of signed word d	6	12/14	+4	+8	+8	+8			x	x	x	x	x			
	AI reg,val	Add value to register R + V → R	8	14					x		x	x	x	x	x			
	C s,d	Compare source and destination	1	14	+4	+8	+8	+8			x	x	x					
	CB s,d	Compare byte source and destination	1	14	+4	+6	+8	+8			x	x	x			x		
	CI reg,val	Compare register with value	8	14					x		x	x	x					
	DEC d	Decrement destination d - 1 → d	6	10	+4	+8	+8	+8			x	x	x	x	x			
	DECT d	Decrement destination d - 2 → d	6	10	+4	+8	+8	+8			x	x	x	x	x			
	DIV s,rn	(rn,rm) / s → rn; Remainder → rm	9	var*	+4	+8	+8	+8							x			
	INC d	Increment destination d + 1 → d	6	10	+4	+8	+8	+8			x	x	x	x	x			
	INCT d	Increment destination d + 2 → d	6	10	+4	+8	+8	+8			x	x	x	x	x			
	LI reg,val	Load reg with val immediately	8	12					x		x	x	x					
	MOV s,d	Move source to destination	1	14	+4	+8	+8	+8			x	x	x					
	MOVB s,d	Move byte source to destination	1	14	+4	+6	+8	+8			x	x	x			x		
	MPY s,rn	Multiply s * rn → (rn,rm)	9	52	+4	+8	+8	+8										
	NEG d	Negates destination	6	12	+4	+8	+8	+8			x	x	x	x	x			
	S s,d	Subtract d - s → d	1	14	+4	+8	+8	+8			x	x	x	x	x			
	SB s,d	Subtract byte d - s → d	1	14	+4	+6	+8	+8			x	x	x	x	x	x		
Bitwise Logic	ANDI reg,val	bitwise AND	8	14					x		x	x	x					
	CLR d	set zero >0000 → d	6	10	+4	+8	+8	+8										
	COC s,d	Check 1 corr. ∃ 1 coc 0 → Equ=0	3	14	+4	+8	+8	+8					x					
	CZC s,d	Check 0 corr. ∃ 1 czc 1 → Equ=0	3	14	+4	+8	+8	+8					x					
	INV d	inverts d bitwise	6	10	+4	+8	+8	+8			x	x	x					
	ORI reg,val	bitwise OR	8	14					x		x	x	x					
	SETO d	set one >FFFF → d	6	10	+4	+8	+8	+8										
	SOC s,d	Set 1 corresponding to s (log. OR)	1	14	+4	+8	+8	+8			x	x	x					
	SOCB s,d	Set 1 corresponding to s MSB	1	14	+4	+6	+8	+8			x	x	x			x		
	SZC s,d	Set 0 corr. to s (log. Not(A) AND B)	1	14	+4	+8	+8	+8			x	x	x					
	SZCB s,d	Set 0 corresponding to s MSB	1	14	+4	+6	+8	+8			x	x	x			x		
	XOR s,reg	eXclusive OR (not immediate!)	3	14	+4	+8	+8	+8			x	x	x					
Branching	B d	Branch to address s: d → PC	6	8	+4	+8	+8	+8										
	BL d	Branch and link: PC → R11; d → PC	6	12	+4	+8	+8	+8				*Jxx query these bits, no changes*						
	BLWP d	Branch-and-Load-Workspace-Pointer*	6	26	+4	+8	+8	+8										
	JEQ offset	Jump if Equal	2							8;10			1					
	JGT offset	Jump if Greater Than	2							8;10		1						
	JH offset	Jump if Higher (unsigned)	2							8;10	1		0					
	JHE offset	Jump if Higher or Equal (unsigned)	2							8;10	1		1					
	JL offset	Jump if Lower (unsigned)	2							8;10	0		0					
	JLE offset	Jump if Lower or Equal	2							8;10	0		1					
	JLT offset	Jump if Less Than	2							8;10		0	0					
	JMP offset	Jump relative unconditionally	2							8;10								
	JNC offset	Jump if No Carry	2							8;10				0				
	JNE offset	Jump if Not Equal	2							8;10			0					
	JNO offset	Jump if No Overflow	2							8;10					0			
	JOC offset	Jump on Carry	2							8;10				1				
	JOP offset	Jump on Odd Parity	2							8;10						1		
	RTWP	Return-with-Worskpace-Pointer *	7	14														
	XOP s,xop#	Extended operations (implicit BLWP)*	9	36	+4	+8	+8	+8									x	
Shift	SLA reg,cnt	Shift left arithmetic, for cnt=0 see docs	5	var*							x	x	x	x	x			
	SRA reg,cnt	Shift right arithmetic (signed)	5	var*							x	x	x	x				
	SRC reg,cnt	Shift right arithmetic circular	5	var*							x	x	x	x				
	SRL reg,cnt	Shift right logical (unsigned)	5	var*							x	x	x	x				
CRU	LDCR reg,cnt	Load cnt bits of reg into *R12 (CRU)	4	var*							x	x	x			x		
	SBO cnt	Set cnt bits 0 in *R12	2						12									
	SBZ cnt	Set cnt bits 1 in *R12	2						12									
	TB cnt	Test bit cnt in *R12	2						12				x					
	STCR reg,cnt	Store cnt bits of reg into *R12 (CRU)	4	42-60							x	x	x			x		
Misc	LIMI cnt	Load int. level:0,1 disable; 2-15 enable	8						16									x
	LWPI adr	Load adr into WP: adr → WP	8						10									
	STST reg	Store Status-Flags in reg: ST → reg	8	8														
	STWP reg	Store WP in reg: WP → reg	8	8														
	SWPB d	Swap MSB ↔ LSB in d	6	10	+4	+8	+8	+8										
	X d	Execute OpCode at d	6	8 + OpCode at d							X does not change flags, but OpCode at d may							

