Block I
Apollo Guidance Computer (AGC)

How to build one in your basement

Part 3: Processing (PROC) Module

John Pultorak
December, 2004

Abstract

This report describes my successful project to build a working reproduction of the 1964 prototype for the Block I Apollo Guidance Computer. The AGC is the flight computer for the Apollo moon landings, and is the world's first integrated circuit computer.

I built it in my basement. It took me 4 years.

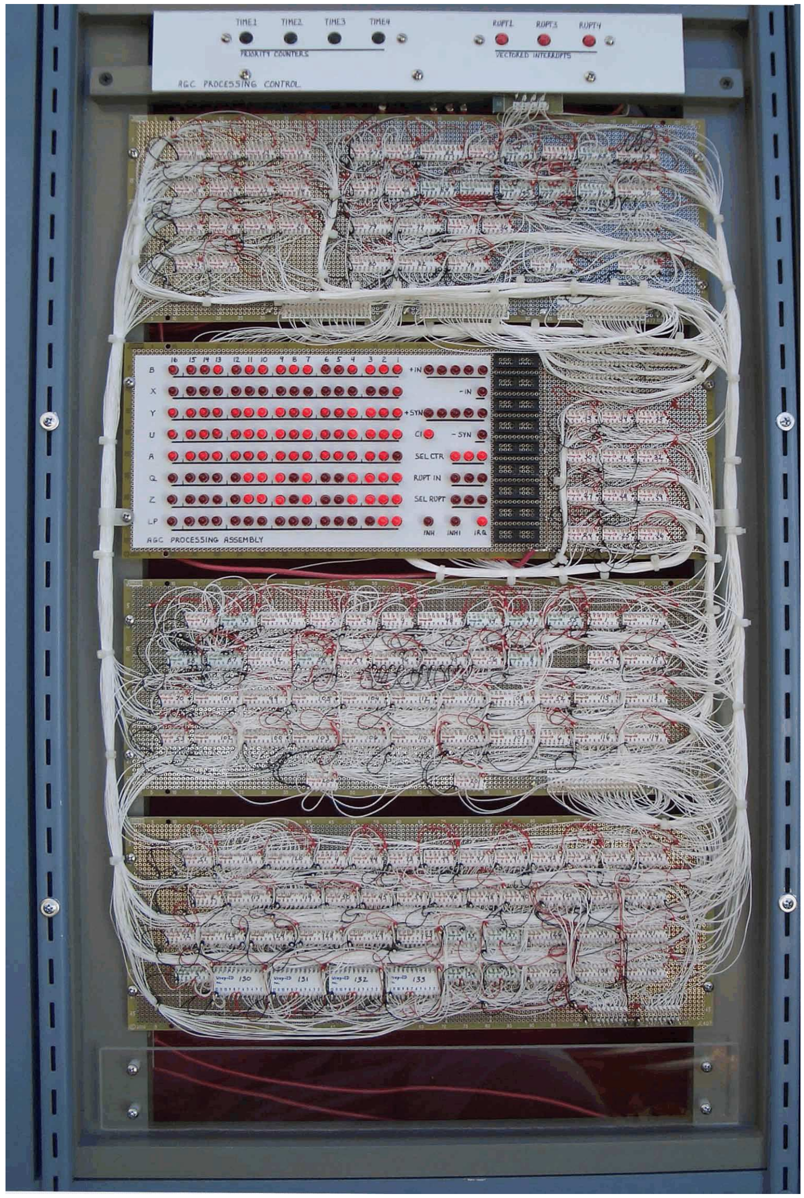
If you like, you can build one too. It will take you less time, and yours will be better than mine.

I documented my project in 9 separate .pdf files:

Part 1	Overview: Introduces the project.
Part 2	CTL Module: Design and construction of the control module.
Part 3	PROC Module: Design and construction of the processing (CPU) module.
Part 4	MEM Module: Design and construction of the memory module.
Part 5	IO Module: Design and construction of the display/keyboard (DSKY) module.
Part 6	Assembler: A cross-assembler for AGC software development.
Part 7	C++ Simulator: A low-level simulator that runs assembled AGC code.
Part 8	Flight Software: My translation of portions of the COLOSSUS 249 flight software.
Part 9	Test & Checkout: A suite of test programs in AGC assembly language.

Overview

The Processing Module (PROC) has 5 subsystems: PMI, ALU, CRG, INT, CTR



PMI (Processing Module external Interface)
The PMI interfaces other processing module subsystems to external AGC modules. 40-pin IDE connectors interface to the other CTL, MEM, and IO modules. Inputs from those modules are buffered to 1 LSTTL load.

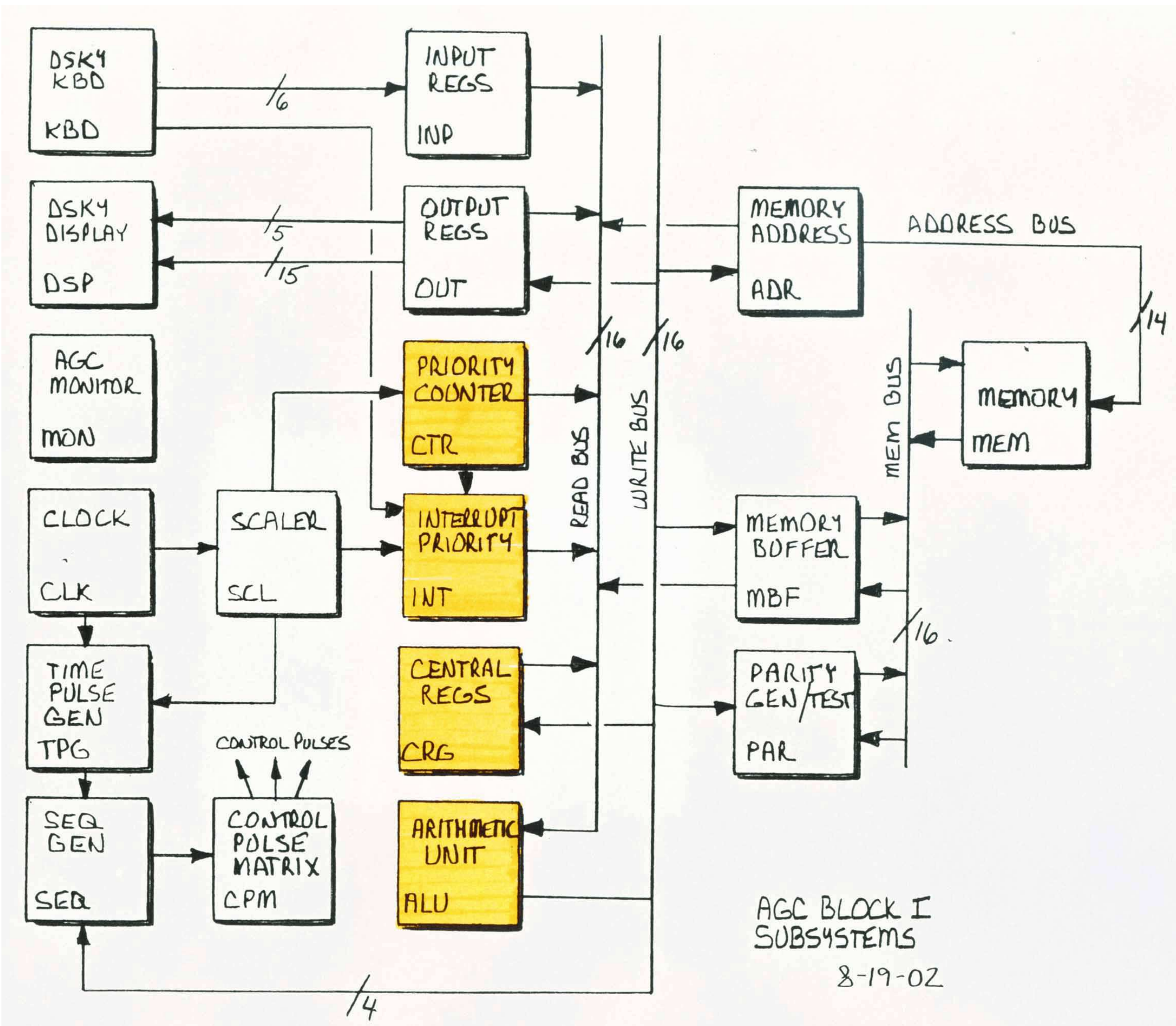
ALU (Arithmetic Logic Unit)
The ALU contains the 16-bit ADDER to perform 1's complement arithmetic and increment the program counter (Z register). The ALU also contains the B and C registers, and logic to inclusive OR the contents of the them with the ADDER. The inclusive OR occurs when the control module (CTL) issues signals to READ the contents of both registers onto the READ bus simultaneously. The ALU transfers data from the READ bus to the WRITE bus for register-to-register transfers and can force the data lines of the WRITE bus to specific states to gate various constants into the AGC registers

CRG (Central Register)
The AGC has four 16-bit “central registers” for general computational use. These are the accumulator (A) for general computation; the program counter (Z) which contains the address of the next instruction; the Q register holding the remainder in the DV instruction, and the return address after TC instructions; and the LP register used to hold the lower product after MP instructions.

