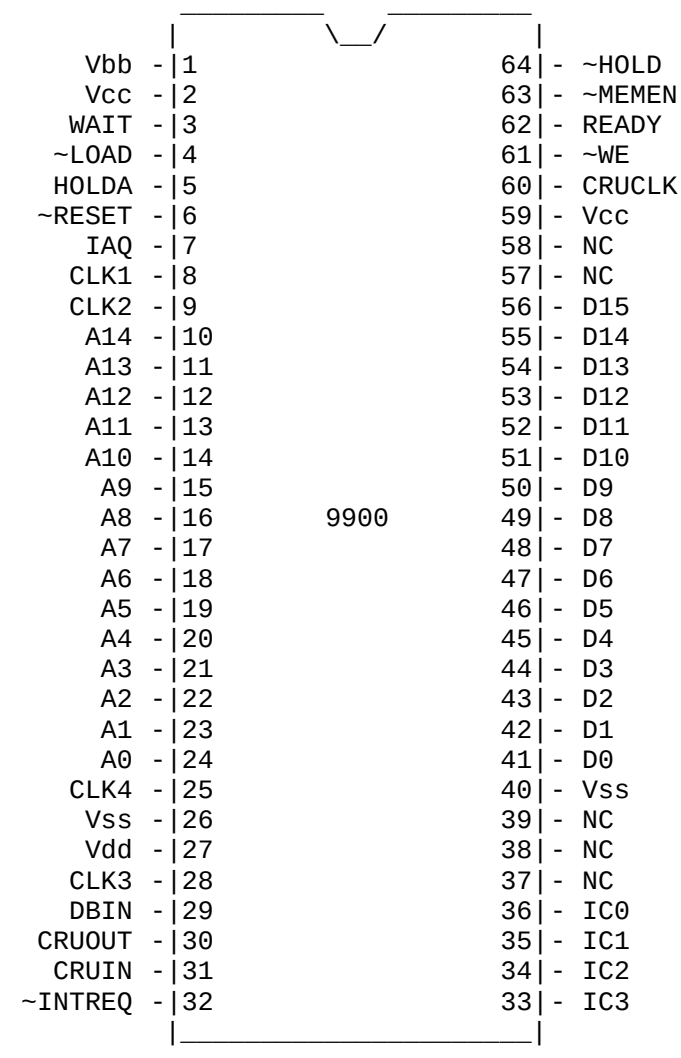
Texas Instruments

```
 99999     99999      000       000
9     9   9     9    0   0     0   0
9     9   9     9   0   0 0   0   0 0
 999999    999999   0  0  0   0  0  0
      9         9   0 0   0   0 0   0
     9         9     0   0     0   0
 9999      9999       000       000
```

9900 MICROPROCESSOR Instruction Set Summary



Written by Jonathan Bowen
Programming Research Group
Oxford University Computing Laboratory
8-11 Keble Road
Oxford OX1 3QD
England

Tel +44-865-273840

Created August 1981
Updated April 1985
Issue 1.2

Mnemonic	Code	IXPVCEAL	F	Z	Description
A s,d	A000	---*****	1	Y	Add
AB s,d	B000	--******	1	Y	Add Bytes
ABS d	0740	---*****	6	Y	Absolute value
AI r,i	0220	---*****	8	Y	Add Immediate
ANDI r,i	0240	-----***	8	Y	AND Immediate
B s	0440	--------	6	N	Branch (PC=d)
BL s	0680	--------	6	N	Branch and Link (R11=PC,PC=s)
BLWP s	0400	--------	6	N	Branch & Load Workspace Ptr (3) (2)
C s,d	8000	-----***	1	N	Compare
CB s,d	9000	--*-***	1	N	Compare Bytes
CI r,i	0280	-----***	8	N	Compare Immediate
CKOF	03C0	--------	7	N	Clock Off
CKON	03A0	--------	7	N	Clock On
CLR d	04C0	--------	6	N	Clear
COC s,r	2000	-----*--	3	N	Compare Ones Corresponding
CZC s,r	2400	-----*--	3	N	Compare Zeros Corresponding
DEC d	0600	---*****	6	Y	Decrement
DECT d	0640	---*****	6	Y	Decrement by Two
DIV d,r	3C00	---*----	9	N	Divide
IDLE	0340	--------	7	N	Computer Idle
INC d	0580	---*****	6	Y	Increment
INCT d	05C0	---*****	6	Y	Increment by Two
INV d	0540	-----***	6	Y	Invert
JEQ a	1300	--------	2	N	Jump if Equal
JGT a	1500	--------	2	N	Jump if Greater Than
JH a	1B00	--------	2	N	Jump if High
JHE a	1400	--------	2	N	Jump if High or Equal
JL a	1A00	--------	2	N	Jump if Low
JLE a	1200	--------	2	N	Jump if Low or Equal
JLT a	1100	--------	2	N	Jump if Less Than
JMP a	1000	--------	2	N	Jump unconditionally
JNC a	1700	--------	2	N	Jump if No Carry
JNE a	1600	--------	2	N	Jump if Not Equal
JNO a	1900	--------	2	N	Jump if No Overflow
JOC a	1800	--------	2	N	Jump On Carry
JOP a	1C00	--------	2	N	Jump if Odd Parity
LDCR s,c	3000	--*--***	4	Y	Load Communication Register

LI	r,i	0200	-----***	8	N	Load Immediate
LIMI	i	0300	*-------	8	N	Load Interrupt Mask Immediate
LREX		03E0	*-------	7	N	Load or Restart Execution
LWPI	i	02E0	--------	8	N	Load Workspace Pointer Immediate
MOV	s,d	C000	-----***	1	Y	Move
MOVB	s,d	D000	--*--***	1	Y	Move Bytes
MPY	d,r	3800	--------	9	N	Multiply
NEG	d	0500	---*****	6	Y	Negate
NOP		1000	--------		N	No Operation (pseudo-operation)
ORI	r,i	0260	-----***	8	Y	OR Immediate
RSET		0360	*-------	7	N	Reset
RT		0458	--------		N	Return (replaces 'B *11',pseudo-op)
RTWP		0380	????????	7	N	Return Workspace Pointer (4)
S	s,d	6000	---*****	1	N	Subtract
SB	s,d	7000	--******	1	N	Subtract Bytes
SBO	a	1D00	--------	2	N	Set Bit to One
SBZ	a	1E00	--------	2	N	Set Bit to Zero
SETO	d	0700	--------	6	N	Set to Ones
SLA	r,c	0A00	----****	5	Y	Shift Left Arithmetic (1)
SOC	s,d	E000	-----***	1	Y	Set Ones Corresponding
SOCB	s,d	F000	-----***	1	Y	Set Ones Corresponding Bytes
SRA	r,c	0800	----****	5	Y	Shift Right Arithmetic (1)
SRC	r,c	0800	----****	5	Y	Shift Right Circular (1)
SRL	r,c	0900	----****	5	Y	Shift Right Logical (1)
STCR	s,c	3400	--*--***	4	Y	Store Communication Register
STST	r	02C0	--------	8	N	Store Status Register
STWP	r	02A0	--------	8	N	Store Workspace Pointer
SWPB	d	06C0	--------	6	N	Swap Bytes
SZC	s,d	4000	-----***	1	Y	Set Zeros Corresponding
SZCB	s,d	5000	-----***	1	Y	Set Zeros Corresponding Bytes
TB	a	1F00	-----*--	2	N	Test Bit
X	s	0480	--------	6	N	Execute the instruction at s
XOP	s,c	2C00	-1------	9	N	Extended Operation (5) (2)
XOR	s,r	2800	-----***	3	N	Exclusive OR
		XXXX				Hexadecimal opcode (16-bit)
		-*01?				Unaffect/affected/reset/set/unknown
				F		Format (1-9, see later)
					Z	Result compared to zero (Y/N)

Mnemonic	IXPVCEAL		Description
	I		Interrupt mask (Bits 12-15)
X	X		Extended operation (Bit 6)
OP	P		Odd Parity (Bit 5)
OV	V		Overflow (Bit 4)
C	C		Carry (Bit 3)
EQ	E		Equal (Bit 2)
A>	A		Arithmetic greater than (Bit 1)
L>	L		Logical greater than (Bit 0)
Rn			Workspace register (TT=00)
*Rn			Indirect workspace register (TT=01)
*Rn+			Indirect auto increment (TT=11)
@nn			Symbolic (direct)
nn(Rn)			Indexed (not R0, TT=10)
nn			Immediate (TT=10)
a			PC relative (PC=PC+2+2*offset)
a			CRU relative (PC=CRU+offset)

BES nn	Block Ending with Symbol
BSS nn	Block Starting with Symbol
BYTE e(,...)	Byte (,byte...)
DEFW nn(,...)	Define Word (,word...)
EVEN	Even word boundary
CRU	Communication Reg (Bits 4-13, R12)
PC	Program Counter (16-bit)
Rn	Workspace Register (16-bit)
n	ditto (if no mode conflict)
WP	Workspace Pointer (16-bit)
ST	Status Register (16-bit)
a	Relative address (-128 to +127)
c	Count (4-bit, 0 to 15)
d	Destination
e	8-bit expression (0 to 255)
i	Immediate
n	Register number (0 to 15)
nn	16-bit expression (0 to 65535)
r	Workspace register
s	Source
0000H to 003FH	Interrupt trap vectors
0040H to 007FH	XOP instruction trap vectors
FFFCH to FFFFH	LOAD function trap vector
XXXBTTDDDDTTSSSS	Format 1 (Arithmetic)
XXXXXXXXOOOOOOOO	Format 2 (Jump)
XXXXXXDDDDTTSSSS	Format 3 (Logical)
XXXXXXCCCCTTSSSS	Format 4 (CRU)
XXXXXXXXCCCCWWWW	Format 5 (Shift)
XXXXXXXXXXTTSSSS	Format 6 (Program)
XXXXXXXXXXXUUUUU	Format 7 (Control)
XXXXXXXXXXXUWWWW	Format 8 (Immediate, word follows)
XXXXXXDDDDTTSSSS	Format 9 (MPY,DIV,XOP)
B	Byte indicator (0=word, 1=byte)
CCCC	Transfer or shift count
DDDD	Destination address
OOOOOOOO	Offset (-128 to +127 words)
SSSS	Source address
TT	Address modifiction
U	Unused
WWWW	Workspace register number
X...X	Op code
(1)	If c=0 bits 12-14 of R0 are used
(2)	R13=old WP,R14=old PC,R15=old ST
(3)	WP=(s),PC=(s+2)
(4)	WP=R13,PC=R14,ST=R15
(5)	WP=(40H+4c),PC=(42H+4c),R11=s