* too complex for this overview, see details in documentation

Unsupported TMS9900 OpCodes on the TI-99/4a: CKON, CKOF, LREX, RSET, IDLE

Assembler				
	DATA word [, ..]	Inserts one or more 16 bit values at current position	DEF label [, ..]	Exposes a label 1-6 characters to the linker for external calls
	BYTE byte [, ..]	Inserts one or more 8 bit values at current position	REF label [, ...]	Imports a label 1-6 characters from another source file
	TEXT 'text'	Inserts text without length or terminator	AORG address	Instruct the loader to continue at address
	BSS val	Reserves val bytes, skipping val bytes in assembling	RORG [offset]	Return to variable loading or change current pos. by offset
	BES val	Like BSS, but a label gets the end address of the block	END [label]	Ends assembling, label specifies the start address to the linker
	EVEN	Inserts >00 if on an odd address to get even	COPY "filename"	Include filename in the assembler code
	RT	Return ... alias for B *R11		

Addressing modes (TS,TD)			Variables and Constants used		Status registers in TMS9900	
R	Register Addressing	00	s,d	any of R,*R,*R+,@L,@L(R)	High	Logically higher than (unsigned)
*R	Register Indirect	01	reg	any register	Gt	Greater than (signed)
*R+	Reg. Indirect Autoincrement	11	rn,rm	two contiguous registers	Equ	Equal value or last operation resulted in zero
@Label	Symbolic Direct	10	val	immediate value 16 bit	Carry	17th bit if result is bigger that >FFFF
@Label(R)	Indexed Addressing	10	offset	jump -128 / +127 Words	Ovf	Overflow in a signed operation
Im	Immediate	-	cnt	value 0 to 15 / 16 (xxCR)	Par	Checks if odd or even number of 1 used
Rel	PC / CRU Relative	-	adr	immediate 16 bit address	Xop	Set during XOP instruction