INT (Interrupt Priority)
The original AGC had five vectored interrupts. This recreation implements three of them: RUPT1, also called T3RUPT which is used as general-purpose timer by the AGC WAITLIST software; RUPT3, also called T4RUPT or DSRUPT, which is used to update the DSKY display at regular intervals; and RUPT4, also called KERUPT, which is triggered by a key press from the user's keyboard. The AGC responds to each interrupt by temporarily suspending the current program, executing a short interrupt service routine, and then resuming the

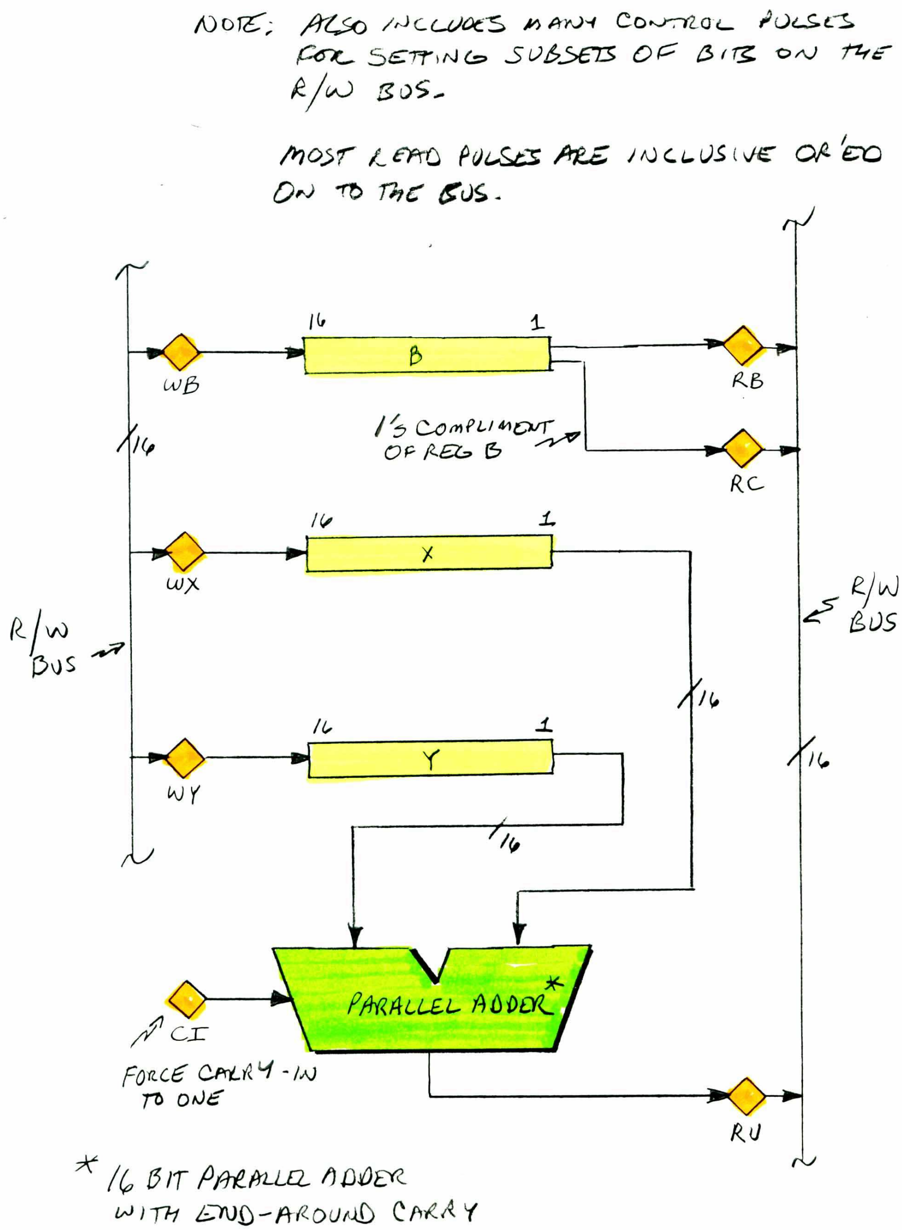
interrupted program.

CTR (Priority Counter)
Twenty memory locations in the original AGC functioned as up/down counters. The counters would increment (PINC) or decrement (MINC) in response to external inputs. Increment or decrement was handled by one 12-step subsequence of microinstructions inserted between any two regular instructions. This replica implements 5 of the counters: OVCTR, an overflow counter incremented or decremented by arithmetic overflow during certain instructions; TIME2 and TIME1, the AGC real-time clock; TIME3, a general purpose timer incremented by a 100Hz signal from the SCALER (SCL); and TIME4, a timer used to update the DSKY display.



The ALU has a 16-bit parallel adder. The addend and augend are supplied by the X and Y registers. The sum, called the U register although it is not really a register at all, is gated to the read bus through the RU (read U) control signal. The WX and WY control signals copy the contents of the write bus into the X and Y registers.

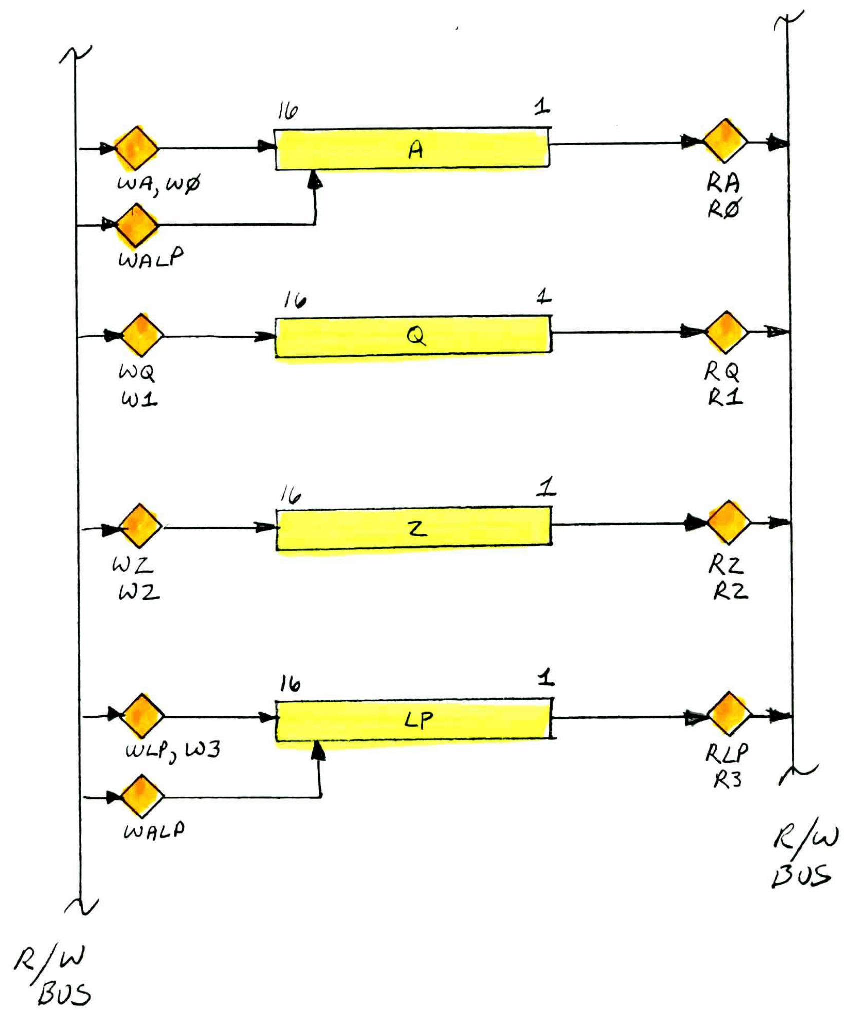
The B register is loaded from the write bus with the WB (write B) signal. The contents of B are output to the read bus with the RB (read B) signal. The inverted output of B can be gated to the read bus with the RC (read C) signal.



The AGC has four user-accessible central registers. The A register is the accumulator; Z is the program counter; Q stores the return address for jumps (TC instruction), and LP stores the lower product (MP instruction).

Central register contents can be output to the read bus by asserting the appropriate read control pulse (RA, RQ, RZ, or RLP). Each register is also mapped to a memory location, with register A mapped to address 0, Q to address 1, Z to 2, and LP to 3. The R0, R1, R2, and R3 control pulses output those registers to the read bus.

