NATIONAL RADIO ASTRONOMY OBSERVATORY
GREEN BANK, WEST VIRGINIA

ELECTRONICS DIVISION INTERNAL REPORT No. 237

THE GPIO BUS INTERFACE UNIT
FOR THE
HP 9826 COMPUTER

RICHARD F. BRADLEY

OCTOBER 1983

NUMBER OF COPIES: 150

THE GPIO BUS INTERFACE UNIT FOR THE HP 9826 COMPUTER

Richard F. Bradley

TABLE OF CONTENTS

THE GPIO BUS INTERFACE UNIT FOR THE HP 9826 COMPUTER

Richard F. Bradley

INTRODUCTION

The GPIO Bus Interface Unit was designed to be a versative interface between the HP 9826 computer and the Universal Local Oscillator System. The interface also provides eight TTL switch outputs for general control applications. The modular design of the interface permits direct expansion for future projects.

This report describes the design and use of the GPIO Bus Interface Unit. Detailed information about the circuit design, unit construction and software support are provided in this report. Information concerning the circuit board wire layout is not included here but is available elsewhere.

CIRCUIT DESIGN DETAILS

The circuit for the GPIO Bus Interface Unit is divided into three sections. The GPIO BUFFER CARD contains all the necessary buffers and pull-up resistors for the data and control lines of the GPIO BUS. A path enable circuit and a pulse power-up circuit is also included on this card. The ULO DECODE CARD contains all the circuitry necessary to pass a group of ten 16-bit words to the Universal Local Oscillator System. The GENERAL PURPOSE OUTPUT (GPO) CARD contains the eight bit latch and supporting circuits for the buffered TTL switches. Two spare card locations are provided for future expansion. A block diagram indicating data flow is shown in Figure 1. A detailed discussion of each card now follows.

BLOCK DIAGRAM

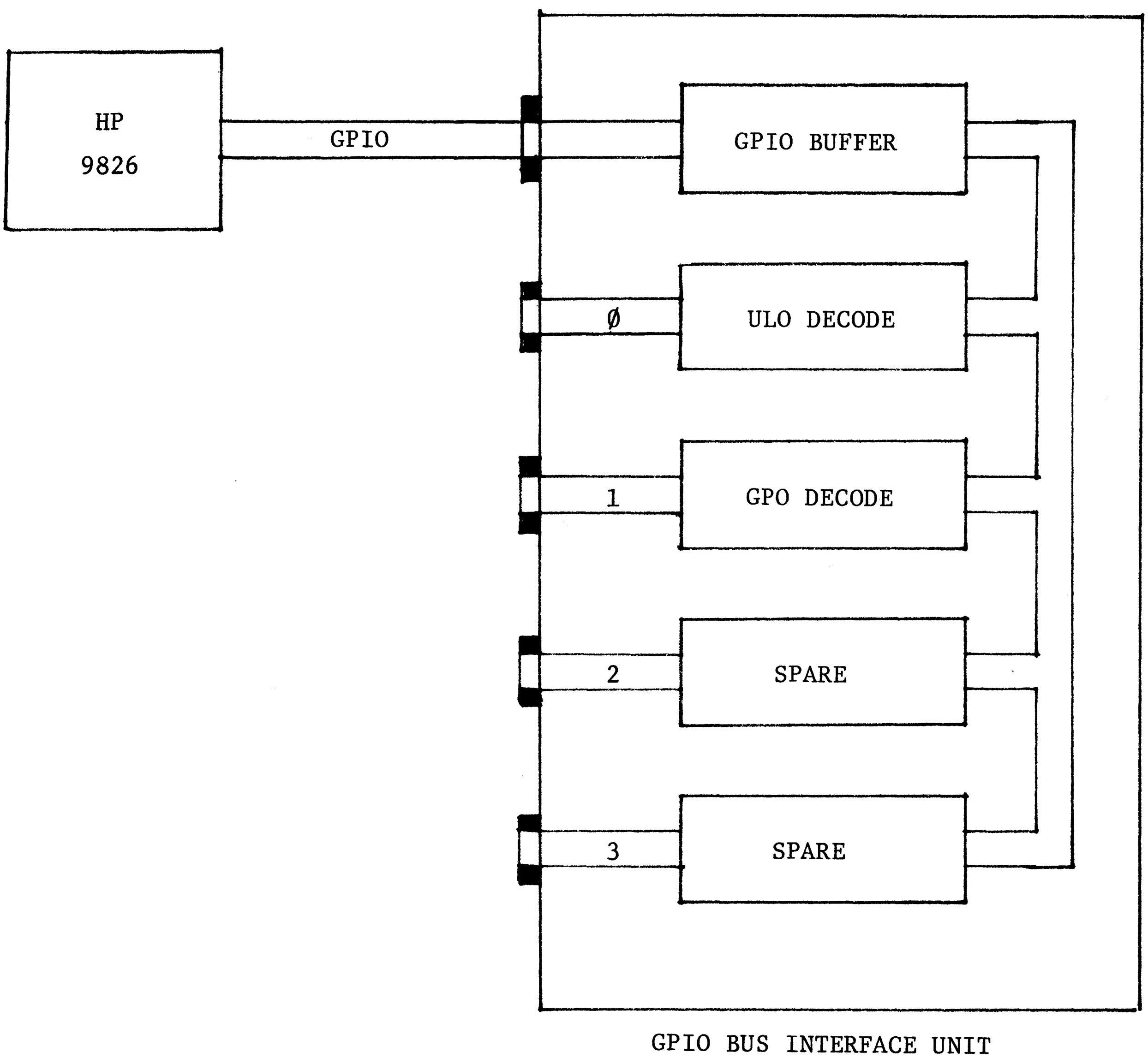


Figure 1

A. GPIO Buffer Card

A schematic diagram for the GPIO Buffer card is shown in Figure 2. The part list is given in Table 1. The "DO" lines, which transfer 16 bits of data from the computer to the external device, contains leveling resistors U1, U2, U3 and U4. Buffers U12, U13 and U14 are used to provide computer isolation for these lines. The "DI" lines directing 16 bits of data from the external device to the computer, are buffered by U6, U7 and U8 for isolation. The PCTL, I/O, PRESET, CTLØ and CTL1 control lines have leveling resistors U10 and U11 and buffers U14 and U9.

A binary-to-decimal decoder, U5, generates four enable signals from the CTLØ and CTL1 lines. These signals are used to enable the ULO decode and GPO output circuits. The power-up pulse circuit, U15 and U16, shown in Figure 3, generates a pulse on pin 5 at INITIAL POWER UP. The pulse is used to reset the GPO and ULO decode circuits for initialization. The GPIO lines from the computer are connected to the GPIO buffer card through three ribbon wire lines. Figure 4 shows the physical layout of the IC chips for this board. Note the ribbon connections:

DATA IN (DI) LINES ... 4F
DATA OUT (DO) LINES ... 4B
CONTROL LINES 4D

All other connections to this card are through the card edge connector.

GPIO BUFFER CARD

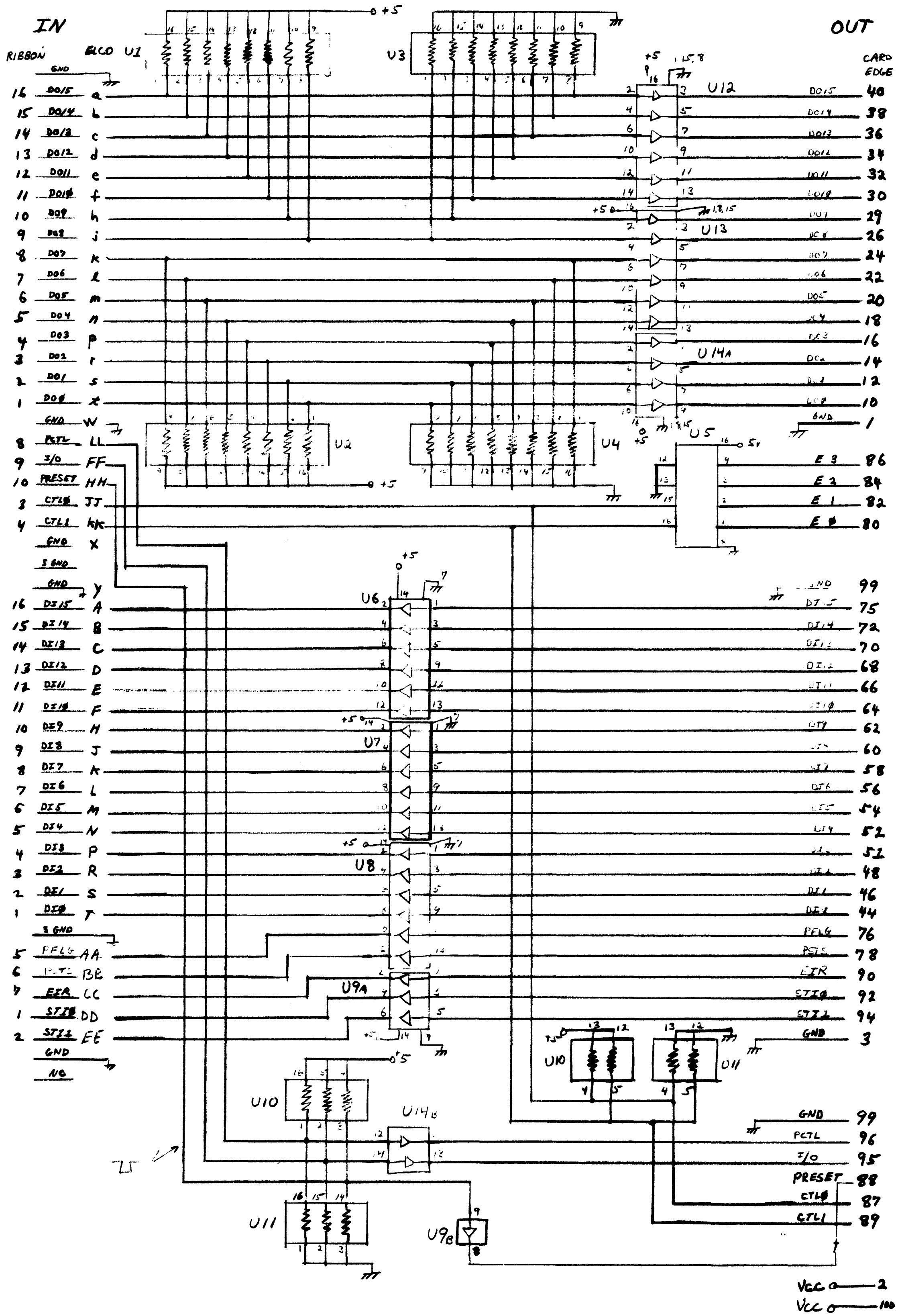


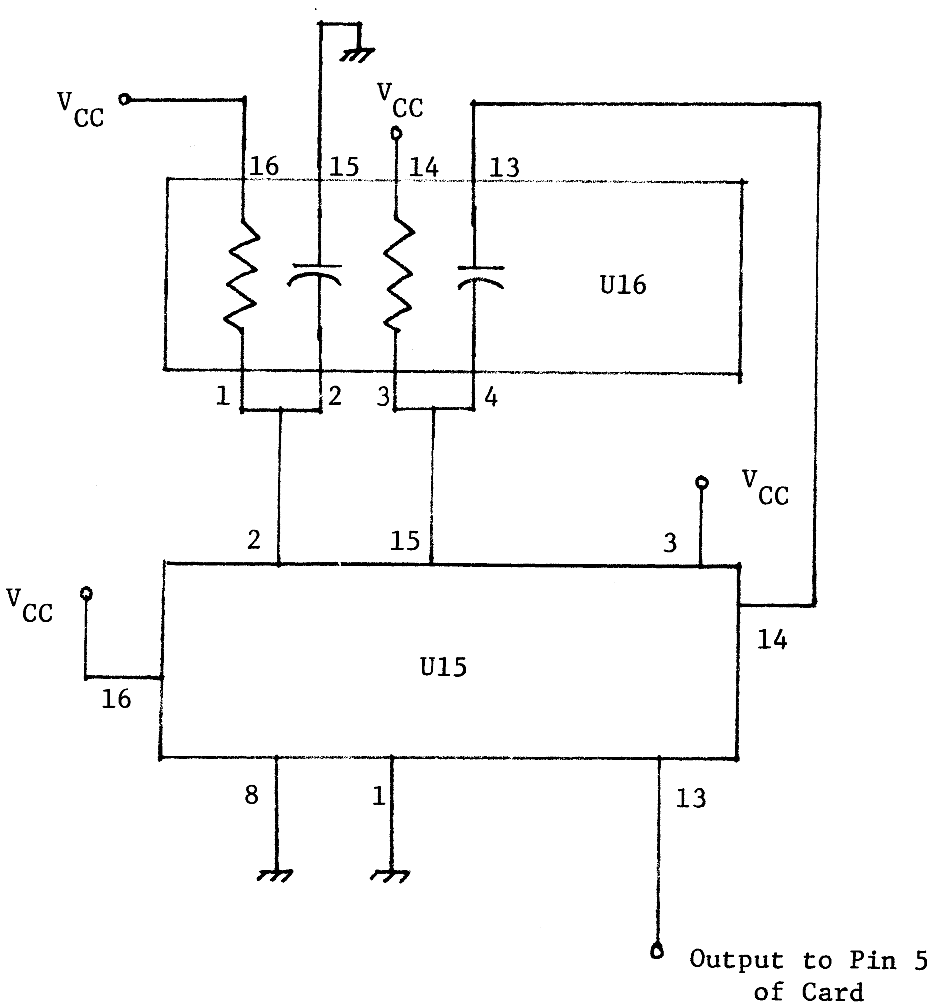
Figure 2

TABLE 1

GPIO Buffer Card

				GND	VCC
U1	Beckman	898-3-R220	4C	None	
U2	Beckman	898-3-R220	4A	None	
U3	Beckman	898-3-R330	3C	None	
U4	Beckman	898-3-R330	3A	None	
U5	7442		2C	8	16
U6	7407		1G	7	14
U7	7407		1F	7	14
U8	7407		1E	7	14
U9	7407		1D	7	14
U10	Beckman	898-3-R220	4E	None	
U11	Beckman	898-3-R330	3E	None	
U12	74365		2B	8	16
U13	74365		3B	8	16
U14	74365		2A	8	16
U15	74123		2F	8	16
U16	Discrete Components		2E	None	

POWER-UP PULSE CIRCUIT



<u>Discrete Components of U16</u>:

1-16	220 KΩ, 1/4 watt
2-15	0.02 μF
3-14	100 KΩ, 1/4 watt
4-13	10 μF elect.

Figure 3

GPIO BUFFER CARD

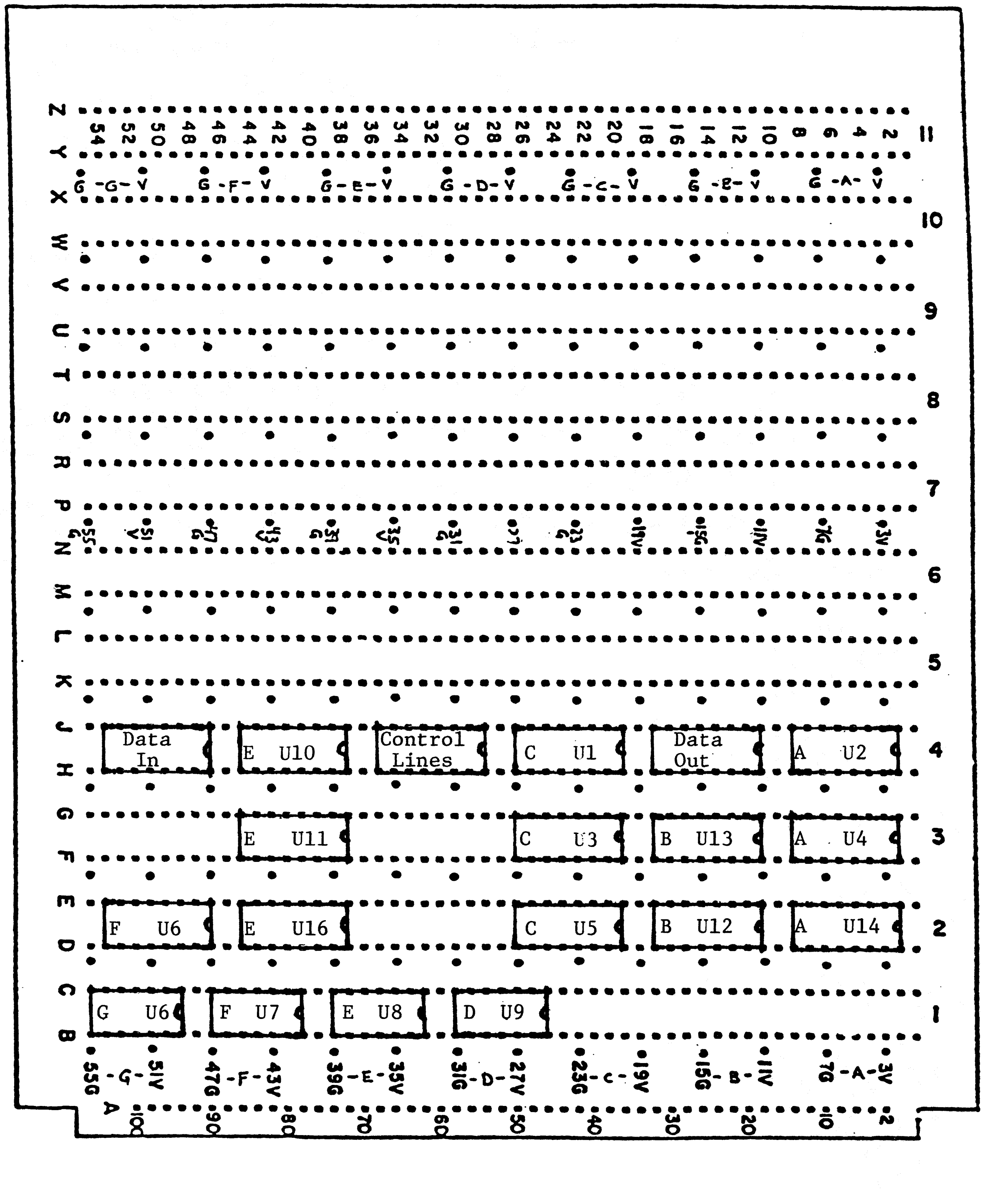


Figure 4

B. ULO Decode Card

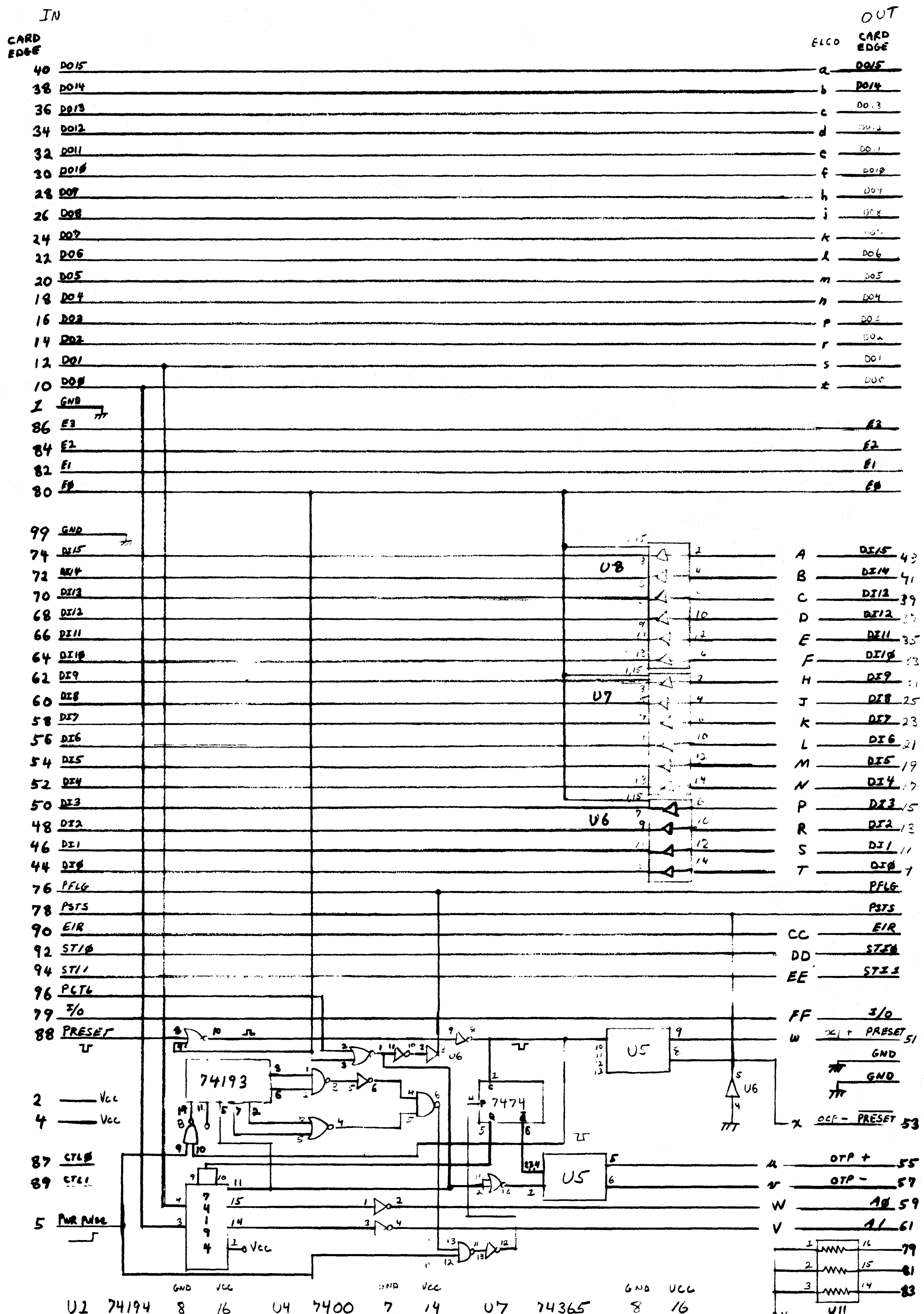
The schematic diagram for the ULO deocde circuit is shown in Figure 5. Table 2 contains the parts list. The data in (DI) lines are buffered by Tri-State enable buffers U8, U7 and U6. The "DO" line bits Ø and 1 are latched by U1 and inverted by U10 to provide the address to the ULO system. The words sent to the ULO are counted by the binary counter U2. Control of the address latch is done by flip-flop U3. This flip-flop allows either clocking of the latch or counting of the words. Operation of this system is described later. The power-up pulse is used to zero the counter and preset the flip-flop. The PCTL and PRESET lines from the GPIO are buffered by line drivers U5 to reduce any noise related problems in the line between the interface and the ULO. Resistor bank U11 provides pull-ups for the hard wired part of the address line. PFLG is generated by the PCTL pulse for handshaking with the computer.

TABLE 2

ULO Decode Circuit

IC #	Type	Board #	Description
U1	74194	1D	Shift Register
U2	74193	1E	Binary Counter
U3	7474	2E	D Flip-Flop
U4	7400	2F	NAND Gates
U5	8830	2D	Line Driver
U6	74365	1A	Tri-State Buffer
U7	74365	1B	Tri-State Buffer
U8	74365	1C	Tri-State Buffer
U9	7402	2B	NOR Gates
U10	7402	2C	Inverters
U11	898-3-R220	1F	220 Ω Resistors

ULO DECODE CARD



		GND	VCC			GND	VCC			GND	VCC
U1	74194	8	16	U4	7400	7	14	U7	74365	8	16
U2	74193	8	16	U5	8830	7	14	U8	74365	8	16
U3	7474	7	14	U6	74365	8	16	U9	7402	7	14
								U10	7404	7	14

Figure 5

ULO DECODE CIRCUIT

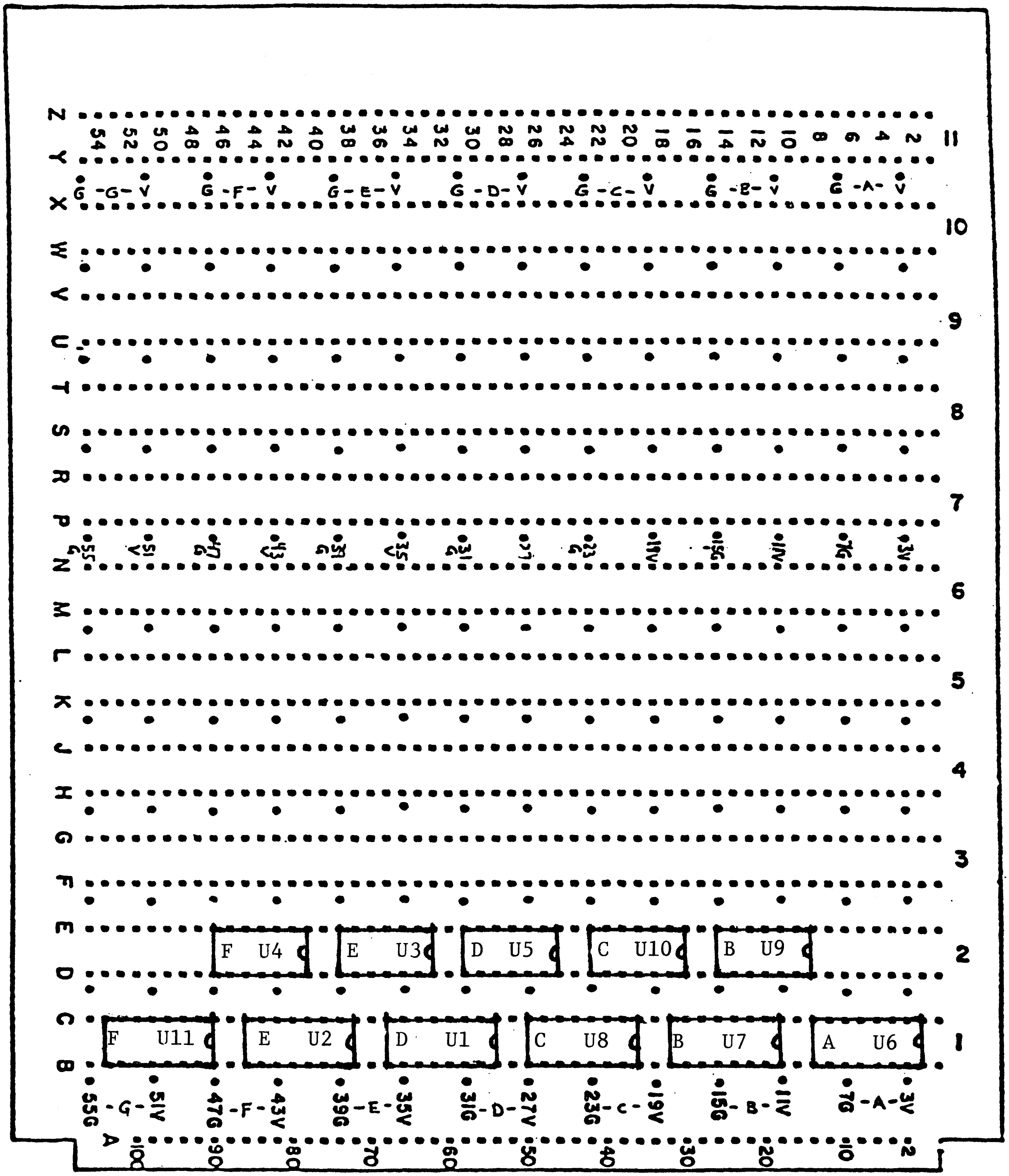


Figure 6

C. GPO Card

The schematic diagram for the GPO card is illustrated in Figure 7. The part list is given in Table 3. The "DI" lines are buffered by U7, U8 and U9 Tri-State buffers. These lines are not used in the switch circuit but are provided for future expansion. The first eight bits of the "DO" line (lower byte) are latched by registors U3 and U4. This latch provides local storage of the switch position. The PRESET pulse clears the register and the PCTL is used to clock the data onto the register. The output of the register is buffered by U5 and U6. The PFLG signal is generated from the PCTL signal for handshaking with the computer. The power-up line clears the register. The circuit shown in Figure 8 drives the LED indicators on the front panel of the interface unit. The component layout for this card is shown in Figure 9.

TABLE 3

GPO Card

IC #	Type	Board #	VCC	GND
U1	7427	3F	14	7
U2	7404	3G	14	7
U3	74194	1B	16	8
U4	74194	1A	16	8
U5	74365	2C	16	8
U6	74365	2D	16	8
U7	74365	1D	16	8
U8	74365	1E	16	8
U9	74365	1C	16	8
U10	7400	3D	14	7
U11	7404	3E	14	7

GPO CARD

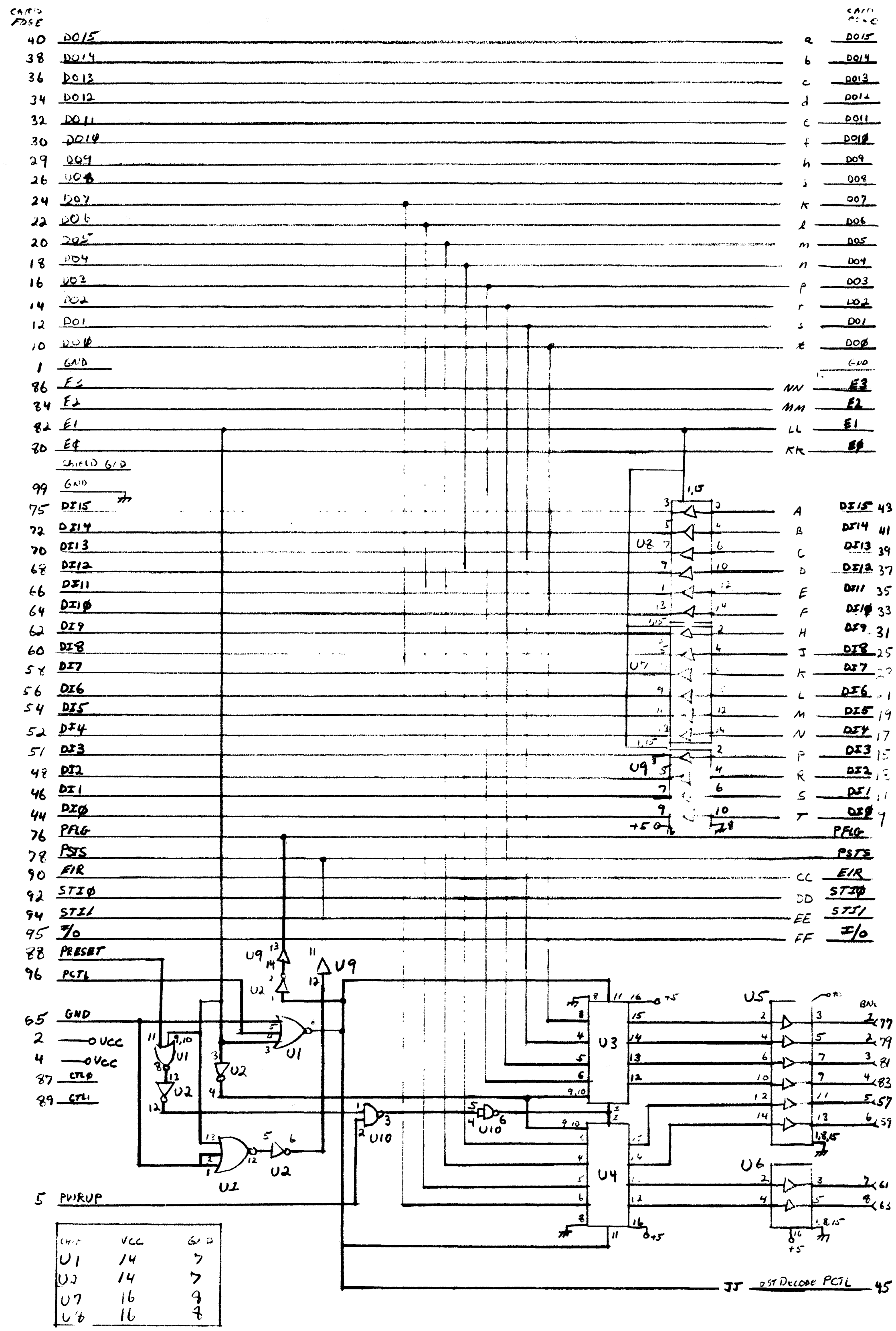


Figure 7

ADDITIONAL CIRCUIT FOR THE LED INDICATORS

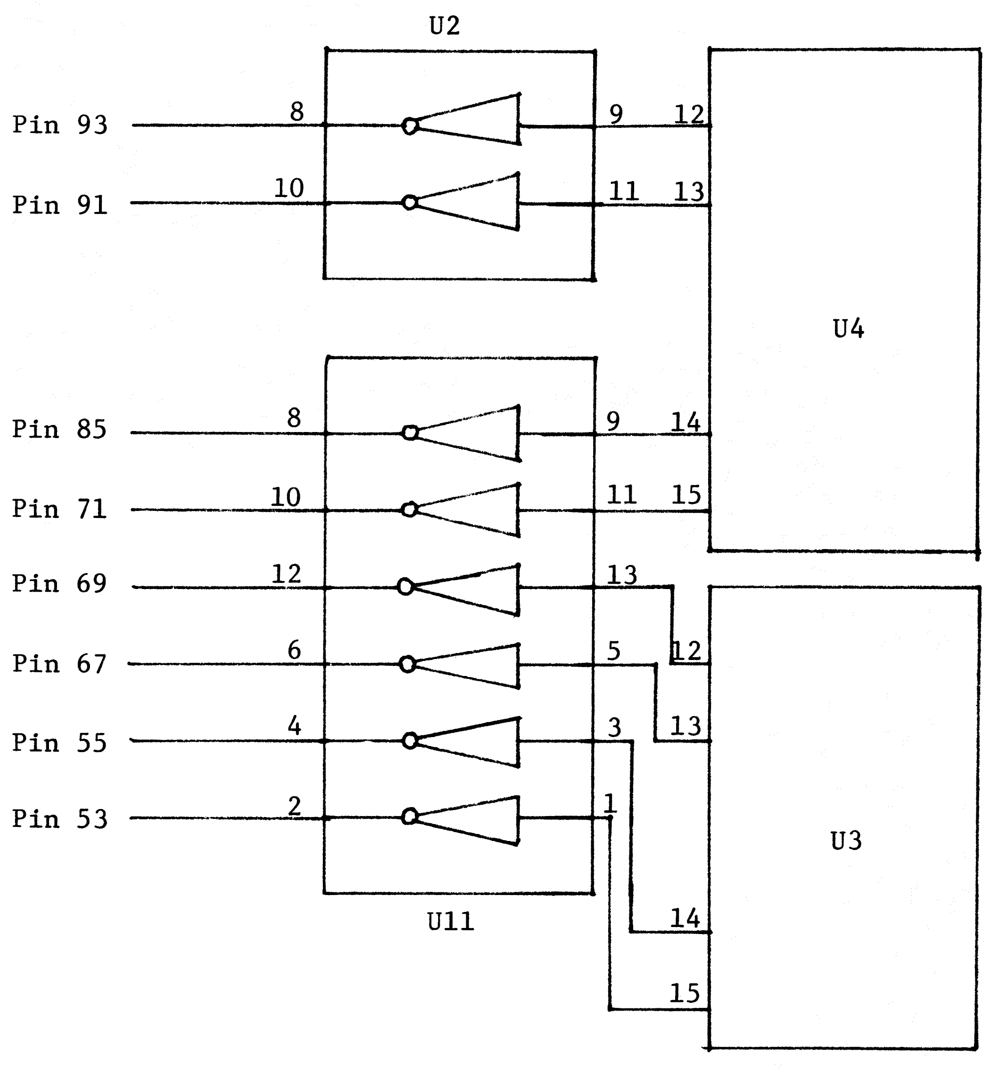


Figure 8

GPO CARD

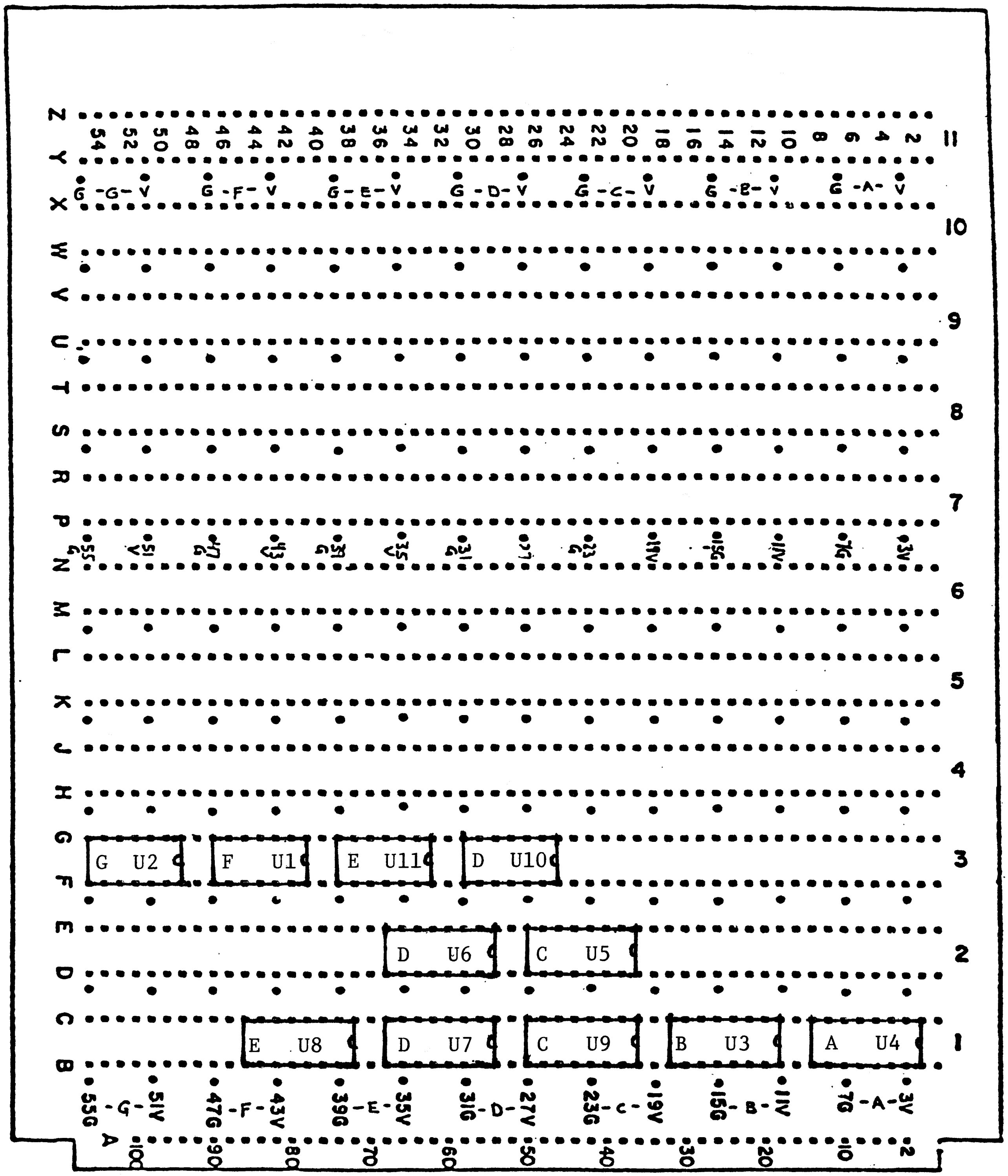


Figure 9

CIRCUIT OPERATION

A. Setting the ULO System

The ULO system used to control the HP synthesizer accepts ten 16-bit words of data. The ULO OTAØ command puts the ULO into the DATA RECEIVE MODE. This OTAØ command must be constructed in the following way. To set the ULO frequencies, the first thing to do is enable the ULO DECODE CIRCUIT. This is done by sending a control signal CTLØ=1 and CTL1=1 to the interface. The software command is as follows:

```
CTL1=1
CTLØ=1
CONTROL 12, 2; CTL1 * 2 + CTLØ
```

Upon power-up or after ten words of data have been sent to or received from the ULO, the address register, U1, will be clocked on each PCTL pulse. The PCTL pulse is provided by the computer when a word of data is on line. Thus, the next thing that should occur is a general GPIO output statement which contains the address information. The software statement is as follows:

```
OUTPUT @ GPIO USING "#,W";Ø
```

The address for the ULO is hard wired except for the two least significant bits which are taken from the GPIO. The address is sent over the GPIO "DO" lines bit 1 and bit Ø. Note that if another word was sent to the interface, the address register would be clocked and its address changed. To lock the address on the register, a PRESET pulse should be sent to the interface. The software command is as follows:

```
PRESET=1
GPIO=12
CONTROL GPIO;PRESET
```

This command has locked the address register and has sent an OCP pulse to the ULO system. This pulse indicates to the ULO that ten words of data follows. This completes the OTAØ forming statements.

Each word of data is accompanied by a PCTL pulse which clocks the counter U2 and used as an OTP pulse for the ULO system. The software statements are as follows:

```
OUTPUT @ GPIO USING "#,W";OUT
```

where "OUT" is the variable containing the data to be sent to the ULO. Remember that ten such statements must follow the PRESET command. Details of the data word will be discussed later.

After the tenth word has been sent to the ULO system, the counter decode circuit then shifts the PCTL from the OTP pulse forming circuit, back to the address register. The ULO decode circuit is now ready to repeat the above series of commands to set the ULO again, or do any other function.

B. Reading Knob Position

To read the knob position from the ULO system, the INA3 statement must be formed. The first command sent to the interface must be the ULO decode circuit enable. The software for the procedure is:

```
CTL1=1
CTLØ=1
CONTROL 12, 2; CTL1 * 2 + CTLØ
```

Upon power-up or after ten words have been sent to or received from the ULO, the address register U1 will be clocked on each PCTL pulse. The PCTL pulse is provided by the computer when a word of data is on line. The next command to the interface must be a general GPIO output statement which contains the address information. The software is:

```
OUTPUT @ GPIO USING "#,W";3
```

The address for the ULO is hard wired except for the two least significant bits which are supplied by the computer. The address is sent over the GPIO "DO" lines bit 1 and bit Ø. Note that if another word is sent to the interface at this time, the address register would be clocked and the data changed. To lock the address on the register, a PRESET pulse should be sent to the interface unit. The software for this PRESET pulse is:

```
PRESET=1
GPIO=12
CONTROL GPIO;PRESET
```

This command has locked the address register, and has sent an OCP pulse to the ULO system. This pulse indicates to the ULO that ten words of data follow. The above statement completes the OTA3 command.

When the computer receives a word of data from the ULO, the corresponding handshake PCTL pulse is used to clock the word counter U2 and to generate an OTP pulse for the ULO. The software command for each word is as follows:

```
ENTER @ GPIO USING "#,W"; Word
```

where "word" is the variable containing the data received from the ULO. Remember that ten statements of this form must follow the PRESET command. After the tenth word has been sent to the computer, the counter decode circuit then shifts the PCTL from the OTP pulse forming circuit, back to the address register. The ULO decode circuit is now ready to repeat the above series, or change modes.

C. Setting the GPO Switches

Setting the TTL switch outputs of the interface is a rather simple process as compared to setting the ULO. The interface must first be set to the GPO mode. This is done by setting CTLØ=Ø and CTL1=1. The corresponding software is:

```
CTLØ=Ø
CTL1=1
CONTROL 12, 2; CTL1 * 2 + CTLØ
```

The GPO output card is now active. The next step is to decide on the switch positions and generate a corresponding binary word using the software techniques described later. Once the word is known, it can now be sent to the GPIO output card which latches the word to the BNC outputs. The data is also displayed on the front panel of the interface unit. The software to implement the above event is:

```
OUTPUT @ GPIO USING "#,B";GPO
```

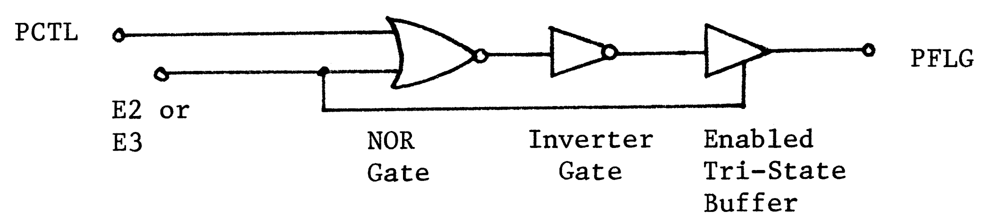
where "GPO" is a variable containing the 8-bit binary coded switch positions. To set all switches to zero voltage, the PRESET pulse is used. This PRESET pulse can be used at any time as long as the GPO card is enabled. The software command is:

```
PRESET=1
CONTROL GPIO: PRESET
```

Note also that the power-up pulse will set all BNC outputs to zero voltage when the interface unit power is turned on.

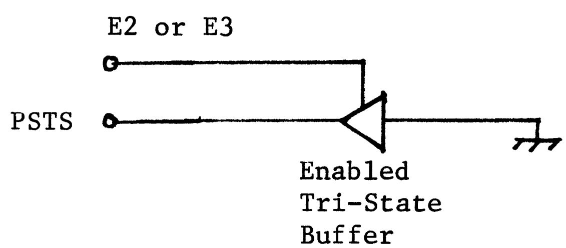
REQUIREMENTS FOR ADDITIONAL BOARDS

When adding additional boards to the interface unit, the following circuit must appear on each new board. This circuit is the minimum needed for full handshaking. More complex handshaking signals may be used to replace this circuit, if desired.



Remember that all Data In Lines should be isolated from the GPIO buffered bus by Tri-State buffers.

Furthermore, the following circuit may be added to the board if board status is required.



Logic Levels (Interface)

Line	Logic High	Logic Low	Pulse
DO∅-DO15	0 V	5 V	-
DI∅-DI15	0 V	5 V	-
PCTL	-	-	
I/O	0 V	5 V	-
PRESET	-	-	
CTL∅	0 V	5 V	-
CTL1	0 V	5 V	-
PFLG	-	-	
PSTS	0 V	5 V	-
EIR	0 V	5 V	-
STI∅	0 V	5 V	-
STI1	0 V	5 V	-
PWR PULSE	-	-	

Logic Levels (ULO)

OTB1-OTB16	0 V	5 V	-
INB1-INB16	0 V	5 V	-
ADB7-ADB16	5 V	0 V	-
OTP+A	-	-	
OTB-A	-	-	
OCP+A	-	-	
OCP-A	-	-	

PHYSICAL CONSTRUCTION DETAILS

The GPIO Bus Interface Unit was built in a 5 1/4" chassis designed by A. Shalloway. Figure 10 shows the physical layout of the parts. All circuit boards and most of the logic lines throughout the chassis are wirewrap. The 5 volt power supply is a Power Products PM 542. The fuse is a Buss GMW 2. The GPIO connection is a 56 pin exposed Elco. The ULO and GPO connectors are 56 pin protected Elco. The switch outputs for the GPO are through standard BNC connectors. Two additional edge card connectors with VCC and ground attached are included for future circuits. Tables 4, 5 and 6 list the pin layout for the interface unit and ULO unit.

TOP VIEW OF GPIO BUS INTERFACE UNIT

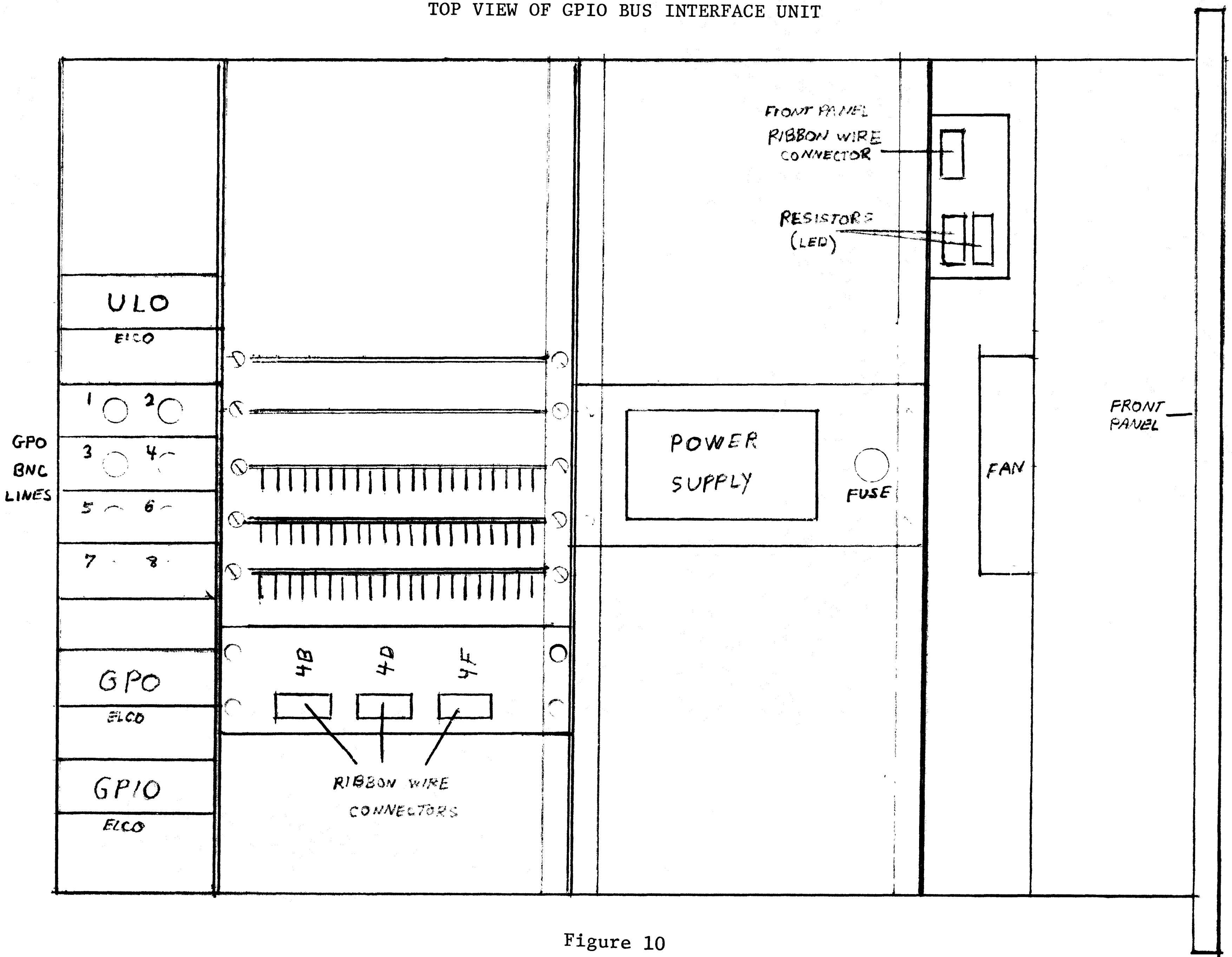


Figure 10

GPO CARD

Input Lines

Data	
Line	**Pin #**
DO15	40
DO14	38
DO13	36
DO12	34
DO11	32
DO1Ø	30
DO9	29
DO8	26
DO7	24
DO6	22
DO5	20
DO4	18
DO3	16
DO2	14
DO1	12
DOØ	10
DI15	75
DI14	72
DI13	70
DI12	68
DI11	66
DI1Ø	64
DI9	62
DI8	60
DI7	58
DI6	56
DI5	54
DI4	52
DI3	51
DI2	48
DI1	46
DIØ	44

Control	
Line	**Pin #**
PFLG	76
PSTS	78
EIR	90
STIØ	92
STI1	94
I/O	95
Preset	88
PCTL	96
CTLØ	87
CTL1	89
PWR UP	5
E3	86
E2	84
E1	82
EØ	80
GND	65

GPO CARD

Output Lines

Data

Line	Elco
DO15	a
DO14	b
DO13	c
DO12	d
DO11	e
DO1Ø	f
DO9	h
DO8	j
DO7	k
DO6	ℓ
DO5	m
DO4	n
DO3	p
DO2	r
DO1	s
DOØ	t
DI15	A
DI14	B
DI13	C
DI12	D
DI11	E
DI1Ø	F
DI9	H
DI8	J
DI7	K
DI6	L
DI5	M
DI4	N
DI3	P
DI2	R
DI1	S
DIØ	T

BNC Lines

BNC #	Pin #
1	77
2	79
3	81
4	83
5	47
6	59
7	61
8	63

Control

Line	Pin	Elco
E3	86	NN
E2	84	MM
E1	82	LL
EØ	80	KK
PFLG	76	--
PSTS	78	--
STIØ	92	DD
STI1	94	EE
EIR	90	CC
I/O	95	FF
PD PCTL	45	JJ

GPIO BUFFER CARD

Input Lines

Data		
GPIO Line	Elco	Ribbon
DO15	a	4B 16
DO14	b	4B 15
DO13	c	4B 14
DO12	d	4B 13
DO11	e	4B 12
DO1Ø	f	4B 11
DO9	h	4B 10
DO8	j	4B 9
DO7	k	4B 8
DO6	ℓ	4B 7
DO5	m	4B 6
DO4	n	4B 5
DO3	p	4B 4
DO2	r	4B 3
DO1	s	4B 2
DOØ	t	4B 1
DI15	A	4F 16
DI14	B	4F 15
DI13	C	4F 14
DI12	D	4F 13
DI11	E	4F 12
DI1Ø	F	4F 11
DI9	H	4F 1Ø
DI8	J	4F 9
DI7	K	4F 8
DI6	L	4F 7
DI5	M	4F 6
DI4	N	4F 5
DI3	P	4F 4
DI2	R	4F 3
DI1	S	4F 2
DIØ	T	4F 1

Control		
GPIO Line	Elco	Ribbon
PCTL	LL	4D 8
I/O	FF	4D 9
PRESET	HH	4D 10
CTLØ	JJ	4D 3
CTL1	KK	4D 4
PFLG	AA	4D 5
PSTS	BB	4D 6
EIR	CC	4D 7
STIØ	DD	4D 1
STI1	EE	4D 2
GND	W	–
GND	Y	–

GPIO BUFFER CARD

Output Lines

Data		Control	
Line	Pin #	Line	Pin #
DO15	40	E3	86
DO14	38	E2	84
DO13	36	E1	82
DO12	34	EØ	8Ø
DO11	32	PFLG	76
DO1Ø	30	PSTS	78
DO9	29	EIR	90
DO8	26	STIØ	92
DO7	24	STI1	94
DO6	22	PCTL	96
DO5	20	I/O	95
DO4	18	PRESET	88
DO3	16	CTLØ	87
DO2	14	CTL1	89
DO1	12	PWR UP	5
DOØ	10		

DI15	75
DI14	72
DI13	70
DI12	68
DI11	66
DI1Ø	64
DI9	62
DI8	60
DI7	58
DI6	56
DI5	54
DI4	52
DI3	51
DI2	48
DI1	46
DIØ	44

ULO DECODE CARD

Input Lines

Data

Line	Pin #
DO15	40
DO14	38
DO13	36
DO12	34
DO11	32
DO1Ø	30
DO9	28
DO8	26
DO7	24
DO6	22
DO5	20
DO4	18
DO3	16
DO2	14
DO1	12
DOØ	10
DI15	74
DI14	72
DI13	70
DI12	68
DI11	66
DI1Ø	64
DI9	62
DI8	60
DI7	58
DI6	56
DI5	54
DI4	52
DI3	50
DI2	48
DI1	46
DIØ	44

Control

Line	Pin #
E3	86
E2	84
E1	82
EØ	80
PFLG	76
PSTS	78
EIR	90
STIØ	92
STI1	94
PCTL	96
I/O	79
PRESET	88
CTLØ	87
CTL1	89
PWR UP	5

ULO DECODE CARD

Output Lines

Data				Control			
Line	Elco	Pin #	ULO	Line	Elco	Pin #	ULO
DO15	a	40	J2 R	E3	--	86	--
DO14	b	38	J2 S	E2	--	84	--
DO13	c	36	J2 T	E1	--	82	--
DO12	d	34	J2 U	EØ	--	80	--
DO11	e	32	J2 V	PFLG	--	76	--
DO1Ø	f	30	J2 W	PSTS	--	78	--
DO9	h	28	J2 X	EIR	CC	90	--
DO8	j	26	J2 Y	STIØ	DD	92	--
DO7	k	24	J2 Z	STI1	EE	94	--
DO6	ℓ	22	J2 a	I/O	FF	79	--
DO5	m	20	J2 b	OCP+	w	51	J3 E
DO4	n	18	J2 c	OCP-	x	53	J3 F
DO3	p	16	J2 d	OTP+	u	55	
DO2	r	14	J2 e	OTP-	v	57	
DO1	s	12	J2 f	AØ	W	59	J2 D
DOØ	t	10	J2 h	A1	V	61	J2 C

Line	Elco	Pin #	ULO
DI15	A	43	J3 R
DI14	B	41	J3 S
DI13	C	39	J3 T
DI12	D	37	J3 U
DI11	E	35	J3 V
DI10	F	33	J3 W
DI9	H	31	J3 X
DI8	J	25	J3 Y
DI7	K	23	J3 Z
DI6	L	21	J3 a
DI5	M	19	J3 b
DI4	N	17	J3 c
DI3	P	15	J3 d
DI2	R	13	J3 e
DI1	S	11	J3 f
DIØ	T	9	J3 h

ULO ADDRESS

	ULO Deocde Card
Elco	Pin #
J2 A	--
J2 B	GND
J2 C	61
J2 D	59
J2 E	GND
J2 F	79
J2 H	81
J2 J	GND
J2 K	GND
J2 L	83

SUPPORTING SOFTWARE

The manual operation and test program is included here to give the user an idea of how the various subroutines can be incorporated into a large program. The main program yields no direct GPIO bus usage. Various options are presented in menu form with the softkeys used as option selectors. The choice of a particular option sends control to the corresponding subprogram of the main program. This subprogram then loads the necessary subroutines from the disc, calls these subroutines, and then deletes the subroutines before control is sent back to the main program. It is in these subroutines that all the GPIO action takes place.

These subroutines fall into three categories:

A. Setting the ULO frequencies.

B. Reading the ULO knob position.

C. Setting the switches on the GPO.

Each of these categories will now be discussed in detail.

```
!!!!!!!!!!!!!!!                     !!!!!!!
! MANUAL OPERAT                     ROGRAM !
!               FOR THE                    !
!       GPIO BUS INTERFACE UNIT            !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! This program will allow manual control
! of the GPIO bus interface between the
! HP 9826 computer and the ULO. The general
! purpose output (GPO) BNC is also controlled
! by this program.
!
! PROGRAM INPUTS:
!    SOFTKEYS:
!     k0: Flo   : LOCAL OSCILLATOR FREQ. (MHZ)
!         Fref1: REFERENCE FREQ. ONE     (MHZ)
!         Fref2: REFERENCE FREQ. TWO     (MHZ)
!     k1: Flo   : LOCAL OSCILLATOR FREQ. (HZ)
!         Fref1: REFERENCE FREQ. ONE     (HZ)
!         Fref2: REFERENCE FREQ. TWO     (HZ)
!     k2: Flo   : LOCAL OSCILLATOR FREQ. (MHZ)
!     k3: NONE
!     k4: NONE
!     k5: Sw1$  : SWITCH 1 POSITION
!         Sw2$  : SWITCH 2 POSITION
!         Sw3$  : SWITCH 3 POSITION
!         Sw4$  : SWITCH 4 POSITION
!         Sw5$  : SWITCH 5 POSITION
!         Sw6$  : SWITCH 6 POSITION
!         Sw7$  : SWITCH 7 POSITION
!         Sw8$  : SWITCH 8 POSITION
!     k6: NONE
!     k7: Flo   : X4 MULT. FREQ.
!     K9: NONE
! PROGRAM OUTPUTS: NO DIRECT OUTPUT
! GPIO OUTPUTS/INPUTS: NO DIRECT I/O
!
! VARIABLES
!     I           : counters
!     Flo         : LO/MULT frequency
!     Fref1       : reference freq. one
!     Fref2       : reference freq. two
!     K           : counter
!     X           : temp. storage
!     Positions$: knob position string
!     Sw1$        : switch 1 position
!     Sw2$        : switch 2 position
!     Sw3$        : switch 3 position
!     Sw4$        : switch 4 position
!     Sw5$        : switch 5 position
!     Sw6$        : switch 6 position
!     Sw7$        : switch 7 position
!     Sw8$        : switch 8 position
!
```

```
56   ! SUBROUTINES:
57   !      Mhz            -          uloset"
58   !      Hz                        z"
59   !      Flo       calls
60   !
61   !                     -
62   !
63   !
64   !      Mult      calls    mult
65   !      Bye
66   !      Spin
67   !
68   !
69   ! >>> CLEAR SCREEN <<<
71     FOR I=1 TO 20
72     PRINT
73     NEXT I
74   !
76     PRINT "GPIO Interface Support Software"
77     PRINT
78     PRINT "MANUAL OPERATION AND TEST PROGRAM"
88   !
89   ! >>> SOFTKEY INITIALIZATION <<<
91     ON KEY 0 LABEL "ULO MHz" GOSUB Mhz
100    ON KEY 1 LABEL "ULO Hz" GOSUB Hz
110    ON KEY 2 LABEL "Flo ONLY" GOSUB Flo
120    ON KEY 3 LABEL "UCOUNT" GOSUB Count
130    ON KEY 4 LABEL "KNOB" GOSUB Knob
140    ON KEY 5 LABEL "SWITCH" GOSUB Switch
150    ON KEY 6 LABEL "GCOUNT" GOSUB Gcount
160    ON KEY 7 LABEL "X4 OUT" GOSUB Mult
161    ON KEY 9 LABEL "EXIT" GOSUB Bye
162 !
170 Spin:    DISP "                    *** MENU ***"
175          GOTO Spin
176 !
185 Mhz:     FOR I=1 TO 20
186          PRINT
187          NEXT I
191          INPUT "ENTER LOCAL OSCILLATOR FREQUENCY in MHz",Flo
192          INPUT "ENTER REFERENCE FREQUENCY ONE in MHz",Fref1
193          INPUT "ENTER REFERENCE FREQUENCY TWO in MHz",Fref2
194          LOADSUB ALL FROM "mhzuloset"
195          CALL Mhzuloset(Flo,Fref1,Fref2)
196          DELSUB Mhzuloset
201          RETURN
202 !
211 Hz:      FOR I=1 TO 20
212          PRINT
213          NEXT I
221          INPUT "ENTER LOCAL OSCILLATOR FREQUENCY in Hz",Flo
231          INPUT "ENTER REFERENCE FREQUENCY ONE in Hz",Fref1
241          INPUT "ENTER REFERENCE FREQUENCY TWO in Hz",Fref2
251          LOADSUB ALL FROM "ulohz"
261          CALL Ulohz(Flo,Fref1,Fref2)
271          DELSUB Ulohz
281          RETURN
282 !
```

```
284 Flo:     FOR I=1 TO 20
285          PRINT
286          NEXT I
291          INPUT "ENTER LOCAL OSCILLATOR FREQUENCY in MHz",Flo
301          LOADSUB ALL FROM "flo"
311          CALL Flo(Flo)
321
331
332 !
334 Count:
335          PRINT
336          NEXT I
341          Flo=111111111.1
342          LOADSUB ALL FROM "uloset"
351          FOR K=0 TO 9
361          X=Flo*K
371          CALL Uloset(X,X,X)
381          NEXT K
391          PRINT "END OF TEST"
401          DELSUB Uloset
411          RETURN
412 !
414 Knob:    FOR I=1 TO 20
415          PRINT
416          NEXT I
421          LOADSUB ALL FROM "knob"
422          Position$="off"
431          CALL Knob(Position$)
441          DELSUB Knob
451          RETURN
452 !
454 Switch:  FOR I=1 TO 20
455          PRINT
456          NEXT I
461          PRINT "ENTER SWITCH POSITION AS"
471          PRINT """on""..       logic 0    5 volts"
481          PRINT """off""..      logic 1    0 volts"
491          INPUT "Enter position of switch 1",Sw1$
501          INPUT "Enter position of switch 2",Sw2$
511          INPUT "Enter position of switch 3",Sw3$
521          INPUT "Enter position of switch 4",Sw4$
531          INPUT "Enter position of switch 5",Sw5$
541          INPUT "Enter position of switch 6",Sw6$
551          INPUT "Enter position of switch 7",Sw7$
561          INPUT "Enter position of switch 8",Sw8$
562          LOADSUB ALL FROM "switch"
571          CALL Switch(Sw1$,Sw2$,Sw3$,Sw4$,Sw5$,Sw6$,Sw7$,Sw8$)
581          DELSUB Switch
591          RETURN
592 !
```

```
601 Gcount: FOR I=1 TO 20
602         PRINT
603         NEXT I
611         LOADSUB ALL FROM "gpocount"
621         CALL Gpocount
631         DELSUB Gpocount
641         RETURN
642 !
651 Mult:   FOR I=1 TO 20
661         PRINT
671         NEXT I
681         INPUT "X4 Mult. Freq. in MHZ",Flo
682         LOADSUB ALL FROM "knob"
683         Position$="off"
685         CALL Knob(Position$)
687         IF Position$<>"flo" THEN
688                       NNOT BE SET"
689
69
69
692         DELSUB Knob
694         LOADSUB ALL FROM "mult"
701         CALL Mult(Flo)
711         DELSUB Mult
721         RETURN
722 !
731 Bye:    STOP
741         END
```

A. Setting the ULO Frequencies

One of the most important functions of the GPIO BUS INTERFACE UNIT and its supporting software is its ability to set the frequencies of the ULO. As described in the earlier section on circuit operation, the ULO is looking for ten words of information before it can take action on the data. These ten words must be of the following form:

Bits	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	
Words					Function
1	100's MHz	10's MHz	1's MHz	100's kHz	F_{LO}
2	10's kHz	1's kHz	100's Hz	10's Hz	
3	1's Hz	0.1 Hz	--	--	
4					F_{REF1}
5		S A M E A	S A B O V E		
6					
7					F_{REF2}
8		S A M E A	S A B O V E		
9					
10					SPACER

Note that word ten is only a spacer word and the bits may be set to anything for this use of the ULO. However, this word does have special meaning in the context of the Cassegrain system.

Each word is divided into four BCD digits; thus, the software task is to take a decimal frequency number and convert it into three 16-bit composite BCD words for the ULO. The procedure for doing this is described below.

The best way to understand this algorithm is by way of an example. Assume that the ULO F_{LO} frequency is to be set at 423610537.1 Hz. This number is first divided by 100,000 which moves the decimal point over 5 places yielding 4236.105371. This division is performed by the DIV command which only looks at the quotient of the division and ignores the remainder. Hence, by using this command, the first four digits of the first word have been found to be 4236. This result is then DIV by 1000 to produce a 4, and then MOD (which only looks at the remainder of a division) by 1000 to produce 236. This result is then DIV by 100 giving second digit of 2 and then MOD by 100 giving a result of 36. A DIV by 10 then yields digit three of value 3 and a MOD by 10 yields digit four of value 6.

To produce a composite word of data, the four individual digits are combined in the following way:

Word = (DIGIT 1 * 4096) + (DIGIT 2 * 256) + (DIGIT 3 * 16) + DIGIT 4

where 16, 256, and 4096 are weighting factors for palcement of the BCD digit in the 16 bit binary word.

The original frequency input is then MOD by 100,000 to yeild 105371 and this result DIV by 100 to give 1053 as the BCD digits of word two. Again, the above procedure is used to break the word up into its four individual digits and then recombine them forming the second word.

For the third word, the original frequency is MOD by 10 to give 7100 and once again the above procedure is used to form word three.

The entire algorithm is used three times; F_{LO}, F_{REF1}, and F_{REF2}. This algorithm develops the first 9 words of data. The tenth word which consists of all zeros, is used as a spacer word. This spacer word clocks the data into the ULO memory.

Notice that in the example, all digits were less than or equal to 7. This is important since the above algorithm runs into trouble if a number "8" or "9" is used as DIGIT ONE in any word. The problem stems from the HP 16-bit computer which uses the MSB as a sign bit. However, this problem can be overcome if a number Ø is used in place of 8 and a 1 is used in place of a 9 as the first digit in the word. The sign bit must also be set which implies a negative number. In most computers, as with the HP 9826, negative numbers are stored in two's complement form. Thus, an extra calculation need be performed on the binary word before it is sent to the ULO. The calculation

$$NEWWORD = OLDWORD - 65{,}536$$

yielding a negative number in two's complement form. Note that OLDWORD was found using the previously mentioned word-forming procedure. The new word is thus sent to the ULO.

The subroutines that follow all use this corrected procedure.

```
10   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
11   ! ULO FREQUENCY SET SUBROUTINE !
12   !             HZ               !
13   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
14   !
15   ! This subroutine will set the ULO
16   ! frequencies.  All three frequencies must
17   ! be supplied to the subroutine.  A boundary
18   ! check is performed on the inputs.
19   !
20   ! PROGRAM INPUTS :
21   !     Flo   : LOCAL OSCILLATOR FREQUENCY (HZ)
22   !     Fref1: REFERENCE FREQUENCY ONE     (HZ)
23   !     Fref2: REFERENCE FREQUENCY TWO     (HZ)
24   ! PROGRAM OUTPUTS: NONE
25   ! GPIO OUTPUTS    : TEN 16 bit words for ULO
26   !
27   ! VARIABLES
28   !     Gpio      : set to 12
29   !     Ctl0,Ctl1: interface address
30   !     Preset    : set to 1
31   !     Freqs(3)  : array of ULO frequencies
32   !     Flo       : local oscillator freq.
33   !     Fref1     : reference frequency one
34   !     Fref2     : reference frequency two
35   !     I,J       : counters
36   !     A         : power of 10 position
37   !     Q         : four digits for word
38   !     Dig1      : MSD of word
39   !     Dig2      : digit2 of word
40   !     Dig3      : digit 3 of word
41   !     Dig4      : LSD of word
42   !     Out1      : decimal equiv. of word
43   !
45     SUB Ulohz(Flo,Fref1,Fref2)
46   !
47   ! >>> CHECK BOUNDARY OF INPUTS <<<
49     IF Flo>=500000000 OR Fref1>=500000000 OR Fref2>=500000000 THEN
50     PRINT "ILLEGAL FREQUENCY... ULO NOT SET"
51     SUBEXIT
52     END IF
53   !
55     Gpio=12
56     ASSIGN @Gpio TO 12;FORMAT OFF
57   !
58   ! >>> GIVE CONTROL TO ULO <<<
60     Ctl0=1
61     Ctl1=1
62     CONTROL 12,2;Ctl1*2+Ctl0
63   !
```

```
64   ! >>> SET ULO ADDRESS TO OTA COMMAND <<<
66     OUTPUT @Gpio USING "#,W";0
67   !
68   ! >>> SEND OCP PULSE TO ULO <<<
70     Preset=1
71     CONTROL Gpio;Preset
72   ! >>> PUT FREQUENCIES INTO AN ARRAY <<<
80     DIM Freqs(3)
90     Freqs(1)=Flo
100    Freqs(2)=Fref1
110    Freqs(3)=Fref2
111  !
113    PRINT "LOCAL OSCILLATOR FREQUENCY is",Flo
114    PRINT "REFERENCE FREQUENCY ONE is",Fref1
115    PRINT "REFERENCE FREQUENCY TWO is",Fref2
116  !
117  ! >>> CALCULATE WORDS AND SEND THEM TO ULO <<<
118  !
119  ! >>> J LOOP COUNTS GROUPS OF THREE WORDS <<<
121    FOR J=1 TO 3
130       A=100000
131  ! >>> I LOOP COUNTS WORDS PER GROUP <<<
140       FOR I=1 TO 3
150          Q=Freqs(J) DIV A
170          Dig1=Q DIV 1000
180          Q=Q MOD 1000
200          Dig2=Q DIV 100
210          Q=Q MOD 100
230          Dig3=Q DIV 10
240          Dig4=Q MOD 10
270          Out1=(Dig1*4096)+(Dig2*256)+(Dig3*16)+Dig4
280          IF Dig1=8 OR Dig1=9 THEN Out1=-65536+Out1
290          OUTPUT @Gpio USING "#,W";Out1
300          Freqs(J)=(Freqs(J) MOD A)+.0001
320          A=A/10000
340          NEXT I
350       NEXT J
351  !
353  ! >>> SEND SPACER WORD TO ULO <<<
354    OUTPUT @Gpio USING "#,W";0
355    BEEP 1200,.1
356  !
357    PRINT
358    PRINT
359    PRINT "ULO FREQUENCIES ARE SET"
360  !
362    SUBEND
```

```
10  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20  ! ULO FREQUENCY SET SUBROUTINE !
30  !            MHZ               !
40  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
50  !
60  ! This subroutine will set the ULO
61  ! frequencies.  All three frequencies must
62  ! be supplied to the subroutine.  A boundary
63  ! check is preformed on the inputs.
64  !
65  ! PROGRAM INPUTS :
66  !     Flo  : LOCAL OSCILLATOR FREQUENCY (MHZ)
67  !     Fref1: REFERENCE FREQUENCY ONE    (MHZ)
68  !     Fref2: REFERENCE FREQUENCY TWO    (MHZ)
69  ! PROGRAM OUTPUTS: NONE
70  ! GPIO OUTPUTS   : TEN 16 bit words for ULO
71  !
72  ! VARIABLES
73  !     Gpio      : set to 12
74  !     Ctl0,Ctl1: interface address
75  !     Preset    : set to 1
76  !     Freqs(3)  : array of ULO frequencies
77  !     Flo       : local oscillator freq.
78  !     Fref1     : reference frequency one
79  !     Fref2     : reference frequency two
80  !     I,J       : counters
81  !     A         : power of 10 position
82  !     Q         : four digits for word
83  !     Dig1      : MSD of word
84  !     Dig2      : digit 2 of word
85  !     Dig3      : digit 3 of word
86  !     Dig4      : LSD of word
87  !     Out1      : decimal equiv. of word
88  !
90    SUB Mhzuloset(Flo,Fref1,Fref2)
91  !
92  ! >>> CHECK BOUNDARY OF INPUTS <<<
94    IF Flo>=500 OR Fref1>=500 OR Fref2>=500 THEN
95    PRINT "ILLEGAL FREQUENCY.. ULO NOT SET"
96    SUBEXIT
97    END IF
98  !
99  ! >>> CHANGE INPUT (MHZ) to (HZ) <<<
101   Flo=Flo*1000000
102   Fref1=Fref1*1000000
103   Fref2=Fref2*1000000
104 !
106   Gpio=12
107   ASSIGN @Gpio TO 12;FORMAT OFF
108 !
109 ! >>> GIVE CONTROL TO ULO <<<
111   Ctl0=1
112   Ctl1=1
113   CONTROL 12,2;Ctl1*2+Ctl0
114 !
```

```
115 ! >>> SET ULO ADDRESS TO OTA COMMAND <<<
118   OUTPUT @Gpio USING "#,W":0
119 !
121 ! >>> SEND OCP PULSE TO ULO <<<
122   Preset=1
123   CONTROL Gpio;Preset
124 !
125 ! >>> PUT FREQUENCIES INTO AN ARRAY <<<
130   DIM Freqs(3)
140   Freqs(1)=Flo
150   Freqs(2)=Fref1
160   Freqs(3)=Fref2
161 !
163   PRINT "LOCAL OSCILLATOR FREQUENCY is",Flo,"Hz"
164   PRINT "REFERENCE FREQUENCY ONE is",Fref1,"Hz"
165   PRINT "REFERENCE FREQUENCY TWO is",Fref2,"Hz"
166 !
167 ! >>> CALCULATE WORDS AND SEND THEM TO ULO <<<
168 !
169 ! >>> J LOOP COUNTS GROUPS OF THREE WORDS <<<
171   FOR J=1 TO 3
180      A=100000
181 ! >>> I LOOP COUNTS WORDS PER GROUP <<<
190      FOR I=1 TO 3
200         Q=Freqs(J) DIV A
220         Dig1=Q DIV 1000
230         Q=Q MOD 1000
250         Dig2=Q DIV 100
260         Q=Q MOD 100
280         Dig3=Q DIV 10
290         Dig4=Q MOD 10
320         Out1=(Dig1*4096)+(Dig2*256)+(Dig3*16)+Dig4
330         IF Dig1=8 OR Dig1=9 THEN Out1=-65536+Out1
340         OUTPUT @Gpio USING "#,W";Out1
350         Freqs(J)=(Freqs(J) MOD A)+.0001
370         A=A/10000
390         NEXT I
400      NEXT J
401 !
403 ! >>> SEND SPACER WORD TO ULO <<<
404   OUTPUT @Gpio USING "#,W";0
405   BEEP 1200,.1
406 !
408   PRINT
409   PRINT
410   PRINT "ULO FREQUENCIES ARE SET"
411 !
413   SUBEND
```

```
10   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20   !   ULO LOCAL OSCILLATOR    !
30   ! FREQUENCY SET SUBROUTINE !
31   !           MHZ            !
40   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
50   !
60   ! This program will set the local oscillator
61   ! frequency of the ULO.  ONLY the LO frequency
62   ! is transfered to this subroutine.  The other
63   ! two frequencies are set to zero.  A boundary
64   ! check on the input is performed.
65   !
66   ! PROGRAM INPUTS :
67   !    Flo: LOCAL OSCILLATOR FREQUENCY  (MHZ)
68   ! PROGRAM OUTPUTS: NONE
69   ! GPIO OUTPUTS : TEN 16 bit words for ULO
70   !
71   ! VARIABLES
72   !     Gpio       : set to 12
73   !     Ctl0,Ctl1: interface address
74   !     Preset     : set to 1
75   !     Freqs(3)   : array of ULO frequencies
76   !     Flo        : local oscillator freq.
77   !     Fref1      : set to 0
78   !     Fref2      : set to 0
79   !     I,J        : Counters
80   !     A          : power of 10 position
81   !     Q          : four digits for word
82   !     Dig1       : MSD of word
83   !     Dig2       : digit 2 of word
84   !     Dig3       : digit 3 of word
85   !     Dig4       : LSD of word
86   !     Out1       : decimal equiv. of word
87   !
89     SUB Flo(Flo)
90   !
91   ! >>> CHECK BOUNDARY OF INPUTS <<<
93     IF Flo>=500 THEN
94     PRINT "ILLEGAL FREQUENCY.. ULO NOT SET"
95     SUBEXIT
96     END IF
97   !
98   ! >>> CHANGE INPUT (MHZ) to (HZ) <<<
100    Flo=Flo*1000000
101  !
103    Gpio=12
104    ASSIGN @Gpio TO 12;FORMAT OFF
105  !
106  ! >>> GIVE CONTROL TO ULO <<<
108    Ctl0=1
109    Ctl1=1
110    CONTROL 12,2;Ctl1*2+Ctl0
111  !
```

```
112 ! >>> SET ULO ADDRESS TO DTA COMMAND <<<
115   OUTPUT @Gpio USING "#,W";0
116 !
117 ! >>> SEND OCP PULSE TO ULO <<<
119   Preset=1
120   CONTROL Gpio;Preset
121 !
122 ! >>> PUT FREQUENCIES INTO AN ARRAY <<<
130   DIM Freqs(3)
140   Freqs(1)=Flo
150   Freqs(2)=0
160   Freqs(3)=0
161 !
163   PRINT "LOCAL OSCILLATOR FREQUENCY is",Flo,"Hz"
164 !
165 ! >>> CALCULATE WORDS AND SEND THEM TO ULO <<<
166 !
167 ! >>> J LOOP COUNTS GROUPS OF THREE WORDS <<<
170   FOR J=1 TO 3
180      A=100000
181 ! >>> I LOOP COUNTS WORDS PER GROUP <<<
190      FOR I=1 TO 3
200         Q=Freqs(J) DIV A
220         Dig1=Q DIV 1000
230         Q=Q MOD 1000
250         Dig2=Q DIV 100
260         Q=Q MOD 100
280         Dig3=Q DIV 10
290         Dig4=Q MOD 10
320         Out1=(Dig1*4096)+(Dig2*256)+(Dig3*16)+Dig4
330         IF Dig1=8 OR Dig1=9 THEN Out1=-65536+Out1
340         OUTPUT @Gpio USING "#,W";Out1
350         Freqs(J)=(Freqs(J) MOD A)+.0001
370         A=A/10000
390         NEXT I
400      NEXT J
401 !
402 ! >>> SEND SPACER WORD TO ULO <<<
404   OUTPUT @Gpio USING "#,W";0
405   BEEP 1200,.1
406 !
408   PRINT
409   PRINT
410   PRINT "LOCAL OSCILLATOR FREQUENCY IS SET"
411 !
413   SUBEND
```

```
10  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20  ! X4 MULTIPLIER FREQUENCY   !
30  ! ULO LOCAL OSCILLATOR      !
40  ! FREQUENCY SET SUBROUTINE  !
50  !          MHZ              !
60  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
61  !
62  ! This subroutine will set the ULO LOCAL
63  ! OSCILLATOR frequency to a frequency
64  ! that is 1/4 the input.  Thus, the X4
65  ! multiplier frequency is the input.
66  ! The input is limited to a range of
67  ! 1000 to 2000 MHZ. A boundary check on
68  ! the input is performed. The ULO KNOB
69  ! must be in the "Flo" position for this
70  ! program to function. The other two
71  ! frequencies are set to zero. NOTE:
73  ! ONLY ONE FREQUENCY IS TRANSFERED TO
74  ! THE SUBROUTINE.
75  !
76  ! PROGRAM INPUTS :
77  !    Flo: THE X4 OUT FREQUENCY   (MHZ)
78  ! PROGRAM OUTPUTS: NONE
79  ! GPIO OUTPUTS : TEN 16 bit words for ULO
80  !
81  ! VARIABLES
82  !     Gpio      : set to 12
83  !     Ctl0,Ctl1: interface address
84  !     Preset    : set to 1
85  !     Freqs(3)  : array of ULO frequencies
86  !     Flo       : Local oscillator freq.
87  !     Fref1     : set to 0
88  !     Fref2     : set to 0
89  !     I,J       : counters
90  !     A         : power of 10 position
91  !     Q         : four digits for word
92  !     Dig1      : MSD of word
93  !     Dig2      : digit 2 of word
94  !     Dig3      : digit 3 of word
95  !     Dig4      : LSD of word
96  !     Out1     :decimal equiv. of word
97  !
99    SUB Mult(Flo)
100 !
102 ! >>> CHECK BOUNDARY OF INPUT <<<
103   IF Flo>=2000 OR Flo<1000 THEN
104   PRINT "ILLEGAL FREQUENCY.. ULO NOT SET"
105   SUBEXIT
106   END IF
107 !
108 ! >>> CHANGE INPUT TO 1/4 OF VALUE <<<
109 ! >>> AND (MHZ) to (HZ)            <<<
110   Flo=Flo*250000
111 !
```

```
112   Gpio=12
113   ASSIGN @Gpio TO 12;FORMAT OFF
114 !
116 ! >>> GIVE CONTROL TO ULO <<<
117   Ctl0=1
118   Ctl1=1
119   CONTROL 12,2;Ctl1*2+Ctl0

120 !
121 ! >>> SET ULO ADDRESS TO OTA COMMAND <<<
122   OUTPUT @Gpio USING "#,W":0
123 !
124 ! >>> SEND OCP PULSE TO ULO <<<
126   Preset=1
127   CONTROL Gpio;Preset
128 !
129 ! >>> PUT FREQUENCIES INTO AN ARRAY <<<
131   DIM Freqs(3)
140   Freqs(1)=Flo
150   Freqs(2)=0
160   Freqs(3)=0
161 !
163   PRINT "LOCAL OSCILLATOR FREQUENCY is",Flo,"Hz"
164 !
165 ! >>> CALCULATE WORDS AND SEND THEM TO ULO <<<
166 !
167 ! >>> J LOOP COUNTS GROUPS OF THREE WORDS <<<
170   FOR J=1 TO 3
180      A=100000
181 ! >>> I LOOP COUNTS WORDS PER GROUP <<<
190      FOR I=1 TO 3
200         Q=Freqs(J) DIV A
220         Dig1=Q DIV 1000
230         Q=Q MOD 1000
250         Dig2=Q DIV 100
260         Q=Q MOD 100
280         Dig3=Q DIV 10
290         Dig4=Q MOD 10
320         Out1=(Dig1*4096)+(Dig2*256)+(Dig3*16)+Dig4
330         IF Dig1=8 OR Dig1=9 THEN Out1=-65536+Out1
340         OUTPUT @Gpio USING "#,W";Out1
350         Freqs(J)=(Freqs(J) MOD A)+.0001
370         A=A/10000
390         NEXT I
400      NEXT J
401 !
402 ! >>> SEND SPACER WORD TO ULO <<<
404   OUTPUT @Gpio USING "#,W";0
405   BEEP 1200,.1
406 !
408   PRINT
409   PRINT
410   PRINT "ULO FREQUENCY IS SET"
411 !
413   SUBEND
```

```
1    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2    ! ULO FREQUENCY SET TEST SUBROUTINE !
3    !               HZ                  !
5    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
6    !
7    ! NOTE.. THIS SUBROUTINE FOR ULO TEST ONLY
8    !
10   ! This subroutine will set the ULO
11   ! frequencies.  All three frequencies must
12   ! be supplied to this subroutine. NO boundary
13   ! check is performed on the input. Hence any
14   ! number can be entered as a frequency.
15   !
16   ! PROGRAM INPUTS :
17   !     Flo  : LOCAL OSCILLATOR FREQUENCY (Hz)
18   !     Fref1: REFERENCE FREQUENCY ONE    (Hz)
19   !     Fref2: REFERENCE FREQUENCY TWO    (Hz)
20   ! PROGRAM OUTPUTS: NONE
21   ! GPIO OUTPUTS   : TEN 16 bit words for ULO
22   !
23   ! VARIABLES
24   !     Gpio      : set to 12
25   !     Ctl0,Ctl1: interface address
26   !     Preset    : set to 1
27   !     Freqs(3)  : array of ULO frequencies
28   !     Flo       : local cscillator freq.
29   !     Fref1     : reference frequency one
30   !     Fref2     : reference frequency two
31   !     I,J       : counters
32   !     A         : power of 10 position
33   !     Q         : four digits for word
34   !     Dig1      : MSD of word
35   !     Dig2      : digit 2 of word
36   !     Dig3      : digit 3 of word
37   !     Dig4      : LSD of word
38   !     Out1      : decimal equiv. of word
39   !
40     SUB Uloset(Flo,Fref1,Fref2)
41   !
42     Gpio=12
43     ASSIGN @Gpio TO 12;FORMAT OFF
44   !
45   ! >>> GIVE CONTROL TO ULO <<<
46     Ctl0=1
47     Ctl1=1
48     CONTROL 12,2:Ctl1*2+Ctl0
49   !
50   ! >>> SET ULO ADDRESS FOR OTA COMMAND <<<
51     OUTPUT @Gpio USING "#,W";0
52   !
53   ! >>> SEND OCP PULSE TO ULO <<<
54     Preset=1
60     CONTROL Gpio;Preset
61   !
```

```
62  ! >>> PUT FREQUENCIES INTO AN ARRAY <<<
70    DIM Freqs(3)
80    Freqs(1)=Flo
90    Freqs(2)=Fref1
100   Freqs(3)=Fref2
101 !

103   PRINT "LOCAL OSCILLATOR FREQUENCY is",Flo
104   PRINT "REFERENCE FREQUENCY ONE is",Fref1
105   PRINT "REFERENCE FREQUENCY TWO is",Fref2
106 !
107 ! >>> CALCULATE WORDS AND SENT THEM TO ULO <<<
109 !
110 ! >>> J LOOP COUNTS GROUPS OF THREE WORDS <<<
111   FOR J=1 TO 3
120      A=100000
121 !
122 ! >>> I LOOP COUNTS WORDS PER GROUP <<<
130      FOR I=1 TO 3
140         Q=Freqs(J) DIV A
160         Dig1=Q DIV 1000
170         Q=Q MOD 1000
190         Dig2=Q DIV 100
200         Q=Q MOD 100
220         Dig3=Q DIV 10
230         Dig4=Q MOD 10
260         Out1=(Dig1*4096)+(Dig2*256)+(Dig3*16)+Dig4
270         IF Dig1=8 OR Dig1=9 THEN Out1=-65536+Out1
280         OUTPUT @Gpio USING "#,W";Out1
290         Freqs(J)=(Freqs(J) MOD A)+.0001
310         A=A/10000
330         NEXT I
340      NEXT J
341 !
342 ! >>> SEND SPACER WORD TO ULO <<<
344   OUTPUT @Gpio USING "#,W";0
345   BEEP 1200,.1
346 !
348   PRINT
349   PRINT
350   PRINT "ULO FREQUENCIES ARE SET"
351 !
353   SUBEND
```

B. Reading the ULO Knob Position

This simple subroutine looks at the position of the ULO knob. Only one word of data is needed but the interface must transfer ten words of data to keep its internal counter synchronized. The subroutine looks at the first four bits of the word and decodes them into their corresponding knob position.

Bit	13	14	15	16	Position
	0	0	0	1	MOD
	0	0	1	0	F_{REF1}
	0	0	1	1	F_{LO}
	0	1	0	0	F_{REF2}
	0	1	1	1	SYNTH IN LOCAL

The program then prints the position of the knob on the CRT.

```
1    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2    ! ULO KNOB POSITION SUBROUTINE !
3    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
4    !
5    ! This subroutine will determine the position
6    ! of the ULO knob. The subroutine will print
7    ! the position of the knob on the CRT and send
8    ! a variable containing the positional info
9    ! back to the main program.
10   !
11   ! PROGRAM INPUTS : NONE
12   ! PROGRAM OUTPUTS: Position$
13   ! GPIO INPUTS    : TEN 16 bit words
14   !
15   ! VARIABLES
16   !     Gpio      : set to 12
17   !     Ctl0,Ctl1: interface address
18   !     Preset    : set to 1
19   !     Word(10)  : array of 10 ULO data words
20   !     Position&: string containing position info
21   !     I         : word counter
22   !
24     SUB Knob(Position$)
25   !
26   ! >>> GIVE CONTROL TO ULO <<<
28     Ctl0=1
30     Ctl1=1
40     CONTROL 12,2;Ctl1*2+Ctl0
41   !
50     Gpio=12
60     ASSIGN @Gpio TO 12;FORMAT OFF
70   !
71   ! >>> SET ULO ADDRESS TO INA 3 COMMAND <<<
80     OUTPUT @Gpio USING "#,W";3
81   !
82   ! >>> SEND OCP PULSE TO ULO <<<
84     Preset=1
90     CONTROL Gpio;Preset
91   !
100    DIM Word(10)
101  !
102  ! >>> I LOOP COUNTS WORDS INPUT FROM ULO <<<
110    FOR I=1 TO 10
120       ENTER @Gpio USING "#,W":Word(1)
130       NEXT I
131  !
140    PRINT
141  !
```

```
142 ! >>> DECODE WORD TO DETERMINE POSITION <<<
150   SELECT Word(1)
160   CASE 1
170      PRINT "ULO switch in ""MOD"" position"
171      Position$="mod"
180   CASE 2
190      PRINT "ULO switch in ""Fref1"" position"
191      Position$="fref1"
200   CASE 3
210      PRINT "ULO switch in ""Flo"" position"
211      Position$="flo"
220   CASE 4
230      PRINT "ULO switch in ""Fref2"" position"
231      Position$="fref2"
240   CASE 7
250      PRINT "HP Synthesizer in ""LOCAL"" mode"
251      Position$="local"
260   END SELECT
261 !
270   SUBEND
```

C. Setting the Switches on the GPO

This subroutine controls the position or logic level of the BNC switch outputs. The switch positions are as follows:

Position	Logic Level	LED Indicator	Voltage Level
"ON"	0	ON	5 V
"OFF"	1	OFF	0 V

The algorithm is straightforward. It takes each switch position as a bit in the binary word of the GPIO. The higher order byte is set to zero. Each bit is multiplied by a weighting factor and then the bits and weighting factors are added together to form the word. Only one word is sent through the GPIO for this operation.

The GPO count routine is a very simple program that counts from Ø to 255. The binary equivalent of the count forms the lower byte of the 16-bit word and it is then sent to the GPO.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! GPO BNC SWITCH SUBROUTINE !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! This subroutine controls the GPO BNC
! output of the interface unit.  Inputs
! to this subroutine are the switch
! positions.  NOTE: SWITCH POSITIONS MUST
! BE ENTERED AS "ON" OR "OFF".
! "ON" IS A LOGIC 0 WITH BNC VOLTAGE=5v
! "OFF " IS A LOGIC 1 WITH BNC VOLTAGE=0v
! Values not given to the subroutine are
! considered "OFF".  A warning message is
! printed in response to such an input.
!
! PROGRAM INPUTS :
!    Sw1$ : SWITCH 1 POSITION
!    Sw2$ : SWITCH 2 POSITION
!    Sw3$ : SWITCH 3 POSITION
!    Sw4$ : SWITCH 4 POSITION
!    Sw5$ : SWITCH 5 POSITION
!    Sw6$ : SWITCH 6 POSITION
!    Sw7$ : SWITCH 7 POSITION
!    Sw8$ : SWITCH 8 POSITION
! PROGRAM OUTPUTS: NONE
! GPIO OUTPUTS    : 16 bit word to set GPO
!
! VARIABLES
!     Sw1$       : switch 1 position
!     Sw2$       : switch 2 position
!     Sw3$       : switch 3 position
!     Sw4$       : switch 4 position
!     Sw5$       : switch 5 position
!     Sw6$       : switch 6 position
!     Sw7$       : switch 7 position
!     Sw8$       : switch 8 position
!     Sw1        : switch 1 logic level
!     Sw2        : switch 2 ligic level
!     Sw3        : switch 3 logic level
!     Sw4        : switch 4 logic level
!     Sw5        : switch 5 logic level
!     Sw6        : switch 6 logic level
!     Sw7        : switch 7 logic level
!     Sw8        : switch 8 logic level
!     Ctl0,Ctl1: interface address
!     Gpo        : decimal equiv. of word
!
  SUB Switch(Sw1$,Sw2$,Sw3$,Sw4$,Sw5$,Sw6$,Sw7$,Sw8$)
!
!>>> GIVE CONTROL TO GPO <<<
  Ctl0=0
  Ctl1=1
  CONTROL 12,2;Ctl1*2+Ctl0
!
  ASSIGN @Gpio TO 12;FORMAT OFF
!
```

```
78  ! >>> CONVERT SWITCH POSITION TO LOGIC LEVEL <<<
110   SELECT Sw1$
120      CASE "off"
130         Sw1=1
140      CASE "on"
150         Sw1=0
160      CASE ELSE
161         Sw1=1
170         PRINT "WARNING.. SWITCH 1 set to ""OFF"" by default"
180   END SELECT
181 !
190   SELECT Sw2$
200      CASE "off"
210         Sw2=1
220      CASE "on"
230         Sw2=0
240      CASE ELSE
250         Sw2=1
260         PRINT "WARNING.. SWITCH 2 set to ""OFF"" by default"
270   END SELECT
271 !
280   SELECT Sw3$
290      CASE "off"
300         Sw3=1
310      CASE "on"
320         Sw3=0
330      CASE ELSE
340         Sw3=1
350         PRINT "WARNING.. SWITCH 3 set to ""OFF"" by default"
360   END SELECT
361 !
370   SELECT Sw4$
380      CASE "off"
390         Sw4=1
400      CASE "on"
410         Sw4=0
420      CASE ELSE
430         Sw4=1
440         PRINT "WARNING.. SWITCH 4 set to ""OFF"" by default"
450   END SELECT
451 !
460   SELECT Sw5$
470      CASE "off"
480         Sw5=1
490      CASE "on"
500         Sw5=0
510      CASE ELSE
520         Sw5=1
530         PRINT "WARNING.. SWITCH 5 set to ""OFF"" by default"
540   END SELECT
541 !
```

```
550   SELECT Sw6$
560      CASE "off"
570         Sw6=1
580      CASE "on"
590         Sw6=0
600      CASE ELSE
610         Sw6=1
620         PRINT "WARNING.. SWITCH 6 set to ""OFF"" by default"
630   END SELECT
631 !
640   SELECT Sw7$
650      CASE "off"
660         Sw7=1
670      CASE "on"


680         Sw7=0
690      CASE ELSE
700         Sw7=1
710         PRINT "WARNING.. SWITCH 7 set to ""OFF"" by default"
720   END SELECT
721 !
730   SELECT Sw8$
740      CASE "off"
750         Sw8=1
760      CASE "on"
770         Sw8=0
780      CASE ELSE
790         Sw8=1
800         PRINT "WARNING.. SWITCH 8 set to ""OFF"" by default"
810   END SELECT
811 !
813   PRINT
814   PRINT "switch 1",Sw1$,"logic",Sw1
815   PRINT "switch 2",Sw2$,"logic",Sw2
816   PRINT "switch 3",Sw3$,"logic",Sw3
817   PRINT "switch 4",Sw4$,"logic",Sw4
818   PRINT "switch 5",Sw5$,"logic",Sw5
819   PRINT "switch 6",Sw6$,"logic",Sw6
820   PRINT "switch 7",Sw7$,"logic",Sw7
821   PRINT "switch 8",Sw8$,"logic",Sw8
822   PRINT
823 !
824 ! >>> CREATE DECIMAL EQUIVALENT <<<
825 !
827   Gpo=(128*Sw8)+(64*Sw7)+(32*Sw6)+(16*Sw5)+(8*Sw4)+(4*Sw3)+(2*Sw
830   PRINT "binary output to gpo is",Gpo
831 !
832 ! >>> SEND WORD TO GPO <<<
840   OUTPUT @Gpio USING "#,B";Gpo
850   BEEP 1000,.1
851 !
860   SUBEND
```

```
10   !!!!!!!!!!!!!!!!!!!!!!!!
20   ! GPO COUNT SUBROUTINE !
30   !!!!!!!!!!!!!!!!!!!!!!!!
40   !
50   ! This subroutine will test the GPO by
60   ! applying a binary count to the BNC output
70   !
80   ! PROGRAM INPUTS : NONE
90   ! PROGRAM OUTPUTS: NONE
100  ! GPIO OUTPUT    : BINARY COUNT ON GPO BNC
110  !
120  ! VARIABLES
130  !     Ctl0,Ctl1  : interface address
140  !     I          : counter
150  !
160  !
170    SUB Gpocount
180  !
190    PRINT "THIS PROGRAM WILL TEST THE GPO"
200    PRINT "BY COUNTING FROM 0 TO 255"
210    PRINT
220    DISP "FIVE SECONDS PAUSE"
230    PRINT
240  !
250    WAIT 5
260  !
270  ! >>> GIVE CONTROL TO GPO <<<
280    Ctl0=0
290    Ctl1=1
300    CONTROL 12,2;Ctl1*2+Ctl0
310  !
320  ! >>> BINARY COUNT FROM 0 to 255 <<<
330    ASSIGN @Gpio TO 12:FORMAT OFF
340    FOR I=0 TO 255
350       OUTPUT @Gpio USING "#,B":I
360       DISP "Output is a binary",I
370       WAIT .5
380       NEXT I
390  !
400    BEEP 1500,1
410  !
420    PRINT
430    PRINT "TEST COMPLETE"
440  !
450    SUBEND
```

USER NOTES

As a final note, the following points should be kept in mind when using or modifying the GPIO BUS INTERFACE UNIT:

1. The ULO must be in the computer control mode for this interface to work. Be sure to check the ULO switch position before using the interface.
2. Note that the voltage levels on the BNC outputs and the LED indicators correspond directly.
3. For a detailed description of the ULO circuit, see Electronics Division Internal Report No. 144 by D. Schiebel.
4. Remember that each additional board must include at least the simple handshaking circuit described in this report. A more complex circuit may be used in place of the simple one, if desired.
5. The power-up pulse line may be used on additional boards to clear registers, zero counters, etc.
6. The least significant digits of the address lines for the ULO are hard wired to 31. Digits 32 may also be used.
7. BE SURE THAT THE LOGIC LEVEL SWITCHES OF THE HP 98622A GPIO ARE SET TO CORRESPOND TO THE LOGIC LEVELS OF THIS INTERFACE UNIT. SEE PAGE 19.

CREDITS

I would like to thank the following people for their help and ideas concerning this project:

R. Norrod - For his help in the overall design of this project and for his helpful hints on designing the digital circuits.

J. Coe - For his help in forming the future plans for this project and with the overall design ideas.

D. Schiebel - For his helpful descriptions of microprocessor data storage and bus structure and for details of the ULO circuit.

J. Turner - For his help with wirewrapping and the physical layout and construction of the chassis.