WEST SIDE ELECTRONICS

APPLETIME MODEL APT-1

P.O. BOX 636, CHATSWORTH, CALIFORNIA 91311

1.0 INTRODUCTION

1.1 The APPLETIME Real Time Clock is a peripheral device for the Apple II computer; it plugs directly into any of the peripheral slots on the Apple II motherboard. Once installed, it provides continuous time-of-day information to the computer. This information can be used by any machine language or BASIC program that the user writes.

1.2 Power to operate the clock can come from either the computer's internal power supply, an external supply, or a battery back-up located on or off the board.

2.0 INSTALLATION

2.1 Various options may be selected before installing the APPLETIME in the Apple II. As supplied, all APPLETIME boards use 24 Hour BCD format and crystal timebase. Any of these default conditions can be changed as follows (see Figure 1).

2.2 DATA FORMAT. As each clock digit is read, the information is presented to the computer through the 8 bit data bus. The lower four bits consist of the BCD representation of the particular digit. The upper four bits are programmable via diodes D2-D5. This programming allows the user to select various data formats. In particular, if the diodes are all installed to represent zeros, then the computer will read the clock digits as simple decimal numbers (0-9). If diodes D2 and D3 are connected for ones, then the data will be in the form of the ASCII codes for the characters "0" - "9." Other programming possibilities include 50/60 Hz or 12/24 hour selection indication.

2.3 TIME FORMAT. With no jumper at the 12 HR location, time will be read in a 24 Hour format. This is desirable because it is more informative and easier to program with. If a small jumper wire is installed as indicated, the clock will revert to a 12 Hour display. In this mode there is also leading zero blanking. Therefore, when the first digit is a zero, the software must be able to recognize a BCD 15 (or ASCII "?") as a zero.

2.4 TIMEBASE SELECTION. As normally supplied, the APPLETIME runs off an extremely stable crystal oscillator. This oscillator can be adjusted to provide better than one minute per year accuracy. Adjustment is simplified by placing an accurate frequency counter at point TP and adjusting C2 for a reading of 3.579540 MHz. DO NOT ADJUST TO CRYSTAL FREQUENCY.

If power-line timebase operation is desired, IC5 must be removed and an insulated jumper wire connected between points A and B. Note: If IC5 is removed, it must be protected by placing it in conductive foam or wrapping in aluminum foil. Alternatively, pin 1 could be bent out perpendicular to the other pins and then IC5 replaced into its socket.

2.5 50 Hz SELECTION. If power-line timebase is selected and the APPLETIME is used where the power line frequency is 50 Hz, pin 14 of IC1 must be cut off or bent out away from the socket.

2.6 Once the various options have been selected and installed, the APPLETIME should be plugged into one of the peripheral slots on the Apple II. If the external power supply is used it should be connected and plugged into an AC outlet. A 9V battery should also be connected and secured to the board.

3.0 OPERATION

3.1 The clock information will appear to the computer as ordinary memory. Each digit will reside in a particular location which is dependent upon which slot the APPLETIME is plugged into. The exact addresses are given in Table II. A read instruction (PEEK function in BASIC) to these addresses will return the value of the desired digit. For example, if the APPLETIME is plugged into slot 7 and the BASIC statement X=PEEK (-16137) is executed, then the computer will assign to the variable X the value of the Hours Tens digit. If the time was 10:45:35 and the data format was set for decimal numbers, X would then have the value of 1.

Thus in order to read the clock six separate read instructions must be executed. The results of these interrogations must be stored somehow to reconstruct the entire six digit time format. This can be done by creating either a six digit real number (e.g. 120000) or a six character string ("120000"). Working with six digit numbers in Integer BASIC is a little tricky although if seconds are not needed, four digit numbers are simple to handle. As a character string, the familiar colons (12:00:00) or spaces (12 00 00) are easily added to make the time easier to read.

3.2 Actually, things are not quite that simple. Since the clock chip requires a certain amount of time to switch from one digit to another, we must address the digit we want prior to actually reading its data. This can be done in a straightforward way as shown in the BASIC example below:

```
  1 CALL -936
 10 HT= PEEK (-16217):HT= PEEK (-16217)
 15 HU= PEEK (-16218):HU= PEEK (-16218)
 20 MT= PEEK (-16219):MT= PEEK (-16219)
 30 MU= PEEK (-16223):MU= PEEK (-16223)
 35 ST= PEEK (-16222):ST= PEEK (-16222)
 40 SU= PEEK (-16221):SU= PEEK (-16221)
 90 VTAB 12: TAB 16
100 PRINT HT;HU;":";MT;MU;":";ST;SU
110 GOTO 10
```

APPLETIME
in slot 2

However, there is a little trick that will save a lot of typing and at the same time read the clock twice as fast. This is done by using "anticipated addresses." The above example then becomes:

```
  1 CALL -936
 10 HT= PEEK (-16217):HT= PEEK (-16218)
 15 HU= PEEK (-16219)
 20 MT= PEEK (-16223)
 30 MU= PEEK (-16222)
 35 ST= PEEK (-16221)
 40 SU= PEEK (-16217)
 90 VTAB 12: TAB 16
100 PRINT HT;HU;":";MT;MU;":";ST;SU
110 GOTO 10
```

APPLETIME in slot 2

In the first part of line 10 we address the HT digit as before, giving the clock time to put out the correct data. The second part however addresses the HU digit; but the data read from the clock will be from the last addressed digit (or HT). Therefore the correct data is assigned to the variable HT. When line 15 is executed, the clock is now putting out the HU digit even though we ask for the MT digit. Thus, as we read each clock digit, we use the "anticipated address" of the next digit to be read.

3.3 Setting the clock to the correct time is accomplished with the three switches labelled FAST, SLOW, and HOLD. By running a program similar to the above examples, you can display the time on the screen and then use the setting switches to synchronize to a local time standard. When first setting the clock, use the FAST switch to go through an entire 24 hour period before stopping at the desired hours digits.

4.0 CIRCUIT DESCRIPTION

4.1 The APPLETIME is a hardware digital clock that can be read by the computer. The timekeeping function is handled by IC1, a digital MOS clock chip. Like other clock IC's, this device accepts a 60 Hz timebase signal and performs all of the counting functions required to keep track of hours, minutes, and seconds. Three control inputs (HOLD, FAST SET, and SLOW SET) are used to synchronize the clock to the correct time. Once set, the clock will continue counting as long as its 12 volt power is maintained.

4.2 The 60 Hz timebase for the clock is provided by IC5, an oscillator/divider circuit. When power-line timebase is used, this is replaced by a passive circuit to drive the clock from the 60 Hz power line.

4.3 In order to read the clock, each of the six digits must be examined one at a time. These digits will be referred to as Hours Tens (HT), Hours Units (HU), Minutes Ten (MT), Minutes Units (MU), Seconds Tens (ST), and Seconds Units (SU). Thus a normal time display would appear as:

HT HU : MT MU : ST SU

Each digit is presented by the clock chip as BCD (Binary Coded Decimal) data on its output lines (pins 2-5). Since only one digit's BCD data can be made available at a time, some means is needed to tell the chip which digit to display. Digit select lines DX, DY, and DZ (pins 26, 27, and 28) provide this function. Table I shows which digit is presented according to the status of the digit select lines.

4.4 In order to interface the clock chip to the computer, it is required that the computer be able to direct the clock chip as to which digit to display and then read this information via the data bus. For simplicity, the digit select control is accomplished using the lower three bits of the address bus. This information is latched by IC3 whenever an address within the peripheral slot's DEVICE SELECT area is referenced. Thus each digit is assigned a corresponding address (see Table II) which can be read in the normal fashion.

The device select signal also enables tri-state buffer IC4 to place the clock's output lines on the data bus. The clock chip supplies the lower half-byte while diodes D2-D5 provide selection of the upper half-byte (see section 2.2). IC2a,b, and c provide voltage translation from the TTL circuitry to MOS levels.

TABLE I. Digit select codes.

DIGIT SELECT LINE	DIGIT DISPLAYED HT	HU	MT	MU	ST	SU
DX	1	0	1	1	0	1
DY	1	1	0	0	1	1
DZ	1	1	1	0	0	0

TABLE II. Clock digit locations

SLOT #	DIGIT HT	HU	MT	MU	ST	SU
0	C087 (-16249)	C086 (-16250)	C085 (-16251)	C081 (-16255)	C082 (-16254)	C083 (-16253)
1	C097 (-16233)	C096 (-16234)	C095 (-16235)	C091 (-16239)	C092 (-16238)	C093 (-16237)
2	C0A7 (-16217)	C0A6 (-16218)	C0A5 (-16219)	C0A1 (-16223)	C0A2 (-16222)	C0A3 (-16221)
3	C0B7 (-16201)	C0B6 (-16202)	C0B5 (-16203)	C0B1 (-16207)	C0B2 (-16206)	C0B3 (-16205)
4	C0C7 (-16185)	C0C6 (-16186)	C0C5 (-16187)	C0C1 (-16191)	C0C2 (-16190)	C0C3 (-16189)
5	C0D7 (-16169)	C0D6 (-16170)	C0D5 (-16171)	C0D1 (-16175)	C0D2 (-16174)	C0D3 (-16173)
6	C0E7 (-16153)	C0E6 (-16154)	C0E5 (-16155)	C0E1 (-16159)	C0E2 (-16158)	C0E3 (-16157)
7	C0F7 (-16137)	C0F6 (-16138)	C0F5 (-16139)	C0F1 (-16143)	C0F2 (-16142)	C0F3 (-16141)

Upper number is location in hex
(Lower number is argument for BASIC PEEK)

This program illustrates many other
techniques pertinent to the use of the APPLETIME

```
  1 DIM TIME$(8)
 10 GOTO 500
100 X= PEEK (BASE+7): POKE 2056, PEEK (BASE+6)+176
110 POKE 2057, PEEK (BASE+5)+176
120 POKE 2059, PEEK (BASE+1)+176
130 POKE 2060, PEEK (BASE+2)+176
140 POKE 2062, PEEK (BASE+3)+176
150 POKE 2063, PEEK (BASE+7)+176
190 RETURN
500 CALL -936
510 INPUT "SLOT NUMBER ",S
520 BASE=-16256+16*S
530 POKE 2058,186: POKE 2061,186: POKE 2064,0
540 CALL -936
550 GOSUB 800
600 GOSUB 100
610 VTAB 12: TAB 16: PRINT TIME$
620 GOTO 600
800 VTAB 10: TAB 14: PRINT "************"
810 VTAB 11: TAB 14: PRINT "*          *"
820 VTAB 12: TAB 14: PRINT "*          *"
830 VTAB 13: TAB 14: PRINT "*          *"
840 VTAB 14: TAB 14: PRINT "************"
850 RETURN
```

Line 1	Initializes a string variable TIME$ as the first variable.
Lines 100-190	Subroutine to read the clock and place time in variable TIME$.
Lines 510-520	Input to calculate base address dependent upon APPLETIME slot.
Line 530	Places colons between Hours, Minutes, and Seconds digits.
Line 600-620	Main program to read and display time on screen.

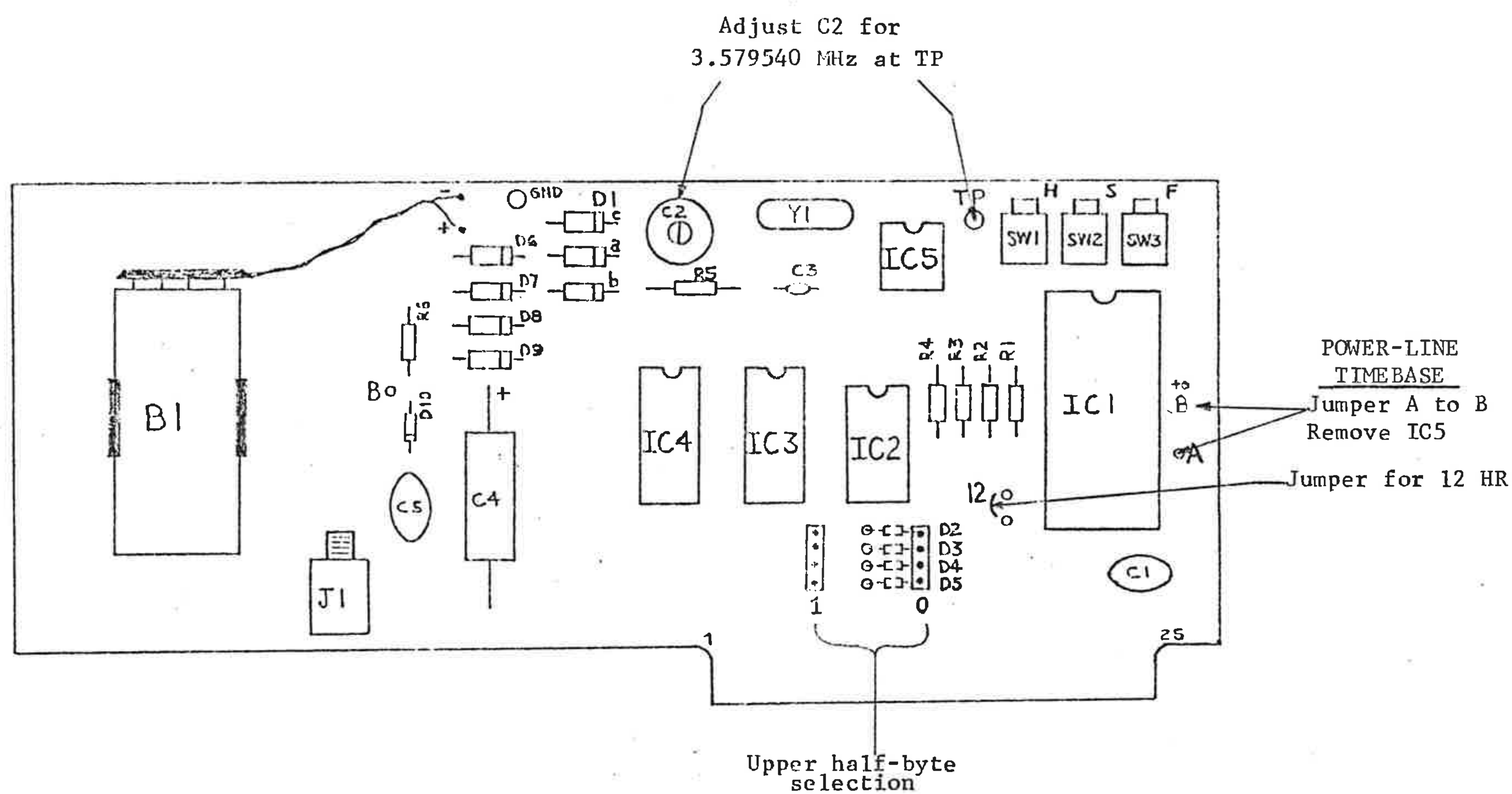
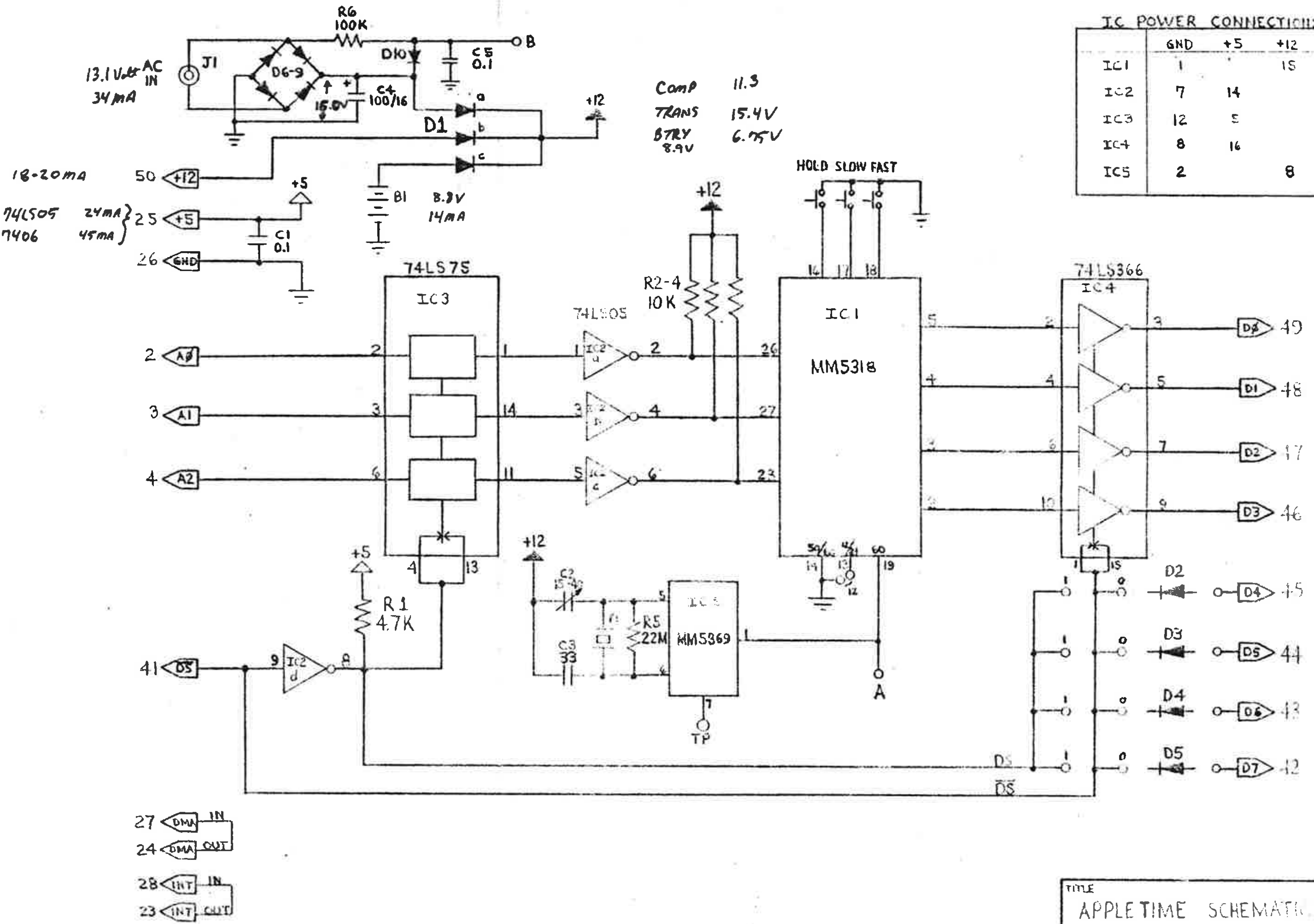


Figure 1. Component placement



IC POWER CONNECTIONS

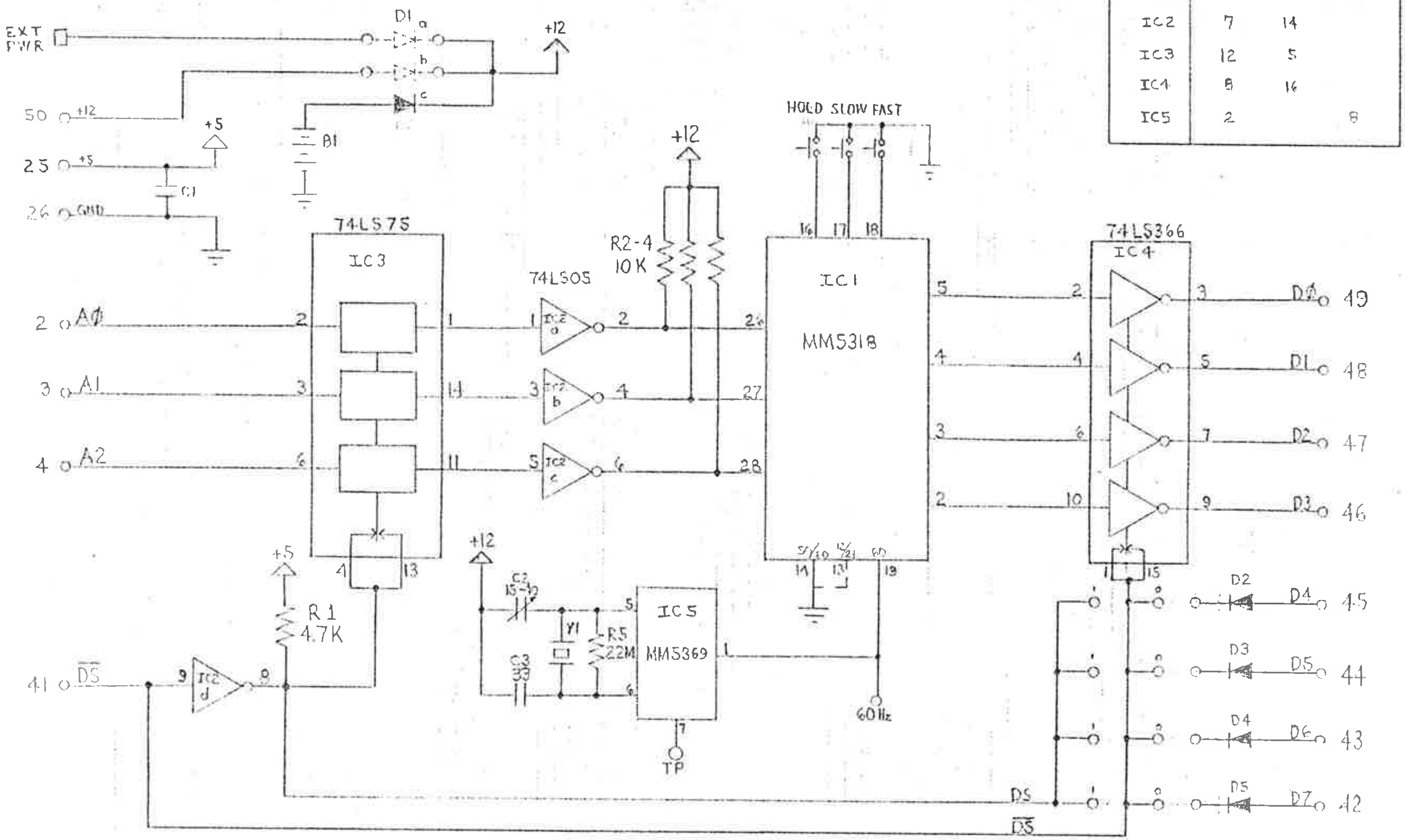
	GND	+5	+12
IC1	1		15
IC2	7	14	
IC3	12	5	
IC4	8	16	
IC5	2		8

TITLE APPLETIME SCHEMATIC

DWG NO AT-100

DATE 7/3/78 DRAWN BY

REV 1: 1/19/79 — POWER CHANGES/AC [illegible]



IC POWER CONNECTIONS

	GND	+5	+12
IC1	1		15
IC2	7	14	
IC3	12	5	
IC4	8	16	
IC5	2		8

TITLE APPLETIME SCHEMATIC

DWG NO. AT-100

DATE 7/3/78 DRAWN BY JM