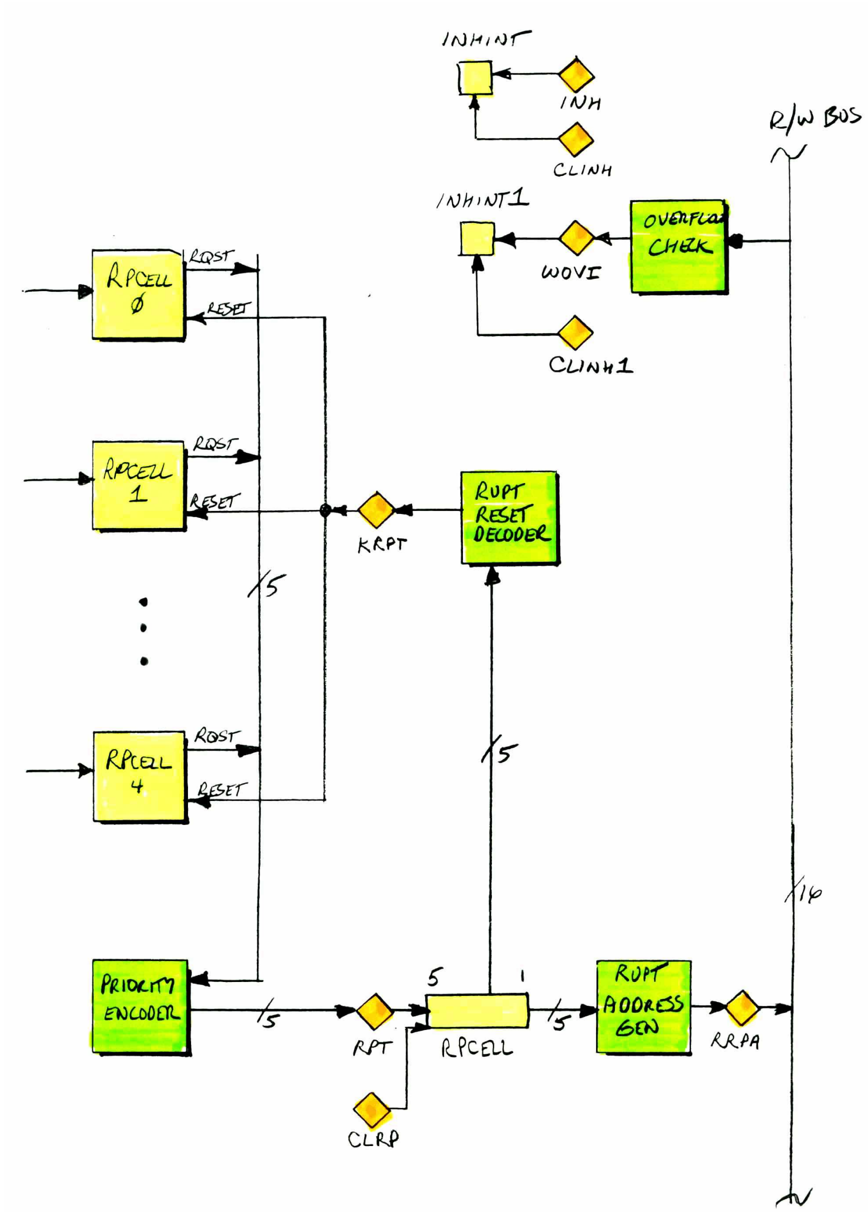
The write bus contents can be loaded into a central register with a write control pulse. WA and WALP load the A register; WQ, the Q register; WZ the Z register; and WLP and WALP the LP register. The W0, W1, W2, and W3 control pulses mapped to memory addresses 0,1,2, and 3 also load those registers.



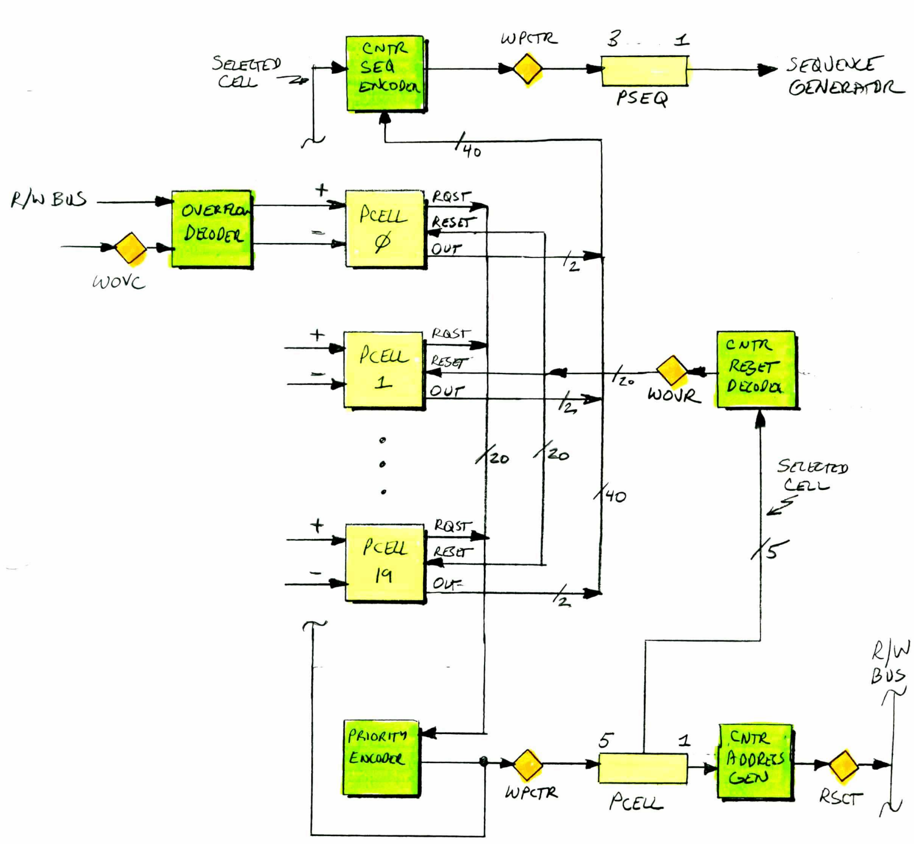
The interrupt priority subsystem manages vectored interrupts. Five interrupts (0-4) are implemented. Each interrupt is latched by its own “RP cell” flip-flop. Signals from all RP cells feed into a priority encoder; a combinational logic array that outputs the code of the highest priority interrupt in the RP cells. When RPT is asserted, the priority code is latched into RPCELL. This is decoded into an address which is the interrupt vector; the address is written to the read bus when RRPA is asserted.

After the interrupt code has been loaded into RPCELL, asserting KRPT (knock-down RPT) causes the RP cell for that interrupt to be reset. This causes the next highest priority interrupt to be decoded by the priority encoder.

The INHINT and INHINT1 flip-flops inhibit interrupts.



The priority counter logic design is similar to the interrupt subsystem. Up (+) and down (-) count input signals feed into 20 PCELLs, one PCELL for each counter. The PCELLs feed into the priority encoder which outputs the code of the highest priority PCELL having a up or down input set. The PCELL code is written to the PCELL register when WPCTR is asserted.



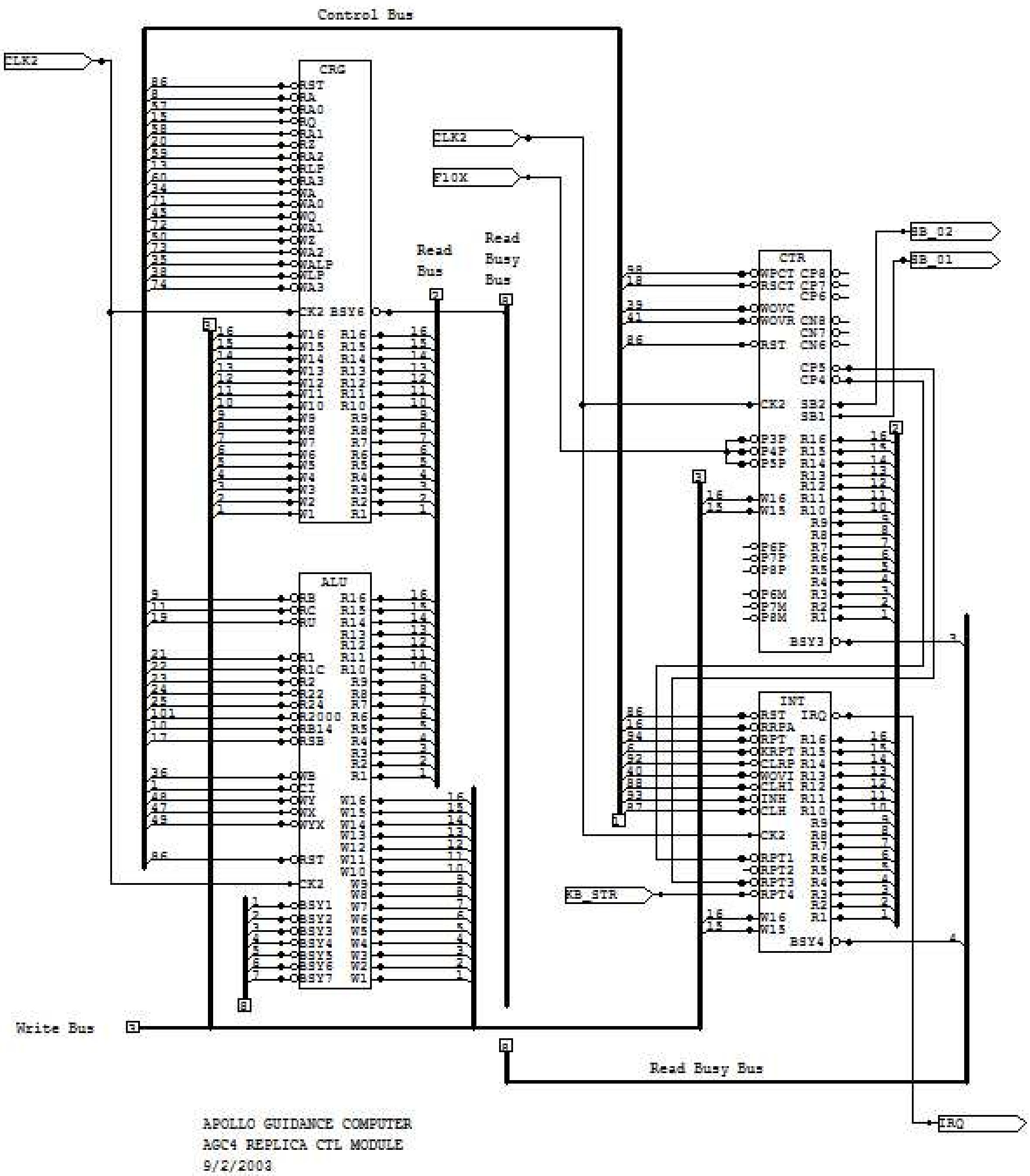
The PCELL memory address, derived from PCELL, is written to read bus when RSCT is asserted. After the code is latched into the PCELL register, the corresponding PCELL is reset by asserting WOVR.

The up or down code for the selected PCELL is written to PSEQ when WPCTR is asserted. This code feeds to the control logic on the CTL module which selects the PINC (increment) or MINC (decrement) instruction subsequence to bump the priority counter up or down.

The PINC and MINC subsequences are inserted between normal instruction subsequences. A SHINC subsequence implements a bit-shift which is used to load telemetry bits into the AGC and assemble them into words. SHINC is not implemented in this AGC replica.

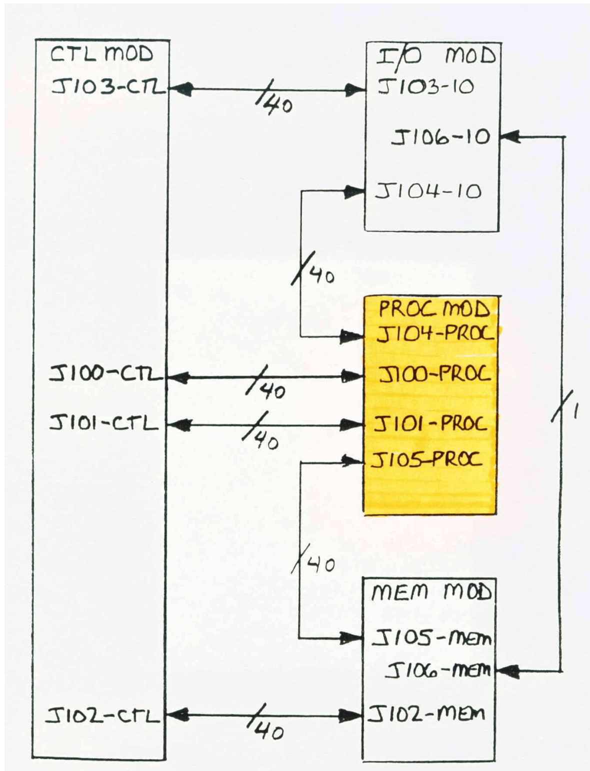
PROC Internal Subsystem Interconnections

This diagram shows internal interconnections for the subsystems in the PROC module.



PROC Module External Interfaces

The PROC module interfaces to the CTL, MEM, and IO modules through 40-pin IDE ribbon cables.

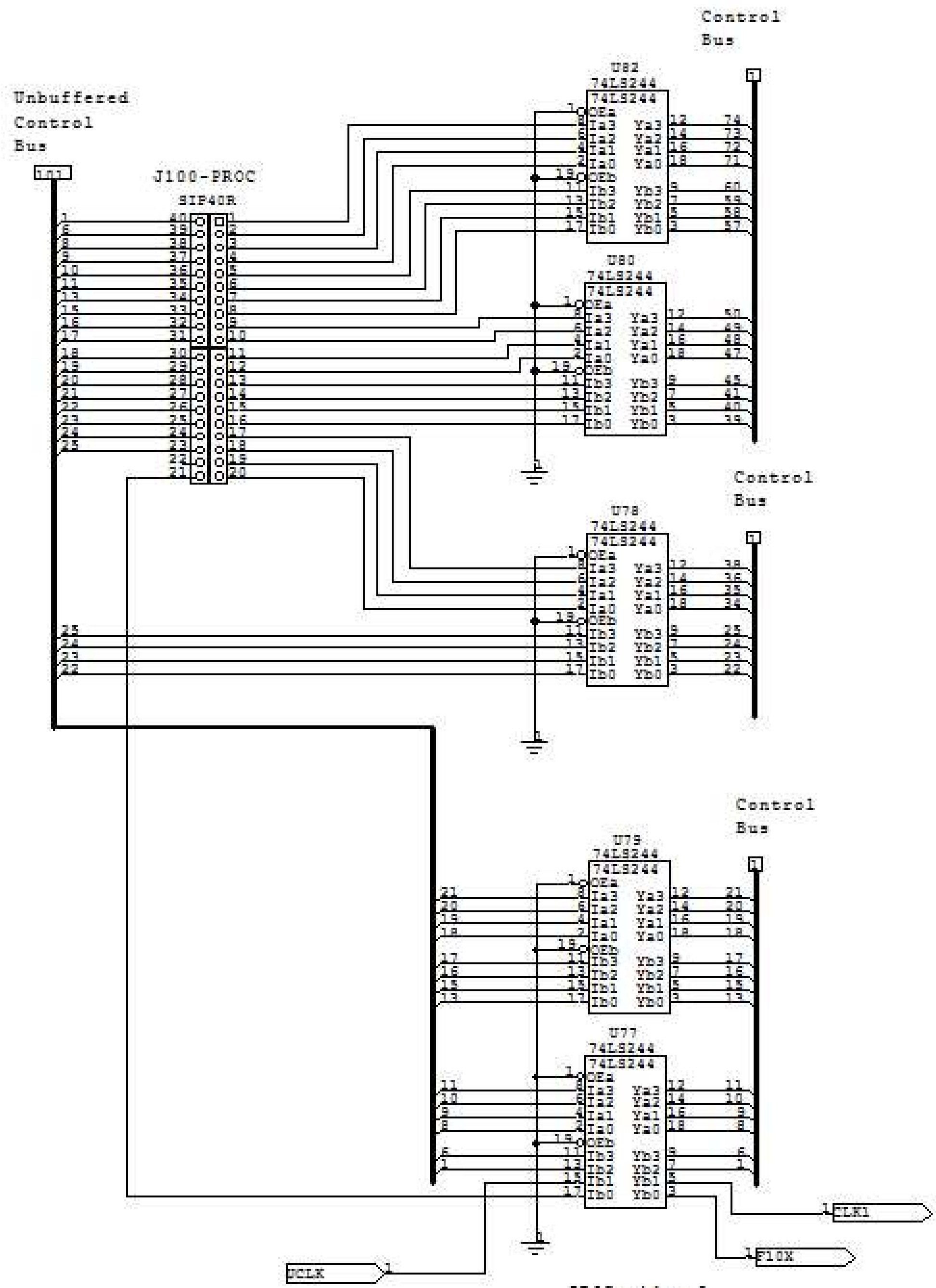


AGC MODULE INTERCONNECTIONS

J100-PROC: PROC-to-CTL I/F
J100 is a 40-pin IDE cable that connects the PROC module to the CTL module.

INPUTS (to PROC):

PIN	signal	full name	state definition
1	WA3	WRITE ADDR 3 (74)	0=Write reg at address 3 (LP)
2	WA2	WRITE ADDR 2 (73)	0=Write reg at address 2 (Z)
3	WA1	WRITE ADDR 1 (72)	0=Write reg at address 1 (Q)
4	WA0	WRITE ADDR 0 (71)	0=Write reg at address 0 (A)
5	RA3	READ ADDR 3 (60)	0=Read reg at address 3 (LP)
6	RA2	READ ADDR 2 (59)	0=Read reg at address 2 (Z)
7	RA1	READ ADDR 1 (58)	0=Read reg at address 1 (Q)
8	RA0	READ ADDR 0 (57)	0=Read reg at address 0 (A)
9	WZ	WRITE Z (50)	0=Write Z
10	WYx	WRITE Y NO RESET (49)	0=Write Y (do not reset)
11	WY	WRITE Y (48)	0=Write Y
12	WX	WRITE X (47)	0=Write X
13	WQ	WRITE Q (45)	0=Write Q
14	WOVR	WRITE OVF (41)	0=Write overflow
15	WOVI	WRITE OVF RUPT INH (40)	0=Write overflow RUPT inhibit
16	WOVC	WRITE OVF CNTR (39)	0=Write overflow counter
17	WLP	WRITE LP (38)	0=Write LP
18	WB	WRITE B (36)	0=Write B
19	WALP	WRITE A/LP (35)	0=Write A and LP
20	WA	WRITE A (34)	0=Write A
21	F10X	F10 SCALER ONESHOT	1=timed out (100.0 Hz)
23	R24	READ 24 (25)	0=Read 24
24	R22	READ 22 (24)	0=Read 22
25	R2	READ 2 (23)	0=Read 2
26	R1C	READ 1 COMP (22)	0=Read 1 complimented
27	R1	READ 1 (21)	0=Read 1
28	RZ	READ Z (20)	0=Read Z
29	RU	READ U (19)	0=Read sum
30	RSCT	READ CNTR ADDR (18)	0=Read selected counter address
31	RSB	READ SIGN (17)	0=Read sign bit
32	RRPA	READ RUPT ADDR (16)	0=Read RUPT address
33	RQ	READ Q (15)	0=Read Q
34	RLP	READ LP (13)	0=Read LP
35	RC	READ C (11)	0=Read C
36	RB14	READ BIT 14 (10)	0=Read bit 14
37	RB	READ B (9)	0=Read B
38	RA	READ A (8)	0=Read A
39	KRPT	KNOCK DOWN RUPT (6)	0=Knock down Rupt priority
40	CI	SET CARRY IN (1)	0=Carry in



PROC side of
CTL/PROC I/F

J101-PROC: PROC-to-CTL I/F

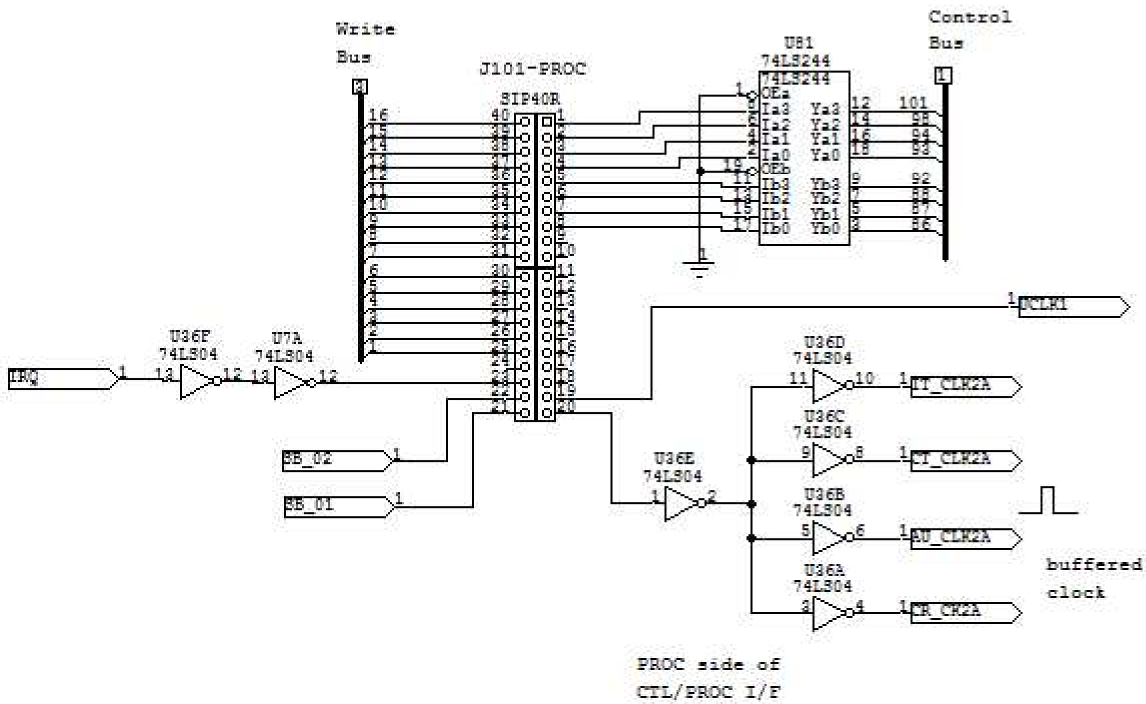
J101 is a 40-pin IDE cable that connects the PROC module to the CTL module.

INPUTS (to PROC):

PIN	signal	full name	state definition
1	R2000	READ 2000 (101)	0=Read 2000
2	WPCTR	WRITE PCTR (98)	0=Write PCTR (latch priority counter seq)
3	RPT	READ RUPT (94)	0=Read RUPT opcode
4	INH	SET INHINT (93)	0=Set INHINT
5	CLRP	CLEAR RPCELL (92)	0=Clear RPCELL
6	CLINH1	CLEAR INHINT1 (88)	0=Clear INHINT1
7	CLINH	CLEAR INHINT (87)	0=Clear INHINT
8	GENRST	GENERAL RESET (86)	0=General Reset
19	CLK1	CLOCK1	1.024 MHz AGC clock 1 (normally low)
20	CLK2	CLOCK2	1.024 MHz AGC clock 2 (normally low)

OUTPUTS (from PROC):

PIN	signal	full name	state definition
21	SB_01	SUB SEL 01	SB_01 is LSB; SB_02 is MSB
22	SB_02	SUB SEL 02	00=no counter; 01=PINC; 10=MINC
23	IRQ	INT RQST	0=interrupt requested.
25	WB_01	WRITE BUS 01	(lsb)
26	WB_02	WRITE BUS 02	
27	WB_03	WRITE BUS 03	
28	WB_04	WRITE BUS 04	
29	WB_05	WRITE BUS 05	
30	WB_06	WRITE BUS 06	
31	WB_07	WRITE BUS 07	
32	WB_08	WRITE BUS 08	
33	WB_09	WRITE BUS 09	
34	WB_10	WRITE BUS 10	
35	WB_11	WRITE BUS 11	
36	WB_12	WRITE BUS 12	
37	WB_13	WRITE BUS 13	
38	WB_14	WRITE BUS 14	
39	WB_15	WRITE BUS 15	US (overflow) bit
40	WB_16	WRITE BUS 16	SG (sign) bit


Write Bus
J101-PROC
SIP40R
U81
74LS244
Control Bus
PCLK1
U36F
74LS04
U7A
74LS04
IRQ
U36D
74LS04
IT_CLK2A
U36C
74LS04
CT_CLK2A
SB_02
SB_01
U36E
74LS04
U36B
74LS04
AU_CLK2A
U36A
74LS04
CR_CK2A
buffered clock
PROC side of CTL/PROC I/F

J104-PROC: PROC-to-IO I/F
J104 is a 40-pin IDE cable that connects the PROC module to the IO module.

INPUTS (to PROC):

PIN	signal	full name	state definition
40	RB_01	READ BUS 01	(lsb)
39	RB_02	READ BUS 02	
38	RB_03	READ BUS 03	
37	RB_04	READ BUS 04	
36	RB_05	READ BUS 05	
35	RB_06	READ BUS 06	
34	RB_07	READ BUS 07	
33	RB_08	READ BUS 08	
32	RB_09	READ BUS 09	
31	RB_10	READ BUS 10	
30	RB_11	READ BUS 11	
29	RB_12	READ BUS 12	
28	RB_13	READ BUS 13	
27	RB_14	READ BUS 14	
26	RB_15	READ BUS 15	US (overflow) bit
25	RB_16	READ BUS 16	SG (sign) bit
22	BUSY2	READ BUS BUSY	0=OUT register output to read bus
21	BUSY1	READ BUS BUSY	0=INP register output to read bus
20	KB_STR	KEY STROBE	1=key pressed strobe; to KEYRUPT. Key data is valid on the negative edge of KB_STR. Data is latched until the next keypress.

OUTPUTS (from PROC):

PIN	signal	full name	state definition
1	WB_01	WRITE BUS 01	(lsb)
2	WB_02	WRITE BUS 02	
3	WB_03	WRITE BUS 03	
4	WB_04	WRITE BUS 04	
5	WB_05	WRITE BUS 05	
6	WB_06	WRITE BUS 06	
7	WB_07	WRITE BUS 07	
8	WB_08	WRITE BUS 08	
9	WB_09	WRITE BUS 09	
10	WB_10	WRITE BUS 10	
11	WB_11	WRITE BUS 11	
12	WB_12	WRITE BUS 12	
13	WB_13	WRITE BUS 13	
14	WB_14	WRITE BUS 14	
15	WB_15	WRITE BUS 15	US (overflow) bit
16	WB_16	WRITE BUS 16	SG (sign) bit

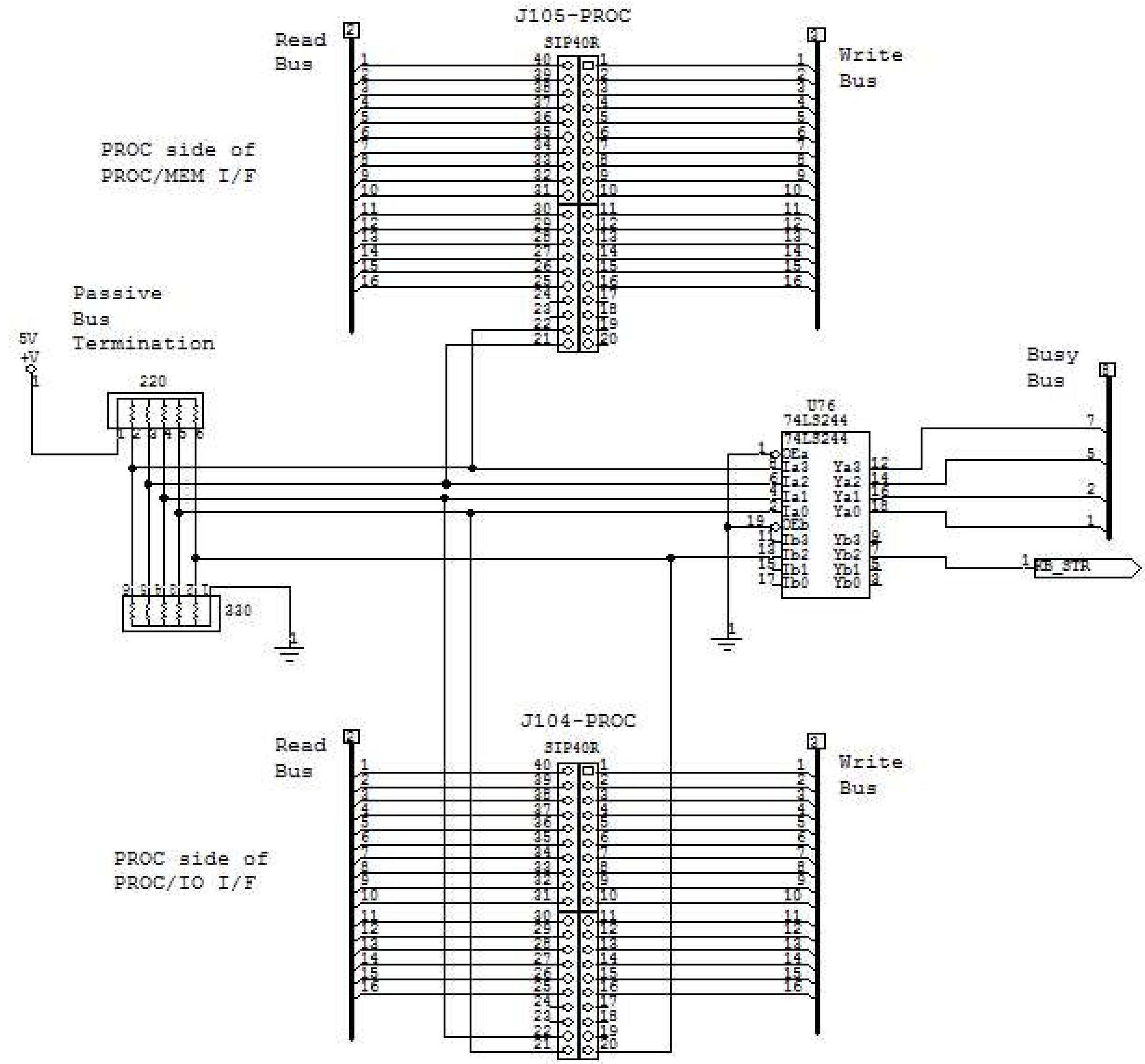
J105-PROC: PROC-to-MEM I/F
J105 is a 40-pin IDE cable that connects the PROC module to the MEM module.

INPUTS (to PROC):

PIN	signal	full name	state definition
40	RB_01	READ BUS 01	(lsb)
39	RB_02	READ BUS 02	
38	RB_03	READ BUS 03	
37	RB_04	READ BUS 04	
36	RB_05	READ BUS 05	
35	RB_06	READ BUS 06	
34	RB_07	READ BUS 07	
33	RB_08	READ BUS 08	
32	RB_09	READ BUS 09	
31	RB_10	READ BUS 10	
30	RB_11	READ BUS 11	
29	RB_12	READ BUS 12	
28	RB_13	READ BUS 13	
27	RB_14	READ BUS 14	
26	RB_15	READ BUS 15	US (overflow) bit
25	RB_16	READ BUS 16	SG (sign) bit
22	BUSY7	READ BUS BUSY	0=BNK register output enabled to read bus
21	BUSY5	READ BUS BUSY	0=G register output enabled to read bus

OUTPUTS (from PROC):

PIN	signal	full name	state definition
1	WB_01	WRITE BUS 01	(lsb)
2	WB_02	WRITE BUS 02	
3	WB_03	WRITE BUS 03	
4	WB_04	WRITE BUS 04	
5	WB_05	WRITE BUS 05	
6	WB_06	WRITE BUS 06	
7	WB_07	WRITE BUS 07	
8	WB_08	WRITE BUS 08	
9	WB_09	WRITE BUS 09	
10	WB_10	WRITE BUS 10	
11	WB_11	WRITE BUS 11	
12	WB_12	WRITE BUS 12	
13	WB_13	WRITE BUS 13	
14	WB_14	WRITE BUS 14	
15	WB_15	WRITE BUS 15	US (overflow) bit
16	WB_16	WRITE BUS 16	SG (sign) bit


J105-PROC
SIP40R
Read
Bus
Write
Bus
PROC side of
PROC/MEM I/F
Passive
Bus
Termination
5V
+V
220
330
Busy
Bus
U76
74LS244
74LS244
OEa
Ia3
Ia2
Ia1
Ia0
OEb
Ib3
Ib2
Ib1
Ib0
Ya3
Ya2
Ya1
Ya0
Yb3
Yb2
Yb1
Yb0
KB_STR
J104-PROC
SIP40R
Read
Bus
Write
Bus
PROC side of
PROC/IO I/F

PROC CONTROL PANEL PUSHBUTTONS

RUPT1	Set the RUPT1 flip-flop (FF). Simulates a TIME3 overflow. Triggers a T3RUPT.
RUPT3	Set the RUPT3 flip-flop (FF). Simulates a TIME4 overflow. Triggers a T4RUPT (DSRUPT).
RUPT4	Set the RUPT4 flip-flop (FF). Simulates a DSKY keypress. Triggers a KEYRUPT.
TIME1	Set the TIME1 flip-flop (FF). Increments the low-order word of the AGC real-time clock.
TIME2	Set the TIME2 flip-flop (FF). Increments the high-order word of the AGC real-time-clock.
TIME3	Set the TIME3 flip-flop (FF). Increments the general purpose timer.
TIME4	Set the TIME4 flip-flop (FF). Increments the display update timer.

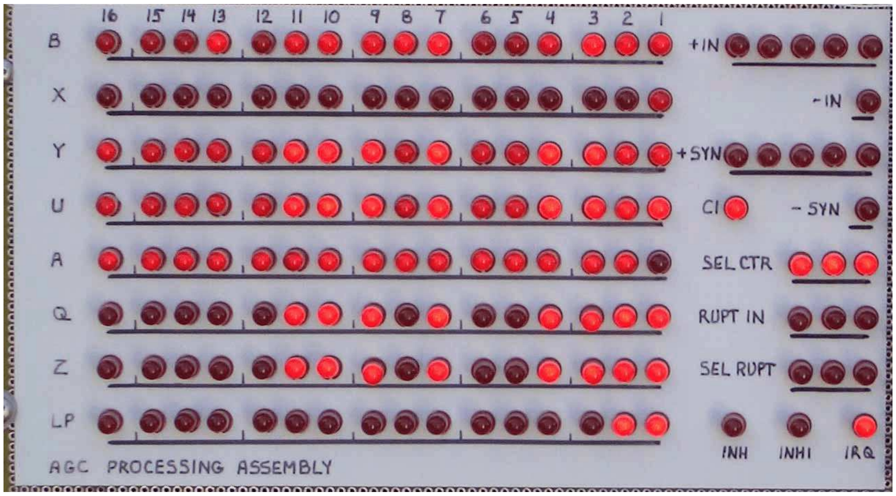
PROC CONTROL PANEL CONNECTIONS

PIN	signal	state definition
1	RUPT1	GND=set RUPT1 FF
2	RUPT3	GND=set RUPT3 FF
3	RUPT4	GND=set RUPT4 FF
4	TIME1	GND=set TIME1 FF
5	TIME2	GND=set TIME2 FF
6	TIME3	GND=set TIME3 FF
7	TIME4	GND=set TIME4 FF
8	GND	

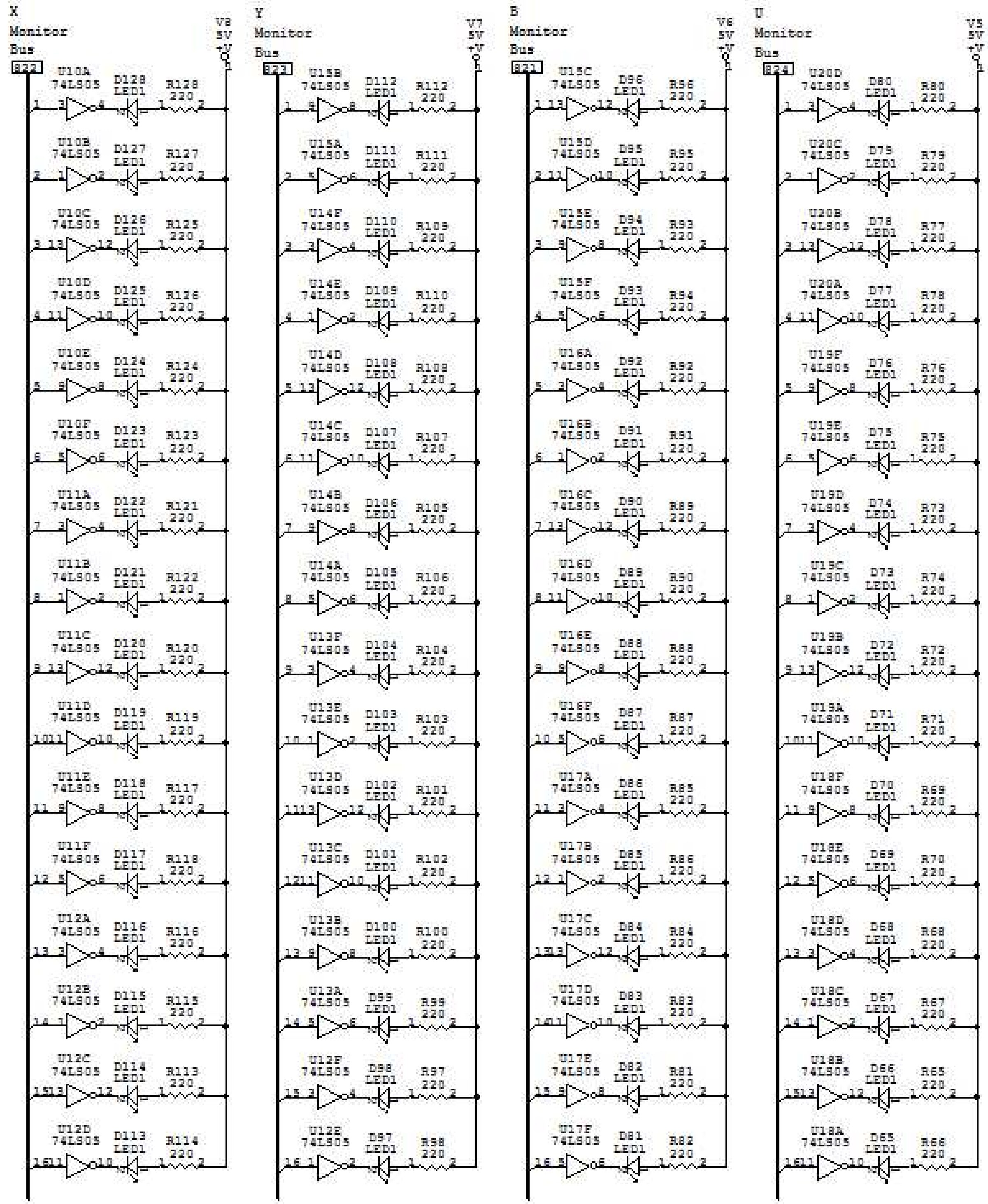
PROC INDICATORS

The PROC module has a panel of indicator lamps (LEDs) to show the state of PROC registers and critical logic signals.

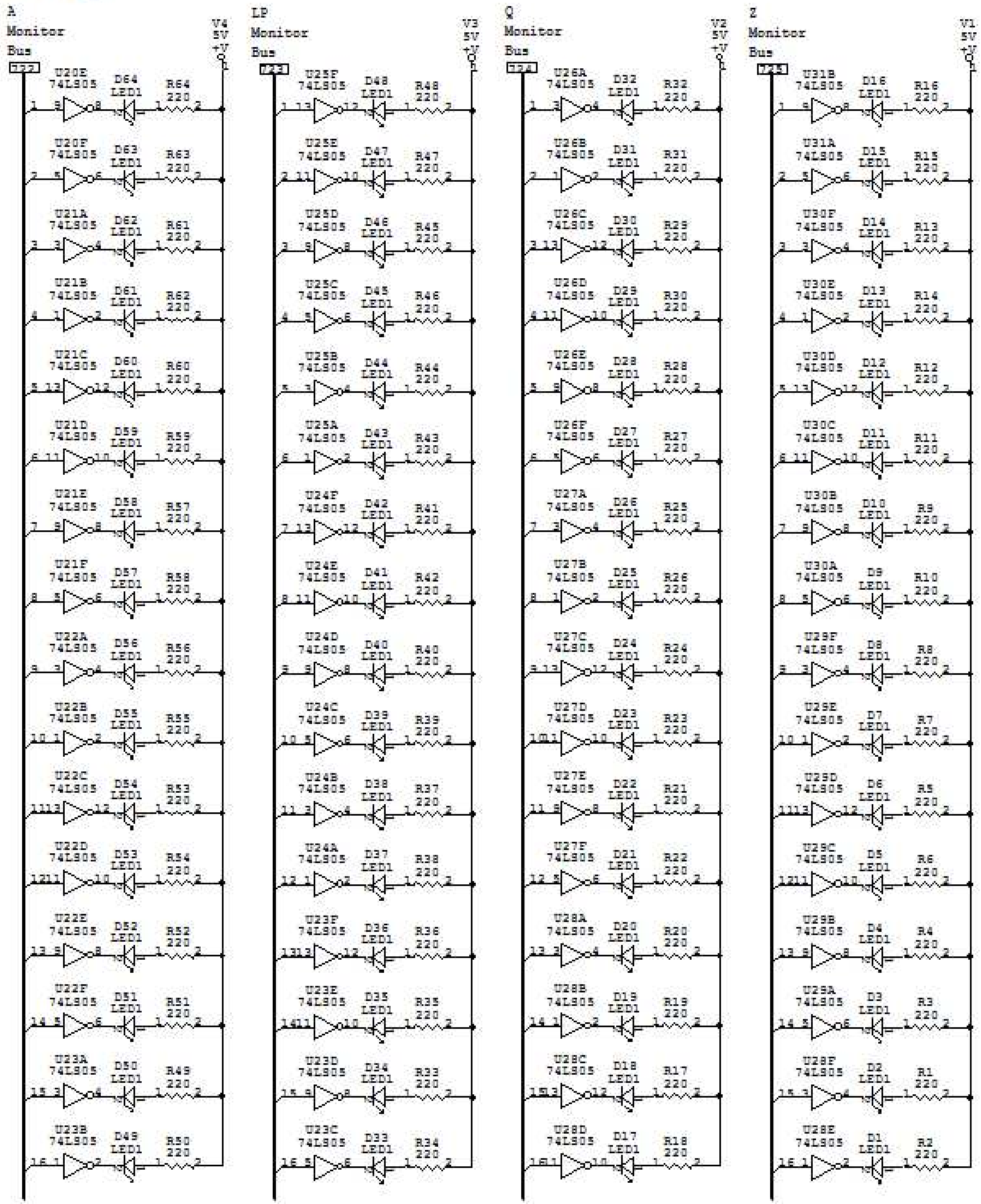
These indicator lamps show the current state of all registers and some additional, important logic signals produced by the PROC module. AGC numbers are represented in octal, so all register lamps are in groups of three. At the time the photo was taken the AGC was running the COLOSSUS 249 flight software load, executing Verb 16, Noun 36: a monitor verb which displays the AGC real time clock.

