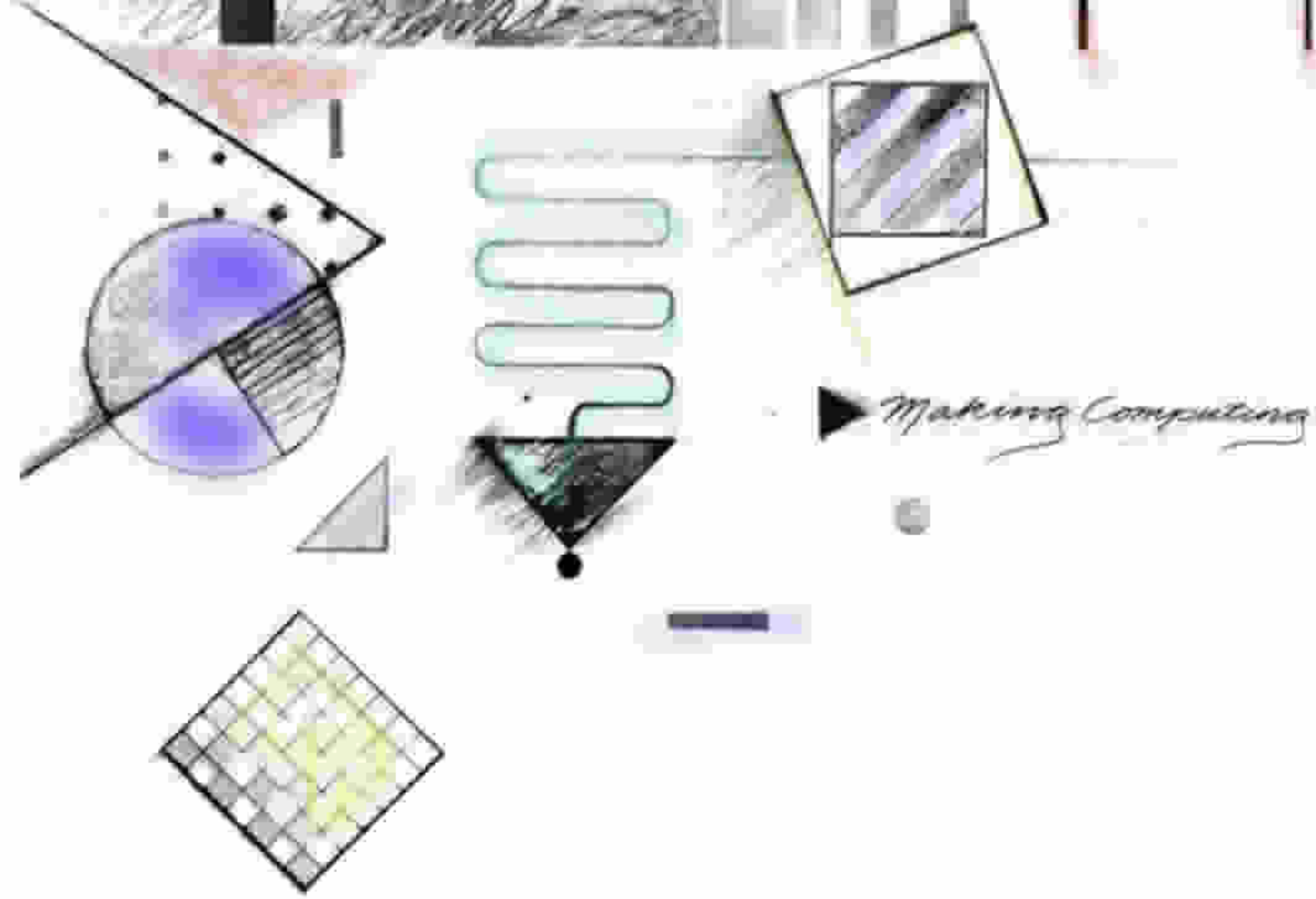


Making Computing Logical

Ridge 32C

multi-user workstation

RIDGE

# technical specifications

***THE RIDGE 32C.*** ***Ridge Computers offers the highest performance per dollar in the industry. And in engineering applications, performance is paramount.***

***Until now, engineers were forced to timeshare on an expensive mainframe located in an air-conditioned room. With the Ridge 32C, they can have a 32-bit computer, capable of executing up to 8 million instructions per second, right in their own office. It's all possible through the Ridge proprietary Reduced Instruction Set Computer (RISC) design.***

***The Ridge 32C also supports the simultaneous operation of four high-resolution graphics terminals, so an engineering team can quickly see results and modify their design.***

***And, through the Ridge UNIX™ System V/4.2 BSD-based multi-user operating system and Ethernet, engineers can share their system with other members of their team.***

### System Overview

- True 32-bit CPU and virtual memory architecture
- 4-stage pipelined architecture
- Affordable, multi-user system
- In the office operation

### Ridge Operating System

- UNIX System V and Berkeley 4.2
- High performance file system
- Fortran 77, C, Pascal, Mainsail™ and assembler languages
- Symbolic debugger

### Virtual Memory System

- 4 Gbytes addressable code and data
- 1-8 Mbytes of memory
- 4 Kbyte demand-paged virtual addressing
- 375 ns memory cycle time

### Mainframe Performance

- 8 million instructions per second (MIPS) peak performance
- 1.5 million operations per second (measured by Whetstone benchmark)

### Graphics and Networking

- High-resolution raster graphics via the Ridge monochromatic display with 1024 x 800 pixels resolution
- High-resolution color monitor, 19" display with 1024 x 768 resolution
- 4 RS-232 ports are available for other devices
- 2 parallel printer/plotter ports
- Ethernet available for networking several systems

### Applications

- A complete and continually growing library of third-party software applications (mechanical and electrical CAD, graphics, databases, etc.)

### Companion Enclosure

- Operates in typical office environment with standard electrical power
- Compact and easily movable
- Style-compatible with Ridge 32C
- Removable top for easy cleaning of tape head
- Accommodates up to 890 Mbytes of hard disc storage and tape system

### Tape System

- Microprocessor-controlled ½" tape drive, 1600 or 3200 bpi tape densit
- Fully automatic tape loading
- Offers both streaming and start/sto operating modes
- Accommodates 7, 8.5, and 10.5-incl reel sizes

### Operating Environment

- Temperature: 5° to 40°C
- Relative Humidity: 5% to 80%, non condensing

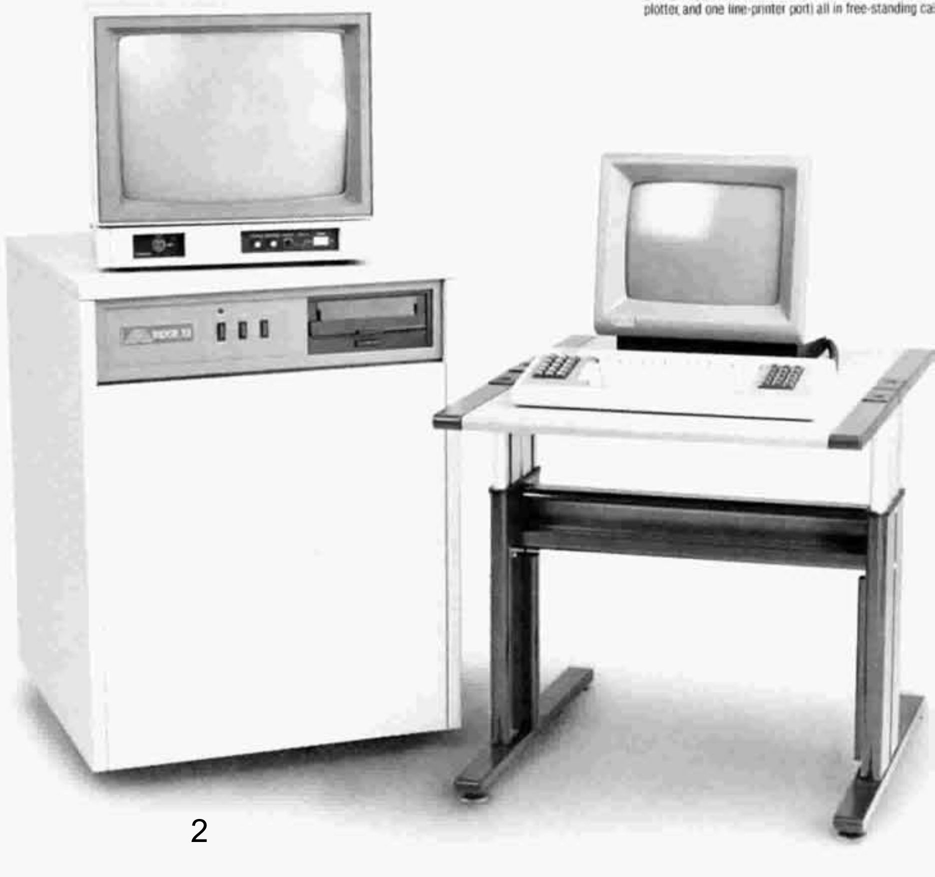
## SPECIFICATIONS

| *Product* | *Height* | *Width* | *Depth* | *Weight* | *V* |
|---|---|---|---|---|---|
| *Ridge 32C* | *35.5"* | *26.5"* | *28"* | *300 lb* | |
| *Companion Enclosure* | *35.5* | *26.6* | *35.5* | *125* | |
| *Tape* | | | | *80* | |
| *Disc* | | | | *137* | |

## CONFIGURATION INFORMATION

| *Model No.* | *Product* | *Description* |
|---|---|---|
| *3241* | *Ridge 32C* | *Base System* with 60 Mbyte Winchester disc* |
| *3242* | *Ridge 32C* | *Base System* with 142 Mbyte Winchester disc* |
| *3243* | *Ridge 32C* | *Base System* plus Companion Enclosure (395 with 445 Mbyte Winchester disc* |
| *3511* | *Disc System* | *Disc Controller plus Companion Enclosure with 445 Mbyte Winchester disc* |
| *3512* | *Disc Expansion Module* | *445 Mbyte Winchester disc* |
| *3531* | *Tape System* | *Magnetic tape system (includes interface and controller)* |
| *3532* | *Tape Drive* | *Magnetic tape drive and cabling* |
| *3951* | *Companion Enclosure* | *Cabinet, contains power distribution and contro unit for peripherals* |

* includes Ridge 32 CPU, 1 Mbyte main memory, 1 Mbyte disc drive, universal interface (four RS-232, one printer plotter, and one line-printer port) all in free-standing ca



RIDGE

Ridge Computers
2451 Mission College Blvd.
Santa Clara, CA 95054
(408) 986-8500

Specifications/configurations subject to change without notice.

© Ridge 1984 5/84 5M Printed in U.S.A. RC127

# RIDGE TO DAC INTERFACE

## *OPERATION*

Data is passed from the Ridge's DR11 board to the DACS through a 64K x 16 buffer of dynamic memory. The memory is hardwired as a FIFO complete with Push and Pop counters connected as its address lines. The arithmetic value - Push counter minus Pop counter -is continually calculated to give an indication  of how full the FIFO is at any moment.

In Push operations, 16 bit data values are loaded from the DR11 to the Interface buffer memory. The "primed' direct memory access (DMA) as described in the DR11 quide is used during Push operations. As many as 64K words can be loaded in one DMA operation, The DAC words should be loaded in a consistent channel sequence.

In Pop operations, the 16 bit data values are loaded from the buffer memory, in the same order in which they were pushed (first in is first out), to the DACs. The DACs are loaded in two operations. First, a set of registers are loaded one channel at a time with the 16 bit data words on the SELECT pulses.  Second, another set of registers are loaded from the first set on the CONVERT pulse. All channels are loaded at once. The second register set is directly connected to the DAC inputs for conversion to analog form.

The Interface derives the CONVERT pulses from a Sample Rate counter (12 bits) clocked by a 16 MHz Crystal oscillator. Immediately after the CONVERT pulse, or, if a Push operation is in progress, immediately after the current push cycle, the Pop operations occur with the SELECT pulses generated. Loading 4  Dacs is done in 4 consecutive Pop cycles after which control is returned to the DMA Push operation. The DACs are then loaded and ready for the next CONVERT pulse to arrive.

In this way the data can be pushed onto the FIFO buffer as fast as possible in a DMA while the the data is popped off the buffer at a set sample rate.

The Interface also includes a MIDI interface, provisions for interrupt driven ADC reads, and provisions for reading the buffer memory one address at a time and setting control parameters

## *MEMORY CYCLE GENERATOR*

The Interface uses dynamic memory for its buffer. Dynamic memory requires a number of specifically sequenced signals for each read or write cycle, RAS - row address strobe, MUX address multiplexer, CAS - column address strobe, WRT - write pulse. Along with these, several signals are needed to load the DACs, clock the data registers, clock the push and pop counters, synchronize with the DR11's DMA signals. See the timing diagram for an illustration of how the main signals line up in time.

The memory access signals are generated by a counter (31) and a shift register (22) clocked by a 16 MHz oscillator (36). During non-active times the counter (31) rests at 0100. The high third bit disables both the counter (31/7) and the shift register data in (22/1). At the start of a memory access cycle a pulse is generated in the 74153 (11/7) which clocks the flip flop (40/11). A load signal for the counter is activated and the counter resets to 0000 synchronous with the 16 MHz clock. The counter then counts up to 0100 where it stops again. The third counter bit (31/12) is low for 250 ns. This 250 ns pulse is then shifted through the shift register to generate the Q0 through Q7 signals. See the timing diagram. All the memory access signals are the derived from these Q signals. Only one signal burst is generated per counter load as the counter's third output bit stops the process on going high.

The status of the PUSH and POP signals determine when the next signal burst is to happen, (11),

| **PUSH** | **POP** | |
|---|---|---|
| **0** | **0** | next cycle when one of the two goes high |
| **0** | **1** | next cycle at end of Q7 as long as POP is active |
| **1** | **0** | current cycle stops at end; PUSH goes to 0 at the end of each cycle to wait for the DMA signals to reactivate it; next cycle when push goes high again. |
| **1** | **1** | state disallowed by gate 47/8. |

## *PUSH*

Data is pushed into the Interface's FIFO buffer through a Direct Memory Access (primed) controlled by the DR11. The user initiates a primed DMA transfer from the DR11 control registers. The following DR11 signals are then generated. See the DR11 documentation for more information. A GO pulse starts the transfer and the READY signal goes low for the duration of the DMA. BUSY goes high followed by the END pulse. Data from the DR11 register (DO) is loaded into holding registers at the interface (18,19) on the front edge of the END pulse, At the same time the interfaces PUSH signal is clocked high (46/6). This starts the first memory write signal burst.

The memory load ends at the final edge of the WRT pulse which turns off the PUSH signal (46/3). At the same time a CYCLE REQUEST (49/13) signal is generated at the interface to be sent to the DR11 to start another BUSY And END signal. In the meantime, the low PUSH allows a POP sequence of memory reads to occur if the convert strobe has occurred (77/3). POP memory access is given high priority since the DACs need to be ready for the next CONVERT STROBE The END pulse from the DR11 again loads the data (DO) holding registers and readies the PUSH signal (46/6) but the actual memory write cycle may be delayed until any POP sequence is finished. After the PUSH cycle the WRT signal again ends the cycle and generates the CYCLE REQUEST signal. This repeats for the entire DMA. Finally, at the end of the DMA transfer the READY line goes high which prevents more CYCLE REQUEST signals PUSH goes low after the last data has been written into memory and remains low.

Each memory access lasts for about 750 ns. A POP cycle for four DAC channels lasts for about 3 uses. POP cycles occur at the sample rate - 44KHz = 1/ 23 usec.

The FULL signal will also inhibit the monostable generating the CYCLE REQ signals. After a POP cycle the FULL signal will go low and a CYCLE REQUEST is generated

## POP

The DACs have two parallel sets of registers at its data inputs. The first set is loaded one channel at a time by the SELECT STROBE. At the CONVERT STROBE the contents of the first register is loaded into the second register set. The contents of the second register set is then converted.

The CONVERT STROBE is generated in the interface by a programmable counter clocked by a 16 MHz crystal oscillator (36.37,38,39,49).

The final edge of the CONVERT STROBE also sets a flip flop (77/5) to ready a POP cycle. If a PUSH cycle is not currently in progress then POP goes high immediately. If a PUSH cycle is in progress, then POP is activated only after PUSH goes low and goes high. A high POP releases the asynchronous load on the CHANNEL down counter (8) which has been loaded with the number of DAC channels to be used. A memory read cycle and a SELECT STROBE then occurs for each DAC channel. When the CHANNEL counter reaches zero count the POP signal is released.

Note that the POP signal masks cut the PUSH signal so that All DAC channels are loaded consecutively without interruption. Both the CONVERT STROBE and the POP signal can be inhibited by a low RUN signal of the cuntrol sinals set by the user to start and stop the conversion process.

Note that the POP cycles also generate a RAS-only memory refresh for the dynamic memory
No WRT signals occur during the POP cycles.

# RIDGE TO DAC INTERFACE

## *REGISTERS*

| F2 | F1 | REG | READ (7/4,5,6,7) | WRITE (chip 42) |
|---|---|---|---|---|
| 0 | 0 | 0 | ADC 0 | BUFFER MEMORY WRITE |
| 0 | 0 | 1 | BUFFER MEMORY OUT | BUFFER MEMORY READ |
| 0 | 1 | 0, 1 | CONTROL REGISTER | CONTROL REGISTER (8 bits) |
| 1 | 0 | 0 | PUSH MINUS POP | MIDI LOAD (8 bits) |
| 1 | 1 | 0 | ADC 1 | SAMPLE RATE (12 bits) |
| 1 | 1 | 1 | MIDI READ | BUFFER MEMORY ADDRESS |

F3 is used to generate the data load signal for Register Writes. Normally F3=0. To write into an interface register first load the data into the DR11 D0 register; second, set F2,F1,F3 = x,x,l; finally set F2,F1,F3 = x,x,0. (x stands for the codes given in the chart above).

## *Control Register*

| BIT | SIGNAL | DESCRIPTION |
|---|---|---|
| 0 | RUN | Set high to enable the DAC conversions and buffer memory POPs. |
| 1 | CHNL#0 | The number of DAC channels to be loaded. |
| 2 | CHNL#1 | 00=4 channels, 01=1 channel, 10=2 channels, 11=3 channels. |
| 3 | REG | Register bit to allow extra interface registers as shown in the chart above. |
| 4 | MIDI DAV | In control reads a high shows MIDI data available at MIDI in. In control writes a high enables an interrupt from MIDI DAV signal. |
| 5 | ADC | In control read, a high shows data available from one of the ADCs. In control writes a high enables an interrupt from the ADCs. |
| 6 | EMPTY | In control reads, a high indicates an empty buffer FIFO. |
| **7** | FULL | In control read, a high shows FIFO is full. In control writes, a high enables an interrupt on FIFO full. |

## *SINGLE MEMORY READ AND WRITE*

The user can can access single memory addresses in the buffer by loading an address and then initiating either a write or a read cycle. These single memory acceses can occur only when there is no PUSH or POP operation in progress.

To load the buffer memory address follow the following steps:

1. Put the 16 bit address on the DI lines of the DR11.
2. F3,F2,F1,REG = 1111
3. F3,F2,F1,REG = Oxxx

Two registers (16,17) are loaded with the address and flip-flop #10/5 assures that the next single memory access, be it a read or write, will get its address lines from registers #16 and #17 instead of from the pop or push counters.

To start a single memory write cycle follow the following steps

1. Put the 16 bit data on the DI lines of the DR11.
2. F3,F2,F1,REG = 1000
3. F3,F2,F1,REG = Oxxx

This action generates a WR pulse (62/15) which will start a PUSH cycle The address for this cycle comes from the PUSH COUNTERS unless a memory address write was done immediately preceding the above action. In either case, the push counters are incremented at the end of the cycle.

To start a single memory read cycle follow the following steps:

1. F3,F2,F1,REG = 1001
2. F3,F2,F1,REG = 0xxx
3. Pick up the 16 bit data from the DO lines of the DR11.

This action generates a RD pulse (42/14) which will start a POP cycle. The address for this cycle comes from the POP COUNTERS unless a memory address write was done immediately preceding the above action. In either case the pop counters are incremented at the end of the cycle. CHNL#1,O should be set equal to 01 before starting the single memory read.

# SUMMARY OF PROGRAMS ON THIS DISK

## SEPT. 1986 (written in C)

### *MIDI PLAY:*

Use in the form " midiplay < mididata ".
Will generate midi out commands from the interfaces midi circuitry. Mididata is a list of standard midi data along with timing data in milliseconds. Timing of the data is done with the Ridge timers in an interrupt system.

### *DAC:*

Use in the form " dac file ", where 'file' is a sound data file. Will feed a sound data file to the interface buffer and then to the Xerox DACs. Starts by asking for number of channels and sample rate. Uses the DR11 DMA capability.

A number of programs can be used to generate simple sound data files:

CONSTANT: creates a file called CDATA made up of a repeated 16 bit constant.

SAWTOOTH; creates a file called STDATA made up of one cycle of a sawtooth type waveform.

SQUARE: creates a file called SQDATA made up of one cycle of a squarewave.

### *DAC WRT:*

A version of the DAC program to be used for testing purposes. Use in the form "dac wrt file " where file is a sound data file. Starts an infinite loop of sending the sound data file to the dacs through DMA transfers. Useful for checking the circuit signals on an oscilliscope. Be sure and disable the 'Empty' and 'Full' flipflops in the circuit.

### *TEST:*

Used to test writing to and reading from the interface registers.

## *MIDIREAD:*

Midi data comning into the interface is set up to interrupt the Ridge. The incoming data is read and placed in a data byte file called "dataread". The data is read in packets - after receiving a data word a small delay is generated (consecutive mini bytes come at a maximum speed of 0.33 ms) and then the interface is tested for the arrival of another data word. If another byte is ready for pickup, the byte is read, placed in a temporary array, and then the delay/test is repeated. If the test indicates no bytes have arrived, then it is assumed that the data packet of fast consecutive bytes is finished and there is time to do the following jobs before the next data packets arrives: First, a 3 byte time code is calculated using the times(O) function. This code is placed in the midi data file in the form "f1, , hibyte, lowbyte". Next, the time data and the temporary array is loaded into the "dataread" file, finally, the program waits for the next interrupting data arrival. The process is stopped by hitting the DEL key.

To read the data file 'dataread' use: od -x /usr/local/drll/dataread

This program has the problem of missing some of the midi codes which arrive at the midi-in port. Data coming in overruns previous data before the program gets around to reading it. Several attempts were made to correct this problem but with no success. It is a timing problem. Both the times(0) function and the array loading take too much time. and the data packets are broken up by the Active Sensing (FE) midi signal coming every 300 ms. .A hardware buffer for the incoming data has been suggested as a solution.

## *ADCTEST*

Used to test the Analogic Analog to Digital converter installed in the interface box. This program does not use DMA transfers. It simply sets F1=0 and F2=1 and ADC/DAC=1 which puts the AN lines on the DR11 Read bus (interface to ridge). Pulsing the F3 signal will then start a single ADC conversion cycle.

The program is set up to initiate two conversion cycles (one for each adc channel) after each hit of the return key. The conversion results are printed out on the screen in both hex and decimal format. Note that the least significant bit is the channel number (the adc output is only 15 bits). The most significant bit is the sign bit. The program converts the ADC output from complementary offset binary to 2's complement binary.

SECTION 7 - DR11 INTERFACE CONTROLLER

Ridge DR11 Interface Specifications

## *Introduction*

The Ridge DR11 is an asynchronous parallel interface for peripheral equipment or an intercomputer link, the interface emulates the DR11W defined by the Digital Equipment Corporation; however, not all of the functions and modes have been implemented, and the timing has been more rigorously defined. The maximum transfer rate of the Ridge DR11 is 2M bytes per second.

## *Interface Signals*

FROM PERIPHERAL EQUIPMENT:

DI0-l5 DATA IN - A 16-bit unidirectional data bus from the peripheral. The high order 8 bits are lines D17 through DIO and low order byte is D1l5 through DI8.

CYCLE REQ A,B CYCLE REQUEST A and B - Either line can initiate a one word transfer, provided READY and BUSY are negated.

ATTN ATTENTION - Halts the data transfer, sets READY, and generates an interrupt to the host CPU.

STATUS A,B,C Status or control signals to be defined by the peripheral device.

## Specifications

TO PERIPHERAL EQUIPMENT:

| | |
|---|---|
| D00-15 | DATA OUT - A 16-bit unidirectional data bus to the peripheral. The high order B bits are lines D07 through DOO and the low order byte is DOlS through DOS. |
| INIT | Initialization or reset command. |
| FUNCT 1.2,3 | Control signals to be defined by the peripheral device. |
| GO | A pulse generated at the beginning of a DMA block transfer. |
| READY | Reset with GO; set when the DMA transfer has |
| BUSY | When BUSY is asserted, the host CPU is processing a CYCLE REQUEST. Another CYCLE REQUEST cannot be issued until BUSY is negated with READY asserted. |
| END CYCLE | A pulse generated when BUSY is negated. |
| ACLO FUNCT2 | Same as FUNCT 2. |

## SIGNALS NOT IMPLEMENTED:

The following signals are included in the DEC DR11W but are not implemented in the Ridge version of the interface:

CO CNTHL Cl CNTHL WC
INC ENB BA INC ENB A00
BURST

## ELECTRICAL INTERFACE:

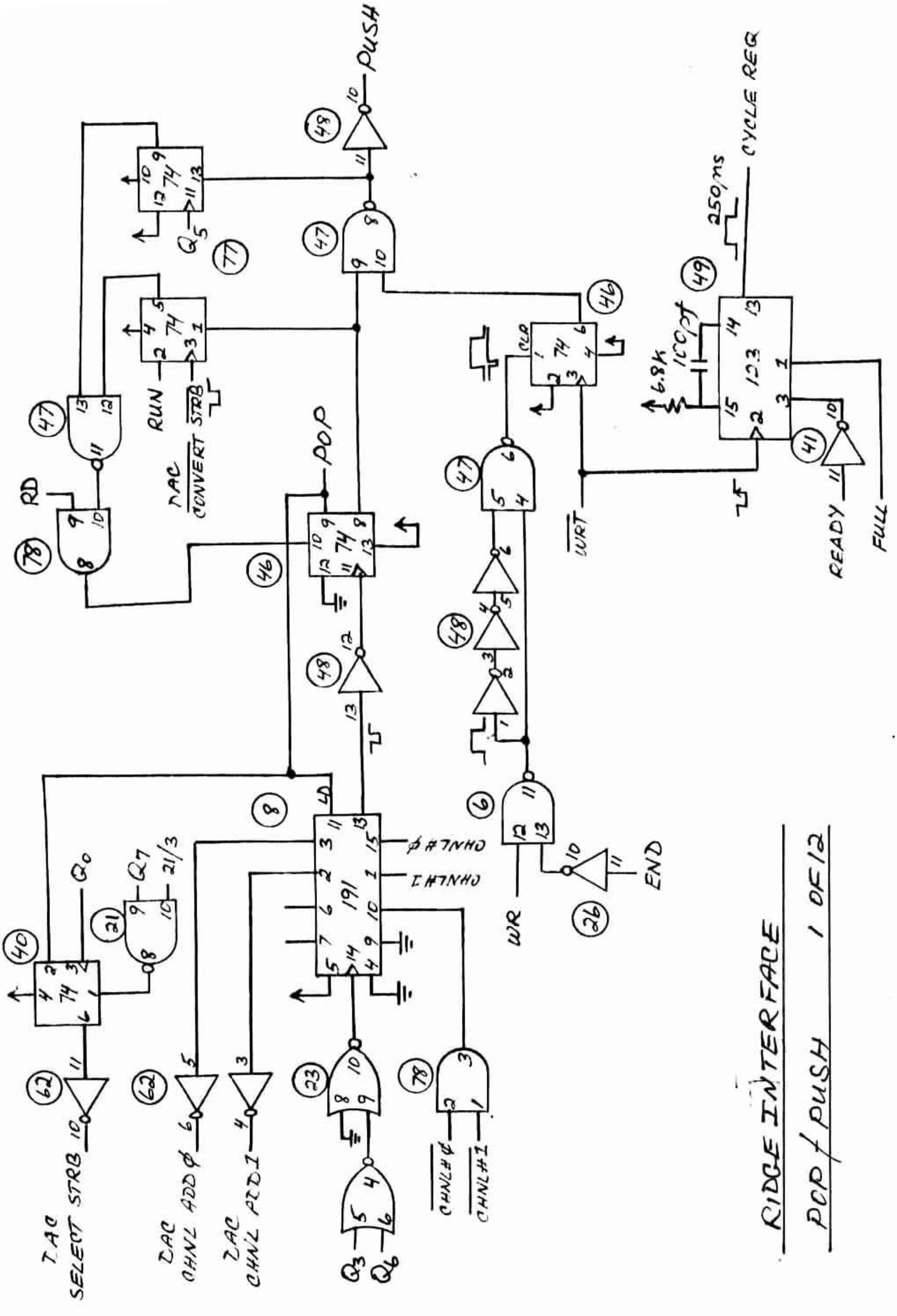
The receivers are Motorola MC3437, National DS8837 or Signetics 8T37 and the drivers are 74S38.

There is one receiver for each input, and there is one driver for each output. The two FUNCT 3 lines each have a driver, and there is a receiver for both CYCLE REQ A and CYCLE REQ B. STAT C is received only on Jl pin 25.

220/330 Ohm termination is installed at both the drivers, the receivers, and all unimplemented lines.

## *RIDGE DR11 PIN ASSIGNMENTS*

| PIN | (DEC) | J1 SIGNAL | J2 SIGNAL |
|---|---|---|---|
| NUMBER | | SIGNAL | SIGNAL |
| 1 | (VV) | +00 15 | +D1 15 |
| 2 | (UU) | +00 00 | +D1 00 |
| 3 | (TT) | +00 14 | +01 14 |
| 4 | (SS - | -i-DO 01 | +D1 01 |
| 5 | (RR) | +00 13 | +01 13 |
| £ | (PP) | +00 02 | +01 02 |
| 7 | (NN) | +00 12 | +01 12 |
| 8 | (MM) | +00 03 | +DI 03 |
| 9 | (LL) | +DO 11 | +DI 11 |
| 10 | (KK) | +00 04 | +DI 04 |
| 11 | (33) | +00 10 | +01 10 |
| 12 | (HH) | +00 05 | +01 05 |
| 13 | (FF) | +00 09 | +DI 09 |
| 14 | (EE) | +00 06 | +DI 06 |
| 15 | (DD) | +00 08 | +01 08 |
| 16 | (CC) | ±00 07 | +01 07 - |
| 17 | (BUY | GROUND | GROUND |
| 18 | (AA) | GROUND | GROUND |
| 19 | (Z | +CYCLE RQ B | GROUND |
| 20 - | (Y) -- | GROUND - | GROUND |
| 21 | (X) - | +END CYCLE | +GO |
| 22 | (W) | GROUND | GROUND |
| 23 | (V) | +STATUS C | +FNCT 1 |
| 24 | (U) | GROUND | GROUND |
| 25 | (T) | | |
| 26 | (5) | GROUND | GROUND |
| 27 | (R) | +STATUS B | +FNCT 2 |
| 28 | (P) | GROUND | GROUND |
| 29 | (N) | +INIT | |
| 30 - | (ML | GROUND | GROUND |
| 31 | (L) | +STATUS A | +FNCT 3 |
| 32 | (K) | | +FNCT 3 |
| 33 | (3) | | |
| 34 - | (H) | GROUND | -GROUND |
| 35 | (F) | +READY | |
| 36 | (E) | GROUND | GROUND |
| 37 | (D) | +ACLO FNCT 2 | +ATTN |
| 38 | *(C)* | GROUND | GROUND |
| 39 | (B) | +CYCLE RQ A | +BUSY |
| 40 | (A) | GROUND | GROUND |



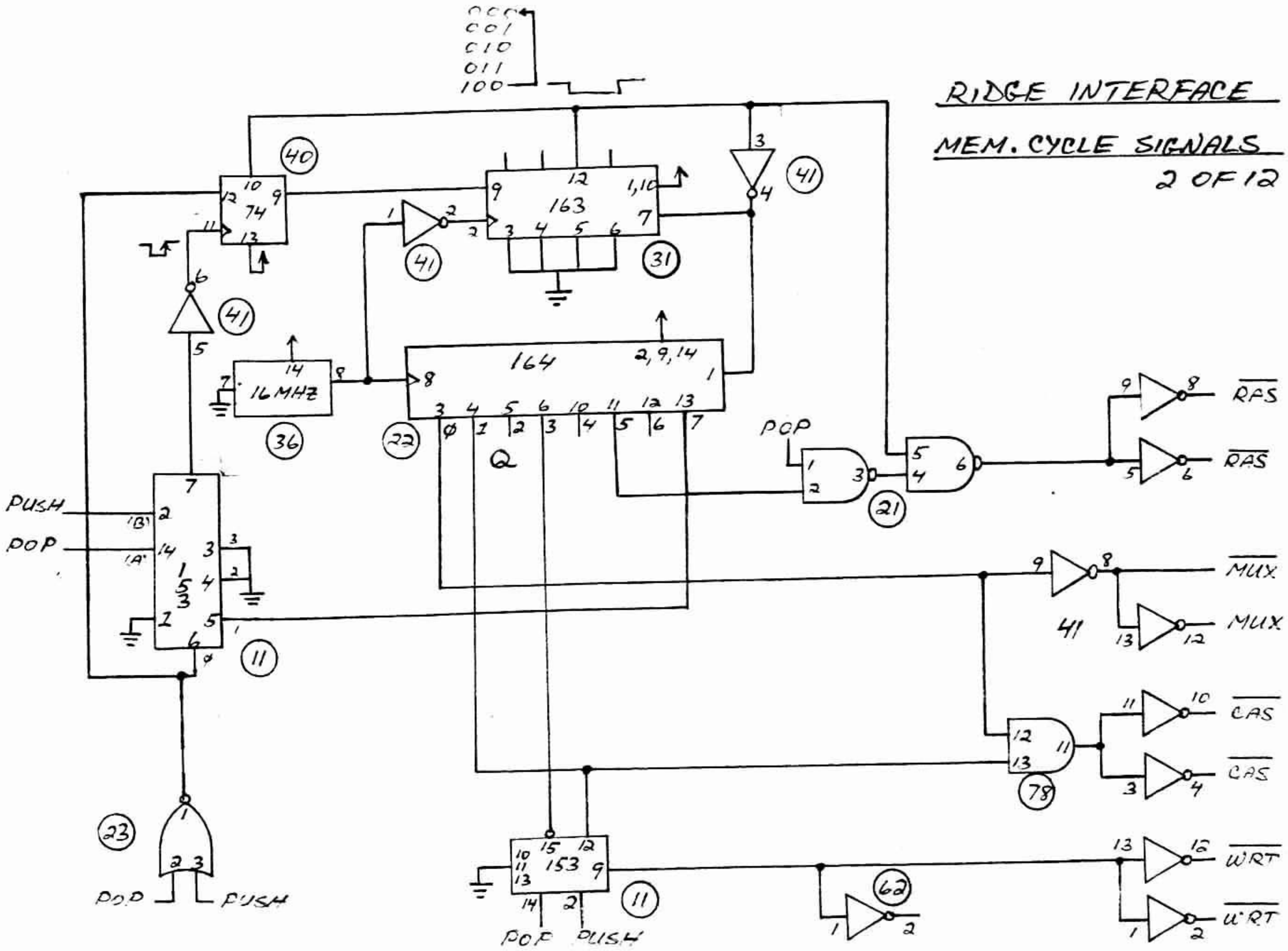
RIDGE INTERFACE

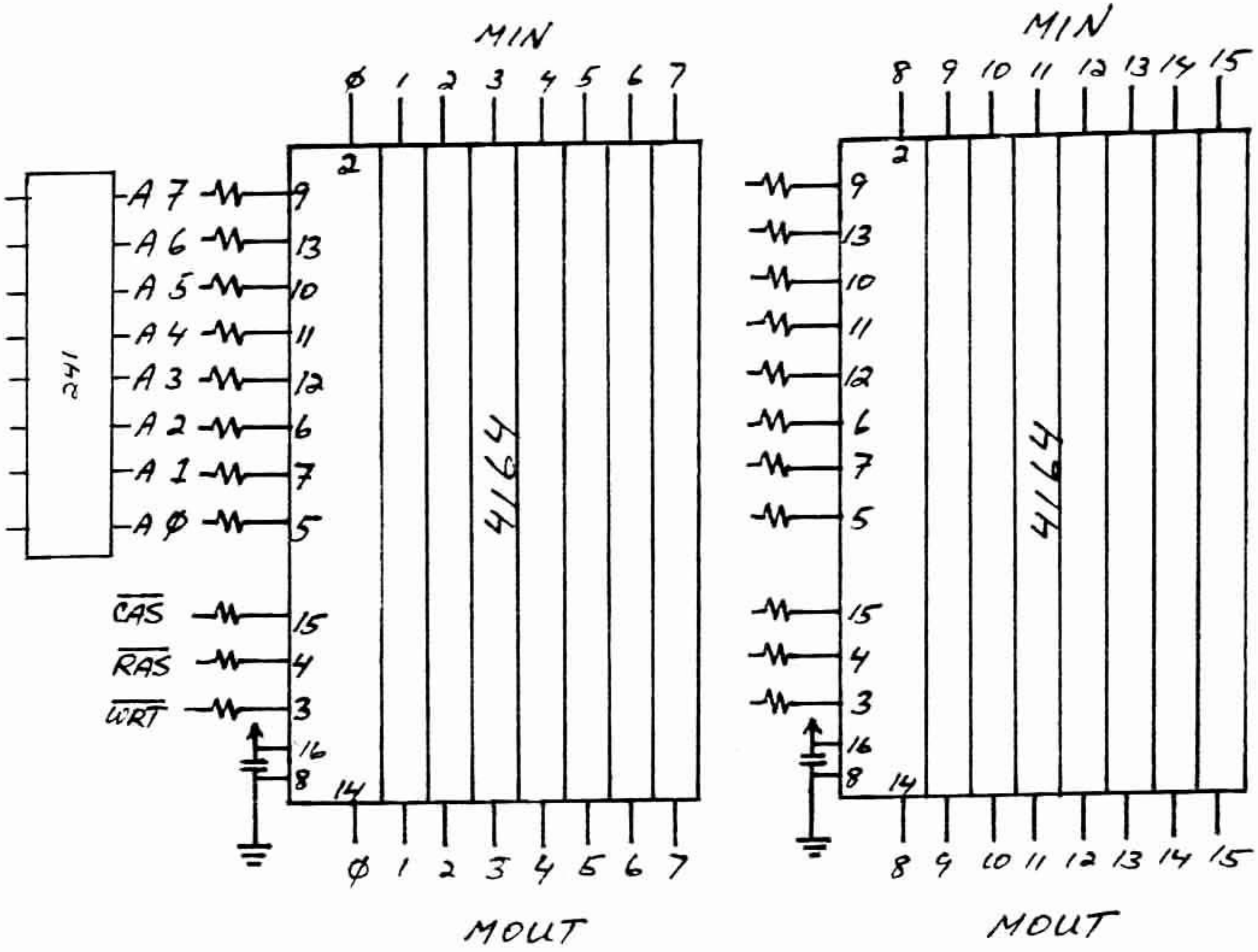
POP / PUSH 1 OF 12

## RIDGE INTERFACE

## MEM. CYCLE SIGNALS

2 OF 12

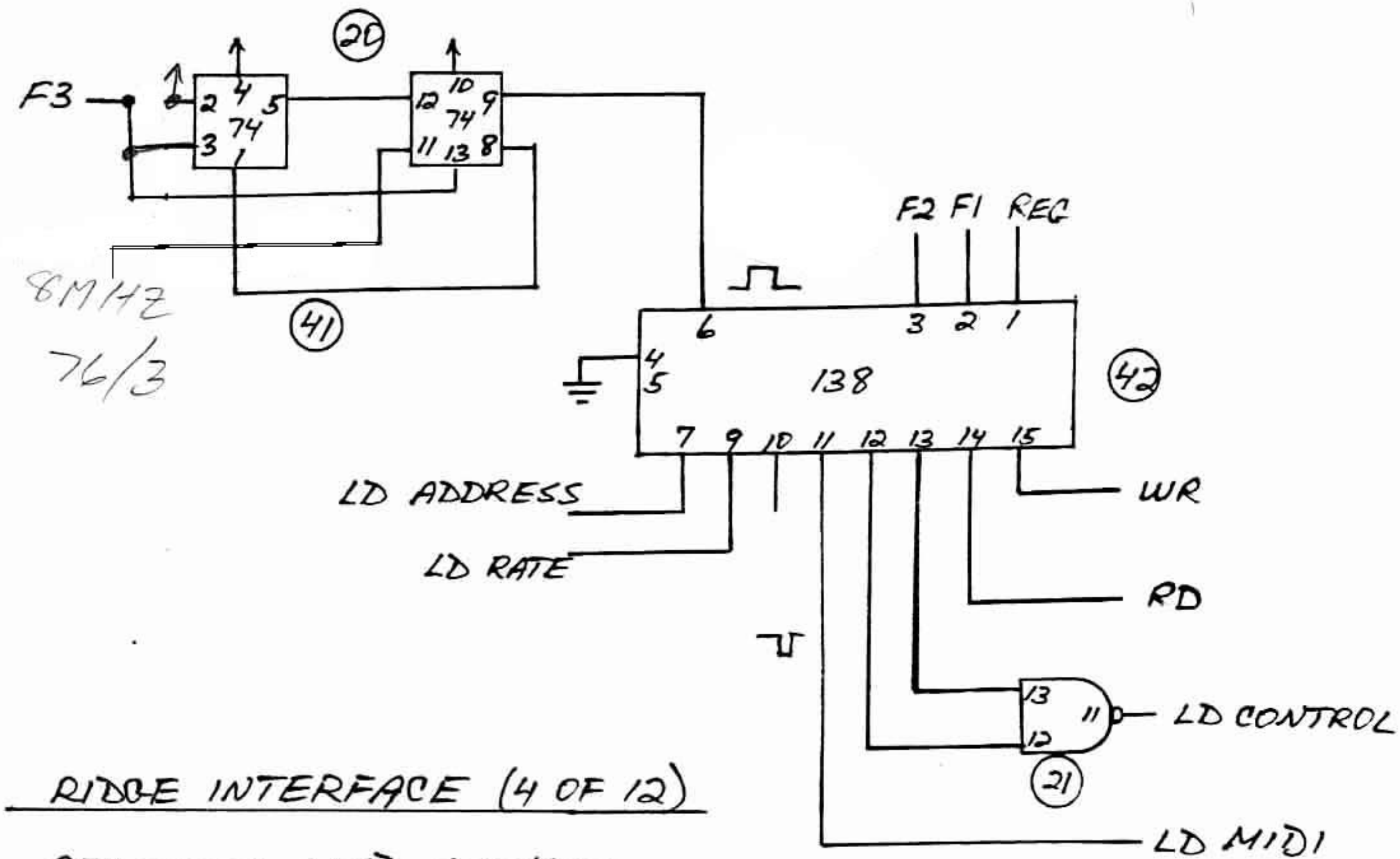




64K x 16 DYNAMIC MEMORY

FIFO MEMORY - RIDGE INTERFACE

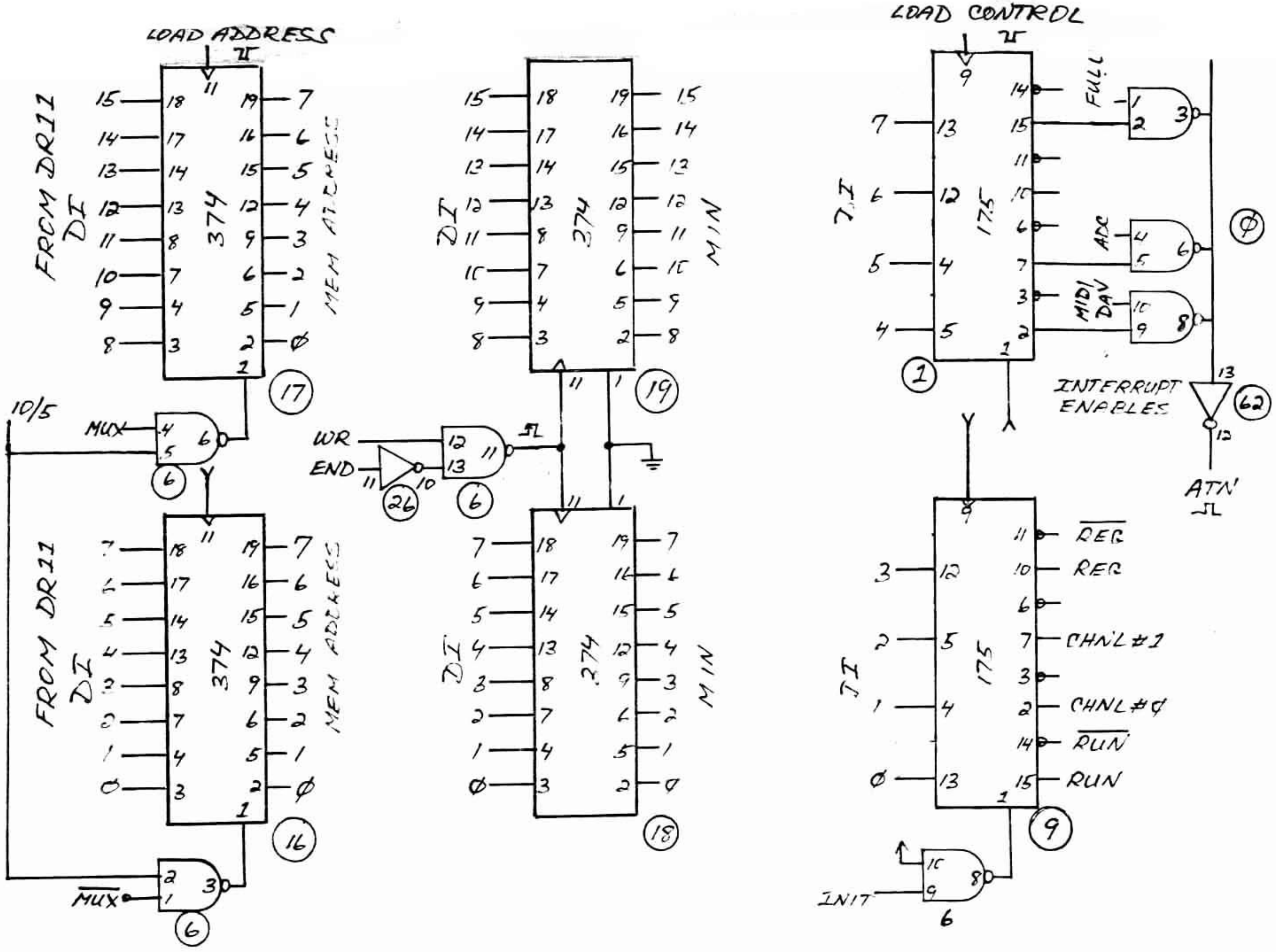
(3 OF 12)

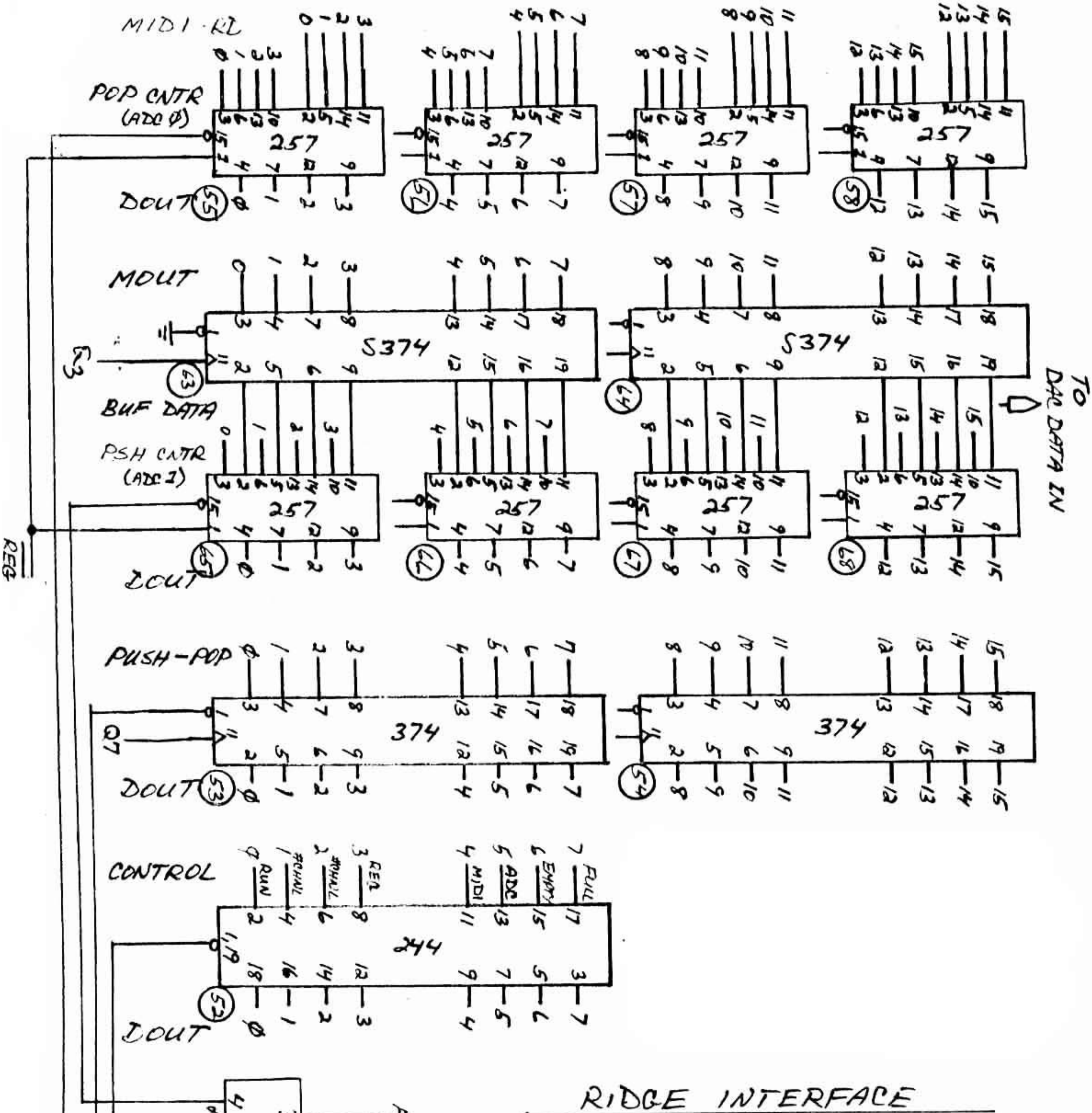


RIDGE INTERFACE (4 OF 12)

REGISTER LOAD SIGNALS

REGISTERS
RIDGE INTERFACE (5 OF 12)