Instruction Format			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	I	Arithmetic	OP Code			B	TD		D				TS		S			
2	II	Jump	OP Code								Signed Displacement of -127 / +128 Words							
3	III	Logical	OP Code						D				TS		S			
4	IV	CRU	OP Code						XFR or Count				TS		S			
5	V	Shift	OP Code								XFR or Count				WS Register Nr.			
6	VI	Program	OP Code												S			
7	VII	Control	OP Code												not used (should be 0)			
8	VIII	Immediate	OP Code											n.u. 0	WS Register Nr.			
			Immediate Value 16bit (Word)															
9	IX	MPY, DIV, XOP	OP Code						D				TS		S			

Hexadecimal Codes			Colors	ASC	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0	0000	0	Transparent	2x	32	33 !	34 "	35 #	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
1	0001	1	Black	3x	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
2	0010	2	Med. green	4x	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
3	0011	3	Light green	5x	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
4	0100	4	Dark blue	6x	96 ´	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
5	0101	5	Light blue	7x	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 ←
6	0110	6	Dark red	Subroutines (with Extended BASIC)									Memory Addresses							
7	0111	7	Cyan	NUMREF	EQU >200C	Gets a numeric parameter							VDPRD	EQU >8800	VDP read data					
8	1000	8	Medium red	NUMASG	EQU >2008	Makes a num.assignment							VDPSTA	EQU >8802	VDP Status					
9	1001	9	Light red	STRREF	EQU >2014	Gets a string parameter							VDPWD	EQU >8C00	VDP write data					
10	1010	A	Dark yellow	STRASG	EQU >2010	Makes a string assignment							VDPWA	EQU >8C02	VDP set read/write addr.					
11	1011	B	Light yellow	XMLLNK	EQU >2018	Assembly rout.in console							RAND16	EQU >83C0	16-bit random number					
12	1100	C	Dark green	VSBW	EQU >2020	Write 1 char to VRAM							RAND8	EQU >83C1	8-bit random number					
13	1101	D	Magenta	VMBW	EQU >2024	Write multi char to VRAM							FAC	EQU >834A	Floating Point Accumulat.					
14	1110	E	Gray	VWTR	EQU >2030	Write 1 byte to VDP Reg.							FREE	EQU >8370	Highest free VDP mem.					
15	1111	F	White	?????	EQU >006A	Return to XB GPL clear							MODE	EQU >8375	Keyboard scan mode					
													KEY	EQU >8375	Key-Code (>FF = none)					

VDP Memory layout	TI BASIC	TI XB	XB256 Sc2	E/A	MiniMem
Screen Image	>0000 -02FF	>0000 -02FF	>0C00 -0FFF	>0000 -02FF	>0000 -02FF
Char Pattern	>0000 -07FF	>0000 -07FF	>1000 -17FF	>0800 -0FFF	>0800 -0FFF
Color Table	>0300 -031F	>0800 -081F	>1800 -181F	>0380 -039F	>0380 -039F
Sprite Attributes	>0300 -0301	>0300 -036F	SoundTable:	>0300 -037F	>0300 -037F
Sprite Patterns	unused	>0000 -07FF	>0980 -0BFF	>0000 -07FF	>0800 -0FFF
Sprite Motion Table	unused	>0780 -07FF		>0780 -07FF	>0780 -07FF

Used Registers:	BLWP <gas>	RTWP	BL <gas>	XOP <gas>,<xop>	(WP) → R13
(gas) → (WP)	(old PC) → (R14)	(R13) → (WP)	(old PC) → (R11)	(gas) → (R11)	(PC) → R14
(gas+2) → (PC)	(ST) → (R15)	(R14) → (PC)	(gas) → (PC)	(>0040 + (xop)*4)→(WP)	(ST) → R15
(Old WP)→ (R13)		(R15) → (ST)		(>0042 + (xop)*4)→(PC)	1 → XOP Stat.Bit

Your personal notes: