BASYS/1™

CMOS Single Board Computer

Assembly Manual

Table of Contents

INTRODUCTION

BASYS/1 Single Board Microcomputer

The BASYS/1 Single Board Microcomputer is a versatile, low-cost computer system on a single printed circuit board. It contains a COSMAC microprocessor, RAM and ROM memory, and flexible I/O circuitry for efficient interfacing to a wide range of devices. Features include straightforward design and construction, simple programming, small size, low power consumption, and wide operating temperature and voltage ranges. As a single-board system, BASYS/1 can be programmed for dedicated applications in security systems, energy management, automotive accessories, amateur radio, clock-timers, portable and battery-operated computers, programmable test equipment, and other electronic projects. When combined with additional boards and accessories, BASYS/1 can form the heart of a powerful microcomputer system, comparable in performance to systems costing much more.

SPECIFICATIONS (options given in parenthesis)

Microprocessor:	COSMAC 1802 (or 1804)
RAM Memory:	256 to 2048 bytes static CMOS (or NMOS). Accepts 2111, 2114, 5114, 6514 (UPD444C), or 6561 series RAMs.
ROM/EPROM Memory:	512 to 8192 bytes static CMOS (or NMOS). Accepts 2704, 2708, 2716, 2732, 2764, or 6654 series ROMs or EPROMs.
Serial I/O:	RS-232C, 20mA current loop, or audio FSK

PARALLEL INPUTS:	8 bits, CMOS (or TTL) compatible, with 4.7K pullup resistors
PARALLEL OUTPUTS:	8 bits, CMOS (or TTL) compatible (optional high current/ high voltage drivers or series resistors)
MULTIPLEXED INPUTS:	80 bits, in 8x10 matrix. Accepts switches, phototransistors, hall-effect devices, etc.
MULTIPLEXED OUTPUTS:	80 bits, in 8x10 matrix. Directly drive 10-digit 7-segment displays, etc.
POWER:	4-6VDC @ 20mA max if all CMOS (5VDC ± 5% @ 300mA max if NMOS and TTL is used). If RS232C used, needs ± 5 to ± 12VDC @ 5mA.
BUS CONNECTOR:	44 pins on 0.156" centers. Includes all CPU address, data and control signals
I/O CONNECTOR:	40 pins on 0.100" centers. Accepts discrete wire or ribbon cable.
TEMPERATURE:	-40 to +85°C if all CMOS (0 to +70°C if NMOS,TTL used)
PHYSICAL:	4.5"x6.5", stackable on 0.6" centers. Weight, 6oz

5/17/82

ASSEMBLY NOTES

IMPORTANT—READ BEFORE BEGINNING ASSEMBLY

TOOLS

You will need the following tools to assemble the BASYS/1 kit:

1. pencil soldering iron, 25 to 40 watts
2. diagonal cutters
3. long-nosed pliers

You may also find a magnifying glass and a small screwdriver helpful.

OPTIONS

BASYS/1 includes a large number of assembly options. They allow the board to be customized to meet specific user requirements (micro-power operation, high current output drivers, etc.). If you are building one of the standard configurations, an appropriate parts list and option sheet is included. Follow the step-by-step assembly instructions carefully, noting any special instructions for the version you are building. For special configurations, you may choose the options yourself. In this case, a typical parts list and a blank options sheet are supplied, which you must fill out yourself. In some cases, options may be mutually exclusive; such cases are clearly identified in the assembly instructions.

ASSEMBLY NOTES

IMPORTANT—READ BEFORE BEGINNING ASSEMBLY

ASSEMBLY

Find a clean, well-lighted place to work. Take your time and work carefully. If you get tired or bored, take a break and come back to it later. While assembling the kit is not difficult, it is tedious and exacting, and mistakes can be difficult to correct.

Unpack the parts and check each one against the parts list. Do not discard any packaging material until all of the parts are accounted for. Leave the integrated circuits in their protective foam until actually used, as they are easily damaged by static electricity. Each part is identified by a Key Number, which is used on the circuit board, the circuit diagram, and in documentation. If any parts are missing, damaged, or defective, please contact TMSI and they will by replaced free of charge. When ordering replacement parts, please use the part numbers and descriptions shown.

Read the entire step first, and then follow the instructions carefully. Check your work, and then place a mark in the space provided to the left of each instruction (✓). Be sure to position the parts exactly as shown in the illustrations. Do not solder parts until the instructions specifically tell you to do so.

SOLDERING

Soldering is the most important skill you will need to assemble the kit. Just one bad solder connection can prevent the kit from operating or make it unreliable. Soldering mistakes can also be very difficult to correct. If you have not soldered before, or are unsure of your soldering abilities, find someone with soldering experience to help you.

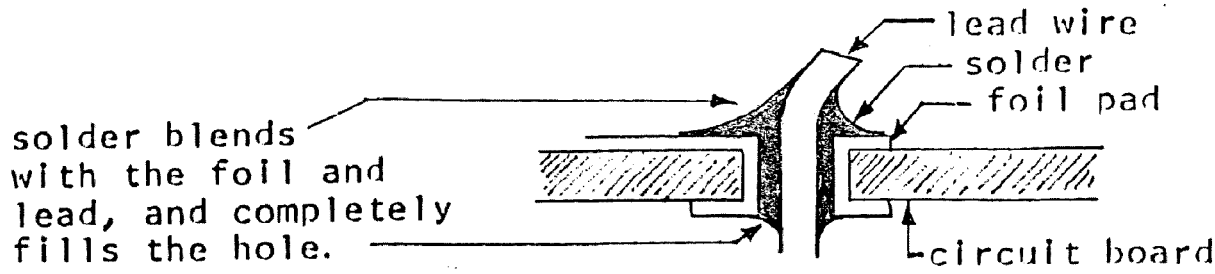
1. Use the right type of soldering iron. It should be rated from 25 to 40 watts and have a chisel or pyramid tip.

2. Keep the soldering iron tip clean. Wipe it occasionally on a damp sponge or cloth and keep it coated with a light coating of solder.

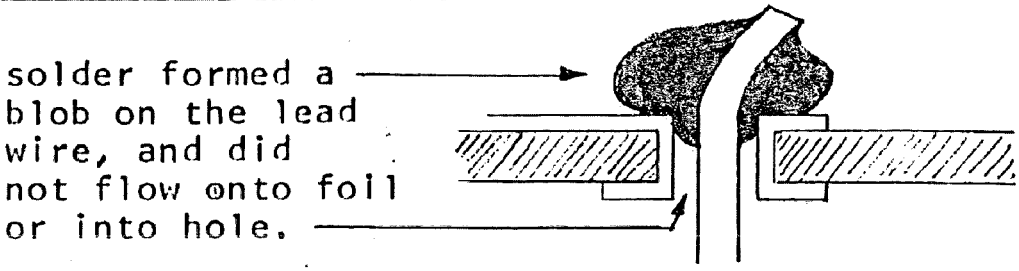
3. Use the solder supplied, or a good quality rosin-core eutectic (60% tin/40% lead) solder. Do NOT use acid core solder.

4. To solder a connection, place the soldering iron tip against the component lead wire AND the foil pad on the circuit board. Apply the solder to the wire and the pad, and NOT to the soldering iron tip itself. When the connection is hot enough, the solder will melt and flow smoothly onto the pad and lead, and into the hole that the lead comes through. Apply just enough solder to make a good connection. Then remove the solder, lift the soldering iron straight up, and let the connection cool.

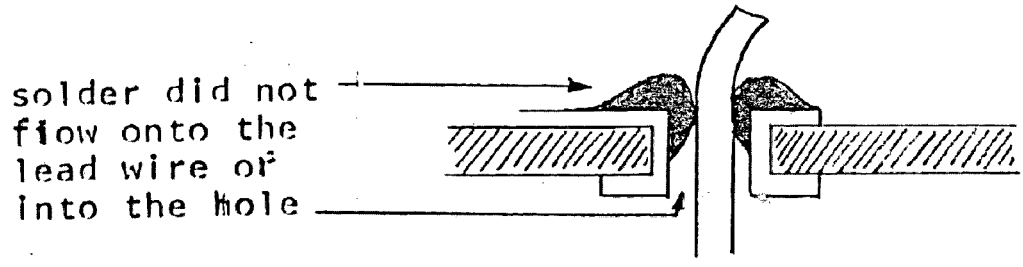
A GOOD SOLDER CONNECTION



POOR SOLDER CONNECTIONS



Too much solder, or too little heat was used. Reheat the connection, and add a little more solder if necessary. Excess solder can be removed by holding the board upside down and letting the excess solder flow down the soldering iron tip.



Too little solder was used, or the lead wire was not heated sufficiently. Reheat the connection and add a little more solder.

5/11/81

UNSOLDERING

If you make a mistake in soldering, use the following proceedures to correct it. Use great care when unsoldering components as it is easy to damage the circuit board from mechanical stress or excessive heat. CAUTION: Heat travels fast. Do not touch a part while you are unsoldering it.

1. Solder bridges - A solder bridge is a short circuit formed by the solder between two adjacent foil pads. They are usually caused by using too much solder or by dragging the tip of the soldering iron across the board when you remove it from a connection. The circuit board is designed so that no two adjacent pads are connected together on the non-component side; thus if you see any such connection, it is a solder bridge. To remove a solder bridge, hold the board with the non-component side down and heat the connection from underneath. The excess solder will melt and flow down the tip of the soldering iron.

2. Resistors, Capacitors, and Diodes - Grasp one of the part's leads with long-nosed pliers and heat the connection with the soldering iron. When the solder melts, gently pull the lead out of the hole. Repeat for the remaining lead. Remove the solder from the hole with the desoldering wick (a fine braided wire) by pressing it against the foil with the tip of the soldering iron. When the solder melts, it will be drawn out of the hole and into the desoldering wick.

3. Transistors and Integrated Circuits not in Sockets - Do NOT try to unsolder these parts intact. It is not worth damaging an expensive circuit board to save an inexpensive part. Replacement parts are readily available from TMSI and other electronic component suppliers. Remove the part by cutting its leads with diagonal cutters as close to the case as possible. The remaining stubs can then be safely removed as described in 2. above.

4. IC Sockets - High quality sockets are supplied for all expensive ICs where their cost is justified. You can replace a single pin or the entire socket using the following method:

   a. Single Pin - Heat the connection and push on the pin with the tip of the soldering iron. When the solder melts the pin will be freed from the plastic body of the socket and can be pushed out with the soldering iron. Grasp the pin with long-nosed pliers and pull it the rest of the way out. Remove the solder from the hole as described in 2. above. A new pin can then be pressed into place and re-soldered.

   b. Entire Socket - Do NOT try to unsolder the socket intact. It is not worth damaging an expensive circuit board to save a socket. Remove the socket by unsoldering each pin as described in 4a. above. Replacement sockets are available from TMSI and other electronic component suppliers. NOTE: Use high quality sockets or the reliability may be reduced, particularly in harsh environments.

5/11/81

OPTIONS SHEET

BASYS/1 - _____

RAM Address Decoding

☐ Starting Address _____ (hex)

XD2

ROM/EPROM Address Decoding

☐ Starting Address _____ (hex)

XD1

ROM/EPROM Memory Options

☐ 512 bytes using _____
☐ 1024 bytes using _____
☐ 2048 bytes using _____
☐ 4096 bytes using _____
☐ 8192 bytes using _____

X5

-5 VDC Bypass Capacitors

☐ used
☐ not used

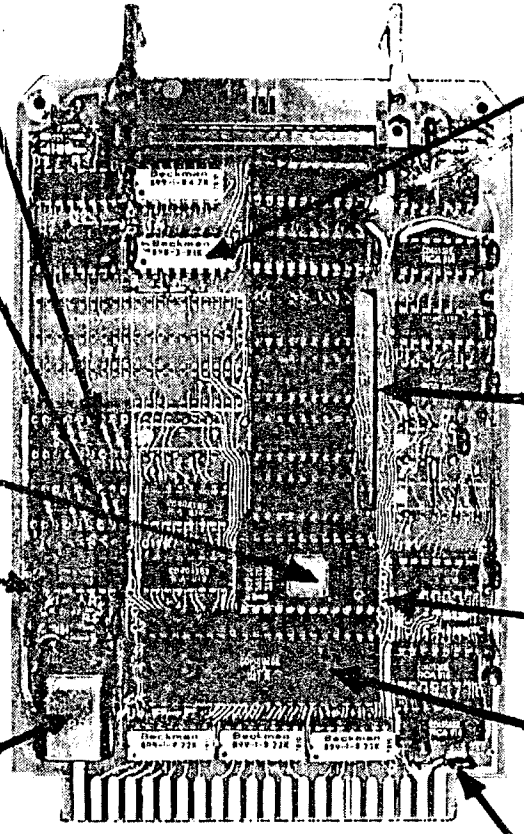
System Clock Options

☐ Crystal Oscillator
freq. _____

☐ RC Oscillator
freq. _____
R2: _____
C3: _____

☐ Schmitt-Trigger Oscillator
freq. _____
R1: _____
C1: _____

☐ External Clock
(jumper X1 open)



Output Port Options

☐ Direct Drive: jumpers at U8
☐ Series Resistors at U8: _____ ohms
☐ Power Driver at U8: _____

RAM Memory Options

☐ 256 bytes using 256x4 RAM type _____
☐ 512 bytes using 256x4 RAM type _____
☐ 1024 bytes using 1024x4 RAM type _____
☐ 2048 bytes using 1024x4 RAM type _____

X6 X3

RAM Write Control Option

☐ RAM Writing Enabled
☐ RAM Write controlled by an external switch

Microprocessor Options

☐ 1802CE (4-6 VDC)
☐ 1802E (3-12 VDC)
☐ 1804
☐ none (I/O, memory board)

+12 VDC Bypass Capacitors

☐ used
☐ not used

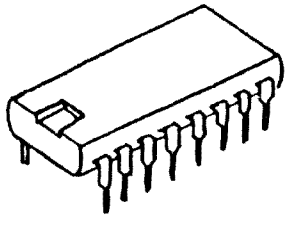
CMOS Logic Option

__ all-CMOS (4-6 VDC, -40 to +85C)
__ mixed technology (5 VDC ± 5%, 0 to 70C). Non-CMOS ICs:

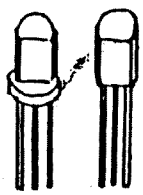
_____ _____ _____

_____ _____ _____

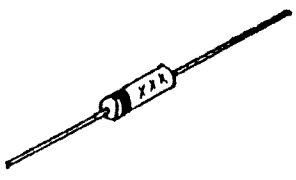
QTY	PART NO.	DESCRIPTION		KEY NO.
		INTEGRATED CIRCUITS -- The part number shown will be on the integrated circuit, but it may also have one or more letters or numbers as prefixes or suffixes. For example, a 4011B may be marked "MC14011BCP". Do not be concerned about the extra letters or numbers.		
1	1802CE	COSMAC CPU, 4-6V	(40-pin)	U5
1	[redacted]	EPROM, [redacted]	[redacted]	U11
4	6514	RAM, 1Kx4, CMOS	(18-pin)	U12-15
1	4001B	quad 2-input NOR, CMOS	(14-pin)	U22
2	4011B	quad 2-input NAND, CMOS	(14-pin)	U21,23
2	4013B	dual D flip-flop, CMOS	(14-pin)	U20,24
1	4020B	14-bit binary counter, CMOS	(16-pin)	U2
2	4042B	4-bit latch, CMOS	(16-pin)	U6,7
1	4069	hex inverter, CMOS	(14-pin)	U1
1	4556B	dual 1-of-4 decoder, CMOS	(16-pin)	U19
1	4N35	optocoupler, transistor	(6-pin)	U26
1	4N30	optocoupler, darlington	(6-pin)	U25
1	74C42	BCD-decimal decoder, CMOS	(16-pin)	U3
2	74C373	8-bit latch, CMOS	(20-pin)	U16,17
1	3-R470	8x470 ohm resistor network	(16-pin)	U8
1	1-R4.7K	1x4.7K ohm resistor network	(14-pin)	U9
3	1-R22K	13x22K ohm resistor network	(14-pin)	U4,10,18
		TRANSISTORS AND DIODES		
1	2N4403	transistor, PNP, signal		Q1
18	1N914	diode, switching		D1-18



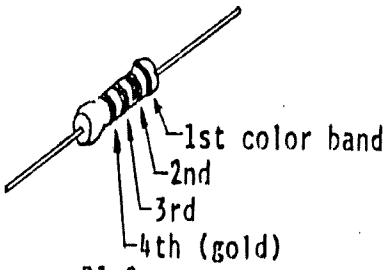
U1-26



Q1



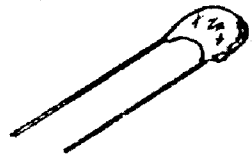
D1-18



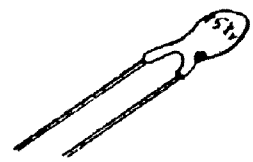
R1-8

PARTS LIST -- BASYS/1K-3F

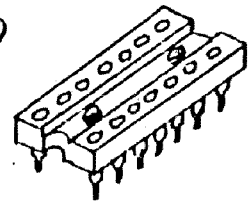
QTY	PART NO.	DESCRIPTION	KEY NO.
	RESISTORS - All are 5% (4th band gold), 1/4 watt		
1	221R5.25	220 ohms (red-red-brown)	R7
1	102R5.25	1.0K ohms (brown-black-red)	R5
2	182R5.25	1.8K ohms (brown-gray-red)	R2,8
2	272R5.25	2.7K ohms (red-violet-red)	R4,6
1	224R5.25	220K ohms (red-red-yellow)	R3
1	106R5.25	10 megohms (brown-black-blue)	R1
	CAPACITORS		
2	200C103	20pF, disc ceramic, NPO, 1KV	C1,2
16	104C250	0.1uF, monolithic ceramic, 25V	C4,5,7-11, 13,15-22
3	156CT250	15uF, tantalum, 16VDC	C6,12,14
	SOCKETS AND CONNECTORS		
1	IC16.3	16-pin IC socket, gold, hi-rel.	JU3
7	IC18.3	18-pin IC socket, gold, hi-rel.	JU8,12-15, JXD1,2
2	IC20.3	20-pin IC socket, gold, hi-rel.	JU16,17
1	IC24.6	24-pin IC socket, gold, hi-rel.	JU11
1	IC40.6	40-pin IC socket, gold, hi-rel.	JU5
1	P40.1RE	40-pin header, rt.angle, ejector	P2
	MISCELLANEOUS		
1	SBC100-X3	PC board for BASYS/1	PCB1
1	2.000MHz	crystal	Y1
6"	--	desoldering wick	
60"	--	solder, eutectic, rosin core	



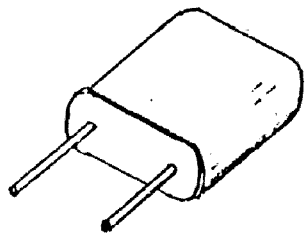
C1-5,7-11, 13,15-22



C6,12,14

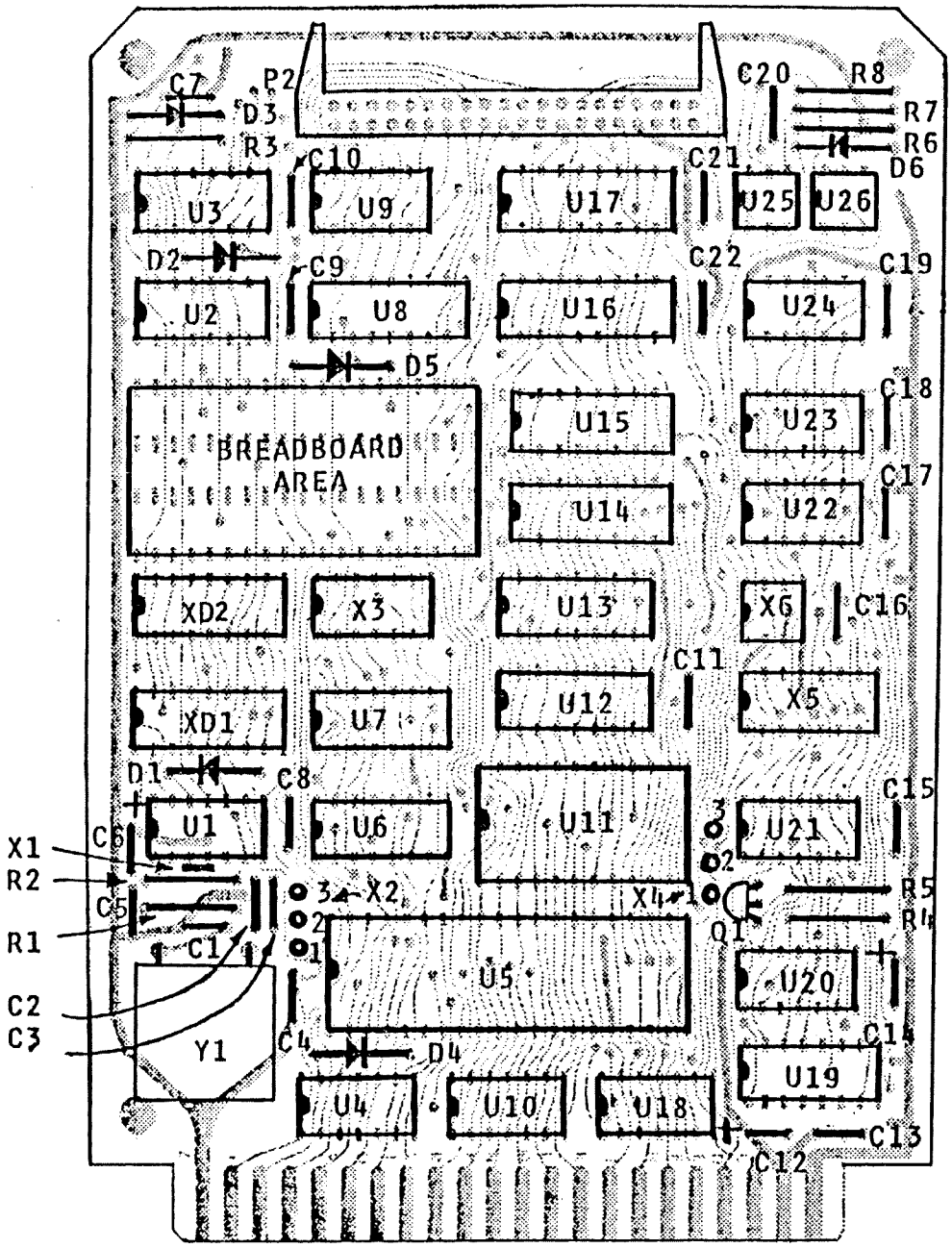


JU3-17



Y1

5/11/81



COMPONENT LAYOUT
SBC100-X3

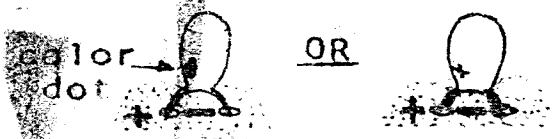
START:

+5 VDC BYPASS CAPACITORS

In each of the following steps, insert a capacitor at the location specified. Be sure that the leads of the capacitor go in the holes at each end of the heavy line printed on the board.

( ) C4: 0.1uF ceramic capacitor
( ) C8: 0.1uF ceramic capacitor
( ) C9: 0.1uF ceramic capacitor
( ) C10: 0.1uF ceramic capacitor
( ) C11: 0.1uF ceramic capacitor
( ) C15: 0.1uF ceramic capacitor
( ) C16: 0.1uF ceramic capacitor
( ) C17: 0.1uF ceramic capacitor
( ) C18: 0.1uF ceramic capacitor
( ) C19: 0.1uF ceramcl capacitor
( ) C21: 0.1uF ceramic capacitor
( ) C22: 0.1uF ceramic capacitor

( ) C14: 15uF tantalum capacitor. Position C14 with its positive (+) lead in the hole marked (+) as shown below:



X1
R2
R1
C2
C3

( ) Solder all the leads to the foil and cut off the excess lead-length. Save the cut leads as they will be used later.

CAUTION: Hold the leads when you cut them so they don't fly toward your eyes.

CONTINUE:

-5 VDC BYPASS CAPACITORS (Optional)

BASYS/1 is normally used with a single +5 VDC power supply. Perform the following steps if you will be using an additional -5 VDC supply:

( ) C5: 0.1uF ceramic capacitor

( ) C6: 15uF tantalum capacitor
Position C6 with its positive lead in the hole marked (+).

( ) Solder the leads to the foil and cut off the excess lead length.

+12 VDC BYPASS CAPACITORS (Optional)

BASYS/1 is normally used with a single +5 VDC power supply. Perform the following steps if you will be using an additional +12 VDC supply:

( ) C13: 0.1uF ceramic capacitor

( ) C12: 15uF tantalum capacitor
Position C12 with its positive lead in the hole marked (+).

( ) Solder the leads to the foil and cut off the excess lead length.

START:

IC SOCKETS AND HEADERS

To insure reliable operation in harsh environments, BASYS/1 uses high quality sockets for all key integrated circuits (CPU, memory, and I/O) and for frequently used jumpers. Sockets are not provided for the remaining ICs. If you wish to provide sockets for these ICs, they should be of similar quality, or reliability may be seriously lowered in harsh environments.

( ) PCB1: Position the PC board as shown, with the silkscreened side up. All parts will be mounted on this side, and all soldering will be done on the back side.

( ) JU5: 40-pin IC socket at U5 Insert the socket with the orientation mark on the left (matching the dot on the silkscreen). If necessary, bend the two corner pins slightly to hold the socket on the board until it is soldered. Then solder all pins to the foil.

CONTINUE:

Mount the following sockets using the same instructions used for JU5:

( ) JU11: 24-pin IC socket at U11

( ) JU12: 18-pin IC socket at U12

(✓) JU14: 18-pin IC socket at U14

( ) JU13: 18-pin IC socket at U13 (optional)

( ) JU15: 18-pin IC socket at U15 (optional)

(✓) JU16: 20-pin IC socket at U16

( ) JU17: 20-pin IC socket at U17

(✓) JU8: 18-pin IC socket at U8

(✓) JU3: 16-pin IC socket at U3

(✓) JXD1: 18-pin IC socket at XD1

( ) JXD2: 18-pin IC socket at XD2

( ) P2: 40-pin right-angle header

( ) Check to see that all of the pins of each socket are soldered and that no solder shorts or "bridges" are present between adjacent pins.

11/11/81

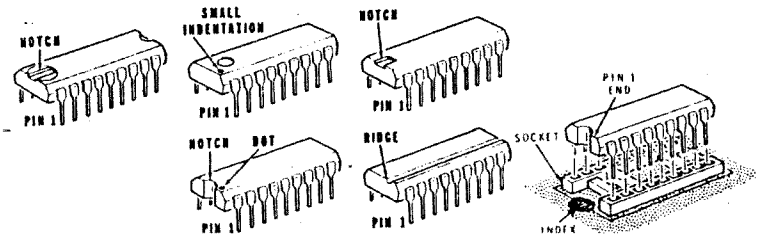
START:

INTEGRATED CIRCUITS

NOTE: The integrated circuits used in BASYS/1 are rugged and reliable components. However, they can be easily damaged by static electricity. Read thru the entire proceedure first, and then perform each step EXACTLY as indicated for each IC. The least sensitive parts will be installed first.

(√) U4: 13x22K resistor network BECKMAN R22K

1. Remove the IC from the protective foam and hold it in one hand.
2. Straighten any bent leads with your other hand. The leads should be parallel to each other and at right angles to the case.
3. Continue holding the IC in one hand, and pick up the circuit board in your other hand.
4. Install the IC at V4 with pin 1 at the index mark as shown below. Press it into its socket or solder it as appropriate.



CONTINUE:

Repeat this proceedure for each of the following ICs:

( ) U10: 13x22K ohm resistor network

( ) U18: 13x22K ohm resistor network

( ) U9: 13x4.7K ohm resistor network

( ) U2: 4020 integrated circuit

( ) U3: 74C42 or 74LS145

( ) U6: 4042

( ) U7: 4042

( ) U19: 4556

( ) U20: 4013

( ) U21: 4011

( ) U22: 4001

( ) U23: 4011

( ) U24: 4013

( ) U17: 74C373 or 74LS373

( ) U16: 74C373 or 74LS373

( ) Check to see that every pin of each IC is properly soldered, or properly seated in its socket. Be sure there are no solder shorts or bridges between adjacent pins.

3/12/81

START:

SYSTEM CLOCK OPTIONS

The system clock can be obtained from any one of four sources: 1. external clock, 2. RC oscillator, 3. Schmitt-trigger oscillator, or 4. crystal oscillator.

( ) Check the Option sheet to see which system clock was supplied with your kit. Then perform the steps described below for that option:

1. EXTERNAL CLOCK
The external clock input is on pin Y of the bus connector P1.

   ( ) C1: jumper wire (use scrap component lead)
   ( ) U1: 4069 or 74C14
   Caution: Static sensitive. Follow IC installation instructions on previous page.
   ( ) R1, R2, C2, C3, Y1, and X1 are not used

2. RC OSCILLATOR
frequency set by R2,C3

$$f(KHz) = \frac{1}{1.4R_2(Kohms)C_3(uF)}$$

   ( ) R2: 1K to 10 megohms
   ( ) R1: 2x to 10x value of R2
   ( ) C3: 300pF to 1uF, non-polar
   ( ) X1: jumper wire (use scrap component lead)
   ( ) U1: 4069
   Caution: Static sensitive. Follow IC installation instructions on previous page.
   ( ) C1, C2, and Y1 are not used

CONTINUE:

3. SCHMITT-TRIGGER OSCILLATOR
frequency set by R1 and C1

$$f(KHz) = \frac{1}{1.7\ R_1(Kohms)\ C_1(uF)}$$

( ) R1: 1K to 10 megohms
( ) C1: any value (no limits)
( ) X1: jumper wire (use scrap component lead)
( ) R2: jumper wire (use scrap component lead)
( ) U1: 74C14
Caution: Static sensitive. Follow IC installation instructions on previous page.
( ) C2, C3, and Y1 are not used

4. CRYSTAL OSCILLATOR
frequency is set by crystal Y1

( ) R1: 10 megohms (brown-blk-blue)
( ) R2: 1.8K ohms (brown-gray-red)
( ) C1: 20pF NPO ceramic capacitor
( ) C2: 20pF NPO ceramic capacitor
( ) Y1: crystal
( ) hold-down wire for Y1 (use scrap component lead)
( ) X1: jumper wire (use scrap component lead)
( ) U1: 4069
Caution: Static sensitive. Follow IC installation instructions on previous page.
( ) C3 is not used

( ) Solder all leads to the foil and cut off the excess lead length.

START:

RESET TIMER

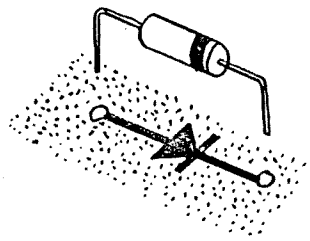
The reset timer resets the computer approximately 20mSec. after power is first applied. The time delay must be longer than the time it takes for the +5 volt power supply to turn on. If your power supply takes more than 20mSec. to stabilize, the value of capacitor C7 may be increased as necessary.

( ) C7: 0.1uF ceramic capacitor

( ) R3: 220K ohms (red-red-yellow)

( ) D3: 1N914 diode (1N4148)

Install the diode with the banded end matching the symbol as shown below:



( ) D1: 1N914 (1N4148)
Install as shown above.

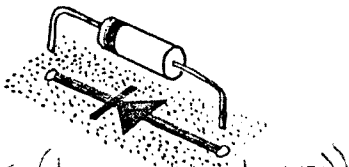
( ) Solder all leads to the foil and cut off the excess lead length.

SERIAL INTERFACE

The serial interface provides an RS-232C, 20mA current loop, or FSK audio I/O for serial communications.

( ) C20: 0.1uF ceramic capacitor

( ) D6: 1N914 diode (1N4148)
Install the diode with the banded end matching the symbol as shown:



1K ohms (brown-black-red)

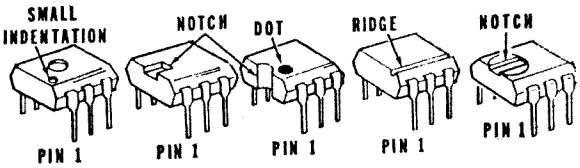
( ) R5: 220 ohms (red-red-brown)

( ) R6: 2.7K ohms (red-violet-red)

( ) R7: 220 ohms (red-red-brown)

( ) R8: 1.8K ohms (brown-gray-red)

(✓) U25: 4N30 or 4N32 IC
Install U25 with pin 1 at the index mark as shown below:



(✓) U26: 4N35 integrated circuit

(✓) Q1: 2N4403 transistor
Install Q2 with the case about 1/8" above the PC board and with the flat side of the case as shown on the silkscreen.

( ) Solder all leads to the foil and cut off the excess lead length.

9/24/81 8/10/82

START:

ROM MEMORY OPTIONS

ROM socket U11 can be wired to accept ROM and EPROM integrated circuits from 512 to 8192 (8K) bytes via jumper wires at X5. Note that some older EPROMs will require +12 VDC and -5 VDC power supplies in addition to the +5 VDC normally used by BASYS/1. If these parts are used, the optional +12 VDC and -5 VDC bypass capacitors must be installed as well.

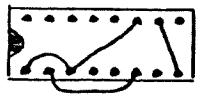
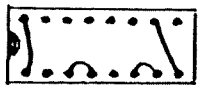
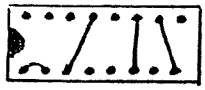
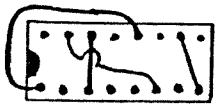
( ) Locate the ROM or EPROM IC, and find the corresponding jumper diagram for X5 on this page. Do not remove the IC from its protective foam yet.

( ) X5: Insert jumper wires at location X5 as shown for your ROM or EPROM IC (use scrap component leads). A 16-pin IC socket (not supplied) may be used at X5 if you expect to change the jumper configuration.

( ) Solder each jumper to the foil and cut off the excess lead length.

( ) U11: ROM or EPROM IC Insert the IC into its socket using the same procedure described earlier for the other integrated circuits.

X5 JUMPER CONFIGURATIONS

size	jumpers	EPROMs	ROMs	notes
512x8		2704 8704	2304 8304	1,3 1,3
"		6604 6654	 1832 CDPR512	2 2 3 3
1Kx8		2708 8708	2308 8308 2608	1 1 1
"		2758 27C58	 1834	
2Kx8		2716 (except 2516	TMS2716 2316	
4Kx8		2732	2332	
8Kx8		68764	8364 6388	 2

Notes:

1. Requires +12 VDC and -5 VDC power supplies and bypass capacitor options in addition to +5 VDC.

2. Has latched addresses, and requires an additional 1N914 diode(D20). Install it as shown at left, using insulating sleeving on each lead. Solder the leads to the plated-through holes.

3. Partially decoded: Will occupy a 1K

START:

RAM MEMORY OPTIONS

BASYS/1 can be assembled with 256 to 2048 bytes of RAM, using either 256x4 or 1024x4 bit RAMs. A write-protect option is provided to disable RAM with an external switch.

( ) X4: Check the option sheet and perform ONE of the following:

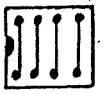
( ) RAM Writing Enabled:
Install a jumper wire between pins 2 and 3 of X4 (use scrap component lead).

( ) RAM Writing Controlled by an External Switch:
Install a jumper wire between pins 1 and 2 of X4 (use scrap component lead).

( ) Check the Option sheet to see which RAM ICs were supplied with your kit. Then perform ONE of the following steps (A or B):

A. RAM part number 2111 or 6561 (256x4) to provide 256 or 512 bytes of RAM memory:

( ) X6: Insert 4 jumper wires at location X6 as shown below (use scrap component leads).



X6 for 256x4 RAMs

( ) X3: Insert 5 jumper wires at location X3 as shown below, (use scrap component leads).

X3 for 256x4 RAMs

CONTINUE:

( ) Solder each jumper to the foil and cut off the excess lead length. Do not solder the unused holes in location X3.

( ) U12: 2111 or 6561
Caution: Static sensitive part. Follow the IC installation instructions given previously.

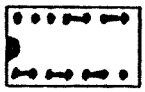
( ) U14: 2111 or 6561
( ) U13: 2111 or 6561 (optional)
( ) U15: 2111 or 6561 (optional)

B. RAM part number 2114L, 5114, or 6514 (UPD444C) (1024x4) to provide 1024 or 2048 bytes of RAM:

( ) X6: Insert 4 jumper wires at location X6 as shown below (use scrap component leads).

X6 for 1024x4 RAMs

( ) X3: Insert 5 jumper wires at location X3 as shown below (use scrap component leads).



X3 for 1024x4 RAMs

( ) Solder each jumper to the foil and cut off the excess lead length.

( ) U12: 2114L, 5114, or 6514(UPD444C)
Caution: Static sensitive part. Follow the IC installation instructions given previously.

( ) U14: 2114L, 5114, or 6514(UPD444C)
( ) U13: 2114L,5114,or 6514(UPD444C)(opt
( ) U15: 2114L,5114,or 6514(UPD444C)(opt

START:

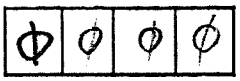
MEMORY ADDRESS DECODING

ROM and RAM are located in the 64K memory map by the placement of diode jumpers in sockets XD1 and XD2 respectively. ROM and RAM addresses are fully decoded and can be independently located at any multiple of the memory size used. For example, a 2Kx8 ROM can be located at 0K, 2K, 4K, 6K, etc.

XD1: ROM Address Decoding

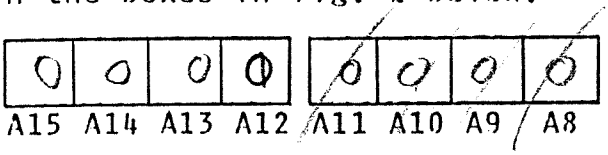
( ) Determine the ROM starting address (a 4-digit hexadecimal number) from the option sheet, and write it in the boxes of fig. 1 below. If no address is specified, use 0000.

FIG. 1



( ) Convert the first two (leftmost) digits of the address into binary, using the Hexadecimal-to-Binary Table provided. Write the result in the boxes in fig. 2 below.

FIG.2



( ) Each box in fig. 2 (containing "1" or "0") represents an address bit (A15-A8). Some of these bits are decoded directly by the ROM, and so are not decoded here. Find the ROM size you are using in fig. 3 and cross out the indicated address bits in fig. 2.

FIG. 3

ROM size:	Cross out boxes:
512x8	A8,A9
1Kx8	A8,A9
2Kx8	A8,A9,A10
4Kx8	A8,A9,A10,A11
8Kx8	A8,A9,A10,A11,A12

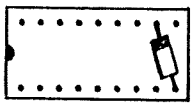
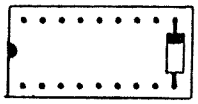
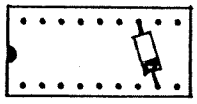
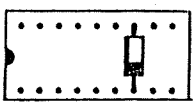
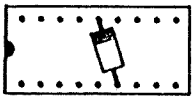
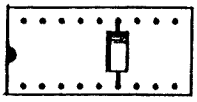
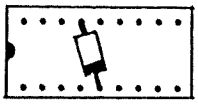
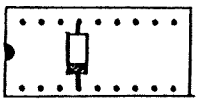
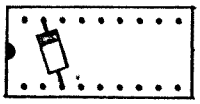
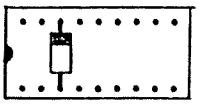
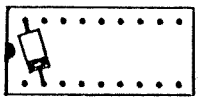
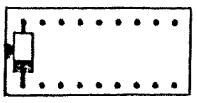
Hexadecimal-to-Binary Table

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

CONTINUE:

In the following steps a diode will be installed in XD1 for each remaining address bit in fig. 2. The diode will be placed in either of two positions, depending on the value ("1" or "0") of the address bit specified in each step. No diode is installed for address bits that were crossed out in the previous step.

Cut the leads of each diode to 1/4" long and bend them by hand to fit into socket XD1 as shown in the drawing for each step. Be careful not to bend the leads too sharply or too close to the case of the diode or it may break. Then carefully insert the diode into the socket exactly as shown, noting the position of the banded end.

Install at XD1 if A-- is:	=0:	=1:	crossed out:
( ) D7: 1N914 if A15 is:			
( ) D8: 1N914 if A14 is:			
( ) D9: 1N914 if A13 is:			
( ) D10: 1N914 if A12 is:			D10 not used
( ) D11: 1N914 if A11 is:			D11 not used
( ) D12: 1N914 if A10 is:			D12 not used

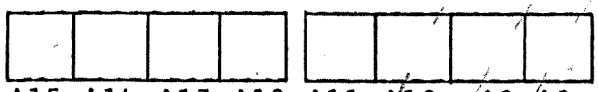
START:

XD2: RAM Address Decoding

( ) Determine the RAM starting address (a 4-digit hexadecimal number) from the Options page, and write it in the boxes in fig. 4 below. If no address is specified, use 0800 (this will place the RAM just above a 2K ROM at address 0000).

FIG. 4 

( ) Convert the first two (leftmost) digits of fig. 4 into binary, using the Hexadecimal-to-Binary Table on the previous page. Write the result in the boxes in fig. 5.

FIG. 5 

( ) Each box in fig. 5 represents an address bit. Some of these bits are decoded elsewhere, and so are not used here. Perform ONE of the following:

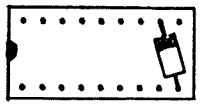
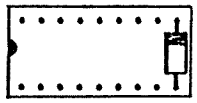
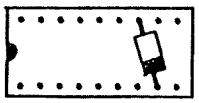
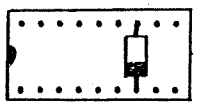
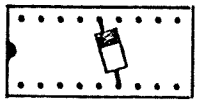
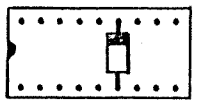
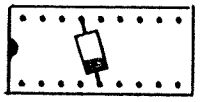
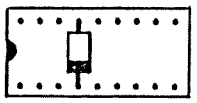
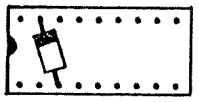
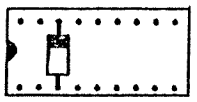
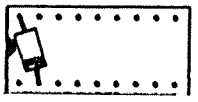
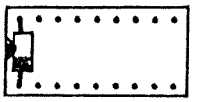
- If 256x4 RAMs are used, cross out box A8 in fig. 5.
- If 1Kx4 RAMs are used, cross out boxes A8,A9, & A10.

In the following steps a diode will be installed in XD2 for each remaining address bit in fig. 5 (except for A9 which is covered separately). The diode will be placed in either of two positions, depending on the value ("1" or "0") of the address bit specified in each step. No diode is installed for address bits that were crossed out in the previous step.

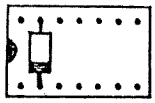
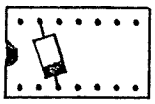
Cut the leads of each diode to 1/4" long and bend them by hand to fit into socket XD2 as shown in the drawing for each step. Be careful not to bend the leads too sharply or too close to the case of the diode or it may break. Then carefully insert the diode into the socket exactly as shown, noting the position of the banded end.

CONTINUE:

Install at XD2

if A-- is:	=0:	=1:	crossed out:
( ) D13: 1N914 if A15 is:			
( ) D14: 1N914 if A14 is:			
( ) D15: 1N914 if A13 is:			
( ) D16: 1N914 if A12 is:			
( ) D17: 1N914 if A11 is:			
( ) D18: 1N914 if A10 is:			D18 not used

Using the same proceedure described above, install diode D19 at location X3 as shown in the following drawings. The position of D19 is determined by the value ("1" or "0") of address bit A9 in fig. 5. Do not install D19 if box A9 is crossed out.

Install at X3:	if A9=0:	if A9=1:	crossed out:
( ) D19: 1N914			D19 not used

Solder both leads to the foil and cut off the excess lead length.

3/12/81

START:

MICROPROCESSOR

BASYS/1 can be assembled with either the COSMAC 1802 or 1804 microprocessor, or with no microprocessor at all (operating as an expansion memory or I/O board for a second BASYS/1 board). Additionally, the 1802 can be operated with either one or two power supply voltages.

( ) Check the Option sheet to see which microprocessor was supplied with your kit. Then perform the steps described below for that option:

1. 1802 microprocessor with one power supply voltage:

   (✓) U5: 1802CD or 1802CE
   Caution: Static sensitive. Follow IC installation instructions given earlier.

   (✓) X2: jumper wire
   Install a jumper wire (use scrap component lead) between holes 2 and 3 of X2. Solder both ends to the foil and cut off the excess lead length.

CONTINUE:

2. 1802 microprocessor with two power supply voltages:

   ( ) U5: 1802D or 1802E
   Caution: Static sensitive. Follow IC installation instructions given earlier.

   ( ) X2: jumper wire
   Install a jumper wire (use scrap component lead) between holes 1 and 2 of X2. Solder both ends to the foil and cut off the excess lead length.

3. 1804 Microcomputer:

   ( ) U5: 1804
   Caution: Static sensitive. Follow IC installation instructions given earlier.

   ( ) X2 is not used

4. No Microprocessor (Memory or I/O Expansion)

   ( ) U5: jumper wire
   Install a jumper wire (use scrap component lead) between pins 1 and 39 of IC socket U5. Pin 1 is in the lower leftmost corner of U5, and pin 39 is the second one from the upper left corner.

   ( ) X2 is not used

START:

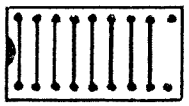
OUTPUT PORT OPTIONS

The parallel output port can be assembled to drive several different types of devices: 1. direct drive for TTL and CMOS ICs, 2. series resistors for driving LEDs and transistors, or 3. high power driver ICs for driving motors, solenoids, high-voltage displays, etc. The output port is configured at location U8. An 18-pin IC socket (not supplied) may be used at U8 if you expect to change options.

( ) Check the Option sheet to see which output port parts were supplied with your kit (the direct drive option uses no parts and so can always be used). Then select ONE of the following steps (1, 2, or 3):

1. Direct Drive Outputs

   ( ) U8: Install 8 jumper wires at location U8 as shown below (use scrap component leads). No jumper is installed between the rightmost pair of pins.



   ( ) Solder each jumper to the foil and cut off the excess lead length.

CONTINUE:

2. Outputs with Series Resistors

x( ) U8: 8x470 ohm resistor network. Install U8 in the leftmost end of location U8. The 2 holes at the right end of location U8 will not be used.

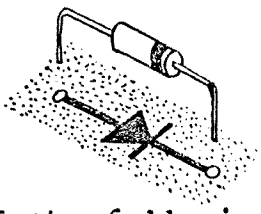
( ) Solder the leads to the foil.

3. Outputs with High Power Driver

( ) U8: 2813, 2983, or 6118 Follow the IC installation instructions given previously.

MISCELLANEOUS

(✓) R4: 2.7K ohms (red-violet-red)



In each of the following steps, install the diode with the banded end matching the symbol as shown above:

( ) D4: 1N914 (1N4148)
( ) D5: 1N914 (1N4148)
( ) D2: 1N914 (1N4148)

( ) Solder all leads to the foil and cut off the excess lead length.

11/11/81
8/10/82

START:

This completes the assembly of BASYS/1. The remaining holes do not need to be soldered, but you may do so now if desired for increased reliability. Be careful not to solder the four rows of holes in the breadboard area, or any holes for options you may later wish to install.

VISUAL INSPECTION

Carefully inspect the board for errors, and correct any that you find:

( ) unsoldered connections

( ) poor solder connections

( ) solder bridges between connections

( ) long leads which could bend and short to other connections

( ) be sure the tantalum capacitors are installed with the (+) lead in the correct hole

( ) check the integrated circuits for proper type and installation. Pin 1 on all ICs is to the left

( ) check transistor Q1 to see that it is installed correctly

FINISH

THEORY OF OPERATION

INTRODUCTION

The following section describes the circuitry and theory of operation for BASYS/1, revisions -X1 through -X3. A knowledge of basic electronics and the fundamentals of digital logic is assumed. Since BASYS/1 supports a large number of specialized options, each circuit is first described for a "typical" configuration. This is then followed by a selection of alternative configurations, each with its benefits and drawbacks. The circuit descriptions cover both "how" things are done, as well as "why" they are done this way, to aid the user in understanding the system and in optimizing it for specific applications.

The circuitry on BASYS/1 can by divided into three functional blocks (see fig.1): CPU, Memory, and I/O. Together, they form a complete microcomputer, requiring only an external power supply and input and output devices. The three blocks communicate with each other via a group of signal lines known as a Bus, which is also routed to edge connector P1 for expansion. Additional memory and I/O boards connected to the bus appear exactly like on-board memory and I/O.

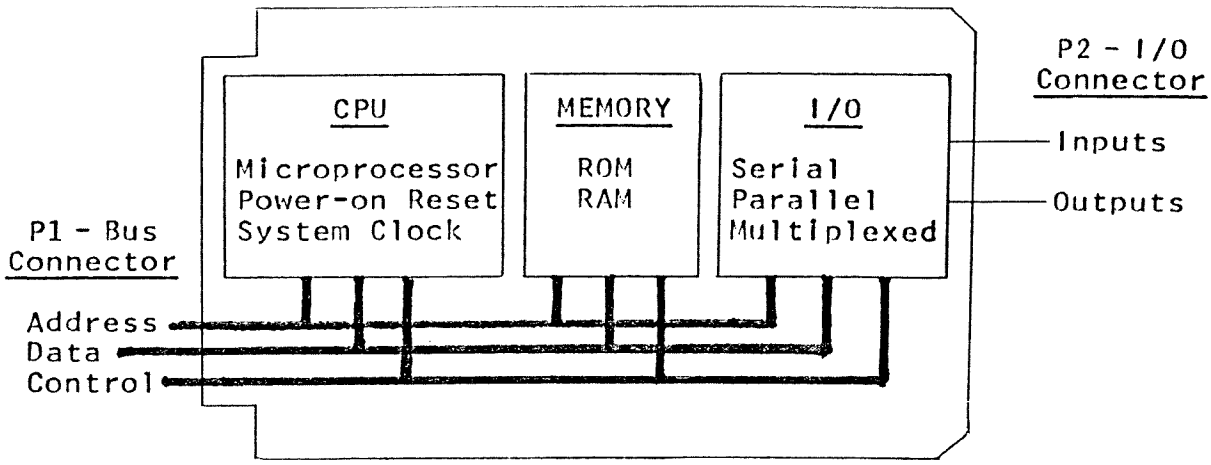


fig. 1 - BASYS/1 Block Diagram

CPU

The CPU (Central Processing Unit) is the system controller: It reads the program, and executes the instructions. The CPU also generates the timing and control signals for the rest of the computer. Virtually all of the CPU logic is implemented by a COSMAC microprocessor (U5, 1802 or 1804). A hex inverter (U1, 4069) performs the remaining CPU functions; Power-on Reset and System Clock. The address, data and control signals of the COSMAC form the Bus, and are routed to the onboard memory and I/O, and to the Bus connector (P1) for offboard use. Each Bus line is provided with a pull-up resistor (resistor networks U4, U10, and U18).

Power-on Reset

When power is first applied, the CPU and I/O circuitry are in a random state, and must be reset to a known state before beginning to execute the program. Sometimes it is also necessary to manually reset the system if, for example, an error in the program causes all other inputs to be ignored. The Power-on Reset circuit performs these functions.

Revisions -X1 and -X2 (see fig.2): Capacitor C7 is initially discharged when power is first applied, and so it holds the input of inverter U1A (pin 13) equal to the supply voltage, i.e. high. This makes the output of U1A low, which pulls the Bus $\overline{CLR}$ line low through diode D1. A low on Bus $\overline{CLR}$ resets the I/O circuitry (via pin 5 of U24) and the COSMAC microprocessor (U5, pin 3). Current through resistor R3 gradually charges C7, until the voltage at the input of U1A drops below its switching threshold (approximately 1/2 of the supply voltage). At this point, the output of U1A switches high, D1 ceases to conduct, and Bus $\overline{CLR}$ is pulled high by its pull-up resistor (pin 3 of U4). Diode D3 insures that C7 will be rapidly discharged to provide a reset pulse even if the power supply voltage drops only momentarily. D3 also protects the input circuit of U1A from the high discharge current from C7. A manual reset can be performed by discharging C7 with

a pushbutton switch connected between pins 22 and 24 of the I/O connector (P2).

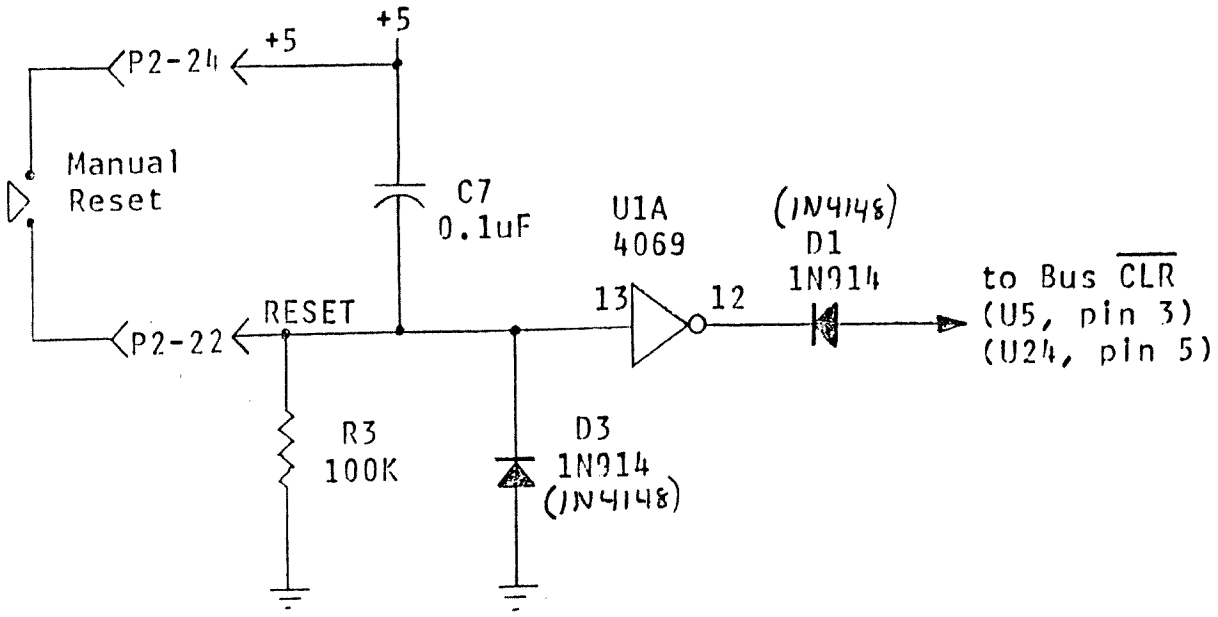


fig. 2 - Power-on Reset (revisions -X1 and -X2)

The length of the Bus $\overline{CLR}$ pulse is determined by the time constant of R3 and C7:

$$t(msec.) \approx 0.7 \times R3(K\ ohms) \times C7(uF)$$

The pulse must be long enough for a) the power supply voltage to stabilize, b) the System Clock to start running, and c) counter U2 (4040 or 4020) in the I/O circuitry to reach a stable state (2304 System clock cycles for rev. -X1; 10,240 for -X2). The pulse must not be unnecessarily long, or Bus $\overline{CLR}$ will have a slow rise time: Some 1802's are sensitive to this and may not reset reliably. Typical values with a 2MHz system clock are C7 = 0.1uF and R3 = 100K.

Revision -X3 (see fig. 3): Capacitor C7 is initially discharged when power is first applied and so it holds the input of inverter U1D (pin 9) low. This causes the output of U1D to be high, which is in turn inverted to a low at the output of inverter U1A (pin 12). The low on the output of U1A pulls the Bus CLR low through diode D1, and thus resets the I/O circuitry (via pin 5 of U24) and the COSMAC microprocessor (U5, pin 3). Current through resistor R3 gradually charges C7 until the voltage at the input of U1D crosses its switching threshold (1/2 of the supply voltage). At this point, the output of U1D switches low, and U1A's output switches high. Diode D1 ceases to conduct, and Bus CLR is pulled high by its pullup resistor (pin 3 of U4). Diode D3 insures that C7 will be rapidly discharged to provide a reset pulse even if the power supply voltage drops only momentarily. D3 also protects the input circuit of U1A from the high discharge current from C7. A manual reset can be performed by discharging C7 with a pushbutton switch connected between ground and the RESET pin on the I/O connector (pins 22 and 34 of P2).

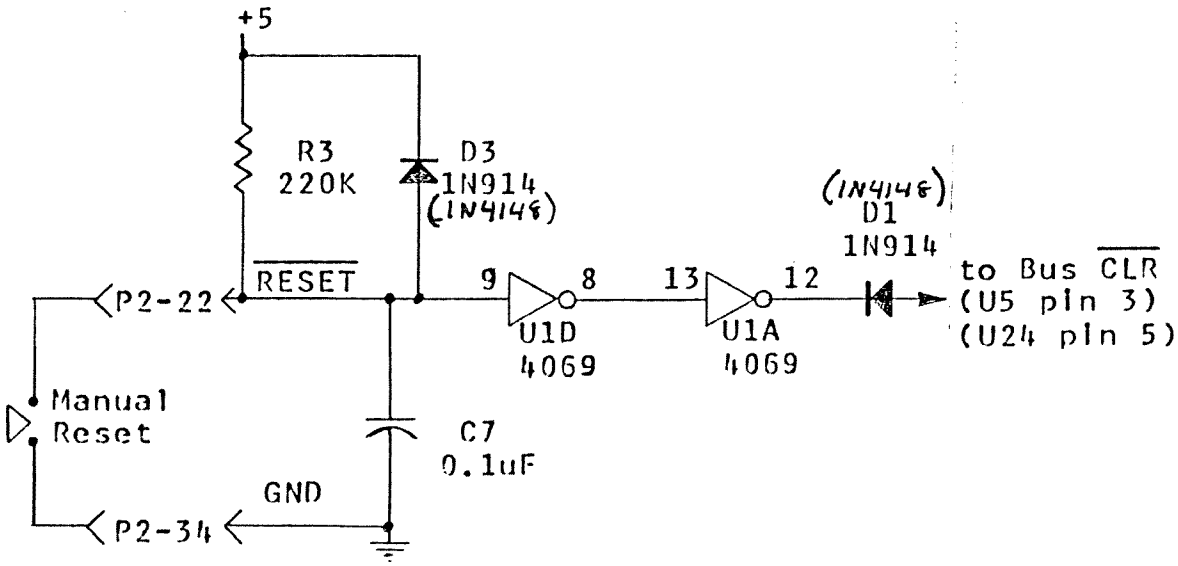


fig. 3 - Power-on Reset (revision -X3)

8/10/82 12/15/81

The length of the Bus $\overline{CLR}$ pulse is determined in the same manner as for revisions -X1 and -X2, and the same minimum pulse length limitations apply. However, the addition of inverter U1D speeds up the transition time, allowing all 1802's to be reliably reset even with a 500 msec. reset pulse. Typical values for rev. -X3 are C7 = 0.1uF and R3 = 220K. The -X3 revision also has reversed the logic level of the external reset switch (now $\overline{RESET}$ instead of RESET), so it is consistent with the DMA, Interrupt, and external flag lines.

System Clock

The System Clock is the master timing reference for the computer, upon which all other timing is based. This clock runs continuously and is unaffected by any other control signals. BASYS/1 may be configured to use any of three System Clock oscillator circuits, or to use an external clock, supplied on the Bus CLOCK line. (Revision -X1 must have the onboard clock disabled by cutting a trace).

The standard System Clock is a quartz crystal oscillator (see fig. 4). Inverter U1B is biased to operate as a linear amplifier by resistor R1. A feedback network consisting of crystal Y1, resistor R2, and capacitors C1 and C2 is also connected across U1B. The circuit oscillates at the parallel resonant frequency of the crystal. Resistor R2 reduces oscillator power consumption and improves stability. Capacitors C1 and C2 provide the crystal's load capacitance. Inverter U1C buffers the output of the oscillator and drives the Bus CLOCK line and the COSMAC clock input (pin 1 of U5). Capacitor C3 is not used.

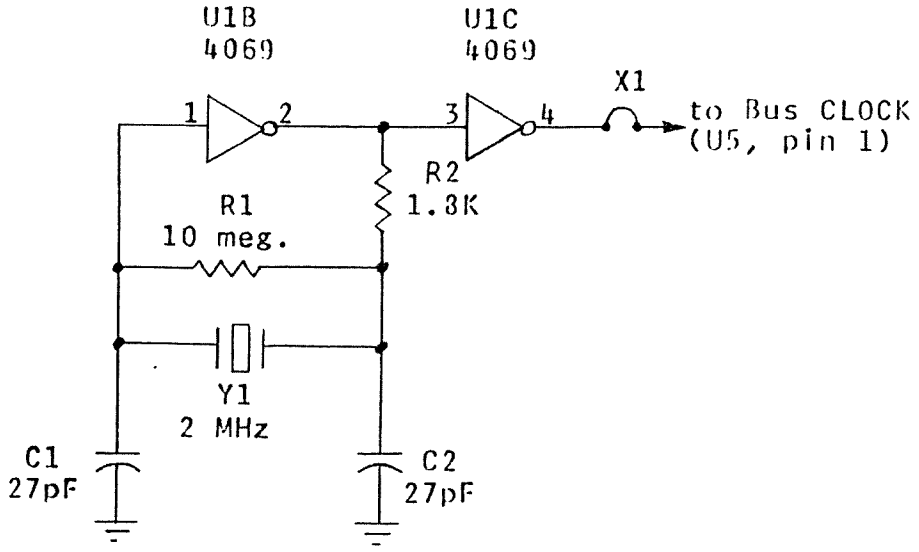


fig. 4 - Crystal-Controlled System Clock

The crystal oscillator provides excellent stability and high speed operation. It can operate from 32kHz to 4MHz at 5vdc (or to 8MHZ at 10vdc) with an accuracy and stability as good as 1 part per million. For critical applications, the series combination of C1+5pF and C2+5pF should match the crystal's load capacitance (typically 10 to 32pF). For example, the crystal in fig. 4 will see a load capacitance of (27+5)/2 = 16pF. The value of C2 may also be adjusted to trim the frequency up or down a few hundred cycles. For micropower applications, R2 should be empirically chosen to optimize power consumption and stability.

For the lowest cost or for testing purposes, an RC-oscillator can be used (see fig. 5). Inverters U1B and U1C, capacitor C3, and resistors R1 and R2 form an RC-oscillator. Crystal Y1 and capacitors C1 and C2 are not used. Below 100 KHz the frequency is :

$$\text{frequency (KHz)} \approx \frac{1}{1.4 \text{ X R2(Kohms) X C3(uF)}}$$

For the best results, C3 should be greater than 200pF, and R1 should be from 2 to 10 times larger than R2. The upper frequency limit is approximately 500 KHz. The lower limit is below 1 Hz: The only practical difficulty is in obtaining a large, non-polarized capacitor for C3. The stability of a CMOS RC-oscillator is surprisingly good -- with precision resistors and capacitor the frequency can be stable to within ± 1% despite wide variations in supply voltage and temperature. For testing purposes, the RC oscillator can be used even if the crystal oscillator is present: Temporarily add C3, and the oscillator will run at the frequency of the RC oscillator components.

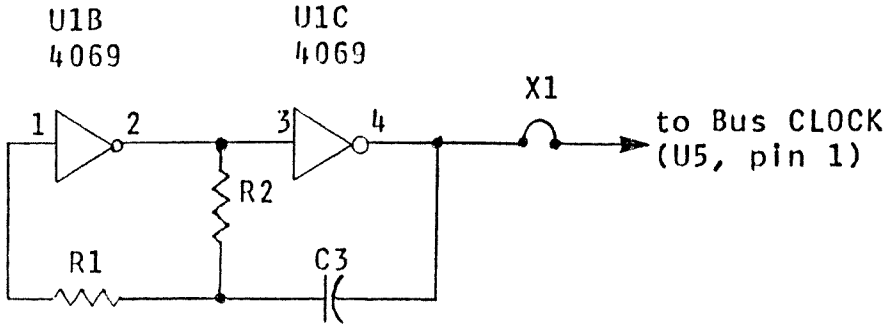


fig. 5 - RC-Oscillator System Clock

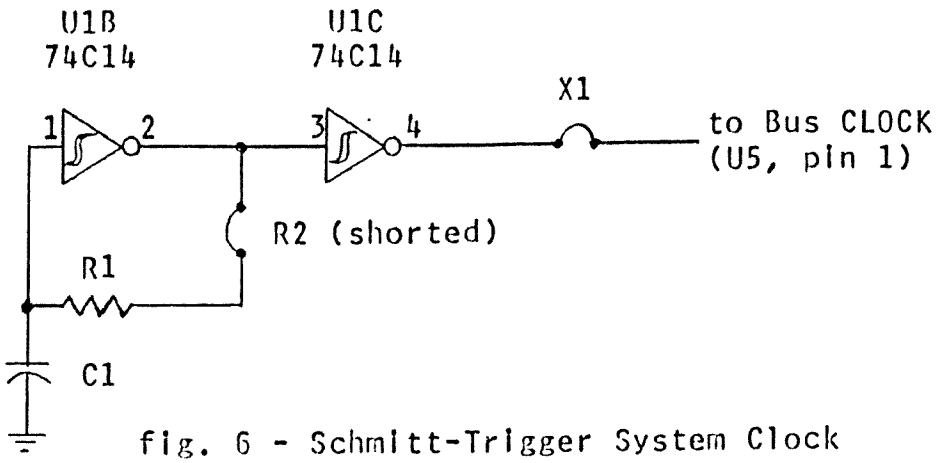


fig. 6 - Schmitt-Trigger System Clock

The third type of System Clock oscillator uses a hex Schmitt-Trigger (40106 or 74C14) in place of inverter U1 (see fig. 6). Only R1 and C1 are used: R2 is replaced by a short circuit, and C2, C3, and Y1 are not used. A Schmitt-trigger has a higher switching threshold when its output is high, and a lower threshold when its output is low. When the output of U1B is high, C1 charges through R1 until the input reaches the higher threshold. The output then switches low, and C1 is discharged through R1 until the lower threshold is reached. The cycle then repeats, forming an oscillator with a frequency of:

$$\text{frequency (KHz)} \approx \frac{1}{1.7 \text{ X R1(Kohms) X C1(uF)}}$$

The range of frequency is an amazing 0 to 3 MHz at 5vdc, and there are no restrictions on the values of R1 and C1. This oscillator is the least stable of the three, and so should be restricted to non-critical applications. For testing and experimentation, a 1 megohm potentiometer can be used for R1 so the clock frequency can be adjusted over a 1,000,000 to 1 range. C1 can be a polar capacitor, so frequencies below 1 Hz can readily be obtained for testing or troubleshooting without an oscilloscope. Use of a Schmitt-trigger for U1 also completely eliminates any possible problems with long reset pulses on -X1 and -X2 boards.

External Clock (see fig. 7): Jumper X1 on board revisions -X2 and -X3 is removed when an external clock is to be used. This jumper is not present on -X1 boards.

The requirements for an external clock are not critical. With a 5vdc supply and a standard 1802, Bus CLOCK is a TTL and CMOS-compatible input, and will accept any ø to 2.5 MHz input with rise and fall times of less than 1uS, and a minimum high and low time of 250 nSec. A 22K ohm pullup resistor is connected on Bus CLOCK to insure TTL compatibility (U4, pin 1). Faster clocks can be used with faster COSMACs, or with higher supply voltages.

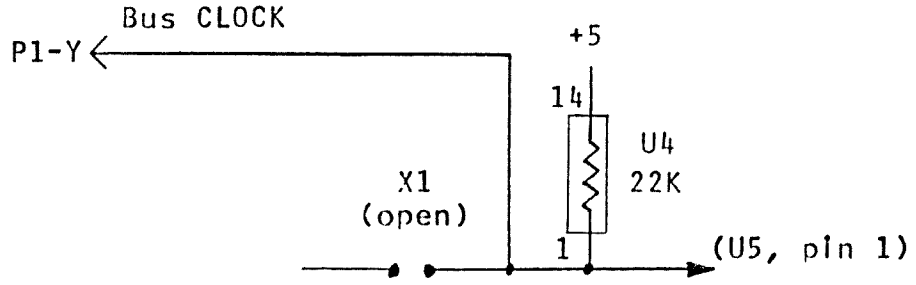


fig. 7 - External System Clock

COSMAC Microprocessor

The COSMAC microprocessor is the heart of a BASYS computer system. It is available in a bewildering array of voltages, speeds, packages, and manufacturers. Since BASYS is compatible with all of these variations, it is worth listing the various options (see fig. 8).

1802 (COSMAC microprocessor only)

package	Standard Speed: (2.5MHz) 4-6 Vdc	4-12 Vdc	Hi-Speed: (4MHz) 4-6 Vdc
ceramic:	CDP1802CD HCMP1802CD SCP1802LD	CDP1802D HCMP1802D SCP1802D	CDP1802AD HCMP1802CD-2
plastic:	CDP1802CE HCMP1802CP SCP1802LE	CDP1802E HCMP1802P SCP1802E	CDP1802ACE HCMP1802CP-2

1804 (COSMAC microprocessor, timer, 2K ROM, 64-byte RAM)

package	Standard Speed: (4MHz) 4-6 Vdc	4-10 Vdc
ceramic	CDP1804CD HCMP1804CD	CDP1804D HCMP1804D
plastic	CDP1804CE HCMP1804CP	CDP1804E HCMP1804P

Notes

1. Speeds assume operation on 5vdc over the -40 to +85C temperature range. Raising or lowering the supply voltage changes the maximum speed accordingly.

2. There is no electrical difference between ceramic and plastic parts. A ceramic package makes the IC slightly more reliable and increases the operating temperature range (-55 to +125C ceramic, -40 to +85C plastic).

3. "CDP-" parts are made by RCA
   "HCMP-" parts are made by Hughes Aircraft Co.
   "SCP-" parts were made by Solid State Scientific, but are no longer manufactured by them.

fig. 8 - Available COSMAC Microprocessors

12/15/81

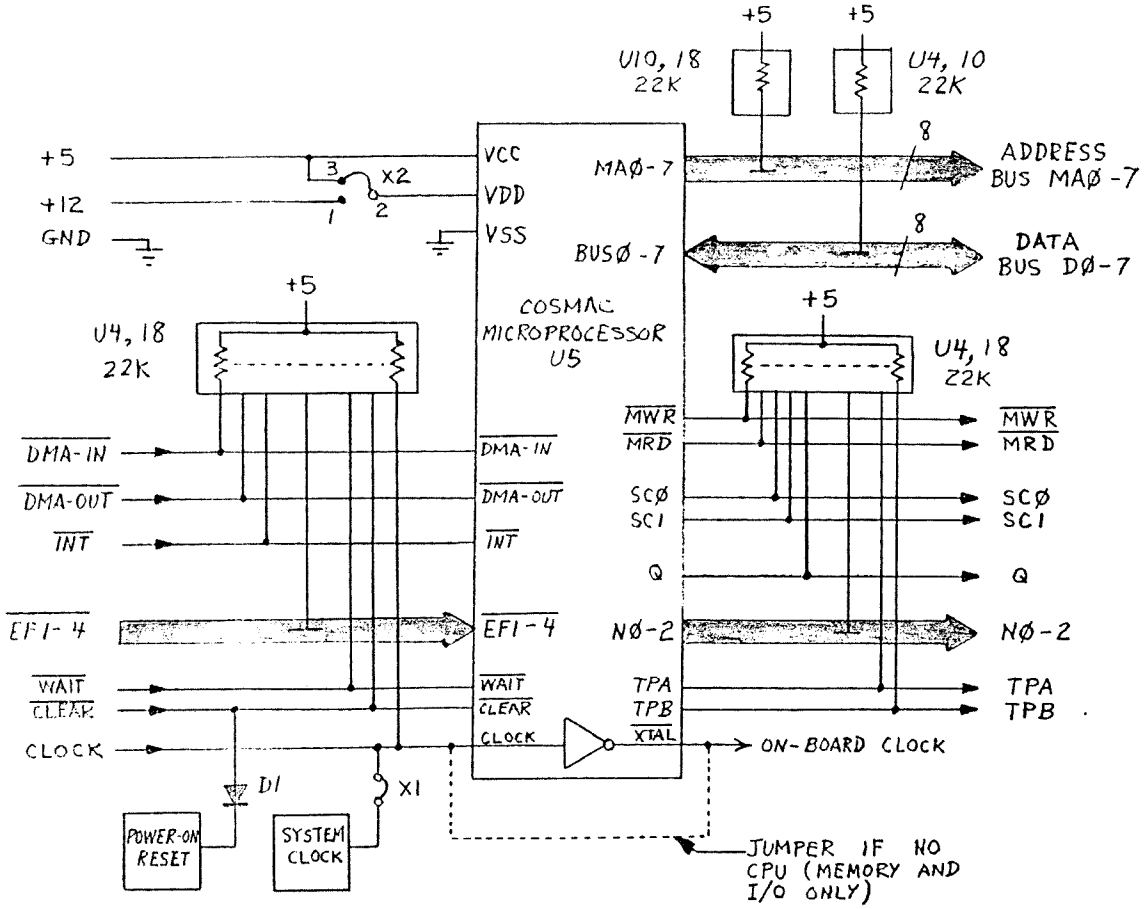


fig. 9 - Microprocessor

The 1802 is the most common and widely-available version of the COSMAC microprocessor. The 1802 is virtually a complete CPU, and lacks only a power-on reset circuit. It also includes a crystal oscillator, four input bits, one output bit, and DMA (Direct Memory Access) logic. The 1804 is a faster, improved 1802, with an expanded instruction set, a hardware timer, 64 bytes of RAM, 2K bytes of mask-programmable ROM, an improved clock oscillator, and a power-on reset circuit. Since the internal ROM can only be programmed by the manufacturer, the 1804

is primarily of interest to large-quantity users. For complete data on the 1802 and 1804, refer to their respective data sheets. The following information is concerned mainly with the hardware aspects of the COSMAC as it is used in BASYS/1, and applies to both parts unless otherwise noted.

The standard COSMAC for BASYS/1 is the 4-6 vdc version of the 1802 in a plastic package. It uses a standard +5vdc power supply, which is by far the most common choice for digital integrated circuits. The standard system clock frequency is 2.000 MHz, which guarantees operation from 4-6 vdc over the entire -40 to +85 C. temperature range. For these conditions, the 1802 will draw less than 2mA.

BASYS/1 does not use the COSMAC's internal clock primarily for reasons of flexibility: It is designed only as a crystal oscillator, and lacks sufficient drive for Bus CLOCK.

Like all CMOS, the performance of the COSMAC is directly related to its power supply voltage. Increasing the supply voltage proportionally increases supply current, output drive, noise immunity, and speed of the internal logic. Thus doubling the supply voltage will double all of these other parameters as well. Power consumption would increase by 4:1 since both supply voltage and current were doubled. The power consumption is also directly proportional to the clock frequency: If the clock frequency in the above example were doubled from 2MHz to 4MHz to take advantage of the higher speed in the internal logic, the power consumption would again double (in addition to the previous 4:1 increase). The reverse is true for a decreasing supply voltage or clock frequency. Thus BASYS/1 can be optimized for a particular set of requirements simply by choosing the supply voltage and clock frequency. COSMACs are also available in 3-12 vdc versions. The advantage of these versions is that they extend the range over which such tradeoffs can be made.

For additional flexibility, the 1802 has two power supply pins: VDD and VCC. These pins may be powered by any voltage within the basic range of the part (i.e. 4-6 or 3-12 vdc), as long as VDD is always equal to or greater than VCC. VDD

powers all of the 1802's internal logic, while VCC powers all of the inputs and outputs. Normally, VDD and VCC are tied together by jumper X2 (X2-2 to X2-3) and the entire board operates from a single supply. Jumper X2 can also be connected to use separate supplies (X2-1 to X2-2) for higher speed or increased noise immunity. With VDD=12VDC for example, the 1802 can be run at 4MHz and yet maintain compatibility with other logic at 5VDC. The dual-supply capability can also be used to save data in the 1802's registers with the help of a small battery or electrolytic capacitor. When power is removed from the system, VCC goes to 0, and the clock stops. At 0 frequency, the power consumption of the 1802 approaches zero, so no current is drawn from VDD. If VDD is kept powered, all of the register contents will remain valid. Care must be taken that VDD is always greater than VCC, and that the battery is not discharged by the normal operating current of VDD.

The 1804 substitutes the EMS (External Memory Select) output for the VCC pin, and has only one power supply on VDD. The EMS signal is not used on BASYS/1, and therfore no jumper is installed at X2. All other pinouts are identical to the 1802, so no changes are necessary.

BASYS/1 can be configured to operate without a CPU, as a memory and I/O expansion board. For this application, simply leave the COSMAC off, and install a jumper between pins 1 and 39 of CPU socket U5. The System Clock may be used by installing X1, or disabled by leaving X1 open. The external CPU may be on another BASYS/1 board, or on a completely different system. In the latter case, either the BASYS Bus can be wired to the other system's bus, or a 40-wire cable can be built to directly connect the BASYS COSMAC socket (U5) to the COSMAC socket in the user's system.

BASYS/1 systems do not use bus drivers for the BASYS Bus. The COSMAC provides much better characteristics as a bus driver and receiver than most of the commonly-used bus interface ICs. Every line on the BASYS bus also has a pull-up resistor to provide additional static protection and to eliminate floating inputs. This technique also makes every feature of the COSMAC available to any board on the bus without exception. It also makes it possible to use a small, low-cost edge connector for the bus, and saves space on the PC boards that would normally be needed for bus interface ICs.

Memory

The memory on BASYS/1 stores information for use by the CPU: Programs, or data to be manipulated by the programs. Two different classes of memory ICs are used: RAM (Random Access Memory) and ROM (Read-Only Memory). Jumper options permit use of many common memory ICs from the CMOS, NMOS, and TTL families, and allow for very flexible memory addressing. A brief summary of the types of memory used in BASYS/1 is given in fig. 10.

ROM (Read Only Memory) - A true mask-programmed ROM is programmed during its manufacture, and cannot be changed. The buyer pays a one-time "mask" charge, and can then buy production quantities at low prices.

PROM (Programmable ROM) - a PROM employs fuses or other non-reversible storage elements. A "PROM Programmer" is used to selectively blow fuses to store data. Once programmed, a bit cannot be erased or unprogrammed.

EPROM (Eraseable PROM) - EPROMs use a reversible charge-storage mechanism to store data. Like the PROM, it is electrically programmed, but can be erased by strong ultraviolet light and re-used many times.

EEPROM (Electrically Eraseable PROM) - EEPROMs are similar to EPROMs, but can be both programmed and erased electrically. If the computer can control the erasing, they become a "read-mostly" memory, and can store data for recovery after a power failure.

RAM (Random-Access Memory) - A better name would have been "read-write memory". You can both read and write into RAM at high speed, so it is used to store temporary data and programs that change often. However, RAMs require power to hold information: If the power is turned off, all of the contents is lost.

fig. 10 - Types of Memory

ROM Memory is used for non-volatile read-only storage. Since ROMs are non-volatile, programs or data stored in them remain valid even with the power off, and are ready the instant power is turned on. It is usually impossible for the computer to alter the contents of a ROM; therefore, it cannot accidentally alter or damage such protected information. ROMs are categorized in fig. 10 by the method used to store information into them.

The standard configuration for BASYS/1 uses a 2716 EPROM (U11) which stores 2K bytes (fig. 11). The 2716 uses a single +5vdc ±5% supply, is relatively low power (50mA), is economically priced, and easily programmed.

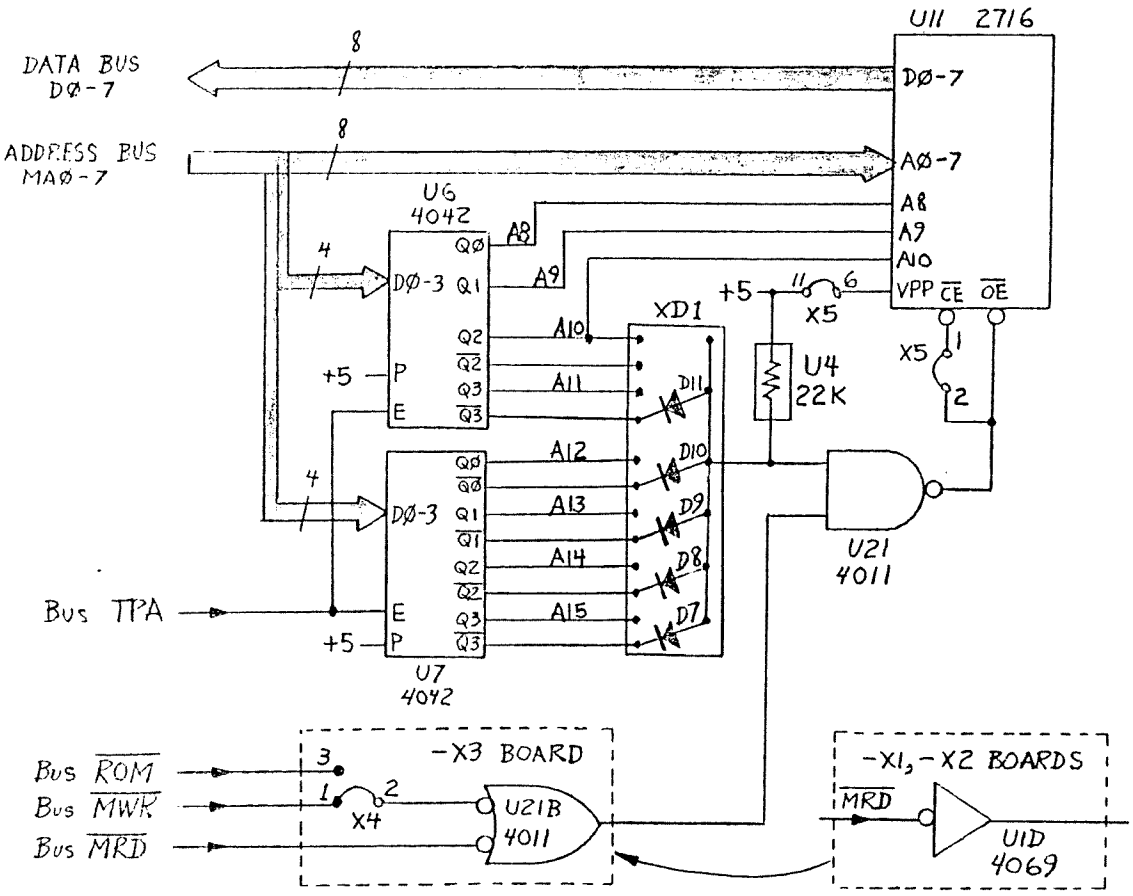


fig. 11 - BASYS/1 with 2716 EPROM

The COSMAC can address any one of 65,536 ($=2^{16}$) memory locations by outputing an appropriate 16-bit address (AØ - A15) and a memory read command ($\overline{MRD}$). Since the COSMAC is an 8-bit CPU, it tends to do things a byte (8 bits) at a time: For addresses, it first outputs the upper byte (A8 - A15), then the lower byte (AØ - A7) on the Memory Address lines (MAØ - MA7). Two timing pulses are also provided to identify which half of the address is currently available: Timing Pulse A (TPA) occurs when the upper byte is present, and Timing Pulse B (TPB) when the lower byte is present.

This technique is called multiplexing, and although somewhat confusing, it has its advantages. It frees 8 pins on the COSMAC, which were put to good use for other functions. It matches the COSMAC instruction set, which organizes memory into 256 pages of 256 bytes each (the page address is the high byte). In systems with more than one memory IC, address decoding logic decodes the high byte to determine which memory IC is to be selected. This decoding logic takes time away from memory access time, so having the high byte available early is an advantage. Finally, BASYS/1 retains the multiplexed addresses on its Bus, which saves wiring and connectors, and so improves cost and reliability.

All memory cycles begin when the COSMAC outputs the page address (high byte) onto the Bus MEMORY ADDRESS lines (MAØ - MA7), followed by Timing Pulse A (on Bus TPA, see fig. 12). When TPA goes high, address latches U6 and U7 (4042) pass the page address through their "D" inputs to their "Q" and "$\overline{Q}$" outputs. When TPA returns low, U6 and U7 latch the high byte. The outputs of U6 and U7 supply all 8 high address bits in both their true (A8 - A15) and inverted form($\overline{A8}$ - $\overline{A15}$) on their "Q" and "$\overline{Q}$" outputs, respectively.

Jumper socket XD1 contains the address selection diodes, D7-12. The number of diodes and their position decode the range of addresses for which U11 will be selected. The number of diodes equals the number of address lines not used directly by U11. A 2716 for example, contains 2048 bytes: $2048 = 2^{11}$, so it has 11 address lines. Thus 5 of the COSMAC's 16 address lines are left over, and so 5 diodes are used in XD1. The diodes and their pull-up resistor (U4, pin 5) form a diode "AND" gate, whose output is high only if the cathodes of all diodes are high. In fig. 11 the diodes are shown connected to A11, $\overline{A12}$, $\overline{A13}$, $\overline{A14}$, and $\overline{A15}$. The AND gate output will go high for any address with all of these bits low i.e. binary ØØØØ ØØØØ ØØØØ ØØØØ to ØØØØ Ø111 1111 1111, hex ØØØØ to Ø7FF, and decimal 0 to 2047). The remaining address bits (AØ to A1Ø) go directly to U11: AØ - A7 connect to Bus MAØ - MA7 since the COSMAC is outputting the low address byte after TPA, and A8 - A1Ø come from the address latch, U6. See the Assembly manual for details on setting the diodes and jumpers for other ROMs or address decoding.

If the COSMAC wishes to read a memory location, it performs a memory read cycle. It generates an active-low output on Bus $\overline{MRD}$. This is inverted by U1D (or U21B on -X3 boards) and enables one input of NAND gate U21A. The other input comes from the address selection diodes in XD1. When MRD is active and the address is within the range selected for U11, the output of U21A goes low and chip-selects U11 (2716). U11 then fetches the contents of the specified address, and places it on the Data Bus, lines DØ - D7. At the end of the memory read cycle, the COSMAC reads this data. $\overline{MRD}$ then returns to the inactive state, and the memory read cycle ends.

Jumper socket X5 provides a number of jumpers to reconfigure socket U11 for many different types of ROM. The -X3 boards are even more flexible, and replace inverter U1D with a NAND gate, U21B. The output of this gate will go high for either a memory read or a memory write

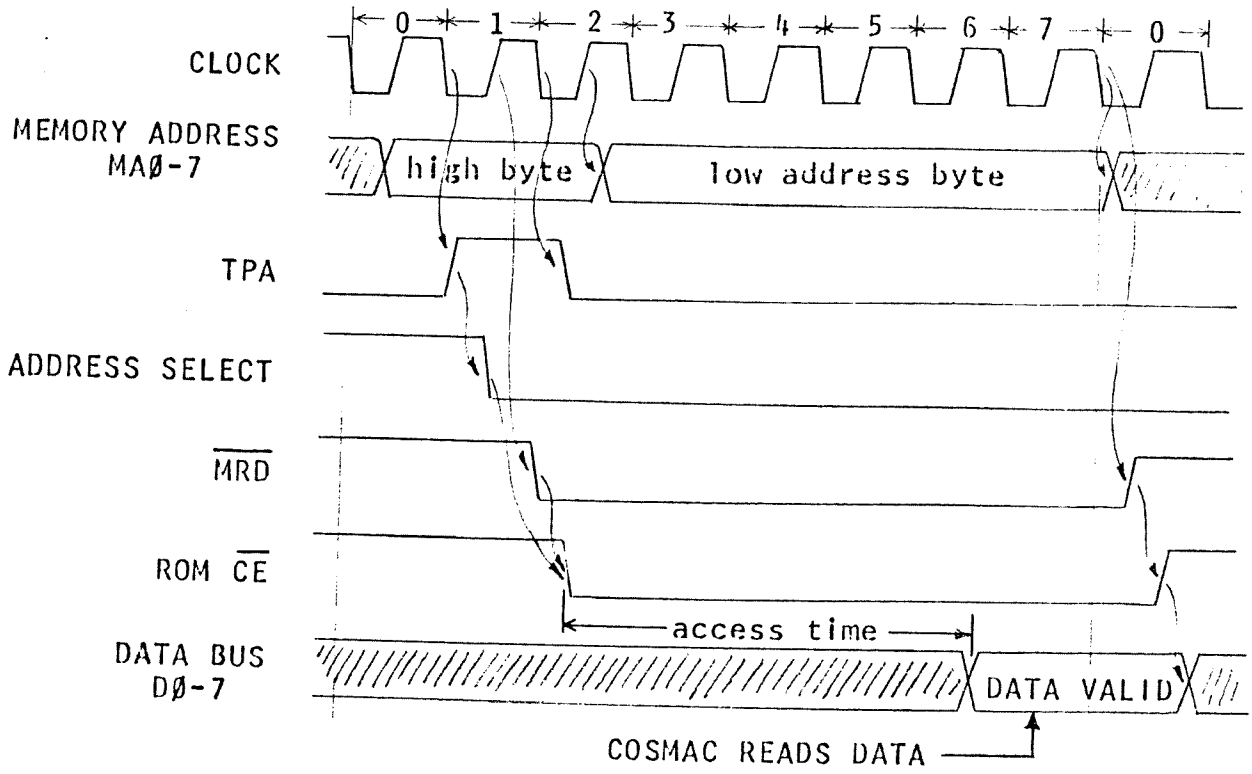


fig. 12 - ROM Memory Read Cycle Timing Diagram

cycle. The Bus $\overline{MWR}$ line also goes to a previously unused pin in X5. These changes now allow U11 to be used with byte-wide RAMs, and make it easier to program PROMs, EPROMs, and EEPROMs in BASYS/1.

Jumper socket X5 also provides for routing +12 and -5 vdc to U11. This is only needed for a few older EPROMs, such as the 2704 and 2708. Pins are defined on the Bus connector (P1) and bypass capacitors (C5, 6, 12, and 13) are provided.

NMOS, CMOS, and TTL parts are available for U11. NMOS parts are generally the most economical and come in the largest variety. Most of them require +5 vdc $\pm$5% and operate over only 0 to 70°C. CMOS parts offer very low power requirements, high noise immunity, wide temperature range, and flexible

power supply voltage selection. They are about twice as expensive a. an equivalent NMOS part, and less available in larger memory sizes. TTL parts are similar to NMOS, but require substantially more power and come in a limited range of sizes. TTL parts are also very fast; this is of no importance in BASYS/1, however. TTL parts do have one unique characteristic that sometimes justifies their use in BASYS/1: When the +5 power supply is disconnected from most Schottky TTL PROMs, all of their inputs and outputs become open circuits (instead of short circuits to ground as in CMOS and NMOS). An appropriate circuit can use the power supply pin as a chip-select. The PROM then draws current only when it is actually being selected -- the rest of the time it uses no power.

RAM Memory is primarily used to store changeable programs and data. It is just as easy for the computer to write into a RAM as it is to read it. However, the information stored in a RAM is volatile: If power is lost, the data is lost. RAM also costs more than an equivalent amount of ROM, and uses more power.

The standard RAM for BASYS/1 is the CMOS 6514 (UPD444C) (or its NMOS equivalent, the 2114). Both are 1Kx4 devices, and so contain 1024 locations of 4 bits each. Since BASYS/1 is an 8-bit system, the RAMs are used in pairs. There are four RAM sockets (U12-15) thus providing 2K bytes of on-board RAM. The 6514 (UPD444C) operates on 4-6vdc at less than 1mA each (operating). The 2114 requires 5vdc $\pm$5% at approximately 50mA each.

RAM circuit operation is very similar to the ROM circuit operation, discussed above (see fig. 13). A memory cycle begins with the COSMAC placing the high byte of an address on the Bus MAØ - 7 lines, and then generating a timing pulse on TPA. Address latches U6 and U7 (4042) latch the high byte, and provide both the true (A8 - A15) and complemented ($\overline{A8}$ - $\overline{A15}$) address bits to diode AND gate XD2. The number of diodes used and their positions determine the RAM addressing exactly the same as for the ROM. The circuit in fig.13 shows one possible arrangement. The five diodes are positioned so that the output of the

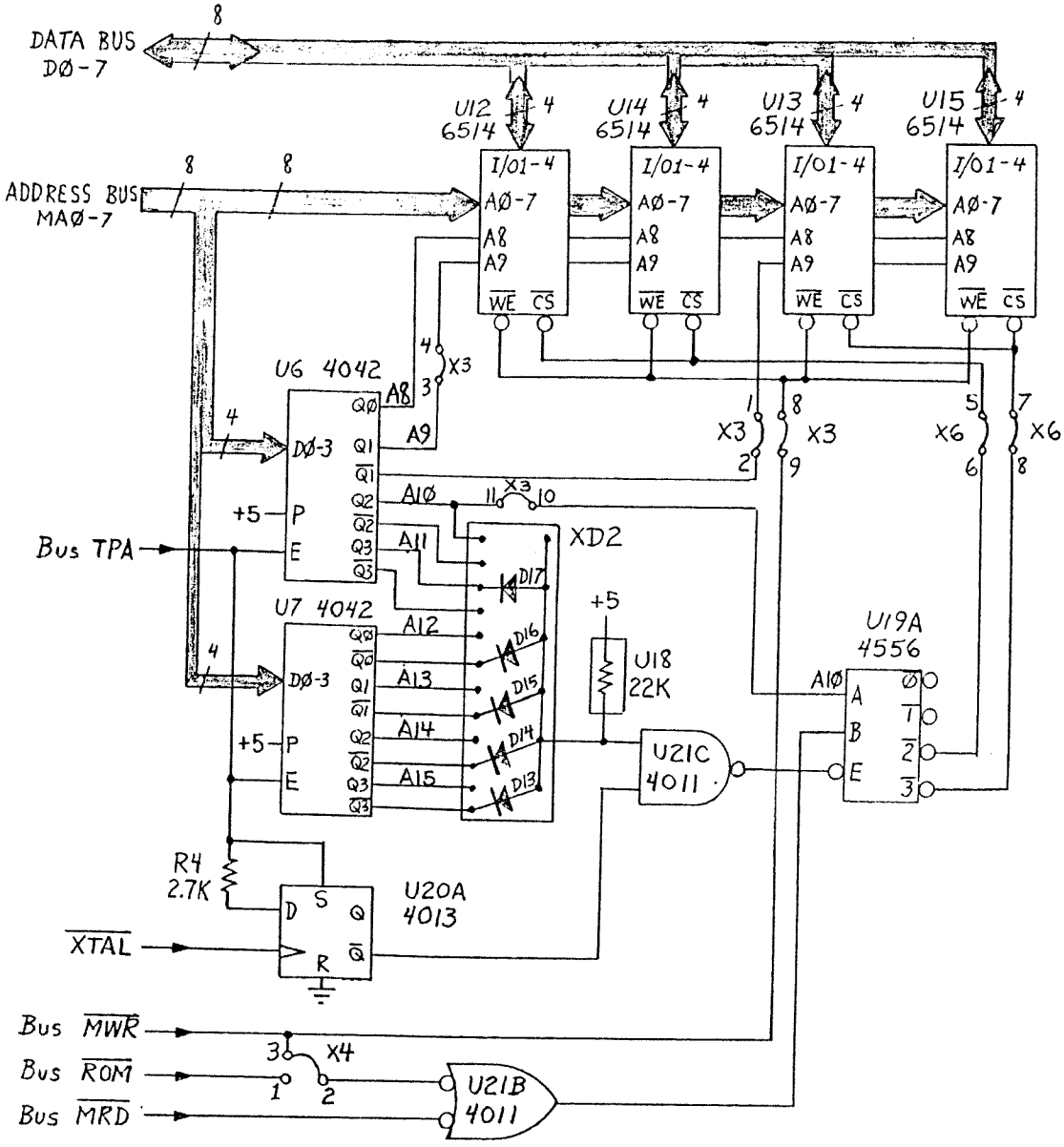


fig. 13 - BASYS/1 using 6514 (UPD444C) CMOS RAM

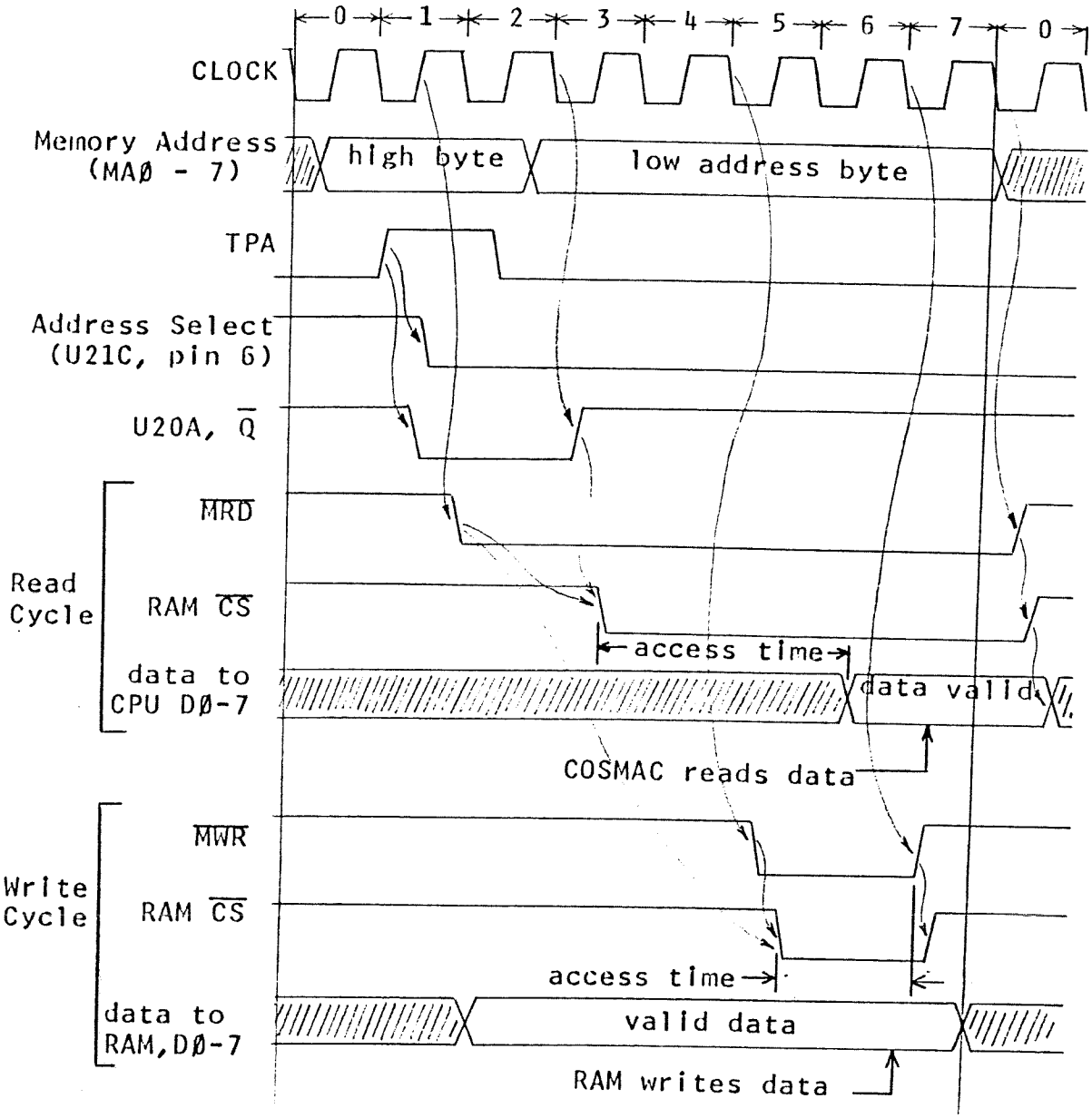


fig. 14 - RAM Memory Read and Write Timing

diode AND gate is high for A15-11 = ØØØØ1. RAM is then selected for any address from binary ØØØØ 1ØØØ ØØØØ ØØØØ to ØØØØ 1111 1111 1111, hex Ø8ØØ to ØFFF, and decimal 2048 to 4095. See the assembly manual for details on setting the address decoding for other starting addresses.

When writing data into RAM, it is important that the address not change afterthe RAM has been selected: Otherwise data may be written into the wrong address. Some RAMs (like the 6514) prevent this problem by latching the address on the falling edge of their chip-select input. Flip-flop U20A (a 4013) delays the RAM chip select long enough to insure that the full address (both high byte and low byte) is ready when the RAM begins its cycle.

Initially U20A is in the reset state, and its $\overline{Q}$ output (pin 2) is high. When Bus TPA occurs, it sets U20A and its $\overline{Q}$ output goes low. U20A remains set when TPA returns low and the COSMAC changes MAØ - 7 from the high byte to the low address byte. At the next falling edge of Bus CLOCK the COSMAC's $\overline{XTAL}$ output goes high and clocks flip-flop U20A. At that time TPA is low, so U20A is reset and $\overline{Q}$ returns high. This high is routed to one input of NAND gate U21C (a 4011). If XD2 has decoded a RAM address, the other input of gate U21C is already high. Then the output of U21C will go low, enabling decoder U19A (a 4556) and chip-selecting the appropriate RAM.

NAND gate U21B (4011) has its two inputs connected to Bus $\overline{MRD}$ and Bus $\overline{MWR}$ through jumper X4. With a jumper between X4-2 and X4-3, the output of U21B will be high for either memory read or memory write cycles. With a jumper between X4-1 and X4-2, memory write cycles are inhibited unless the Bus $\overline{ROM}$ line is externally connected to Bus $\overline{MWR}$. This permits using an external "Memory Protect" switch or logic, for applications such as front panels. When the output of U21B is high, it selects the "2" or "3" outputs of decoder U19A, as determined by address line A10, in the "A" input of U19A: A10 = Ø selects the "3" output, and thus U13 and U15. Once selected, the RAMs perform a read cycle if the write-enable input (pin 10) is not low (i.e. inactive), and a write cycle if write-enable is low (i.e. active). The data for read and write cycles is communicated via the Bus DATA lines, DØ - D7.

I/O

The I/O circuits are the computer's method of communicating with the outside world. BASYS systems provide a variety of types of I/O lines which can directly control serial, parallel, and multiplexed devices.

Serial I/O

Serial I/O transfers blocks of data one bit at a time via a single line or data path. Special "start" and "stop" patterns are used to mark the beginning and end of each data block. Because only one bit at a time is manipulated, serial I/O is simple, but slow. It is usually used for human-interfaced devices, such as printers, data terminals, Teletypes, tape recorders, telephone lines, etc.

BASYS/1 has an optically-isolated serial input and output which can be used for the industry-standard RS-232C or 20mA current loop interfaces, for audio FSK (frequency shift keying), or for special applications requiring an isolated single-bit input or output. The COSMAC's "Q" output and "$\overline{EF4}$" flag input are used, together with software, to make the baud rate (data rate in bits per second) and data format completely programmable. For BASYS/1 boards sold with software, refer to the software documentation for a description of the serial I/O formats supported.

RS-232C (0 to 1200 baud)

Receive (see fig. 15): BASYS/1 receives RS-232C data on Serial-In Common (P2, pin 37); Serial-In (P2, pin 40) is connected to the RS-232C signal ground. When not receiving data ("idle") or when a data bit "1" is being received, Serial-In Common will be at a negative voltage. Current will flow through R6 and the LED in optocoupler U26 (4N30-35). The light from the LED turns the phototransistor in U26 on, which pulls the COSMAC microprocessor's $\overline{EF4}$ input low (U5, pin 21). When a data bit "Ø" or "break" is received, Serial-In

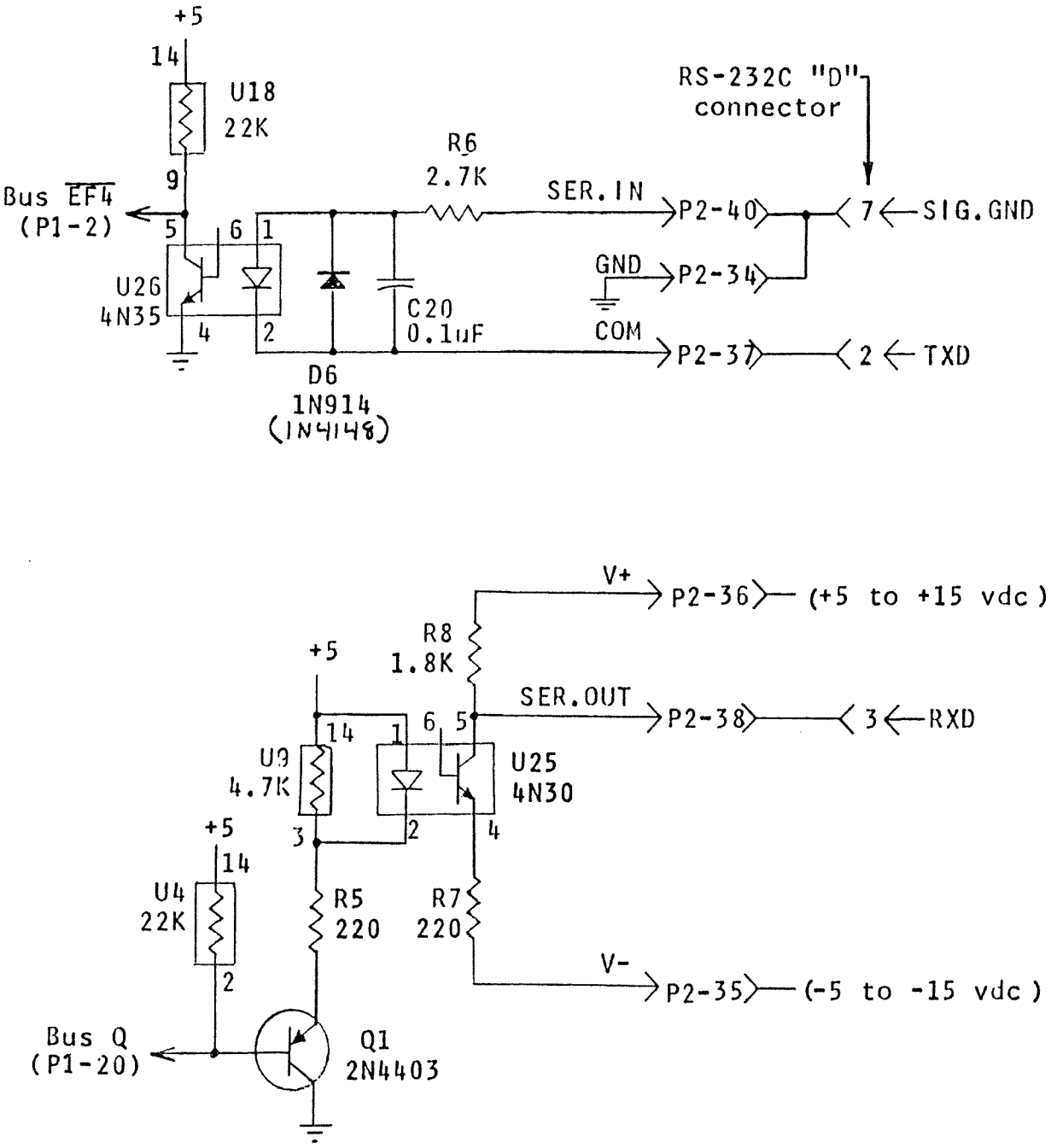
Common will be at a positive voltage. Current through R6 is shunted through D6, the LED in U26 does not light, and the phototransistor will turn off. The $\overline{EF4}$ line (U5, pin 21) is then pulled high by pullup resistor U18, pin 9. Capacitor C20 works with R6 to suppress noise pulses.

Transmit: The COSMAC's "Q" output is low during "idle" or to transmit a data bit "1". This pulls the base of emitter follower Q1 (2N4403) low. Current then flows through the LED in optocoupler U25 (4N30-35), resistor R5, and through Q1 to ground. Light from the LED turns on the phototransistor in U25, pulling Serial-Out (P2, pin38) down to within 2 volts of V- (P2, pin 35). Resistor R7 limits the short-circuit current to an acceptable level. To output a data bit "Ø", the COSMAC's "Q" output is set high. Q1 turns off, current stops flowing through the LED in U25, and the phototransistor turns off. Serial-Out (P2, pin 38) is then pulled up to V+ (P2, pin 36) by resistor R8.

20mA Current Loop (0 to 1200 baud)

Receive (see fig. 16): BASYS/1 uses TTY-IN (P2, pin 39) and Serial-In Common (P2, pin 37) to receive current-loop data. When not receiving data ("idle") or when a data bit "1" is being received, the 20mA current will flow into TTY-IN, through the LED in the optocoupler U26 (4N26 - 4N35) and back out Serial-In Common. The light from the LED turns the phototransistor in U26 "on", which pulls the COSMAC microprocessor's $\overline{EF4}$ input low (U5, pin 21). When a data bit "Ø" or "break" is received, current will cease flowing through the LED in U26, it's phototransistor turns off, and the $\overline{EF4}$ line is pulled high by pullup resistor U18, pin 9. A software program can then interpret the serial data on $\overline{EF4}$.

fig. 15 - RS-232C Interface



RS-232C Interface

GENERAL: Used to exchange serial data between two devices up to 50 ft. apart. 0 to 9600 baud, synchronous or asynchronous, any character length or code. Recommended connector, TRW Cinch DB-25S (female) or DB-25P (male) or equivalent.

POWER SUPPLY: Requires ±5 to ±15vdc @ 5-15mA (typical). May be available on DB-25 connector.

DATA (TXD and RXD) : active low

1 = MARK (Idle) = -5 to -15vdc (from transmitter)
-3 to -25vdc (at receiver)
0 = SPACE (Break) = +5 to +15vdc (from transmitter)
+3 to +25vdc (at receiver)

Terminals and computers output data on Transmit Data (TXD, pin 2); connect them to BASYS/1 as shown. Modems output data on Receive Data (RXD, pin 3); connect to BASYS/1 with pins 2 and 3 interchanged at the DB-25 connector.

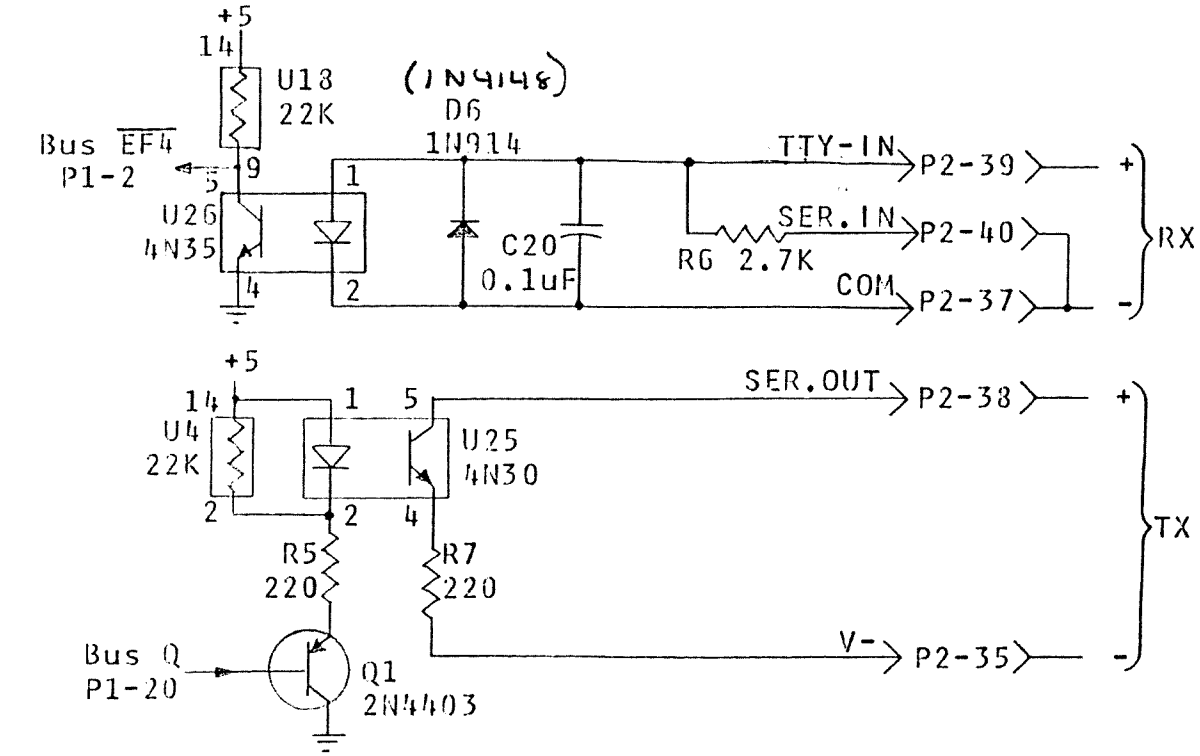
CONTROL (CTS, RTS, DSR, DTR): active high

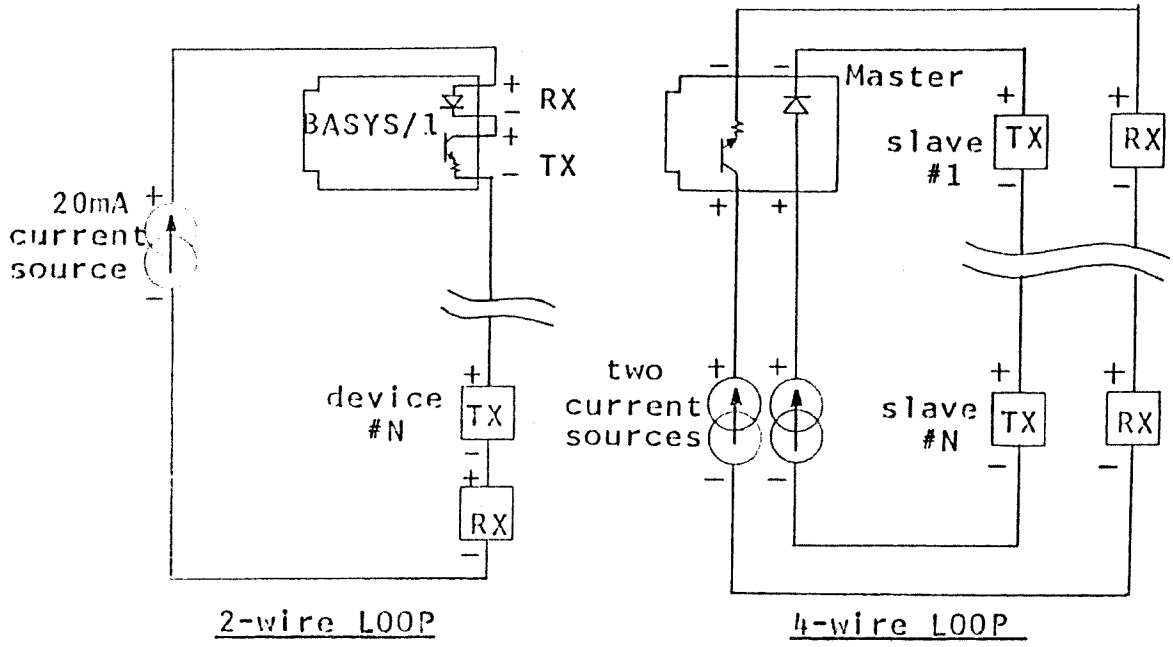
1 = ON (true) = +5 to +15vdc (from transmitter)
+3 to +25vdc (at receiver)
0 = OFF (false) = -5 to -15vdc (from transmitter)
-3 to -25vdc (at receiver)

Some RS-232C devices may require that one or more of these control lines be held ON to operate. Try connecting CTS (pin 5) to RTS (pin 4), and DSR (pin 6) to DTR (pin 20) on the DB-25 connector.

12/15/81
8/10/82

fig. 16 - Current Loop Interface





Current Loop Interface

GENERAL: Used for serial communications between two or more devices over long distances. Data is asynchronous, ASCII or Baudot. The most common format is 7-bit ASCII with even parity, 1 start, and 1 stop bit (2 at 110 baud). There is no generally accepted connector.

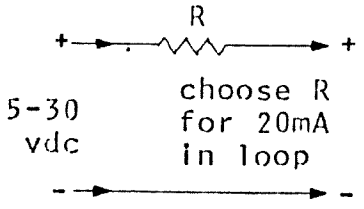
The simplest arrangement is a 2-wire loop (half-duplex). All transmitting and receiving devices are connected in series with a constant current source. Data sent by any transmitter is received by all receivers; thus any key typed on a terminal is immediately printed.

A 4-wire loop (full duplex) adds a second 2-wire loop so that two data transmissions can take place at once. If one station is designated as the master, it can control more than one slave device. The slaves receive only what the master transmits, while the master receives everything transmitted by any slave.

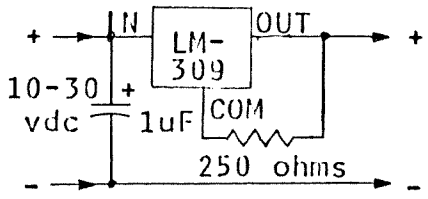
EXTERNAL POWER REQUIRED: 20mA current source; 30vdc max. Two such current sources for full duplex.

DATA: current flowing = 1 = MARK (Idle)
current not flowing = Ø = SPACE (Break)

CONTROL: No separate control lines, Holding a SPACE (current not flowing) for more than a few character times is a BREAK, used to get the attention of a device on the other end. Normal ASCII control codes can then be used.



Simple Current Source



Regulated Current Source

Transmit: The COSMAC's "Q" output is low for "idle" or to transmit a data bit "1". This pulls the base of transistor Q1 (2N4403) low, turning it on. Current then flows through the LED in optocoupler U25 (4N30-35), resistor R5, and through Q1 to ground. Light from the LED turns the phototransistor in U25 "on", and the 20mA from the current loop flows into Serial-Out (P2, pin 38), through U25 and R7, and out V- (P2, pin 35). R7 is chosen to limit the current to 20mA when using a 5vdc power supply for the loop. When the COSMAC's "Q" output goes high, Q1 turns off and current stops flowing through the LED in U25. The phototransistor turns off and interrupts the flow of current in the loop.

Audio FSK (0 to 2.5 KHz)

Receive (see fig. 17): The audio input is connected to Serial-In (P2, pin 40) and Serial-In Common (P2, pin 37). Since the input is optically isolated, the leads are interchangeable and need not be ground referenced. During the positive half cycle (Serial-In + with respect to Serial-In Common) current flows through R6 and the LED in optocoupler U26 (4N28-35). Light from the LED turns U26's phototransistor "on", and pulls the COSMAC's $\overline{EF4}$ input low. During the negative half cycle (Serial-In - with respect to Serial-In Common) current flows through R6 and diode D6, bypassing the LED in U26. The phototransistor turns off, and the $\overline{EF4}$ input of the COSMAC microprocessor is pulled high by its pullup resistor (U18, pin 9). Software can then determine the input frequency by monitoring $\overline{EF4}$.

Transmit: The COSMAC's "Q" output can be programmed to generate a square wave audio output with the appropriate software. The high frequency content (above 2.5 KHz) is attenuated by the frequency response of the optocoupler. When "Q" is low, transistor Q1 (2N4403) is turned "on". Current then flows through the LED in optocoupler U25 (4N26-35), resistor R5, and through Q1 to ground.

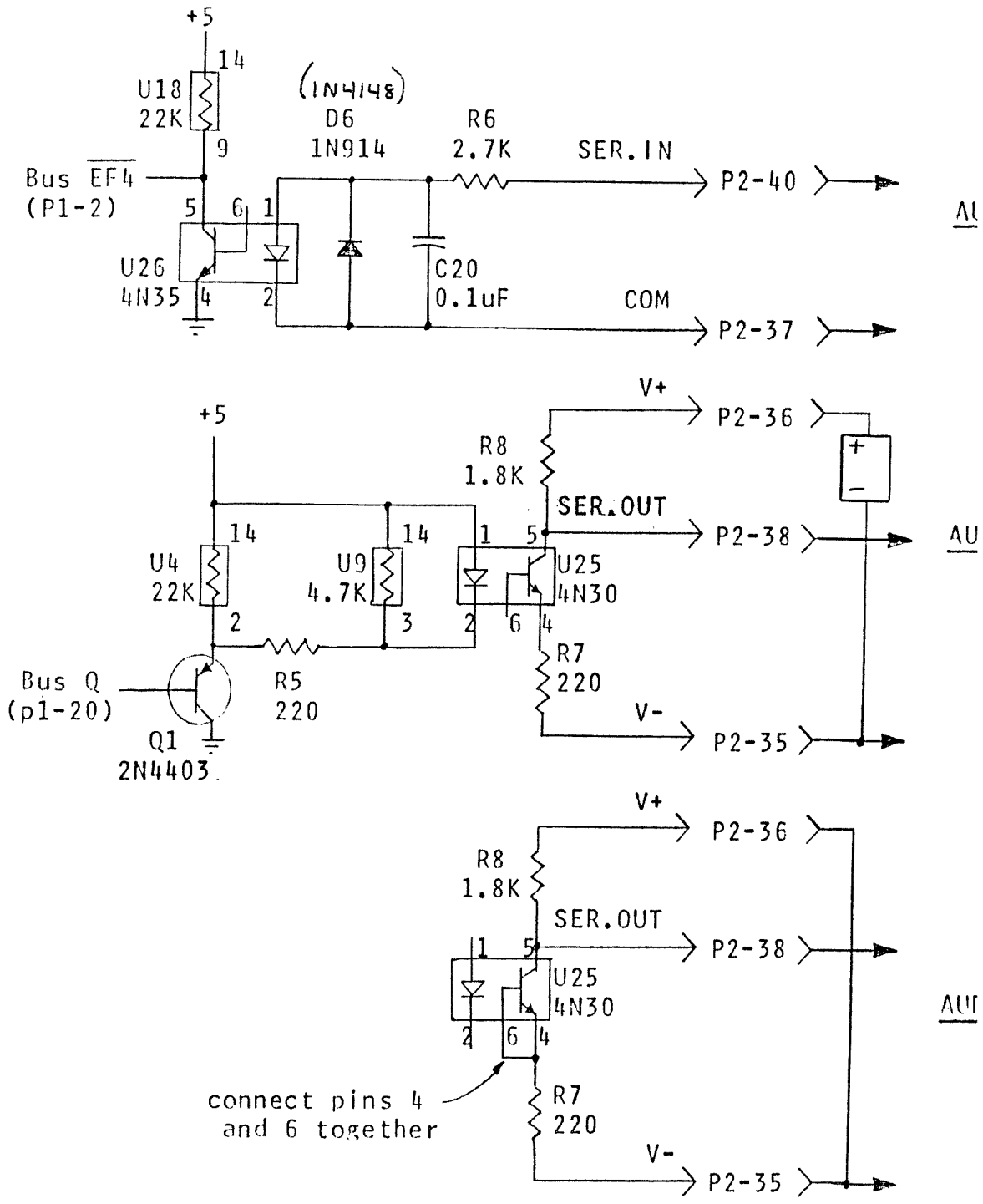
Light from the LED turns the phototransistor "on". When "Q" goes high, transistor Q1 turns off; no current flows through the LED in U25, and the phototransistor then turns off. Sufficient drive is available for small loudspeakers when a photodarlington optocoupler is used for U25.

General Serial Applications: The primary purpose of the serial I/O is to provide at least 1 bit of highly-isolated I/O for data communications in harsh environments. The serial input can withstand up to ±60mA and/or ±25 volts continuously without damage. The serial output can withstand 30 volts in the "off" state and 100mA in the "on" state continuously without damage. The isolation voltage is limited by the spacing between traces on the printed circuit board and should not exceed ±500 volts (peak). Isolation is not sufficient for direct connection to the AC power line.

The optocouplers chosen for U25 and U26 are a compromise between operating speed and input sensitivity. With low input current (RS-232C using ±5 volt power supplies), a high gain coupler must be used (CTR > 50%; 4N29-33, 35-37). High gain couplers tend to be slow, especially if they have darlington phototransistors. The baud rate should be 300 or less for the 4N29-33, 1200 for the 4N35-37. With higher input currents (RS-232C at ±12 volts, current loops, etc.), any optocoupler with a CTR > 10% can be used (4N26-33, 35-37). Baud rates can be up to 2400 baud for the 4N26-28. Speed can also be increased by lowering the resistance of pullup resistor network U18: A 4:1 change (from 22K to 4.7K) will increase the maximum baud rate by approximately 40%.

Resistor R5 sets the LED current to 15mA for optocoupler U25. If BASYS/1 is assembled as an all-CMOS board, this 15mA is approximately 2/3 of the total power consumed. For micropower operation, R5 is increased to 1K, and a darlington optocoupler is used for U25.

fig. 17 - Audio FSK Interface



Audio FSK Interface

AUDIO INPUT

input impedance:	3K ohms
frequency response:	DC - 5KHz
minimum input voltage:	$2V_{rms}$ ($5V_{peak-peak}$)

AUDIO OUTPUT (high-level)

uses external +5 to +24VDC supply

output voltage:	$.9(V+ - V-)-1$ $volts_{peak-peak}$
frequency response:	DC - 5KHz

AUDIO OUTPUT (low-level)

isolated output - no power supply

output impedance:	1.8K ohms
output voltage:	20 $mV_{peak-peak}$
frequency response:	DC - 100KHz

12/4/81

8/10/82

Parallel I/O

Parallel I/O is simple, straightforward, and fast. It assigns an I/O line to every switch, light, motor, relay, or other device that the computer is to monitor or control. Many devices will have more than one line: For example, ASCII is a 7-bit code, so an ASCII-encoded keyboard requires at least 7 parallel input lines. Since the COSMAC is an 8-bit microprocessor, it handles its parallel I/O 8 bits at a time. The BASYS/1 board has one 8-bit parallel input port with optional pullup resistors, and one parallel output port with optional high-power drivers.

Parallel inputs (see fig. 18): The 8-bit parallel input port accepts digital logic levels from TTL or CMOS outputs, open-collector transistors, mechanical switches, or any equivalent sources. Optional pullup resistors can be used, which also help protect against damage from static electricity. The input port on BASYS/1 is read by the COSMAC microprocessor's IN3 instruction. This instruction sets the COSMAC's N lines to N=3 to enable the input port, and writes its contents into the COSMAC's D register and into the memory location specified by R(X).

At the beginning of the IN3 instruction's "execute" cycle, bus SCØ and $\overline{\text{MRD}}$ will go high, and the bus "N" lines (N2, N1, NØ) will change from Ø to a binary 3 (=Ø11). SCØ = 1 goes to the "set" input of flip-flop U20B (1/2 4013) and insures that its Q output is high. N2 = Ø enables decoder U19B (1/2 4556); N1 and NØ are then decoded and the "3" output of U19B goes low. This low is inverted by gate U23A (1/4 4011) and applied to one input of NAND gate U23B (1/4 4011). The other input of U23B (from MRD) is already high, so the output of U23B goes low and enables the data bus drivers of input latch U17 (pin 1, 74C373 or 74LS373). If multiplexed I/O is not being used, the Enable input of U17 (pin 11) will be high, and the data bus (DØ-D7) will change to the l logic levels present on parallel inputs INØ-IN7 (P2, pins 25-32). If multiplexed I/O is being used (see

"Multiplexed I/O"), the Enable input of U17 will be a 1KHz square wave (approximately), and the data bus (DØ-D7) will change to the logic levels present on INØ-IN7 the last time the Enable input of U17 was high. The IN3 instruction loads the data bus into the D register, and simultaneously performs a memory write cycle (bus MWR=Ø) to write this data into memory (refer to the section on RAM memory for a description of the memory write cycle).

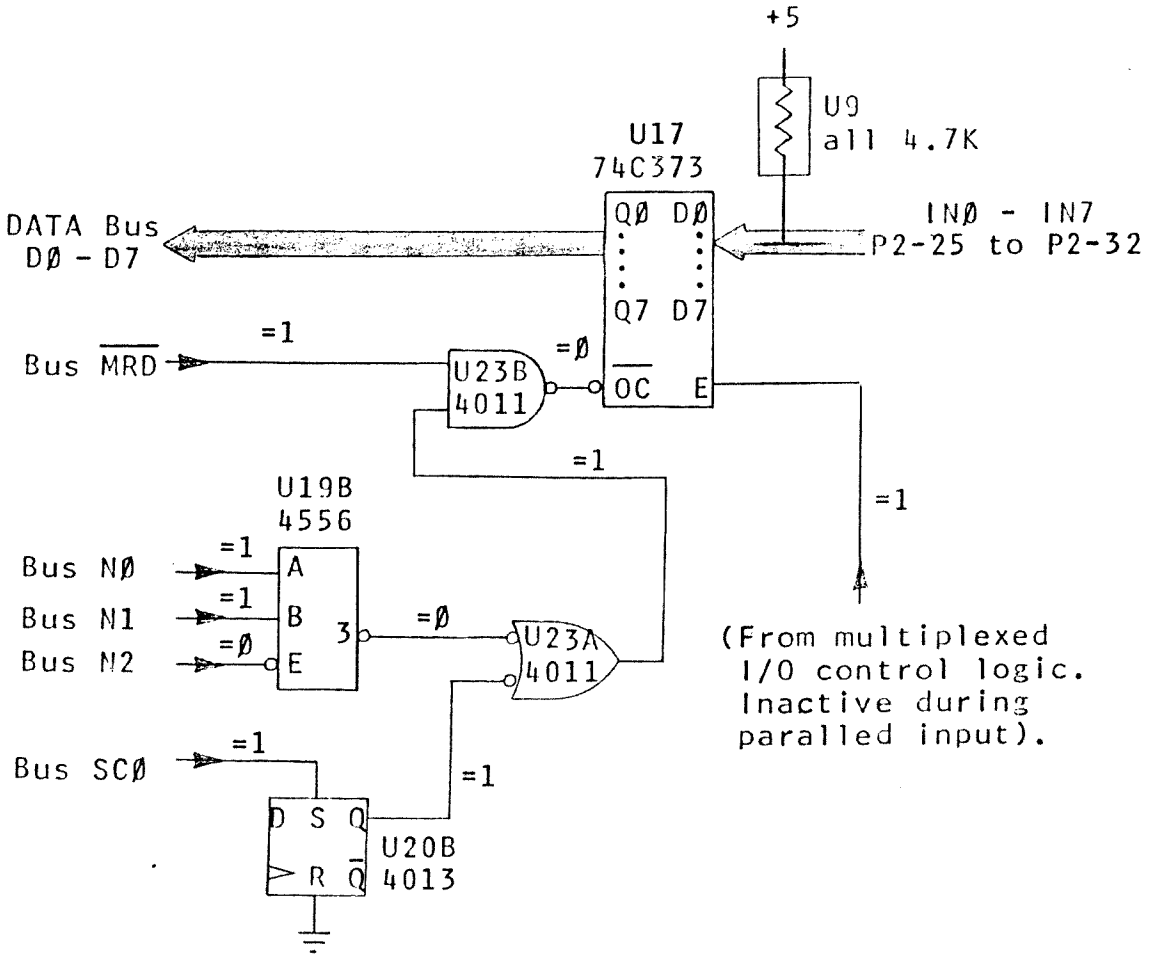


fig. 18 - Parallel Input Port

12/7/81

The input characteristics of IN∅ - 7 are determined by the input latch U17 and pullup resistor network U9. If U17 is a CMOS part (74C373), the equivalent circuit of each input is simply a resistor to +5 vdc. CMOS switching thresholds are quite stable and well-matched between inputs of the same IC, and so IN∅ - 7 can be used as voltage comparators or level detectors for non-critical uses. The pullup resistors serve several purposes: They insure compatibility with any CMOS or TTL device, allow sensing switch contacts or open-collector output status, guarantee that unused inputs will not be left floating, and add protection from static electricity. With CMOS, the input voltage must not exceed the positive supply voltage (usually +5 vdc) or go below ground unless the current is limited to less than 10mA.

A 74LS373 may be used for U17, provided that a +5 vdc ±5% power supply is used. The input characteristics of IN∅ - 7 will now be one low-power Schottky TTL load, plus the pullup resistor. The 74LS373 inputs are compatible with all TTL inputs, but may not be acceptable to some CMOS devices. Some CMOS parts don't have enough drive to reach a TTL "low" logic level, even without a pullup resistor. The 74LS373 inputs are much tougher than CMOS, however: They are diode clamped and can withstand -1.5 to +7 vdc without damage.

Parallel Outputs (see fig. 19): The parallel output port provides eight CMOS and TTL-compatible outputs, each with substantial drive capability. An output driver socket is also provided that can hold series resistors for directly driving LEDs or power transistors, or can hold driver ICs rated at up to ±110 vdc or 600mA per output. Data is written to the output port by the COSMAC's OUT3 instruction. It sets N=3 to select the output latch, and then reads the memory location specified by CPU register R(X). The contents of this memory location is written into the latch, and then R(X) is incremented by one. The parallel output port is also used by the multiplexed I/O: Refer to the next section for details.

An output cycle begins during the "execute" state of the OUT3 instruction. Bus SCØ and MRD go high, and the Bus N lines (N2, N1, and NØ) change from Ø to a binary 3 (=Ø11). SCØ goes to the "set" input of flip-flop U20B and insures that its "Q" output stays high. N2=Ø enables decoder U19B (1/2 4556): NØ and N1 are then decoded and the "3" output of U19B goes low. This low is inverted by U23A and applied to one input of NAND gate U23C (1/4 4011). The other input of this gate connects to Bus TPB. Since TPB is low at this time, the output of U23C remains high.

Meanwhile, Bus $\overline{\text{MRD}}$ goes low, and a memory read cycle begins (see the section on "memory"). When Bus TPB goes high late in the cycle, the data bus will carry the data to be written into the output port from memory. The high on TPB enables U23C, which enables the output of U22A to go high. This enables the output latch U16 to capture the data, thus ending the cycle.

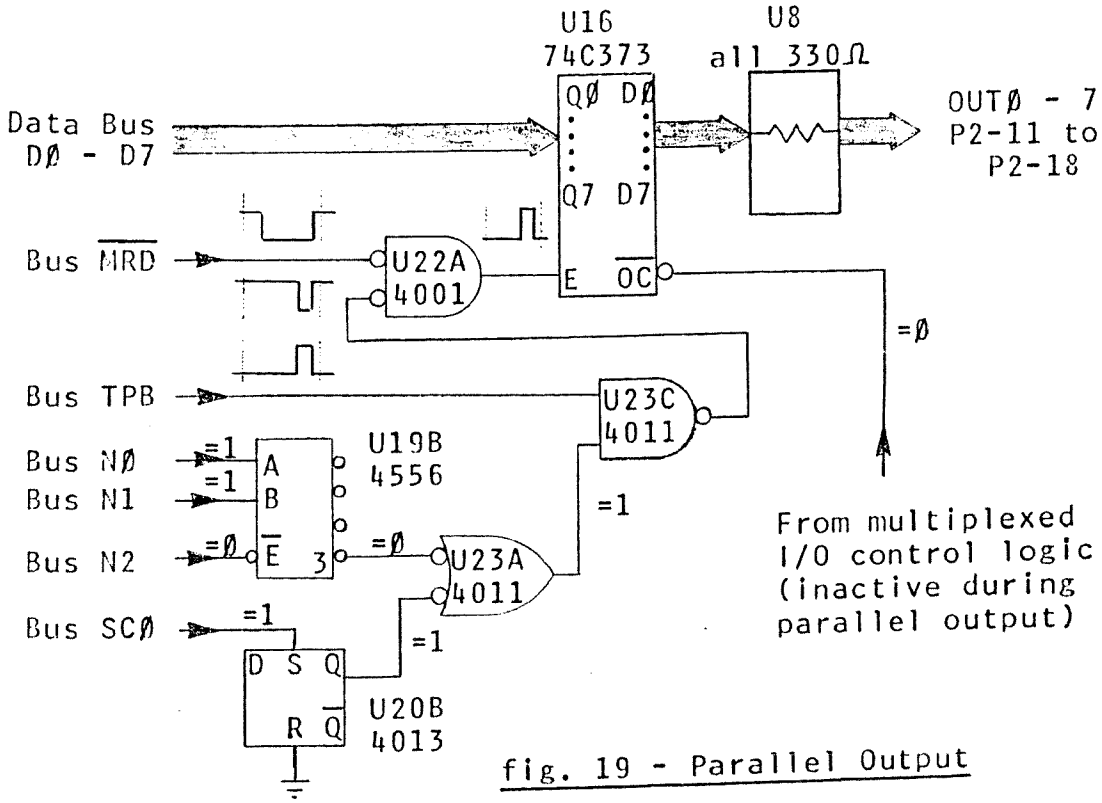


fig. 19 - Parallel Output

12/8/81

With the right choice of components, the output port can directly drive a wide variety of loads (see fig. 20). The CMOS 74C373 has a fan-out of 1 driving standard TTL, and an internal NPN emitter-follower that can source over 20mA at 5 volts. For a modest increase in power supply current, the 74LS373 can increase the fan-out to 10 TTL loads, and source a minimum of 30mA. A 74S373 has even higher drive, but at a much higher power supply current.

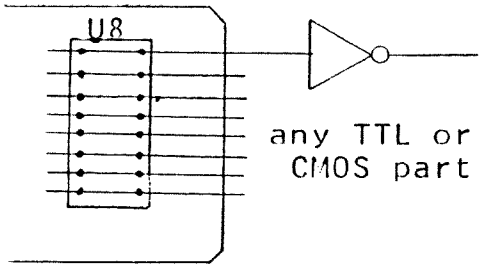
With series resistors at U8, the output port can directly drive LEDs, the base of power transistors, SCR's, and other similar devices. Since the CMOS outputs can go from rail-to-rail (within millivolts of +5 or ground) simple resistor networks can be installed at U8 for D/A (digital-to-analog) conversion.

For higher voltages or currents, a selection of power drivers is available for U8. Most of these contain a darlington power output transistor, and some have snubbing diodes for inductive loads. The output voltage can be as high as 110 volts, and currents up to 600mA per output, either positive or negative. Drivers are also available to meet the special requirements of fluorescent and gas-discharge displays.

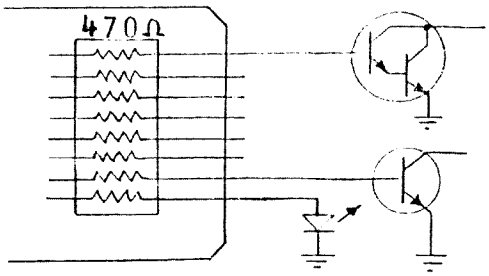
Multiplexed I/O

Multiplexing is a technique for sharing I/O lines between more than one function. It is the most useful when the number of inputs or outputs gets too large to be easily handled by parallel I/O. For example, an 80-key keyboard with a parallel interface would need 80 input lines, or ten 8-bit input ports. Multiplexed I/O solves this problem by arranging the switches into an 8 by 10 matrix, with 10 rows of 8 switches each. Each of the 8 columns goes to one line of an 8-bit input port, and each of the 10 rows is driven by one of 10 output bits. The computer can now select one row of keys at a time, and the input port reads the status of only the keys in that row. Thus the number of I/O lines is reduced from 80 to 18.

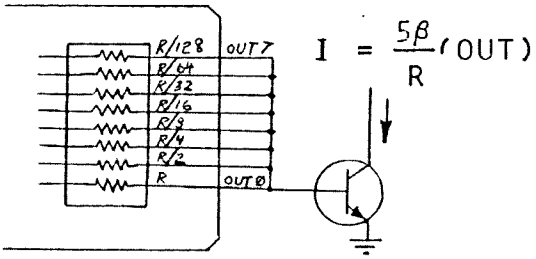
Driving other logic



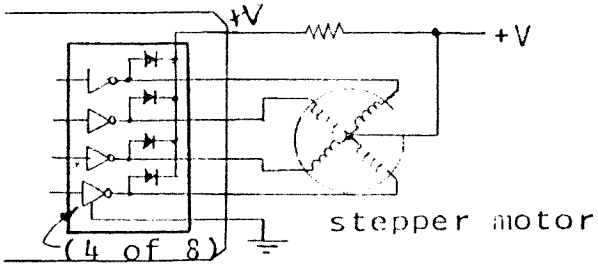
LEDs, transistors



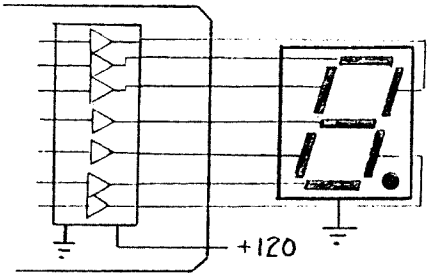
D/A Conversion



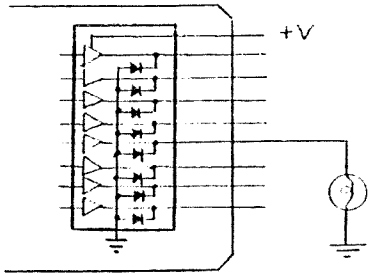
Inductive Loads



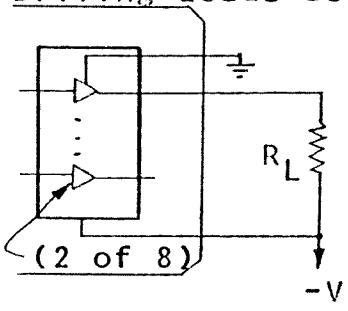
High-Voltage Displays



Ground-Referenced Loads



Driving Loads Below Ground



Driver IC (Sprague #)	Voltage (max)	Current (+=I_{in})
UDN2580A	-50	-500mA
ULN2813A	+50	+600mA
ULN2823A	+95	+500mA
ULN2983A	+80	-500mA
UDN6118A	+85	+40mA
UDN6184A	+120	+70mA
UDN7180A	-110	+20mA
2803	??	500mA

fig. 20 - Parallel Output Applications

12/8/81

BASYS/1 provides multiplexed I/O for 80 inputs (from switches, keyboards, etc.) and for 80 outputs (to 10 digits of 7-segment displays, etc.). The parallel input and output ports are shared between the multiplexed and parallel I/O functions. Special logic has been added to automatically scan the 10 rows and transfer the data to and from memory without software, using the COSMAC's DMA capabilities. This logic also provides a simple hardware timer, useful for real-time applications. The operation of the multiplexed I/O involves inputs, outputs, memory, DMA, and the timer all at once, so it can get quite complicated to follow. Therefore, we'll begin with a simplified explanation, and work up from there.

A simplified block diagram of the multiplexed I/O circuitry is shown in fig. 21. Only four rows are

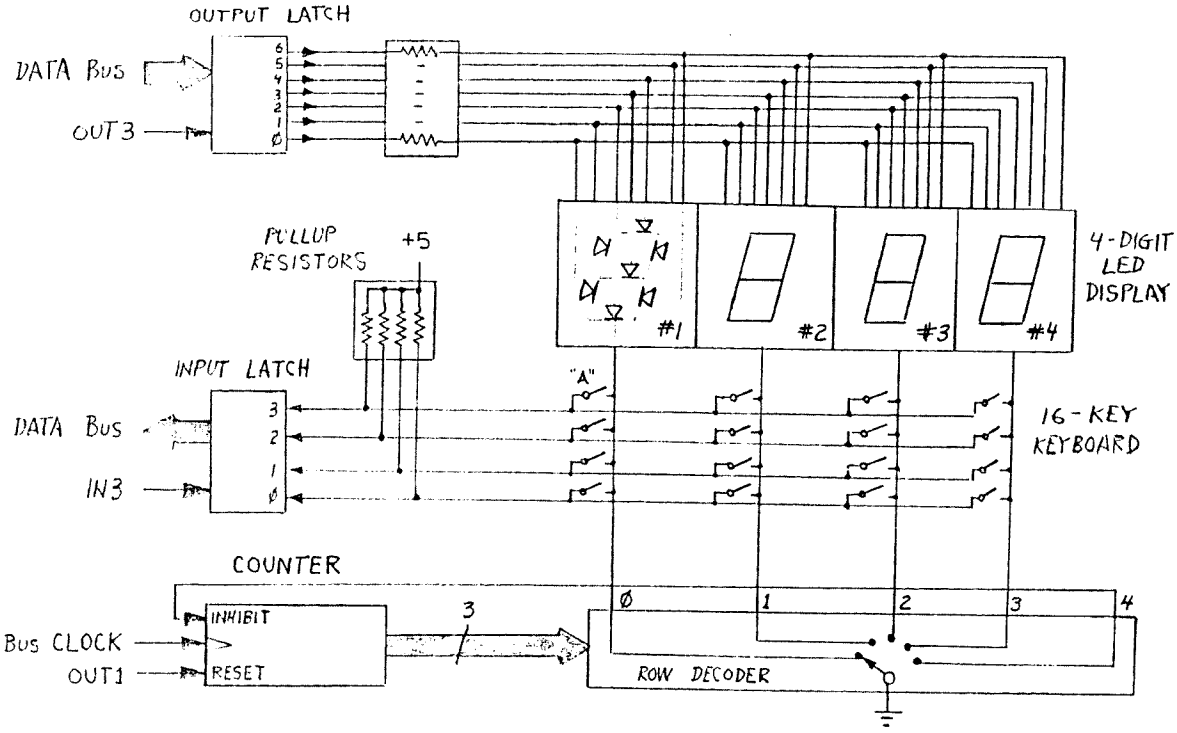


fig. 21 - Multiplexed I/O (Simplified)

shown, and a 16-key keyboard and 4-digit LED display (with 7 segments per digit) are used. Action begins when the computer starts a scan cycle with an OUT1 instruction. This resets the counter to Ø, which in turn sets the row decoder switch to position Ø. The common cathode of the LEDs in digit #Ø is now grounded. The computer can now control each LED in digit #Ø with an OUT3 instruction. To light the lower left segment, for example, the computer would set bit Ø of output port 3 high. Current flows out of the bit Ø output, through the series resistor, the LED, and to ground via row Ø. No LEDs in digits #1 - #3 can light because their cathodes have no path to ground.

Meanwhile, all 4 inputs to the input latch are held high by its pullup resistors. Closing any of the keys in rows 1, 2, or 3 won't change this because they won't provide a path to ground. But if a switch in row Ø is depressed ("A" for example) the corresponding input will be pulled low. The computer can thus read the status of the switches in row Ø with an IN3 instruction.

After a short time (approximately 1 mSec.) the counter will advance, and the row decoder switch will ground row 1. The LEDs in digit #Ø turn off because row Ø is no longer grounded. The computer now outputs the bits for digit #1, and can read the keys in row 1. 1 mSec. later the counter advances again, and the process is repeated for row 2, then row 3, and finally row 4. When row 4 is reached, no LEDs or switches are selected, and the counter is inhibited from advancing any further. The scan is over, and must be started again by another OUT1 instruction from the computer.

There are a number of subtleties involved in making this process work properly. The entire scan must be repeated often enough so that the displays appear to be continuously active, and the keyboard doesn't miss keys. This requires that the scan be repeated at least 30 times per second. Also, the computer must perform the data inputs and outputs at exactly the right time. If for example, the output latch is updated slightly late, the information to be displayed in digit #1 will appear faintly in digit #2 as well. While the COSMAC could maintain such tight control, it would have little time to do anything else. Therefore the multiplexed I/O circuits were modified to minimize the amount of CPU time that was required to maintain a flicker-free display. This was done by using the COSMAC's built-in DMA capability (see fig.22).

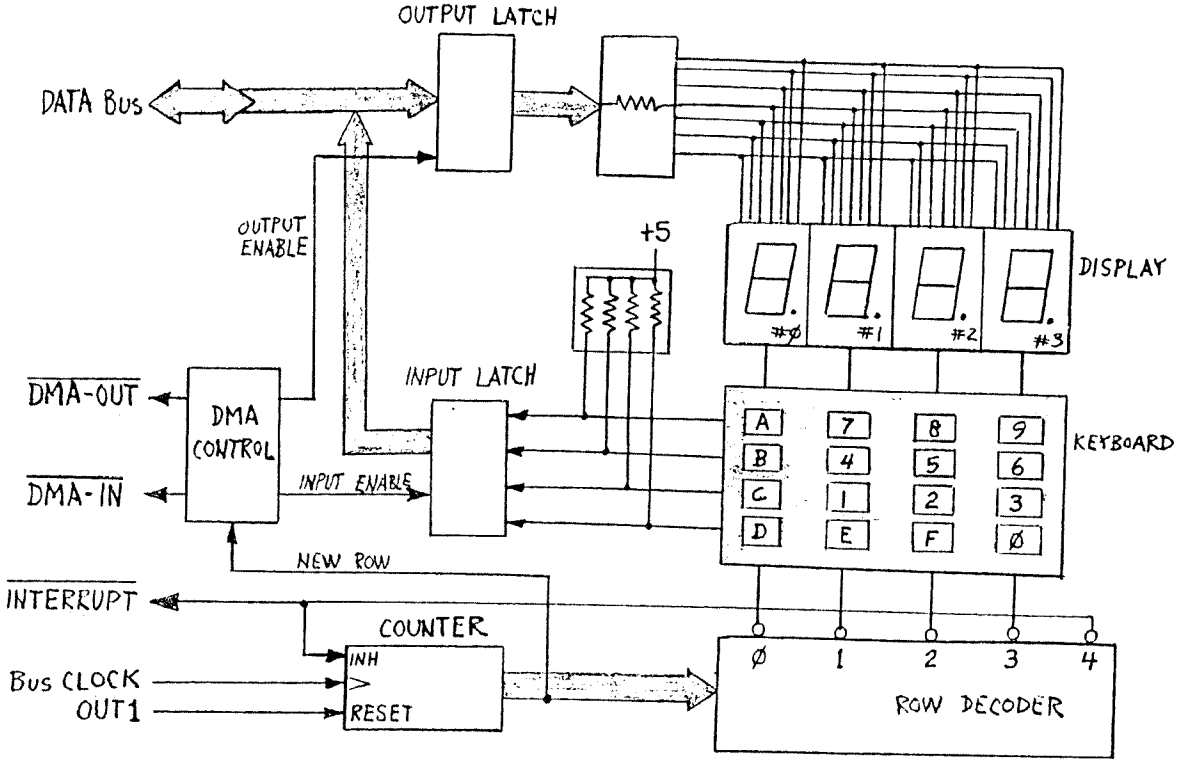


fig. 22 - Multiplexed I/O with DMA (simplified)

DMA (Direct Memory Access) is a way to transfer information directly between memory and I/O without passing through the CPU. The CPU is thus free to execute its program without interruption. DMA works by momentarily suspending CPU operation, doing one (or more) simultaneous memory and I/O cycles, and then letting the CPU resume normal operation. Other than the time lost during the DMA cycles, the CPU is unaware anything unusual has happened. The COSMAC has two DMA control inputs, DMA-IN and DMA-OUT. Register RØ is used as a pointer to the next memory location for DMA. When DMA-OUT goes low, the COSMAC reads the location specified by RØ and outputs it on the data bus. DMA-IN inputs data from the data bus to the location specified by RØ. RØ is incremented after each DMA cycle so successive DMA cycles can be done.

Now we're ready to explain fig. 22. In this simplified example, the COSMAC sets up an 8-byte table in RAM, and points RØ to its base (lowest address). This table will become memory-mapped I/O for the keyboard and display status. Bytes 1, 3, 5, and 7 in the table are set to the values to be displayed in digits #Ø, #1, #2, and #3 respectively. Bytes 2, 4, 6, and 8 are for the keyboard status, and need not be initialized. The COSMAC then starts a keyboard/display scan with an OUT1 instruction as before. OUT1 resets the counter to Ø, and the row selector grounds row Ø. The counter also sends a NEW ROW signal to the DMA control logic. It immediately performs a DMA-OUT cycle to read byte 1 in the table (= digit #Ø) and place it on the data bus. The OUTPUT ENABLE signal captures it in the output latch, where it is displayed in digit #Ø. Next the DMA logic performs a DMA-IN cycle. INPUT ENABLE reads the status of the keys in row Ø and places it on the bus: The DMA-IN cycle then writes it into byte 2 in the table (remember, RØ was incremented after the previous DMA cycle).

About 1 mSec. later the counter advances to row 1, and another NEW ROW signal is generated. The DMA control logic performs a DMA-OUT cycle from byte 3 in the table to display digit #1; then a DMA-IN cycle from key row 1 to byte 4 in the table. This process is repeated for rows 2 and 3. When row 4 is reached, the scan ends; it inhibits the counter from advancing any further and generates an interrupt to the CPU. The interrupt tells the computer that the scan is finished, and that the table in RAM now contains the latest keyboard status. To maintain scanning, the interrupt handler need only reset RØ to the base of the table and execute another OUT1 instruction. And since each interrupt follows the OUT1 by a precise number of cycles, it can be used as a real-time clock.

But we still need a few more improvements to make this a practical circuit. First, there is always some delay between selection of a new row and the DMA-OUT cycle with data for that row. This would cause the data from the previous digit to be momentarily displayed in the new digit. To fix this, we must disable the output latch's outputs during the row change. Second, it isn't practical to read a row of keys immediately after selecting it -- many keyboards just aren't that fast. We should read the key status at the end of the row select time. Finally, the circuit must work with the Bus WAIT and CLEAR signals, and not conflict with other uses of DMA-IN, DMA-OUT, and INTERRUPT.

The final circuit diagram of the multiplexed I/O is shown in fig. 23. It has been expanded to handle 10 rows with 8 bits per row, and uses a 22-byte table in RAM for DMA. As before, an OUT1 instruction begins a scan cycle. It sets the Bus "N" lines to 1 (N2=Ø, N1=Ø, NØ=1) and then pulses Bus MRD low. Decoder U19B decodes the "N" lines and sets its "1" output low to enable one input of gate U22B. The Bus MRD pulse enables the other input, so the output of U22B (labelled "OUT1") is a positive-going pulse. This pulse goes two places: It resets the scan counter, and starts the DMA control logic.

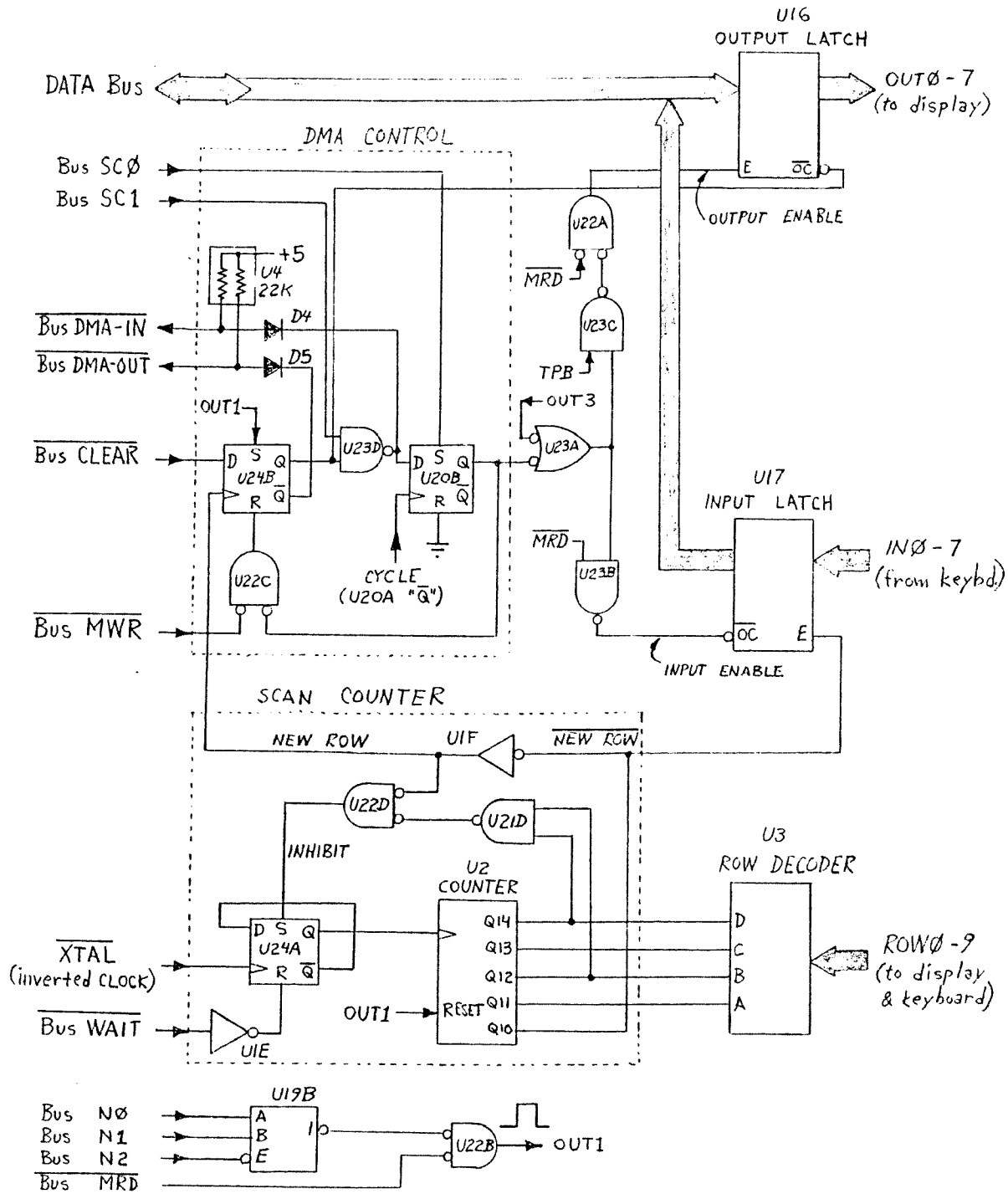


fig. 23 - Multiplexed I/O with DMA

12/15/81

The scan counter is a straightforward binary counter, running at the CPU's clock frequency. It contains an initial divide-by-two stage (U24A) followed by a 14-bit binary ripple-counter (U2). Initially, U2 will be outputting Q10=1, Q12=1, and Q14=1. The INHIBIT signal will be high, U24A will be held "set", and the XTAL clock pulses will be ignored. OUT1 resets the outputs of U2 to Ø; INHIBIT goes low and U24A is free to count (assuming that Bus WAIT is also high). On each subsequent clock pulse, U24A inverts its own output, since its "D" input is tied to its "Q̄" output. The "Q" output is therefore a square wave at 1/2 the XTAL clock frequency. U2 counts these pulses, and its most significant 4 bits (Q11 - Q14) are the Row number. It starts at Ø (following OUT1) and is incremented once every 2048 XTAL clock pulses (1.024 mSec. at 2 MHz). The Q10 output of U2 is the NEW ROW signal: It pulses once per row, and has its falling edge just prior to each change in the Row number.

Once started, the scan counter advances from Row Ø through Row 9. Row decoder U3 decodes each Row number and grounds the selected row of the keyboard and display (ROWØ - ROW9). At the end of each row, the falling edge on NEW ROW captures the status of that key row in input latch U17. Finally Row 10 is reached. Decoder U3 has no ROW10 output, so no key row or display digit is selected. Q12=1 and Q14=1, so the output of NAND gate U21D goes low. U2 continues to count until Q10=1. This is inverted by U1F (so NEW ROW =Ø). Both inputs to U22D are now low, so INHIBIT goes high. The scan counter is now inhibited from counting any further, and the scan is over.

The DMA control logic is responsible for performing two DMA cycles at the beginning of each new row. The first is a DMA-OUT cycle to update the output latch, and the second is a DMA-IN to read the input latch. The timing for these cycles is shown in fig. 24.

The OUT1 pulse starts the process by setting the DMA Request flip-flop (U24B). Its "Q" output goes high, which enables one input of gate U23D, and disables the outputs of latch U16 (OUTØ - OUT7) until it can be loaded with data for the new row. Simultaneously the "$\overline{Q}$" output of U24B goes low, and pulls Bus $\overline{\text{DMA-OUT}}$ low through diode D5. The CPU completes its current instruction, and then performs the requested DMA-OUT cycle: Bus $\overline{\text{MRD}}$ goes low, the location specified by RØ is read and placed on the data bus, and the bus state code lines indicate a DMA cycle (Bus SCØ=Ø, Bus SC1=1).

The DMA Acknowledge flip-flop (U20B) is normally "set", and is waiting for this DMA cycle. When it occurs, the low on Bus SCØ removes the "set" input, and the high on Bus SC1 enables the remaining input of gate U23D to make the "D" input low. (It also pulls Bus $\overline{\text{DMA-IN}}$ low through diode D4 to make the following cycle a DMA-IN.) Shortly after TPA the rising edge of CYCLE (from the RAM control logic) clocks U20B, and its "Q" output goes low. This enables the output of gate U23A, just as if an OUT3 instruction were being executed. Thus the information on the data bus is written into the output latch (U16) as described earlier for the parallel output port. At the end of the DMA-OUT cycle, Bus $\overline{\text{MRD}}$ returns high and RØ is incremented.

The CPU next performs a DMA-IN cycle (although Bus $\overline{\text{DMA-IN}}$ and $\overline{\text{DMA-OUT}}$ are both low, DMA-IN has a higher priority). The state code lines continue to indicate a DMA cycle, the Bus $\overline{\text{MWR}}$ line goes low, and a memory write cycle is performed to the memory location specified by RØ. The data to be written into memory comes from input latch U17: The "Q" output of U20B is still low and Bus $\overline{\text{MRD}}$ is high, so the input latch places its contents on the data bus just as if an IN3 instruction were being executed. Additionally, the low on Bus $\overline{\text{MWR}}$ enables the output of gate U22C to go high. This resets the DMA Request flip-flop (U24B) and enables the outputs of output latch U16 (OUTØ - OUT7).

It also lets Bus $\overline{\text{DMA-OUT}}$ and Bus $\overline{\text{DMA-IN}}$ be returned high by the pullup resistors in U4. At the end of the DMA-IN cycle, Bus $\overline{\text{MWR}}$ returns high and RØ is incremented. The "D" input of U20B is high, so it will be set to its idle state at the next rising edge of CYCLE. The first pair of DMA cycles is now completed, and the CPU resumes normal program execution where it left off.

This process is repeated at the end of each successive row. Each time the scan counter advances to the next row, $\overline{\text{NEW ROW}}$ goes low. This is inverted by U1F into a positive-going edge, which sets the DMA Request flip-flop (U24B) to begin another pair of DMA cycles. When the scan counter finally reaches row 10 and inhibits itself, a total of 22 DMA cycles have been performed: 2 for the initial OUT1, and 2 more for each of the 10 rows.

An entire scan takes 11 x 2048 = 22528 clock cycles (11.264 mSec. at 2 MHz). To maintain scanning to produce a continuous display, the computer will have to execute OUT1 instructions this often. There are two ways of accomplishing this: With a software polling technique, or with interrupts. Polling is the most straightforward approach. The computer starts a scan, and then goes on to perform the next task. When done, it again checks to see if the scan is finished, and if it is, it is restarted. If the scan is not done, the computer waits for the end, and then restarts it. While simple, there are two drawbacks: (1) tasks being performed between scan restarts must take less than the time for a scan, and (2) the excess time after each task is lost waiting for the current scan to end. The status of the current scan is read by looking at RØ.

Interrupts may be used to maintain scanning without any significant loss of time. This is done by tying the Bus $\overline{\text{INT}}$ or I/O connector $\overline{\text{INT}}$ line to $\overline{\text{ROW9}}$. Use of the I/O connector $\overline{\text{INT}}$ allows multiple interrupt sources for larger system applications. There will be precisely 2048 x ROW# clock cycles from OUT1 to $\overline{\text{INT}}$. Register RØ functions as a status register to determine the source of the interrupt if multiple sources are used.

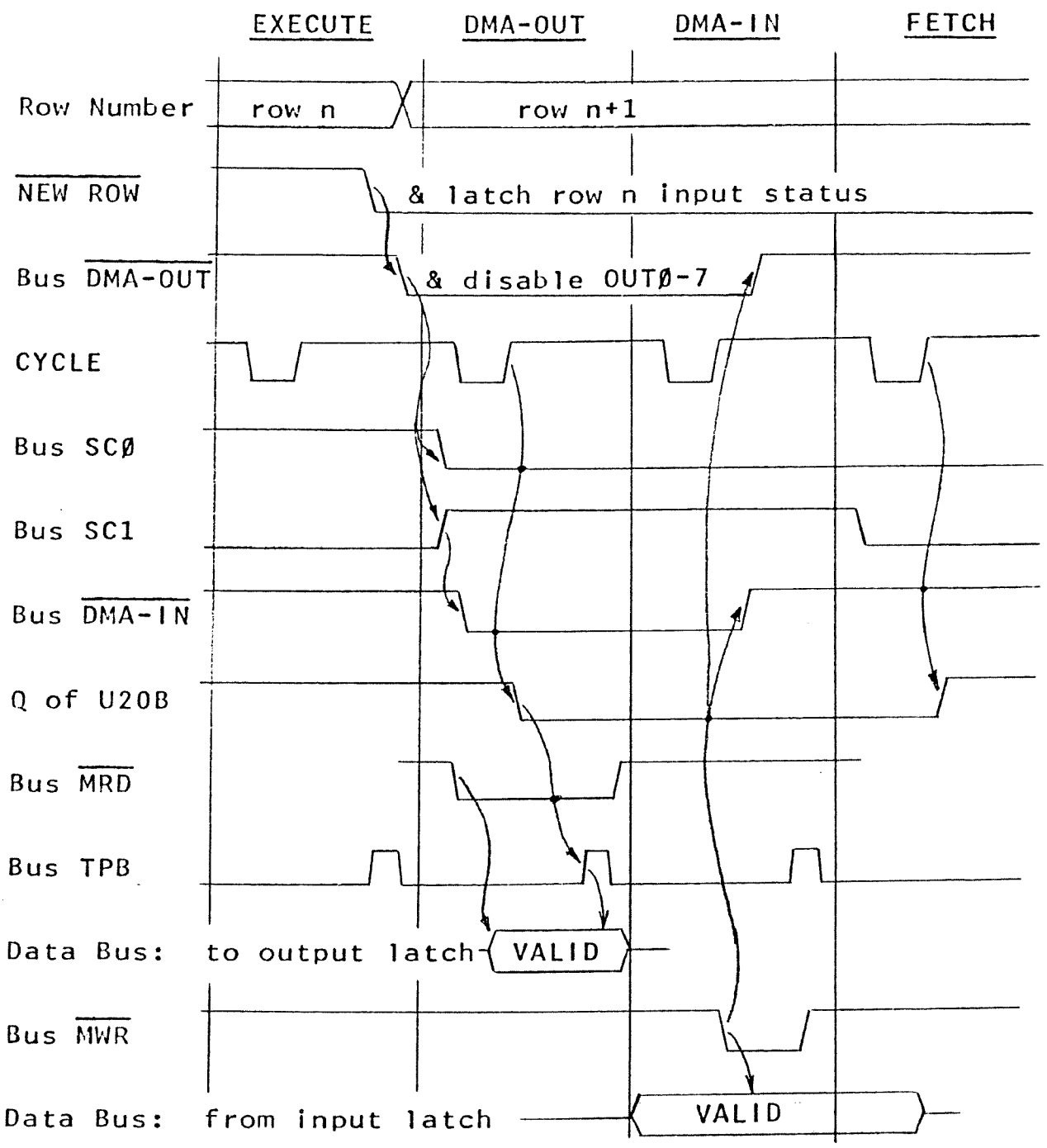


fig. 24 - Multiplexed I/O DMA Timing

12/24/81

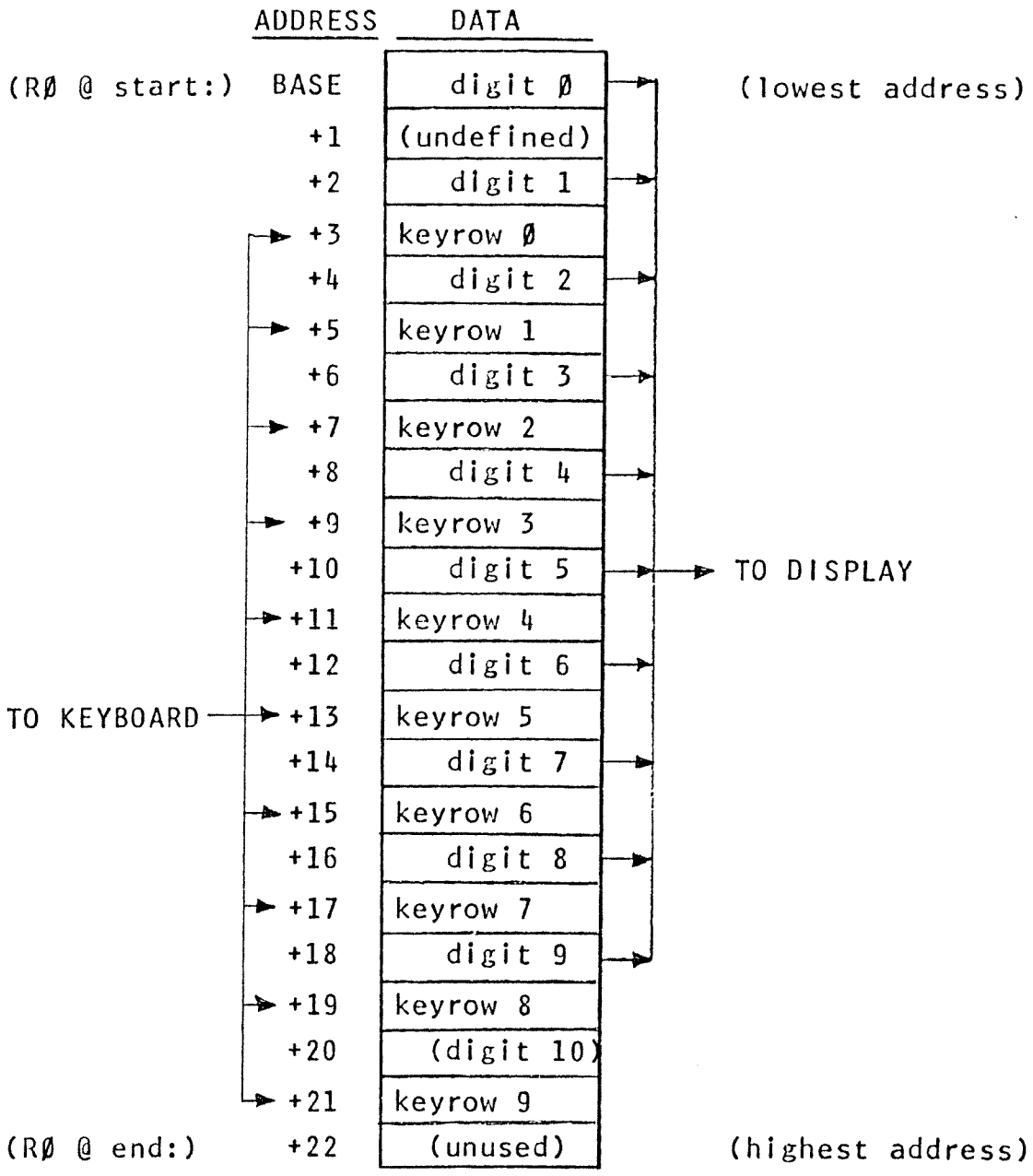


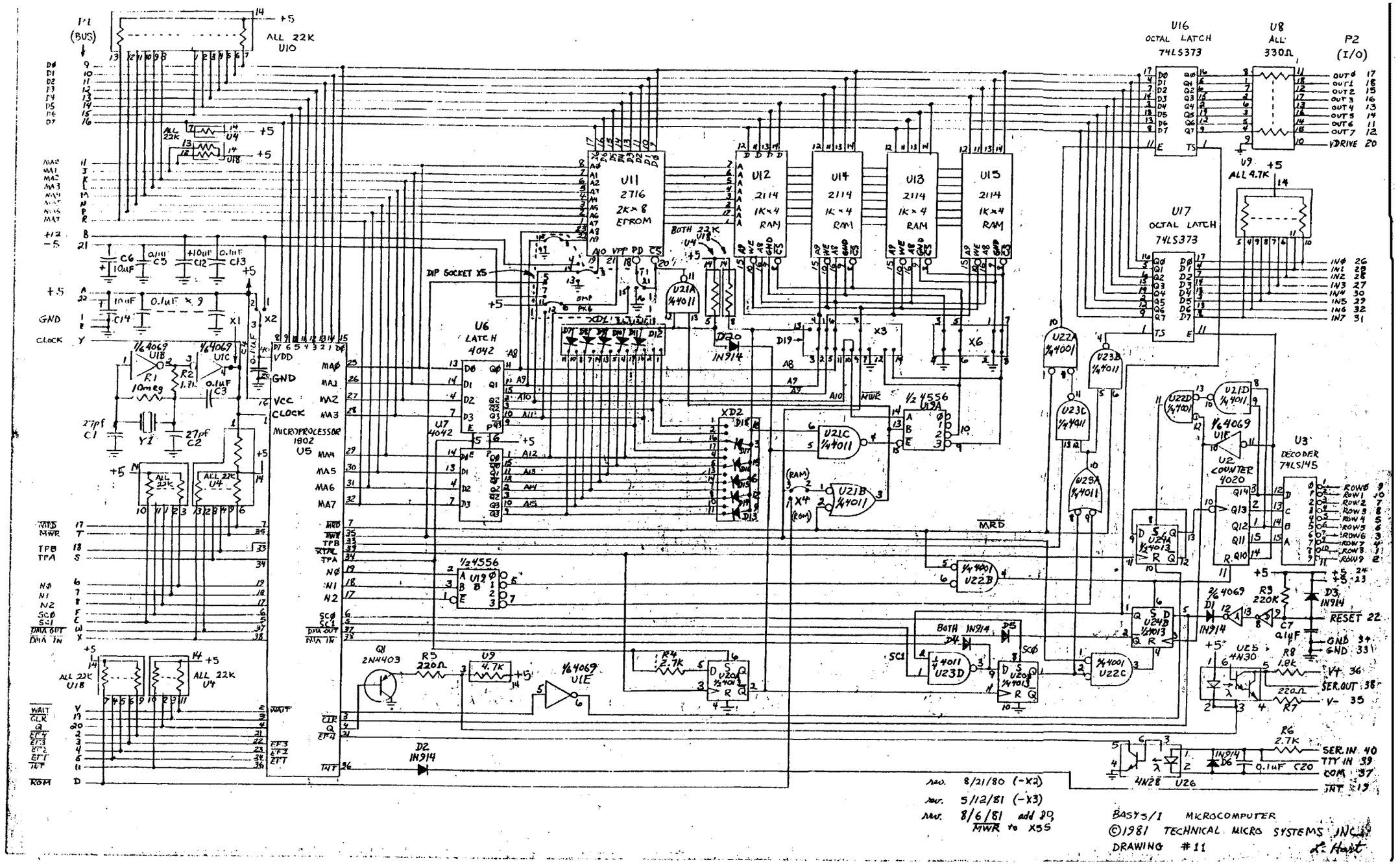
Fig. 25 - Typical DMA Table for Multiplexed I/O

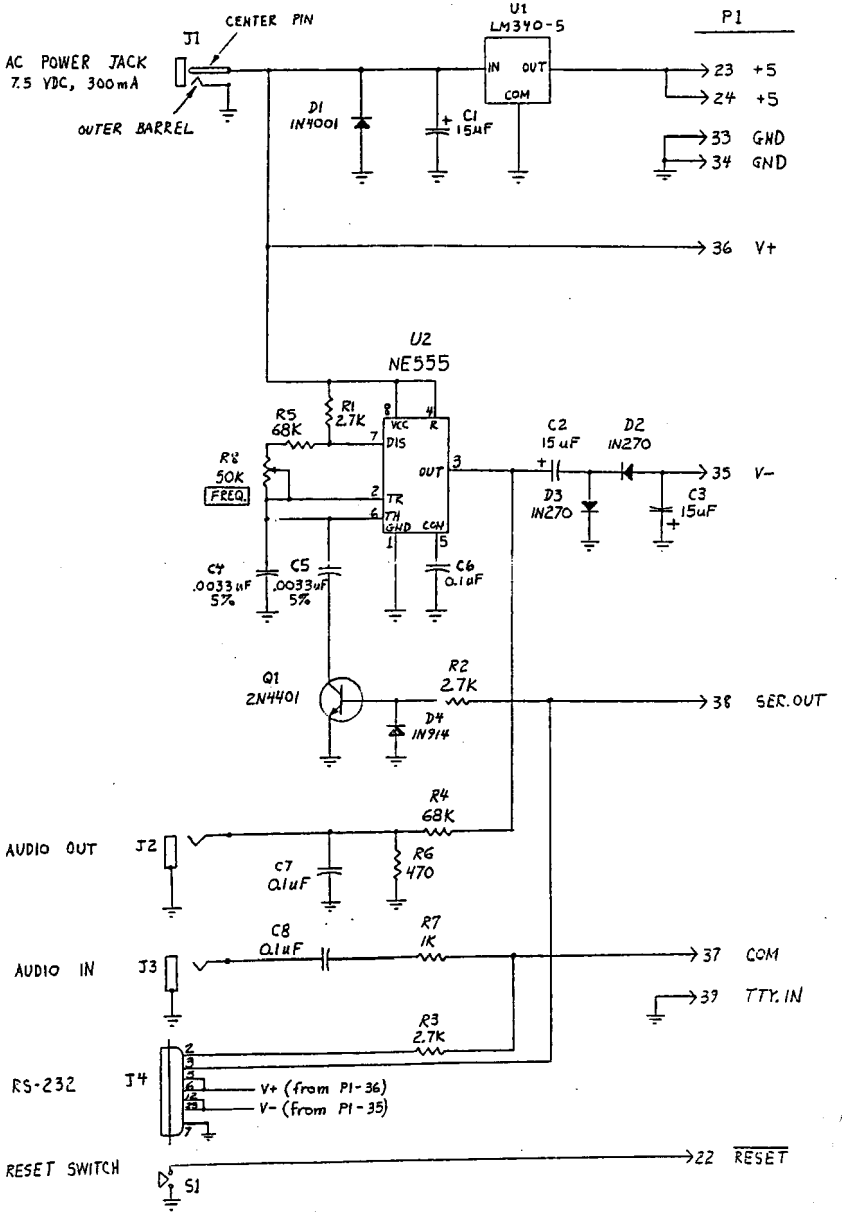
There is one potential source of problems if multiple interrupt sources are used with multiplexed I/O interrupts. If a bus interrupt cycle (Bus SCØ=1, SC1=1) happens to occur just prior to the DMA-OUT cycle for any row, the DMA-OUT cycle will NOT BE PERFORMED. All subsequent DMA's for that particular scan will then be off by one. The odds of this happening are very low, but nevertheless possible. The consequence of this situation would be a momentary flicker in the display, and one scan of erroneous data from the keyboard. The keyboard data should be debounced anyway, so it should not cause errors.

Fig. 25 shows the typical DMA memory assignments for multiplexed I/O. At the beginning of the scan RØ points to the base of the DMA table, which contains the byte to be displayed by the digit in row Ø. Bytes with an even-numbered offset (2,4,6, etc.) contain the bytes for successive digits. After a DMA scan, the contents of the address Base+3 contains the status of the keys in row Ø. Successive odd-numbered addresses contain the status of the successive key rows. At the end of a scan, RØ is left pointing at Base+22. The contents of Base+1 is undefined: It holds the status of INØ-IN7 as read at the time of the OUT1 instruction, when no row was selected. Similarly, the byte at Base+20 will be output for row 10 when no $\overline{\text{ROW}}$ output is active.

The row decoder (U3) is normally a 74C42 for CMOS boards, or a 74LS145 when low power is less important. The CMOS part will drive only small calculator-type displays, or off-board drivers. The 74LS145 has high-current open-collector outputs that can sink up to 24mA at V_{out}=0.5V, or 80mA at V_{out}=3v (max.), and up to 15v. in the off state. The 74LS445 is similar, but good for 7v max.

The -X1 revision boards have only a 9-row multiplexed I/O, and U2 is a 12-bit counter. Each row takes 512 clock cycles, and a complete scan takes 10 x 512 = 5120 cycles. The $\overline{\text{ROW9}}$ output is low at the end of a scan.


P1 (BUS)
P2 (I/O)
U5 MICROPROCESSOR 1802
U6 LATCH 4042
U7 4042
U8 ALL 330Ω
U9 ALL 4.7K
U10 ALL 22K
U11 2716 2K×8 EPROM
U12 2114 1K×4 RAM
U13 2114 1K×4 RAM
U14 2114 1K×4 RAM
U15 2114 1K×4 RAM
U16 OCTAL LATCH 74LS373
U17 OCTAL LATCH 74LS373
U2 COUNTER 4020
U3 DECODER 74LS145
DIP SOCKET X5
Y1
27pf C1
27pf C2
10meg R1
1.2k R2
0.1uF C3
10uF C4
0.1uF C5
10uF C6
10uF C12
0.1uF C13
10uF C14
0.1uF x 9
X1
X2
X3
X4 (RAM)
X6
D1 IN914
D2 IN914
D3 IN914
D4 IN914
D5 IN914
D6 IN914
Q1 2N4403
R3 220Ω
R4 2.7K
R5 220Ω
R6 2.7K
R7 220Ω
R8 1.8K
C7 0.1uF
C20 0.1uF
U25 4N30
U26 4N28
RESET 22
SER.OUT 38
SER.IN 40
TTY IN 39
COM 37
INT 19
rev. 8/21/80 (-X2)
rev. 5/12/81 (-X3)
rev. 8/6/81 add IO, MWR to X55
BASYS/1 MICROCOMPUTER
©1981 TECHNICAL MICRO SYSTEMS INC.
DRAWING #11
A. Hart



PARTS LIST

U1	REGULATOR, 5V. NSC LM340T-5.0 OR EQ.
U2	TIMER, NSC LM555CN OR EQ.
C1-3	CAPACITOR, 15 uF, 16 VDC, TANTALUM, TMSI 156CT160
C4,5	" 0.0033 uF 5%
C6-8	" 0.1 uF, TMSI 104C250
D1	DIODE, TMSI IN4001
D2,3	" , 100 PIV, 200 mA, GERMANIUM, IN270
D4	" TMSI IN914
J1	CONNECTOR, POWER JACK, SWITCHCRAFT PC-712A OR EQ.
J2,3	" MIN. PHONE JACK, SWITCHCRAFT PC-142A OR EQ.
J4	" RS-232, FEMALE, AMP HDP-20 SERIES, P/N 205738-1
Q1	TRANSISTOR, TMSI 2N4401
P1	HEADER, 40-PIN, 2 ROWS OF .025" SQ. POSTS ON 0.1" CENTERS
R1-3	RESISTOR, 2.7K 5% 1/4W CARBON FILM
R4,5	" 68K " " " "
R6	" 470Ω " " " "
R7	" 1K " " " "
R8	POTENTIOMETER, 50K, 20%. 1 TURN CERMET, 3/8" SQ. SPECTROL 63P503, BOURNS 3386P-1-503 OR EQUIV.
S1	SWITCH, MOM. N.O. PUSHBUTTON

NOTES

1. PARTS SPECIFIED FOR 0-70°C OPERATION. FOR -40 ~ +85C, USE U1 = LM140T-5.0 AND U2 = LM555J
2. J1-4 AND S1 ARE CHOSEN FOR SAME HEIGHT ABOVE BOARD.

9/8/82 - ADDED V+ AND V- CONNECTIONS TO PINS 5, 6, 12, AND 25 OF J4

PROTEUS I/O BOARD

SCALE:	APPROVED BY	DRAWN BY C. Hunt
DATE: 7/12/82	P. Hunt 7/12/82	

TECHNICAL MICRO SYSTEMS, INC.
366 CLOVERDALE · ANN ARBOR, MI. 48105

SCHEMATIC DIAGRAM & PARTS LIST

DRAWING NUMBER 34

Technical Micro Systems, Inc.
90 Day Limited Warranty

All Technical Micro Systems, Inc. parts and products carry a ninety (90) day limited warranty. If you have any questions or problems with our products, parts, or service please call or write to us at the address given below and we'll do our best to help. Sorry, no collect calls.

For a period of ninety (90) days after purchase, Technical Micro Systems, Inc. will repair or replace free of charge any part or product found defective in either materials or workmanship. We further warrant that our products will meet the published specifications for a period of ninety (90) days after purchase when assembled and used according to our published instructions. If a defective part or a design error on our part causes a part or product to fail during the warranty period, we will repair or replace it free of charge. Send it to us postpaid and we will reimburse you for the shipping charges.

Our warranty does not cover and we are not responsible for damage caused by abuse, incorrect assembly, defective tools, or unauthorized modifications. Our warranty does not include reimbursement for inconvenience, loss of use, or customer labor.

This warranty includes only parts and products sold by Technical Micro Systems, Inc. and does not cover equipment to which our parts and products may be connected. We are not responsible of accidental or consequental damages. Some states do not allow exclusion of incidental or consequental damages, so this may not apply to you.

Technical Micro Systems, Inc.
366 Cloverdale
Ann Arbor, MI 48105

(313) 994-0784

NOTES