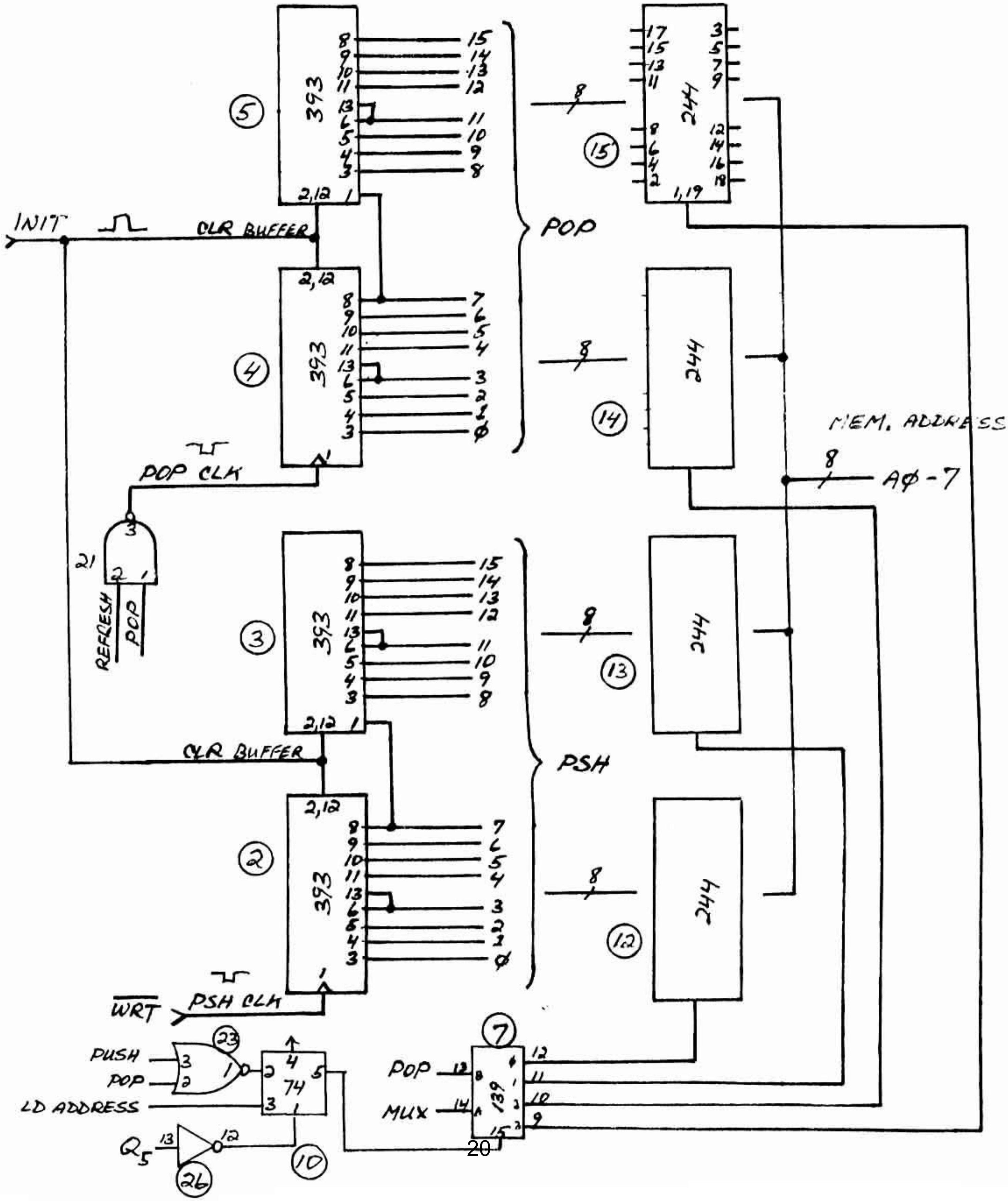


RIDGE INTERFACE

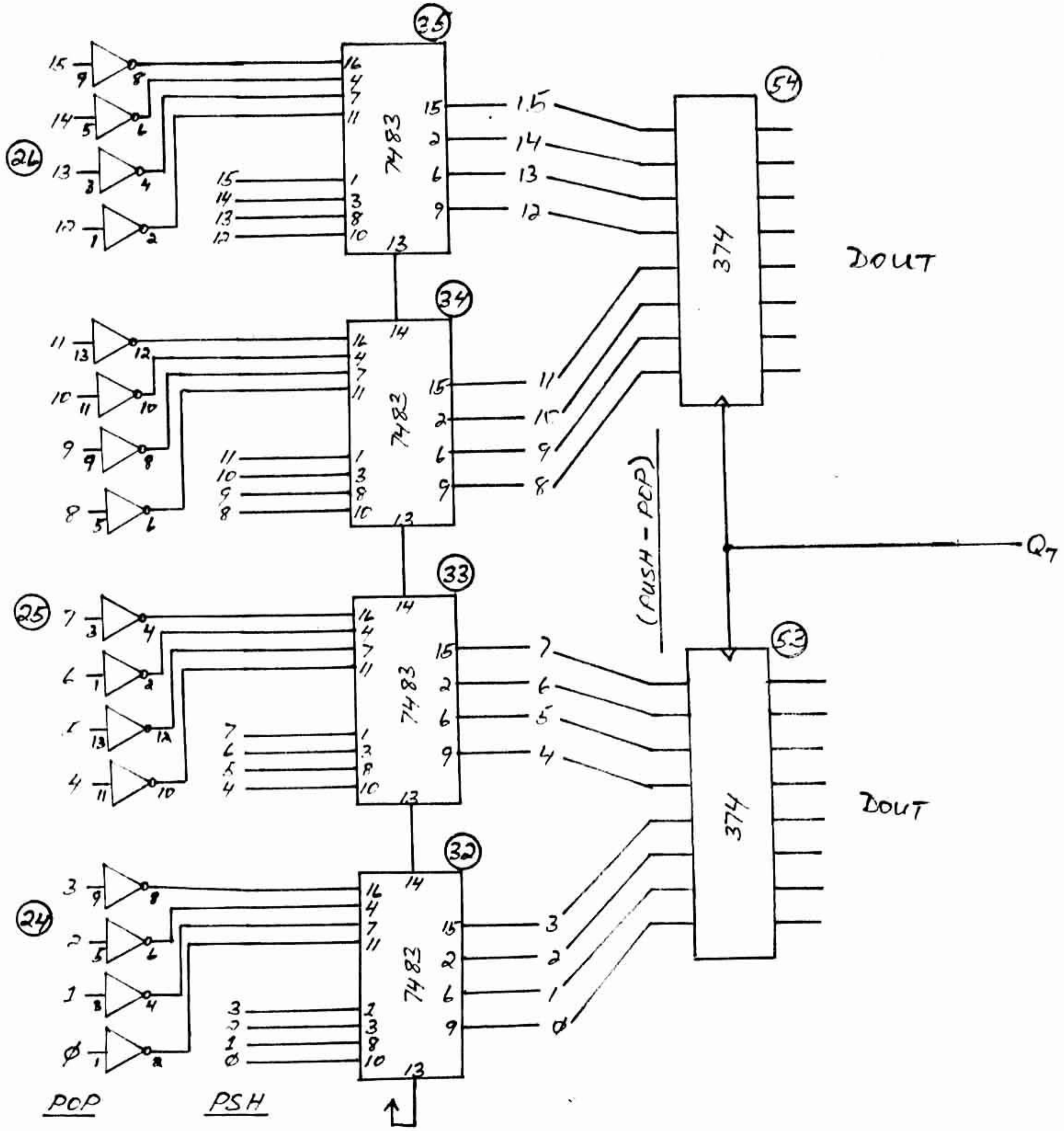
READ REGISTERS

(6 OF 12)

## RIDGE INTERFACE (7 OF 12)
## PUSH - POP COUNTERS



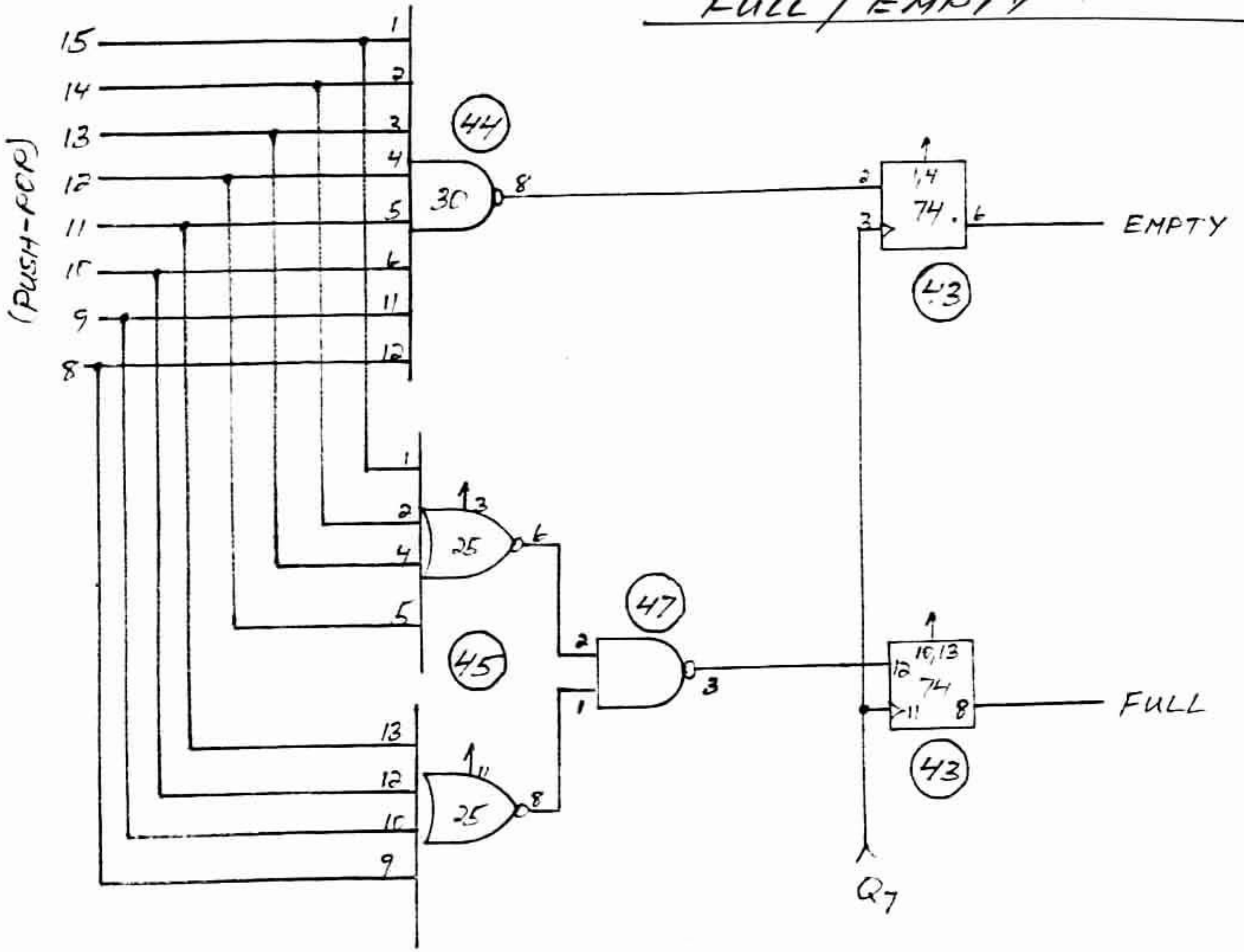
RIDGE INTERFACE (8 OF 12)
(PUSH - POP)



5v - pin 5
GND - pin 12

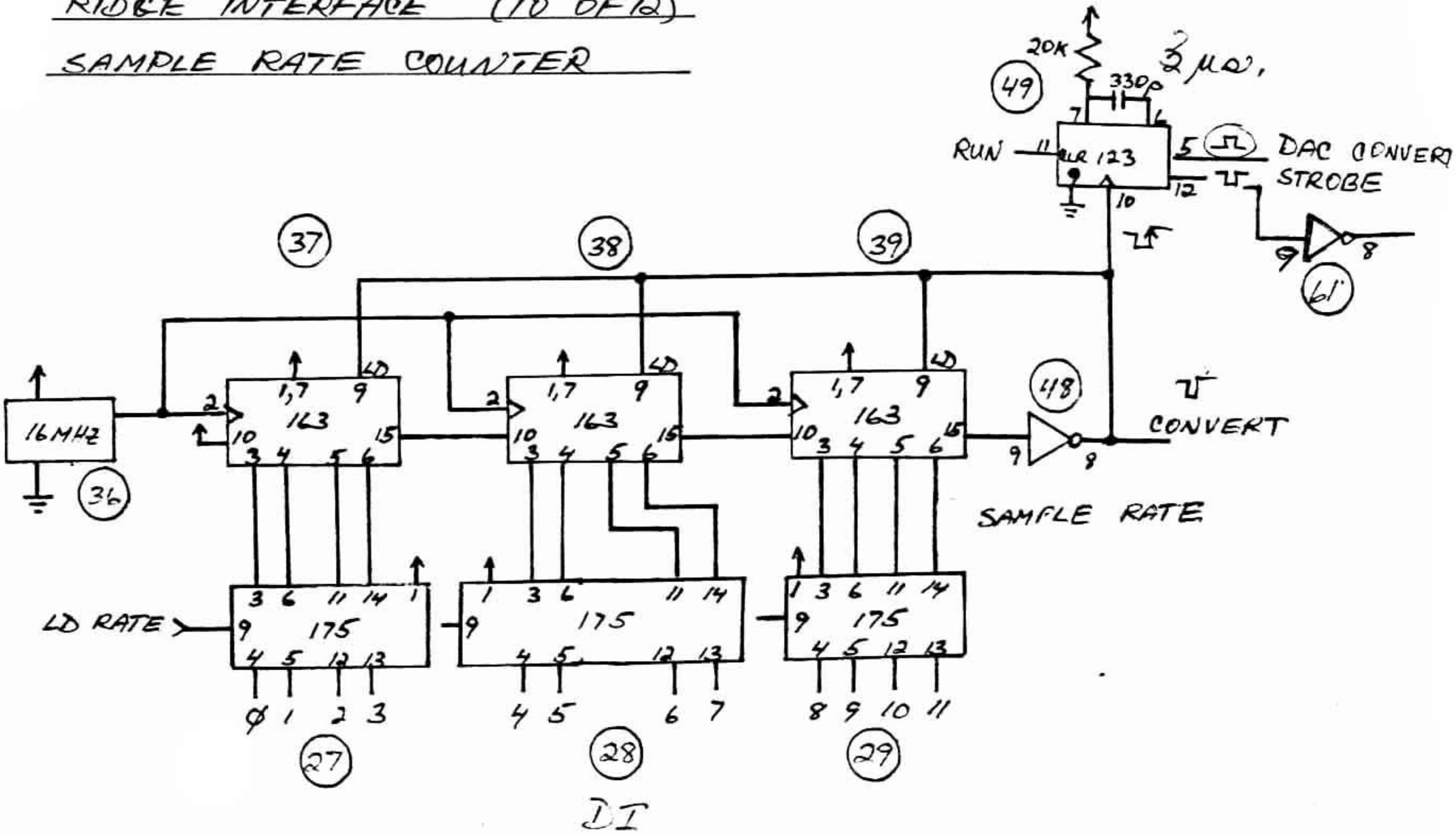
## RIDGE INTERFACE (9 OF 12)

## FULL / EMPTY



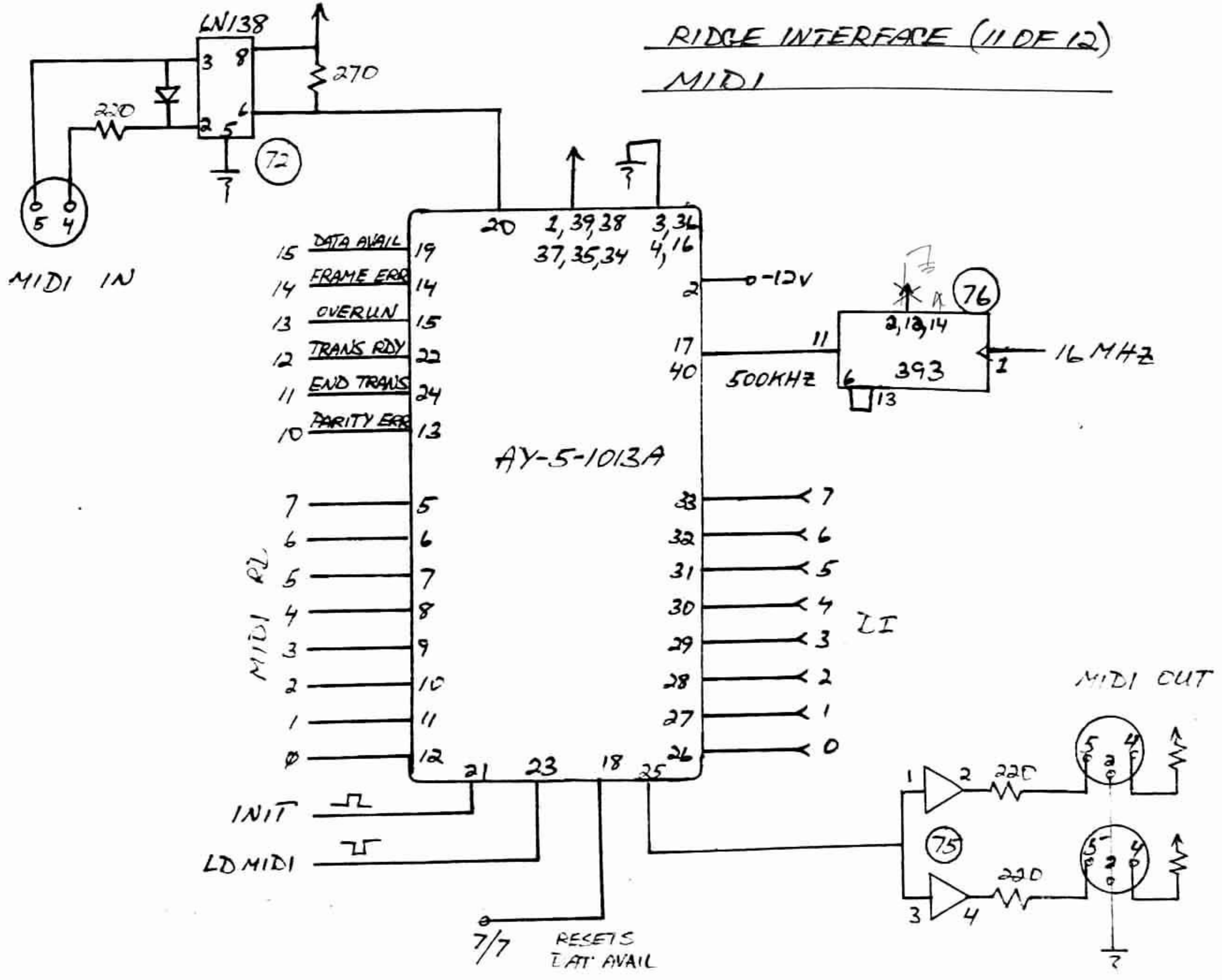
## RIDGE INTERFACE (10 OF 12)

## SAMPLE RATE COUNTER



# RIDGE INTERFACE (11 OF 12)

## MIDI



# RIDGE INTERFACE (12 OF 12)
## BOARD / PIN SIDE

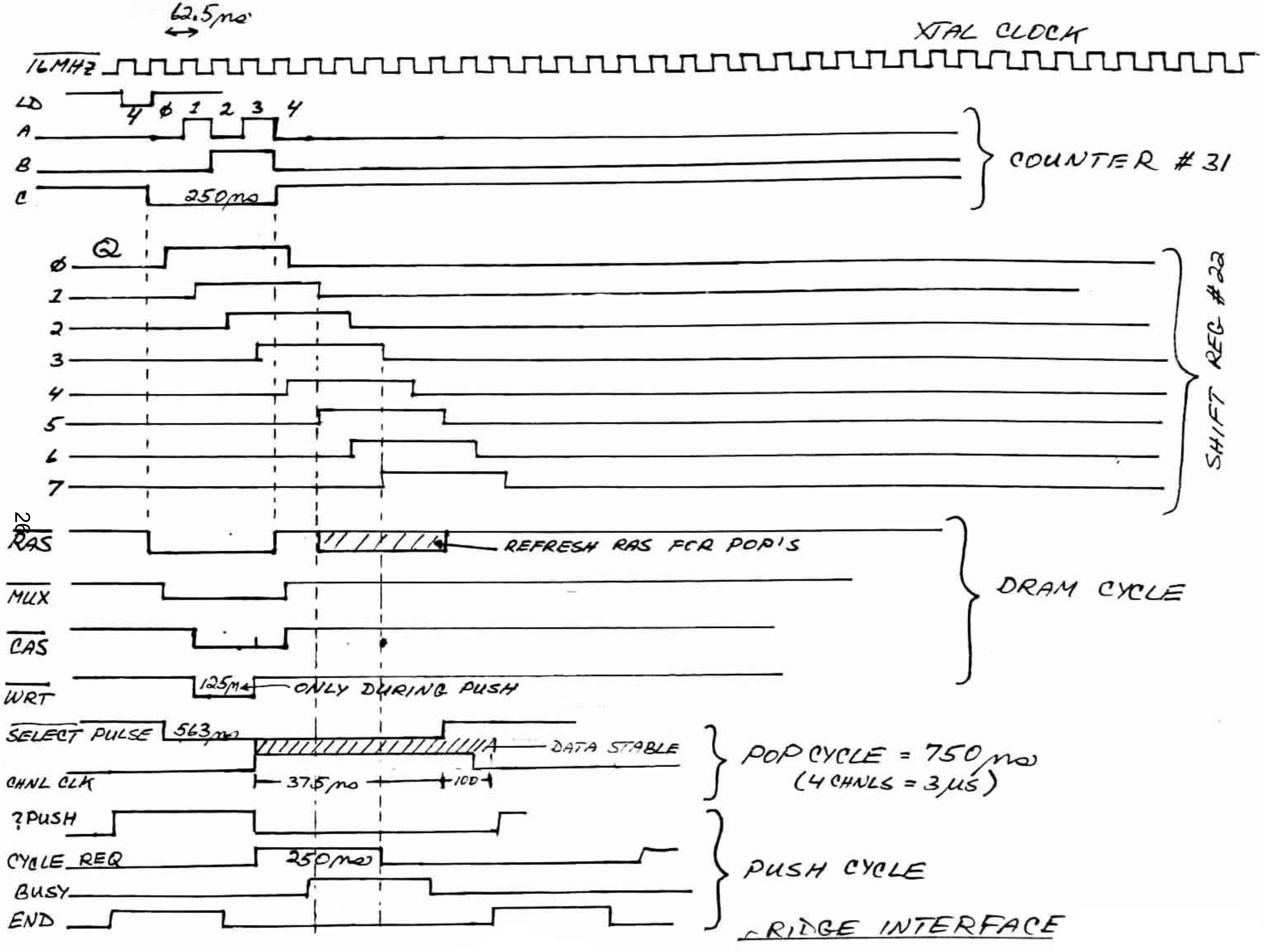
|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| ADC JMP<br>79 | 08<br>78 | 74<br>77 | 393<br>76 | 07<br>75 | MIDI JMP<br>—<br>74 | —<br>73 | LN138<br>72 |  |  |
| ADC JMP<br>69 | 257<br>68 | 257<br>67 | 257<br>66 | 257<br>65 | S374<br>64 | S374<br>63 | 14<br>62 | UART<br>AY-5-<br>1013A |  |
| 59 | 257<br>58 | 257<br>57 | 257<br>56 | 257<br>55 | 374<br>54 | 374<br>53 | 244<br>52 |  |  |
| 123<br>49 | 04<br>48 | 00<br>47 | 74<br>46 | 25<br>45 | 30<br>44 | 74<br>43 | 138<br>42 | 04<br>41 | 74<br>40 |
| 163<br>39 | 163<br>38 | 163<br>37 | XTAL<br>36 | 83<br>35 | 83<br>34 | 83<br>33 | 83<br>32 | 163<br>31 | 30 |
| 175<br>29 | 125<br>28 | 175<br>27 | 04<br>26 | 04<br>25 | 04<br>24 | 02<br>23 | 164<br>22 | 00<br>21 | 74<br>20 |
| 374<br>19 | 374<br>18 | 374<br>17 | 374<br>16 | 244<br>15 | 244<br>14 | 244<br>13 | 244<br>12 | 153<br>11 | 74<br>10 |
| 175<br>9 | 191<br>8 | 139<br>7 | 00<br>6 | 393<br>5 | 393<br>4 | 393<br>3 | 393<br>2 | 178<br>1 | 03<br>0 |

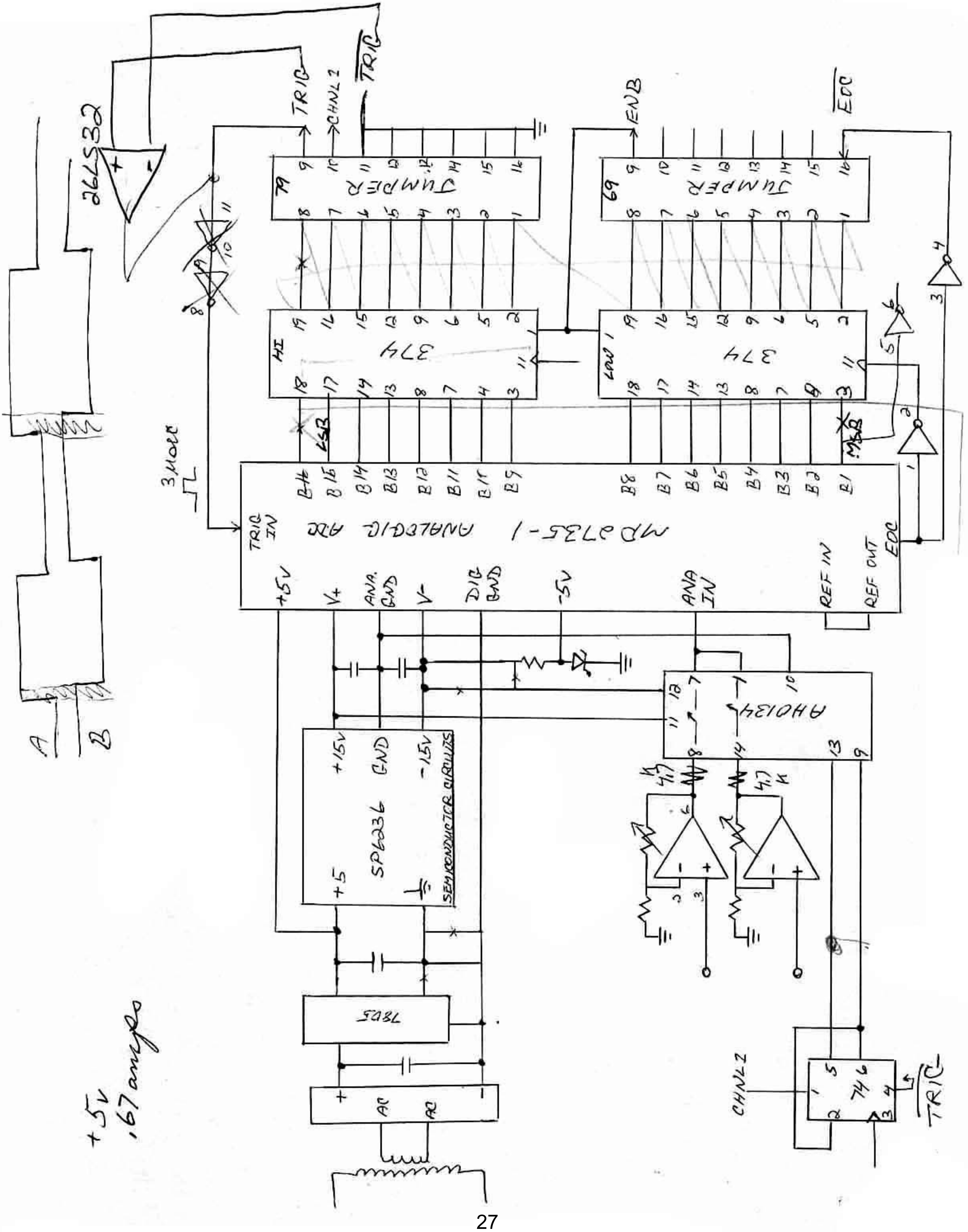
2 1
DR11-J1 DO's
P2

2 1
DR11-J2 DI's
P1

TO DACS
P4

TO LED's
P3


62.5ns
XTAL CLOCK
16MHz
LD
4 ø 1 2 3 4
A
B
C
250ns
COUNTER # 31
Q
ø
1
2
3
4
5
6
7
SHIFT REG #22
RAS
REFRESH RAS FCR POP'S
MUX
CAS
WRT
125ns
ONLY DURING PUSH
DRAM CYCLE
SELECT PULSE
563ns
DATA STABLE
CHNL CLK
375ns
100
POP CYCLE = 750ns
(4 CHNLS = 3µs)
?PUSH
CYCLE REQ
250ns
BUSY
END
PUSH CYCLE
-RIDGE INTERFACE


26LS32
TRIG
CHNL 2
TRIG
ENB
EOC
JUMPER
JUMPER
79
69
374
374
HI
LOW
3 USEC
LSB
MSB
MD2735-1 ANALOG-IG ADC
TRIG IN
B16
B15
B14
B13
B12
B11
B10
B9
B8
B7
B6
B5
B4
B3
B2
B1
+5V
V+
ANA GND
V-
DIG GND
-5V
ANA IN
REF IN
REF OUT
EOC
AH0134
4.7 K
4.7 K
SP6036
+15V
GND
-15V
+5
SEMICONDUCTOR CIRCUITS
7805
AC
AC
+
-
CHNL 2
74
TRIG
A
B
+5V
.67 amps

XEROX

System Products

Analog Instrument Data Sheet

# DA50 DIGITAL - TO - ANALOG CONVERTER

The XDS Model DA50 Digital-to-Analog Converter (DAC) is a compact, high performance unit capable of converting 12-bit or 15-bit digital words into equivalent analog outputs. Resolution is 0.006% of full scale in the 15-bit version. Resolution is 0.049% of full scale in the 12-bit version. Up to sixteen DA50 units can be operated under common digital control to provide up to 256 channels.

The 15-bit DA50 is designed for high precision applications, such as simulation in conjunction with hybrid digital/analog computers, or supplying a computer controlled sequence of high accuracy test voltages during automatic checkout.

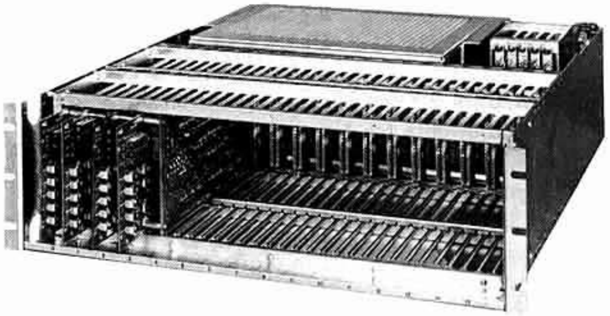
The 12-bit DA50 is suitable for driving galvanometers, operating process control actuators, and similar applications requiring less resolution.

By substituting a variable voltage for the fixed internal reference voltage, the DA50 can also operate as a four-quadrant multiplying DAC, providing an output which is the product of the digital input and the variable voltage. The variable voltage may be the output of another DAC channel.

Each DA50 unit can store from 1 to 16 words of digital input in addressable dual-rank flip-flop registers, converting them to analog output at a Convert strobe. With sixteen DA50 units slaved together, a total of 256 channels can be operated with the same Convert strobe. Conversion may be performed sequentially by updating one channel at a time, or simultaneously on many channels.

The capability of each channel is determined by customer choice of modules. Options are available to accept either 12 bits or 15 bits, and to provide one of these three outputs: ±10 volts, ±100 volts, or ±100 volts multiplying. Option modules may be intermixed within the same chassis in any combination, providing complete flexibility.

Current limiting circuitry is provided at each output to make it short-circuit proof.



Only two digital control inputs are needed: Select Strobe to update digital inputs, and Convert Strobe to change outputs. If single line control is desired, these two functions can be combined through a manual one-time switch setting to operate on the leading and trailing edges of a single strobe. An Inhibit input is also available to provide output data protection in the event a power failure affects the source of digital data. In an XDS Sigma computer controlled system, the DA50 can be controlled by the Model 7910 Analog Output Controller or the Model 7923 Analog/Digital Adapter. In a CF16A computer-controlled system, the DA50 can be controlled by a similar Analog Output Controller (PE25).

Accuracy of conversion is enhanced by a self-contained floating power supply with a ±10 volt reference voltage regulator. The regulator can also be slaved to a customer-supplied voltage.

The DA50 circuitry floats with reference to the digital inputs, and power supply permits the analog outputs to be ground-referenced at the load. This minimizes interference from circulating ground currents.

The DA50 is compatible with all XDS computing equipment, and can be made compatible with any other computing equipment through the use of reliable XDS J or T Series logic modules.

## INHIBIT SIGNAL

Grounding the Inhibit terminal inhibits the Convert signal, preventing the transfer of digital data from the first rank to the second rank of the storage register. This feature makes it possible to maintain the analog outputs in event of power failure affecting the source of digital data. If power continues to be supplied to the DA50 and ground is applied to the inhibit line, the DA50 maintains the analog outputs corresponding to the data content of the second-rank registers. If emergency power is to be supplied to the DA50, it should be provided at the AC power input.

# SYSTEM INTERFACE

## LOCATIONS

DA50 interfaces are found in three locations:

1. On the exposed edges of the control modules at the rear of the chassis, shown in figure 3 and listed in table 5.
2. On the exposed edges of the amplifier auxiliary modules, shown in figure 4 and listed in table 5.
3. On the power terminal boards shown in figure 4 and listed in table 6.

## TYPES OF SIGNALS

The signals at the interfaces with other system units are of four types:

1. Digital inputs, which include the data to be converted, the address of the chassis and the channel, and the two strobe lines, listed in table 5.
2. Analog outputs, representing the converted data, available in two locations, listed in table 3.
3. Customer-supplied optional analog variable for Y inputs, listed in table 6 (used only in the multiplying DAC application).
4. Customer-supplied optional Inhibit and +10 volt external reference signals, listed in table 6.

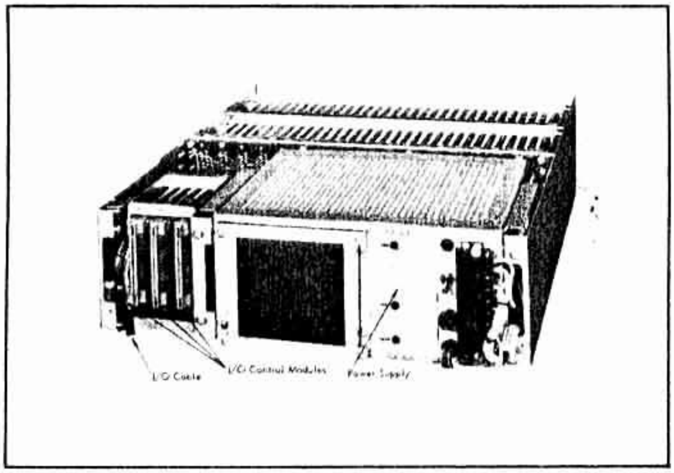


Figure 3. DA50, Rear View

Table 5. Input/Output Connections

| Function | Location* |
|---|---|
| Input Data Bit $2^0$ (sign) | J35-1, A 5 |
| Input Data Bit $2^{-1}$ | J35-2, B 6 |
| Input Data Bit $2^{-2}$ | J35-3, C 8 |
| Input Data Bit $2^{-3}$ | J35-4, D 9 |
| Input Data Bit $2^{-4}$ | J35-5, E 10 |
| Input Data Bit $2^{-5}$ | J35-6, F 12 |
| Input Data Bit $2^{-6}$ | J35-7, G 13 |
| Input Data Bit $2^{-7}$ | J35-8, H 14 |
| Input Data Bit $2^{-8}$ | J35-9, K 16 |
| Input Data Bit $2^{-9}$ | J35-10, L 17 |
| Input Data Bit $2^{-10}$ | J35-11, M 18 |
| Input Data Bit $2^{-11}$ | J35-12, N 20 |
| Input Data Bit $2^{-12}$ | J35-13, P 21 |
| Input Data Bit $2^{-13}$ | J35-14, R 22<br>15 — 24 |
| Chassis Address $2^3$ | J33-1, A 25 |
| Chassis Address $2^2$ | J33-2, B 26 |
| Chassis Address $2^1$ | J33-3, C 28 |
| Chassis Address $2^0$ | J33-4, D 29 |
| Channel Address $2^3$ | J33-5, E 30 |
| Channel Address $2^2$ | J33-6, F 32 |
| Channel Address $2^1$ | J33-7, G 33 |
| Channel Address $2^0$ | J33-8, H 34 |
| Select Strobe | J33-9, K 36 |
| Data Bit $2^{-14}$ | J33-10, L 37 |
| Convert Strobe | J33-11, M 38 |
| Channel 0 Output | J37-1 or E25 12 40 |
| Channel 1 Output | J37-2 or E23 13 41 |
| Channel 2 Output | J37-3 or E22 14 42 |
| Channel 3 Output | J37-6 or E17 15 4[illegible] |
| Channel 4 Output | J37-9 or E12 |
| Channel 5 Output | J37-12 or E6 |
| Channel 6 Output | J37-13 or E4 |
| Channel 7 Output | J37-14 or E2 |
| Channel 8 Output | J37-A or E27 |
| Channel 9 Output | J37-B or E26 |
| Channel 10 Output | J37-C or E24 |
| Channel 11 Output | J37-F or E28 |
| Channel 12 Output | J37-K or E11 |
| Channel 13 Output | J37-N or E5 |
| Channel 14 Output | J37-P or E3 |
| Channel 15 Output | J37-R or E1 |
| Shield Common | Shield Common |

INTERFACE BOARD

Note: Channels 0 to 7 terminate on left side; channels 8 to 15 terminate on right side of module in location J37.

*Connections are made to front-edge connectors on both sides of the cable plug module or to etched solder points on the module.