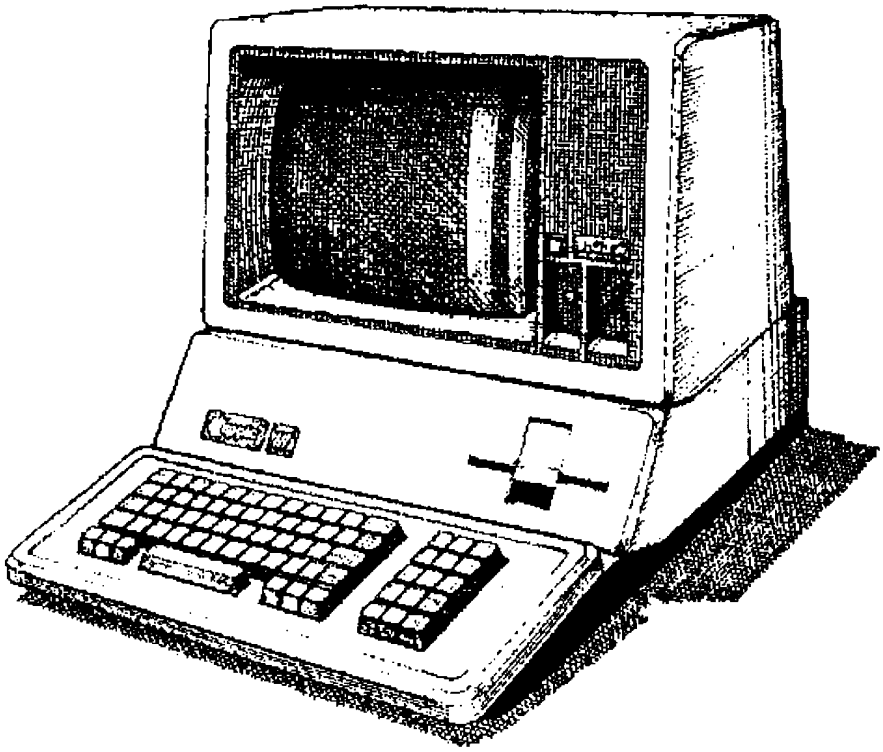
Apple /// Computer Information

Apple /// Service Reference Manual



Section II of II • Servicing Information

Chapter 17 • General Appendix

Written by Apple Computer • 1982

apple computer inc.

APPLE /// SYSTEM OVERVIEW

FEATURING:

[] EXTENDED DISPLAY
[] EXPANDED RAM
[] NEW PROCESSOR DESIGN
[] BUILT IN I/O
[] APPLE II EMULATOR

STARRING: THE APPLE /// PROCESSOR

* 6502 INSTRUCTION SUBSET
* RELOCATABLE BASE REGISTER PAGE
* RELOCATABLE STACK
* 256K BYTE ADDRESS RANGE

VIDEO

* NTSC COLOR COMPOSITE VIDEO
* NTSC B/W COMPOSITE VIDEO
* SYNC
* 4 PRIMARY INDEPENDENT VIDEO LINES
* MIX TO FROM RGB APPLE COLORS
  THREE LINES CAN DRIVE TTL RGB MONITOR
  FOUR INDEPENDENT VIDEO OUTPUTS CAN BE GENERATED

DISPLAY MODES

* GRAY SCALE ON B/W OUTPUT
* RAM CHARACTER GENERATOR (128 CHAR.)
* 40 X 24 CHARACTER B/W TEXT (2K BYTES RAM)
* 80 X 24 CHARACTER B/W TEXT
* 40 X 24 CHARACTER COLOR TEXT
  (16 BACKGROUND, 16 TEXT COLORS)
* 280 K 192 B/W HIRES (8K RAM)
* 560 X 192 B/W HIRES
* 140 X 192 16-COLOR HIRES
* 280 X 192 16-COLOR HIRES WITH 40 X 192 BACKGROUND/FOREGROUND RESOLUTION

I/O

* FOUR APPLE II BUS PERIPHERAL SLOTS
* ONE BUILT IN DISK DRIVE
* CONTROLLER FOR THREE ADDITIONAL DRIVES
* RS 232 PORT (COMPLETE)
* SILENTYPE PORT
* TWO PADDLE PORTS (A/D INPUTS)
* EXTERNAL SOUND JACK (SIX BIT AUDIO, HARDWARE BEEPER
* EXPANDED VIDEO LINES

apple computer inc.

* CLOCK/CALENDER

I/O BLOCK TRANSFER

* PERMITS UP TO 1 PAGE (256 BYTE) FAST I/O TRANSFER WITHOUT DMA HARDWARE ON PERIPHERAL
* TRANSFERS AT UP TO THE RAM CYCLE RATE

APPLE II EMULATION RESTRICTIONS

* NO LANGUAGE OR ROM CARD
* PADDLES ARE DIFFERENT
* ENTER WITH SOFTWARE BUT ONLY RESET WILL EXIT

apple computer inc.

6.1 Development Monitor

The current version of the diagnostic/boot ROM includes a monitor that may be useful for debugging. THE MONITOR IS NOT A PART OF THE SUPPORTED Apple III SOFTWARE AND IT MAY BE CHANGED OR DELETED IN FUTURE VERSIONS OF THE BOOT ROM. MONITOR SUBROUTINES MUST NOT BE CALLED FROM DRIVERS OR INTERPRETERS.

6.2 Monitor Commands

The monitor commands are listed below. Commands are read directly from the keyboard; blanks are not used except as data separators in a {byte list}. Command lines may be up to 79 characters long and are always terminated by a RETURN. Multiple commands may be entered on the same line using a slant (/) as a command separator. Numeric data is always entered in hex. ASCII strings may be entered by enclosing them in single or double quotes. If single quotes (') are used, bit 7 of each byte will be clear; if double quotes (") are used, bit 7 will be set.

```
      {address} ::= numeric value 0..FFFF
{address range} ::= {address} | {address}.{address}
         {byte} ::= numeric value 0..FF | Single byte string
    {byte list} ::= one or more bytes separated by blanks
    {block num} ::= numeric value 0..117
```

{address range}	Memory dump
{address} : {byte list}	Memory store
{byte} < {address range} S	Memory search
{address} < {address range} M	Memory move
{address} < {address range} V	Memory verify
{block num} < {address range} R	Read disk
{block num} < {address range} W	Write disk
{address} G	Subroutine call (JSR)
{address} J	Execute code (JMP)
U	Call user subroutine (JSR $3F8)
X	Repeat command line
RETURN	Continue memory dump

When dumping memory, the output can be suspended then stepped by pressing the space bar; pressing any other key will resume normal output. Pressing the TAB key will terminate processing of the command line.

6.3 Escape Commands

Escape commands may be used during command input to move the cursor or control the display. Escape mode is entered by typing an ESCAPE; the cursor is changed to a flashing plus sign (+) to identify escape mode. Any character that is not an escape command will terminate escape mode.

↑ or K^c	Move cursor up
↓ or J^c	Move cursor down
← or H^c	Move cursor left
→ or U^c	Move cursor right

apple computer inc.

Apple III I/O System Programmer's Guide

L	Clear to end of line
P	Clear to end of page
S	Home cursor and clear screen
4	Set 40 column display
8	Set 80 column display

6.4 Zero Page Locations

58	Window left
59	Window right
5A	Window top
5B	Window.bottom
5C	Horizontal cursor position
5D	Vertical cursor position
74,75	A1
76,77	A2
78,79	A3
7A,7B	A4

6.5 Entering Addresses

adr1	A1, A2, A := adr1		
adr1.adr2	A1, A3 := adr1	A2 := adr2	
adr3<adr1.adr2	A1, A3 := adr1	A2 := adr2	A4 := adr3

6.6 Pointer Usage

Memory dump:	(A1)
Memory store:	(A3)
Memory move:	(A4) := (A1)
Memory verify:	(A4) : (A1)
Memory search:	A4 : (A1)

apple computer inc.

APPLE /// LOGIC SIGNAL SOURCE

The location of the signal source is described first by the page number of the schematic followed by its coordinates.

SIGNAL	LOCATION	MEANING
6551sel*	5-4a	ACIA SEL
a0-a7	4-4d	ADDRESS BUS (EXTERNAL)
a8-a11	4-4c	
a12-a15	4-4b	
abk1	3-2c	ADDRESS BANK 1
abk2	3-2c	ADDRESS BANK 2
abk3	3-2c	ADDRESS BANK 3
abk4	3-2c	ADDRESS BANK 4
ahires	6-4a	AHIRES
aIIlores	6-4a	APPLE II LORES
altstk*	5-1b	ALTERNATE STACK
apple i*	9-4b	APPLE SWITCH 1
apple ii*	9-4b	APPLE SWITCH 2
ar0	3-3b	RAM ADDRESS X
anyky	9-4c	ANYKEY (DEPRESSED)
ar0	3-3d	
ar1	3-3d	
ar2	3-3c	
ar3	3-3c	
ar4	3-3c	
ar5	3-3b	
ar6	3-3b	
audio	8-1b	
ax,ax*	10-3a	ADDRESSS MUX SELECT
axco	5-2a	
bckswl,3	8-4b	BANKSWITCH
bl	10-1b	BLANKING
c08xn-c0Fxn	5-4a	ADDRESS DECODE
c0xxn-c7xxn	5-4b	ADDRESS DECODE
c1m,c1m*	5-4b	CLOCK 1 MEGHERTZ
c14m,*	10-4c	CLOCK 14 MEGHERTZ
c3,5m,c3,5m*	10-4c	
c7m,c7m*	10-4b	
caplk*	9-4b	CAPS LOCK SWITCH
cas0*	3-2c	COLUMN ADDRESS SELECT (RAM)
cas1*	3-2c	
cas2*	3-2c	
cas3*	3-2c	
cas4,7*	3-2c	
cas5,6*	3-2c	
ch80*	6-4a	CHARACTER (80 COLUMN)
clkbat	5-2d	CLOCK BATTERY
clken80	6-2d	CLOCK ENABLE 80 (COLUMN)

apple computer inc.

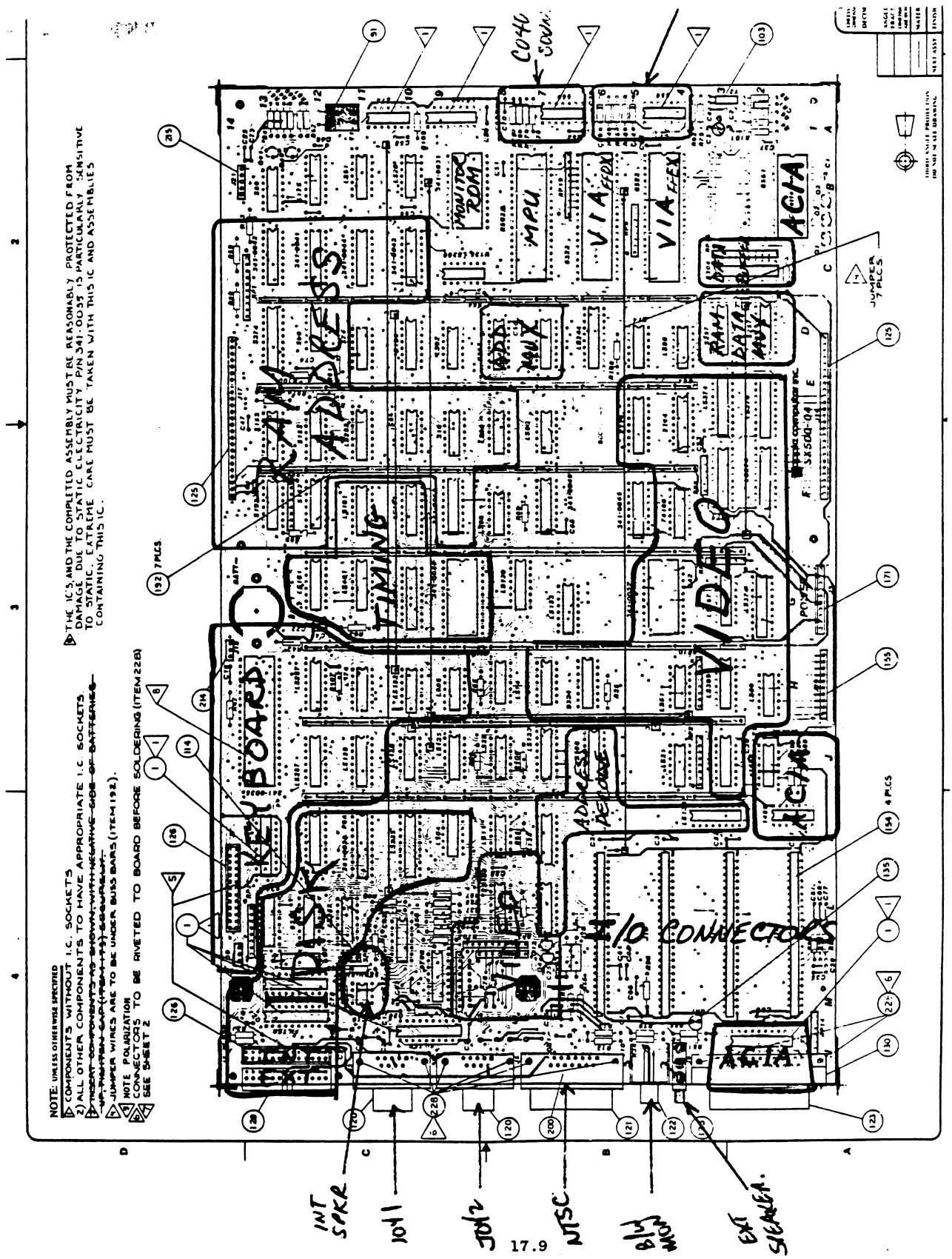
clkirq*	5-1c	CLOCK INTERRUPT
clrkl*	6-4a	
clrstrb*	6-3a	CLEAR STROBE (KEYBOARD)
colrgate	10-1b	COLOR GATE
comp	10-1b	COMPENSATE
control*	9-4b	CONTROL KEY
cs6522	10-4a	
cts	8-1c	CLEAR TO SEND (EIA)
c00x-c07x	5-3a	ADDRESS DECODE
cxxx	5-4b	
c-fxxx	5-4c	
d0-d7	4-2c	
da0,da7	10-3b	
datain	8-1c	EIA
db0-db7	10-4b	
dc0,dc7	6-3d	VIDEO BUS (CHAR GEN OUT)
dcd	8-1c	DATA CARRIER DETECT
devsell,6*	5-4a	DEVICE SELECT
dhires	5-4a	DHIRES
dmal*	4-4d	DIRECT MEMORY ACCESS IN
dmaok	4-4a	DMA OK
dph0,dph3	7-4b	DISK (MOTOR) PHASE X
dsply	10-1b	DISPLAY
dsr	8-1c	DATA SET READY (EIA)
dtrdy,*	9-4c	DATA READY (KEYBOARD)
dtr	8-1c	DATA TERMINAL READY
dv0,dv7	6-4d	VIDEO BUS (RAM LATCH OUT)
dxo,dx7	6-3a	VIDEO BUS (COLOR LATCH OUT)
en257	5-3d	ENABLE RAM BUS TO DATA BUS
en8304	5-3d	ENABLE MPU XCVR
enbl1,e*	7-2c	ENABLE DISK 1 EXTERNAL
enbl1,i*	7-2c	ENABLE DISK 1 INTERNAL
enbl2,e*	7-2b	ENABLE DISK 2 EXTERNAL
enbl3,i	7-2b	ENABLE DISK 3 EXTERNAL
encwrt	7-2c	ENABLE CHARACTER RAM WRITE
enhreg*	3-3c	ENABLE GRAPHICS REGISTER (LATCH)
ensel	7-2c	ENABLE EXT PRINTER SELECT
ensio	7-2c	ENABLE SERIAL I/O
PDL0	5-2a	PADDLE ADDRESS 0
ext*	7-4b	EXTERNAL (DRIVE)
extspk	8-1b	EXTERNAL SPEAKER
ffcx*	5-3c	
ffdx*	5-3c	
ffex*	5-3c	
fieldout	10-1b	
fspace*	5-3b	
gph1	5-3c	
gph2	5-3c	
h0,h3	10-2d	VIDEO STATE
h4,h5	10-2d	VIDEO STATE
hires	5-2a	
hpe*	10-2d	
id0-id7	4-3c	INTERNAL DATA BUS
ind*	3-3a	

apple computer inc.

inh*	5-4c	INHIBIT (ROM)
int*	7-2d	INTERNAL (DRIVE)
blankin	8-3c	
ioen	5-1b	I/O (DEVICES) ENABLE
ionmi*	9-2b	I/O NON MASKABLE INTERRUPT
iosel1,4*	5-4b	I/O SELECT
iostopd*	10-4b	I/O STOPPED
iostrb*	5-4b	I/O STROBE
io sync	4-3b	I/O SYNC
irq1,4*	5-3c	
i r*/w	4-2c	INTERNAL READ/WRITE
i r/w*	4-3c	
kbint*	9-2c	KEYBOARD INTERRUPT
kbo*	5-3a	KEYBOARD STROBE(ENABLE KEY TO DATA)
kreset*	9-4b	RESET KEY
kvcc	9-4b	KEYBOARD VCC
ldps*	10-4c	LOAD PARARELL SHIFT REGISTER
mix	5-2a	
muxl	3-3c	
nmi*	9-1b	
ntsca,b	6-2b	
oe374	6-4a	ENABLE COLOR LATCH OUTPUTS
pa8	4-4b	PROCESSOR ADDRESS 8
pal5	4-4b	PROCESSOR ADDRESS 15
page 2	5-2a	
pcas0*	3-2d	PROM CAS X
pcas1*	3-2d	
pcas2*	3-2d	
pcas3*	3-2c	
pdint*	5-1c	POWER DOWN INTERRUPT (CLOCK)
pdl2	5-2a	PADDLE ADDRESS 2
pdlen	5-2a	PADDLE ENABLE (ADC RAMPSTART)
pdlot*	8-3a	PADDLE OUT (RAMPSTART)
pg2*	6-4a	
ph0	10-3a	PROCESSOR 0
ponrst	9-3a	POWER ON RESET
pras0,3*	3-2b	PROM RAS X
pras1,2*	3-2b	
pras4,5*	3-2b	
pras6,7*	3-2b	
preim	10-4b	PRE 1 MEGAHERTZ
pwrdwn*	5-2d	POWER DOWN
q0,q3	10-3c	Q STATE (ZMEG)
ram r/w	3-1b	
ras, ras*	10-4b	RAM ROW ADDRESS SELECT
ras0,3*	3-1b	
ras1,2*	3-1b	
ras4,5*	3-1b	
ras6,7*	3-1b	
rbl	10-1c	ROM BLANK
rclkpwr	5-2d	CLOCK POWER
rcolrgt	10-1c	ROM COLORGATE
rddata	7-2d	READ DATA (DISK)
rdy	3-3a	READY

apple computer inc.

rdy*	3-3a	
reset*	9-2b	
resetlk*	5-1b	RESET LOCK
rfield	10-1b	
rfsh	10-1b	REFRESH
rgbl,rgb8	6-3b	RED,GREEN,BLUE
romsel*	5-3b	ROM SELECT
romsel1	5-1b	
romsel2	5-1b	
rrfsh	10-1b	ROM REFRESH
rtcwrt	10-1c	ROM TIME CHARACTER RAM WRITE
rts	8-1c	REQUEST TO SEND
rsync	10-1c	ROM SYNC
rwpr	5-1b	
rwrprot	10-1b	
r/w	4-3c	READ WRITE
s399	3-3a	
s50/60	10-1b	SELECT 50 HZ/60 HR
sco	5-1b	SERIAL CLOCK
scr	7-2c	SCROLL
scrn	5-1b	SCREEN
sel2m*	5-1b	SELECT 2 MEG
sel374	6-4a	SELECT ORDER OF 374'S to dvx bux
ser	5-1b	SERIAL DATE
shift*	9-3c	SHIFT KEY
spkr*	5-3a	SPEAKER (STROBE)
sum4	3-3d	
sum3	3-3d	
sum2	3-3d	
sum1	3-3d	
synch	10-1b	
tcwrt	10-1b	TIME CHARACTER RAM WRITE
text	5-2a	
tromsel	4-3a	
tromsel*	5-3d	
tsadb*	4-4d	
txd	8-1c	
usel b	3-2c	MICROPROCESSOR SELECT B RAM BUS
v0,v1	10-2c	
v2,v5	10-2b	
va	10-2c	
vb,vc	10-2c	
vbl	10-1c	
we2114	6-4d	WRITE ENABLE CHARACTER RAM
wrdata	7-2d	WRITE DATE (DISK)
wramen	10-4a	WRITE RAM ENABLE
wrprot	7-2d	WRITE PROTECT
wrreq	7-2d	WRITE REQUEST
x0,x7	9-4c	KEY SCAN
y0,y9	9-4c	KEY SCAN
z0,z7	5-1b	ZERO PAGE ADDRESS BITS
zpage*	4-4b	ZERO PAGE


KEYBOARD
RAM ADDRESS
TIMING
VIDEO
I/O CONNECTORS
MPU
VIA
VIA
ACIA
ACIA
ACIA
ROM
ADDR MUX
RAM DATA MUX
ADDRESS DECODE
JUMPER 7 PLCS
125
171
155
154
4 PLCS
135
130
122
121
120
200
INT SPKR
JOY 1
JOY 2
NTSC
B/W MON
EXT SPEAKER
17.9

apple computer inc.

HOW TO READ PROM (ROM) LOGIC EXPRESSIONS

X′ -- The single quote at the end of an expression means that the state of the signal is true when low, or it may represent the inversion of the state of the signal.

* -- Defines a logic AND operation. The expressions on either side of the asterisk is ANDed.

\+ -- Defines a logic OR operation. The expression on either side of the asterisk is OR'd.

() -- Defines the boundaries of a logic expression. A new expression is defined by what ever is inside of the brackets.

RULES FOR INTERPRETATION

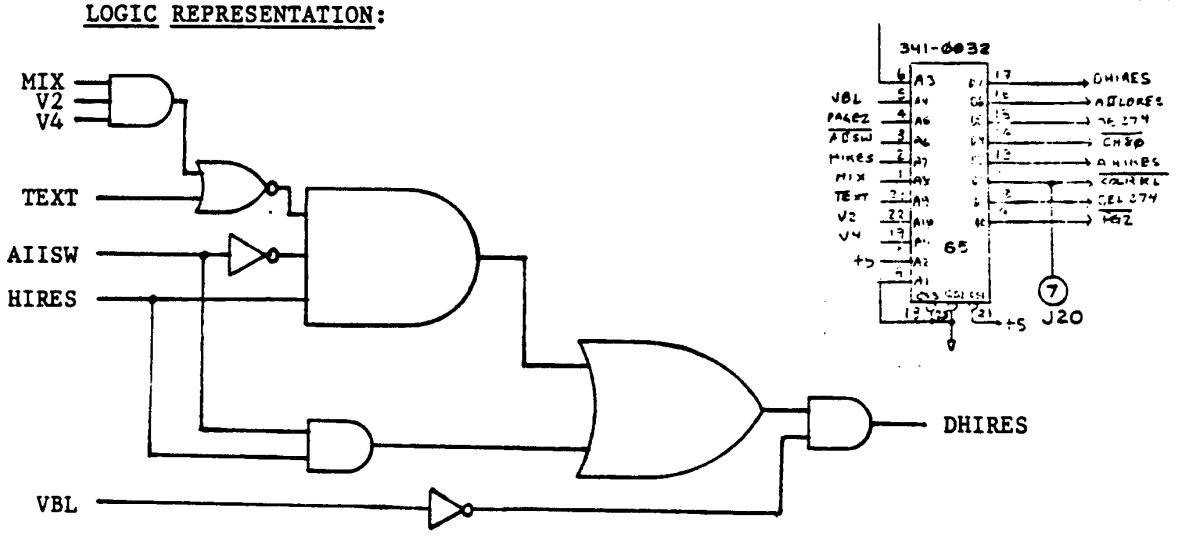
1. Always interpret (transform) the expression within the brackets first.
2. Interpret AND (*) logic operations before OR (+) operations.

EXAMPLE:

Given: INPUTS: AIISW′ HIRES TEXT
MIX V2 V4
VBL

OUTPUT: DHIRES = (AIISW*HIRES*(TEXT+MIX*V2*V4)′+AIISW′*HIRES)*VBL′

LOGIC REPRESENTATION:



apple computer inc.

```
PROM 341-0043

A=A11
B=A12
C=A13
D=A14
E=IOSYNC
F=SEL2M'
G=ABK4
H=PA8
I=A15
J=ZPAGE'
D0=S399
D1=PRDY'
D2=IND'

S399=ZPAGE*PA8'*A15'*A14'*A13'*A12*A11

PRDY'=IOSYNC*SEL2M*ABK4

IND'=(ABK4*(ZPAGE*PA8')')'
```

apple computer inc.

```
PROM 341-0042

A=A13
B=A11
C=A15
D=IND'
E=DHIRES
F=ABK1
G=ABK2
H=ABK3
I=A14
J=AY'
D0=PCAS4,7'
D1=PCAS5,6'
D2=PCAS1'
D3=PCAS2'

PCAS4,7'=(AY*IND'*ABK3'*A15'*(ABK1*ABK2'+ABK1'*ABK2)*(A14'*A13+A14*A13')+AY*
IND*ABK3'*(ABK2'*ABK1'*A15+ABK2'*ABK1+ABK2*ABK1'*A15')*A14'+AY*IND'*ABK1*
ABK2*ABK3'*(A15'*A14'*A13+A15*A14'*A13')+AY*IND*ABK3'*ABK2*(A15'*ABK1+A15*
ABK1')*(A14'*A13'+A14*A13))'

PCAS5,6'=(AY*IND'*ABK3'*(ABK1*ABK2'+ABK1'*ABK2)*(A15'*A14*A13+A15*A14'*A13')+
AY*IND*ABK3'*(ABK2'*ABK1'*A15+ABK2'*ABK1+ABK2*ABK1'*A15')*A14+AY*IND'*ABK1*
ABK2*ABK3'*(A15'*A14)+AY*IND*ABK3'*ABK2*(A15'*ABK1+A15*ABK1')*(A14'*A13+A14*
A13'))'

PCAS1' =(DHIRES *AY'+AY*(ABK1'*ABK2'*ABK3'*IND'+ABK1*ABK2*ABK3)*(A15'*A14'*
A13+A15*A14'*A13')+AY*IND*ABK1'*ABK2'*ABK3'*A15'*(A14'*A13'+A14*A13))'

PCAS2' =(DHIRES *AY'+AY*(ABK1'*ABK2'*ABK3'*IND'+ABK1*ABK2*ABK3)*(A15'*A14)+
AY*IND*ABK1'*ABK2'*ABK3'*A15'*(A14'*A13+A14*A13'))'
```

apple computer inc.

```
ROM 341-0032

A=A
B=B
C=S4
D=VBL
E=PAGE2
F=AIISW'
G=HIRES
H=MIX
I=TEXT
J=V2
K=V4
D0=PG2
D1=SEL374
D2=COLRKL'
D3=AHIRES
D4=CH80'
D5=OE374
D6=AIILORES
D7=DHIRES

PG2=AIISW'*(HIRES*MIX'*TEXT ')'*HIRES*PAGE2*S4*VBL'

SEL374=(VBL'*(AIISW*(PAGE2'*(TEXT +MIX*V2*V4)'+PAGE2*(TEXT +MIX*V2*V4))+
AIISW'*(HIRES*(PAGE2*S4)'+HIRES'*(PAGE2*S4))))'

COLRKL'=(AIISW*TEXT +AIISW'*(HIRES'*(MIX+TEXT ')+HIRES*TEXT '))'

AHIRES=AIISW'*(HIRES*MIX*TEXT )

CH80'=(AIISW'*MIX)'

OE374=(AIISW*HIRES'*(TEXT +MIX*V2*V4)'+AIISW'*MIX'*TEXT )'

AIILORES=AIISW*HIRES'*(TEXT +MIX*V2*V4)'+AIISW'*HIRES*MIX*TEXT

DHIRES=(AIISW*HIRES*(TEXT +MIX*V2*V4)'+AIISW'*HIRES)*VBL'
```

apple computer inc.

PROM 341-0046 U180

```
A=C1M
B=C07X'
C=RAMEN
D=DSPLY
E=IOSTOPD'
F=C-FXXX
G=FSPACE'
H=R/WN
I=RWRPROT
J=SEL2M'
D0=PCS6522
D1=PHASEN
D2=WRAMEN

PCS6522=(IOSTOPD'*C1M+FSPACE'*C1M')'

PHASEN=((SEL2M'*C07X')'*IOSTOPD'*FSPACE+FSPACE*C1M'+C1M'*SEL2M'+C1M'*DSPLY*
RAMEN)'

WRAMEN=RAMEN*(RWRPROT*C-FXXX)'*R/WN'*(DSPLY*C1M')'
```

apple computer inc.

```
PROM 341-0056

A=A11
B=A13
C=A14
D=A15
E=R/WN
F=DHIRES
G=AY'
H=ABK2
I=PRAS1,2
J=PRAS0,3
D0=PCAS0'
D1=PUSELB
D2=PCAS0,3
D3=PCAS3'

PCAS0'=(PRAS0,3 *(DHIRES'*AY'+AY*(A15'*A14'*A13'*A11'*R/WN'+A15'*A14'*A13'*
R/WN+A15*A14'*A13+A15*A14*A13'*A11)))'

PUSELB =PRAS0,3 *(A15'*A14'*A13'*A11+A15*A14*A13'*A11'+A15*A14*A13)+PRAS0,3 *
PRAS1,2 *(A15'*A14'*A13+A15*A14'*A13')+PRAS0,3 *PRAS1,2 '*(A15'*A14'*A13+A15'*
A14*A13')+PRAS0,3 '*PRAS1,2 *(A14'*A13'+A14*A13)+PRAS0,3 '*PRAS1,2 '*A14'

PCAS0,3 =(PRAS0,3 *(DHIRES'*AY'+AY*(A15'*A14'*A13'*A11+A15*A14*A13'*A11'+A15*
A14*A13))+PRAS0,3 *(DHIRES'*AY'+AY*(A15'*A14'*A13'*A11'*R/WN'+A15'*A14'*A13'*
R/WN+A15*A14'*A13+A15*A14*A13'*A11)))'

PCAS3'=(PRAS0,3 *(DHIRES'*AY'+AY*(A15'*A14'*A13'*A11+A15*A14*A13'*A11'+A15*
A14*A13)))'
```

apple computer inc.

PROM 341-0044 RAS

```
A=ABK1
B=ABK2
C=ABK3
D=PA15
E=AY'
F=PA8
G=ZPAGE'
H=DHIRES
I=RFSH
J=ABK4
D0=PRAS0,3
D1=PRAS1,2
D2=PRAS4,5
D3=PRAS6,7


PRAS0,3 =AY'*(DHIRES'+RFSH)+((ABK4*(ZPAGE*PA8')')'+ABK1*ABK2*ABK3)*AY

PRAS1,2 =AY'*(DHIRES+RFSH)+AY*(ABK1'*ABK2'*ABK3'*(ABK4*(ZPAGE*PA8')'*PA15)'+
ABK1*ABK2*ABK3)+AY*ABK3'*(ABK1'*ABK2*ABK4*(ZPAGE*PA8')'*PA15+ABK1*ABK2*(ABK4*(
ZPAGE*PA8')'*PA15)')

PRAS4,5 =RFSH*AY'+AY*ABK2'*ABK3'*(ABK1'*ABK4*(ZPAGE*PA8')'*PA15+ABK1*(ABK4*(
ZPAGE*PA8')'*PA15)')

PRAS6,7 =RFSH*AY'+AY*ABK3'*(ABK1*ABK2'*ABK4*(ZPAGE*PA8')'*PA15+ABK1'*ABK2*(
ABK4*(ZPAGE*PA8')'*PA15)')
```

apple computer inc.

PROM 341-0045 U176

```
A=C5XXN
B=R/WN
C=C6XXN
D=INH'
E=ROMSEL'
F=DMAI'
G=FSPACE'
H=PH2M
I=C7XXN
J=CXXX
D0=EN257
D1=EN8304
D3=RAMEN

EN257=(PH2M*R/WN*ROMSEL'*INH'*FSPACE'*(C5XXN *C6XXN*C7XXN*CXXX)')'

EN8304=(PH2M*FSPACE'*ROMSEL'*DMAI')'

RAMEN=ROMSEL'*INH'*FSPACE'*(C5XXN *C6XXN*C7XXN*CXXX)'
```

apple computer inc.

PROM 341-0055 U175.1

```
A=VA
B=VB
C=VC
D=VA1
E=VB1
F=VC1
G=DHIRES
H=SCR
I=WE2114'
J=VBL
D0=MUX1
D1=MUX2
D2=MUX3
D3=ENHREG'

MUX1=((DHIRES'+SCR*(VA*VA1'+VA'*VA1)+SCR'*VA)*VBL'+VBL)'+VBL*WE2114'*SCR*VC1

MUX2=(DHIRES*(SCR*(VA*VA1*(VB*VB1'+VB'*VB1)'+(VA*VA1)'*(VB*VB1'+VB'*VB1))+VB*
SCR'))'

MUX3=DHIRES*(SCR*((VA*VA1*(VB+VB1)+VB*VB1)*(VC*VC1'+VC'*VC1)'+(VA*VA1*(VB+
VB1)+VB*VB1)'*(VC*VC1'+VC'*VC1))+VC*SCR')

ENHREG'=DHIRES'*WE2114'
```

apple computer inc.

ASCII Conversion Tables

DEC	ASCII	OCTAL	HEX	BINARY 76543210	DEC	ASCII	OCTAL	HEX	BINARY 76543210
0	NUL	000	00	00000000	64	@	100	40	01000000
1	SOH	001	01	00000001	65	A	101	41	01000001
2	STX	002	02	00000010	66	B	102	42	01000010
3	ETX	003	03	00000011	67	C	103	43	01000011
4	EOT	004	04	00000100	68	D	104	44	01000100
5	ENQ	005	05	00000101	69	E	105	45	01000101
6	ACK	006	06	00000110	70	F	106	46	01000110
7	BEL	007	07	00000111	71	G	107	47	01000111
8	BS	010	08	00001000	72	H	110	48	01001000
9	HT	011	09	00001001	73	I	111	49	01001001
10	LF	012	0A	00001010	74	J	112	4A	01001010
11	VT	013	0B	00001011	75	K	113	4B	01001011
12	FF	014	0C	00001100	76	L	114	4C	01001100
13	CR	015	0D	00001101	77	M	115	4D	01001101
14	SO	016	0E	00001110	78	N	116	4E	01001110
15	SI	017	0F	00001111	79	O	117	4F	01001111
16	DLE	020	10	00010000	80	P	120	50	01010000
17	DC1	021	11	00010001	81	Q	121	51	01010001
18	DC2	022	12	00010010	82	R	122	52	01010010
19	DC3	023	13	00010011	83	S	123	53	01010011
20	DC4	024	14	00010100	84	T	124	54	01010100
21	NAK	025	15	00010101	85	U	125	55	01010101
22	SYN	026	16	00010110	86	V	126	56	01010110
23	ETB	027	17	00010111	87	W	127	57	01010111
24	CAN	030	18	00011000	88	X	130	58	01011000
25	EM	031	19	00011001	89	Y	131	59	01011001
26	SUB	032	1A	00011010	90	Z	132	5A	01011010
27	ESC	033	1B	00011011	91	[	133	5B	01011011
28	FS	034	1C	00011100	92		134	5C	01011100
29	GS	035	1D	00011101	93	]	135	5D	01011101
30	RS	036	1E	00011110	94		136	5E	01011110
31	US	037	1F	00011111	95	_	137	5F	01011111

apple computer inc.

DEC	ASCII	OCTAL	HEX	BINARY	DEC	ASCII	OCTAL	HEX	BINARY
32	SP	040	20	00100000	96	`	140	60	01100000
33	!	041	21	00100001	97	a	141	61	01100001
34	"	042	22	00100010	98	b	142	62	01100010
35	#	043	23	00100011	99	c	143	63	01100011
36	$	044	24	00100100	100	d	144	64	01100100
37	%	045	25	00100101	101	e	145	65	01100101
38	&	046	26	00100110	102	f	146	66	01100110
39	'	047	27	00100111	103	g	147	67	01100111
40	(	050	28	00101000	104	h	150	68	01101000
41	)	051	29	00101001	105	i	151	69	01101001
42	*	052	2A	00101010	106	j	152	6A	01101010
43	+	053	2B	00101011	107	k	153	6B	01101011
44	,	054	2C	00101100	108	l	154	6C	01101100
45	-	055	2D	00101101	109	m	155	6D	01101101
46	.	056	2E	00101110	110	n	156	6E	01101110
47	/	057	2F	00101111	111	o	157	6F	01101111
48	0	060	30	00110000	112	p	160	70	01110000
49	1	061	31	00110001	113	q	161	71	01110001
50	2	062	32	00110010	114	r	162	72	01110010
51	3	063	33	00110011	115	s	163	73	01110011
52	4	064	34	00110100	116	t	164	74	01110100
53	5	065	35	00110101	117	u	165	75	01110101
54	6	066	36	00110110	118	v	166	76	01110110
55	7	067	37	00110111	119	w	167	77	01110111
56	8	070	38	00111000	120	x	170	78	01111000
57	9	071	39	00111001	121	y	171	79	01111001
58	:	072	3A	00111010	122	z	172	7A	01111010
59	;	073	3B	00111011	123		173	7B	01111011
60	<	074	3C	00111100	124		174	7C	01111100
61	=	075	3D	00111101	125		175	7D	01111101
62	>	076	3E	00111110	126		176	7E	01111110
63	?	077	3F	00111111	127	DEL	177	7F	01111111