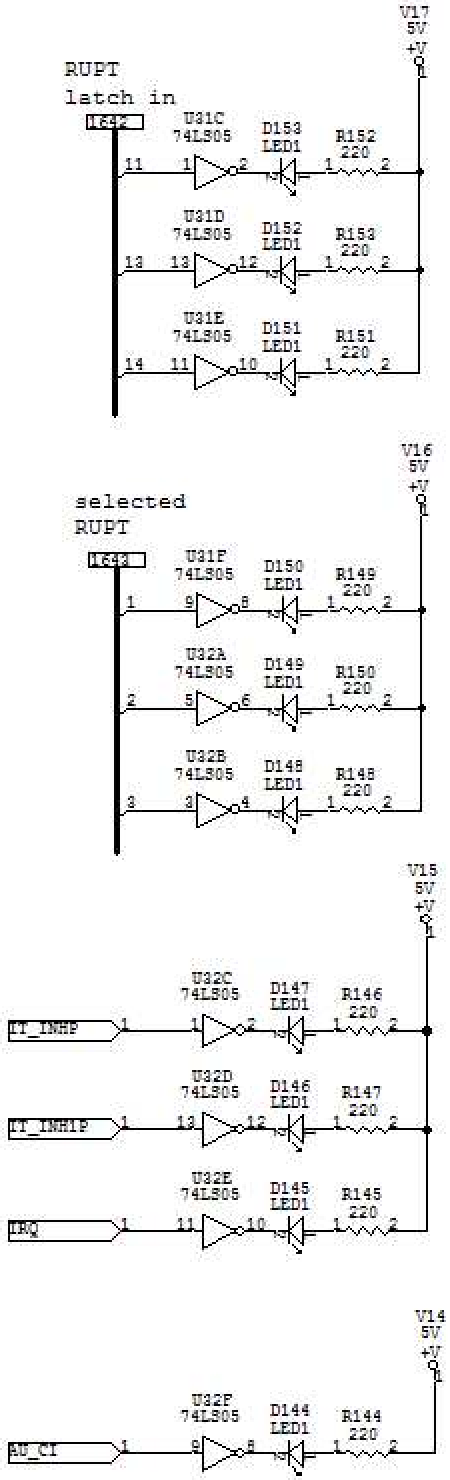


PROC ASSEMBLY
INDICATORS #1

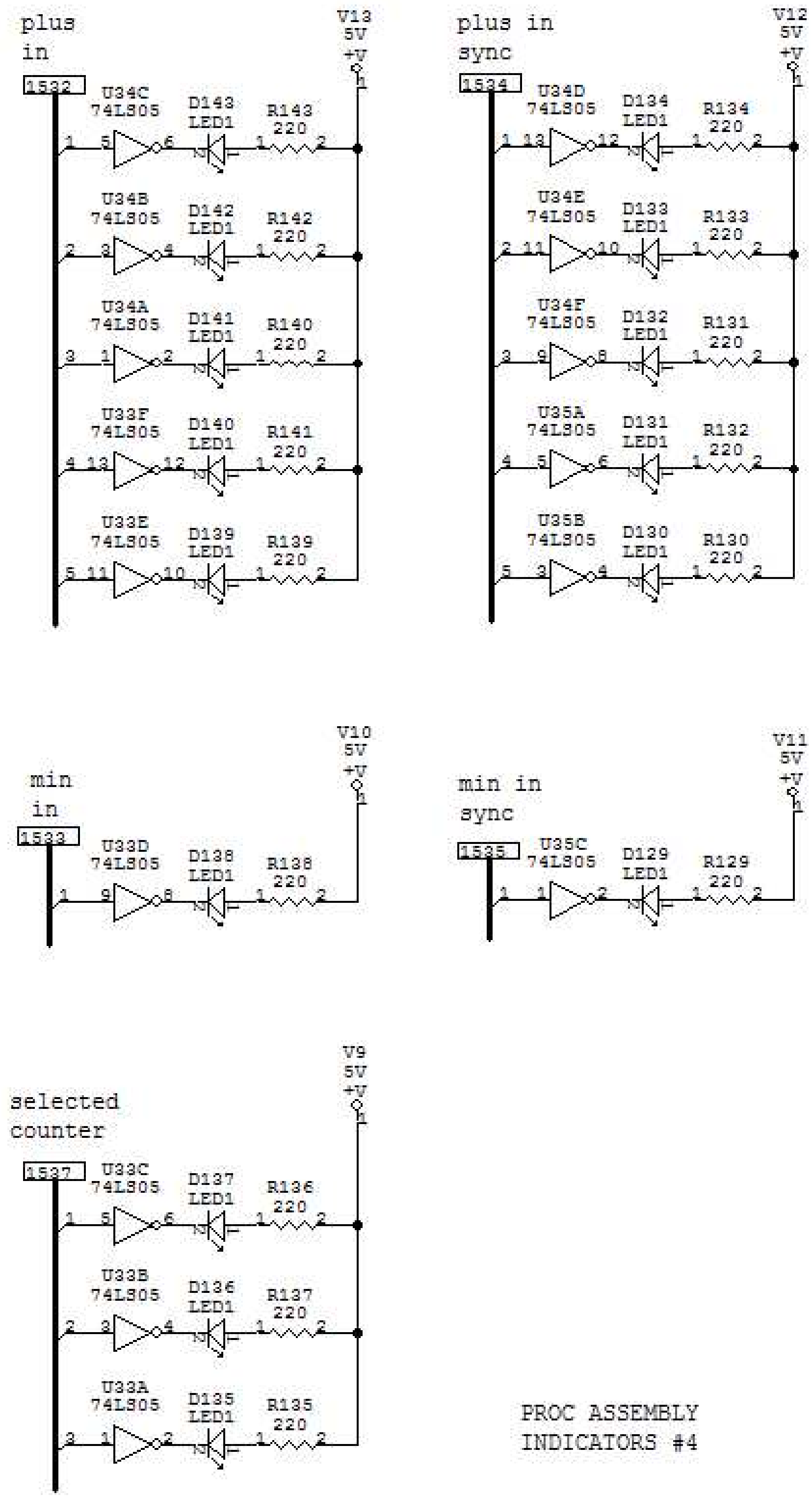


PROC ASSEMBLY
INDICATORS #2





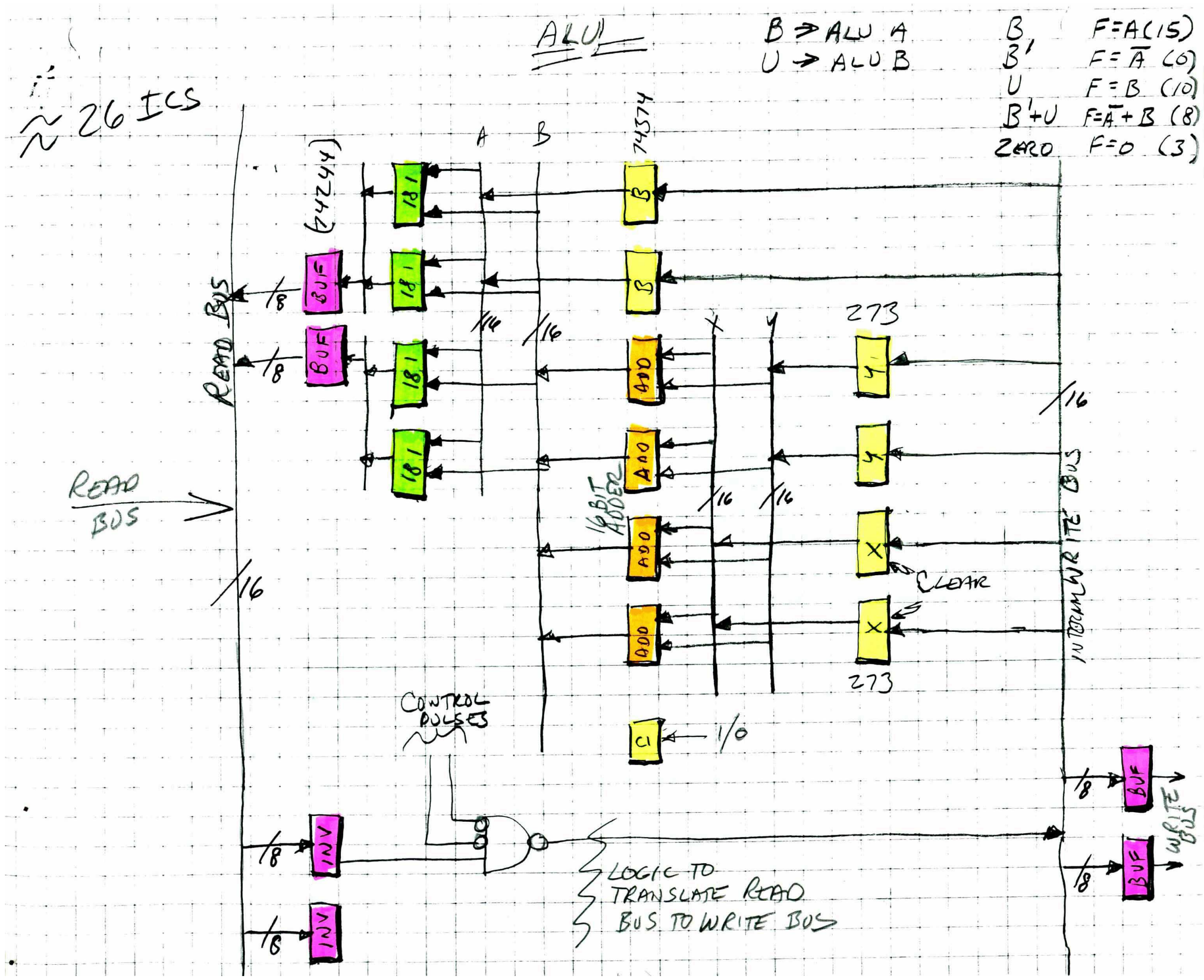
PROC ASSEMBLY
INDICATORS #3



PROC ASSEMBLY
INDICATORS #4

ALU (Arithmetic Logic Unit)

My earliest architectural representation of the ALU logic is shown below:



The ALU contains the 16-bit ADDER (colored orange in the diagram) which performs 1's complement arithmetic, and increments the program counter (Z register). Each orange box is a 4-bit parallel adder; collectively, they add 16 bits. The ADDER uses the X, Y, and U registers:

X: the 16-bit extension register (2 8-bit registers in yellow) that holds one of two inputs to the ADDER.

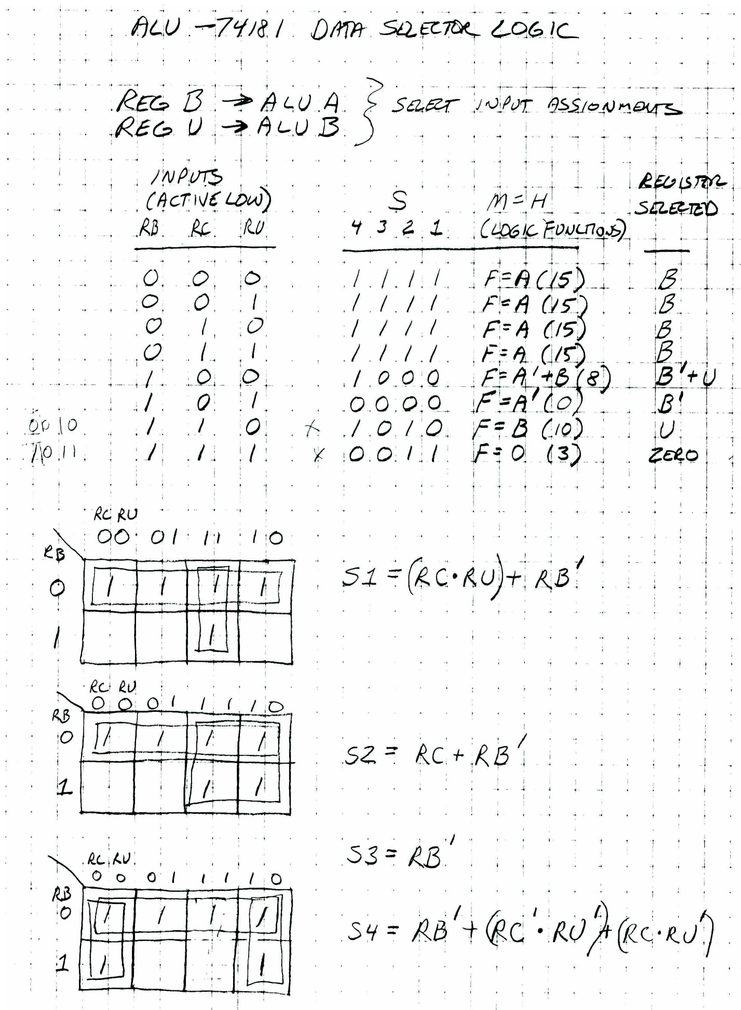
Y: the 16-bit extension register (also in yellow) that holds the other input to the ADDER.

U: the ADDER output (the 1's complement sum of the contents of registers X and Y). Outputs to the bus labeled “B” on the diagram.

The ALU also contains the B and C registers:

B: a general-purpose buffer register (shown as 2 8-bit registers in yellow), also used to pre-fetch the next instruction. At the start of the next instruction sequence, the upper bits of B (containing the next op code) are copied to the SQ register (in CTL), and the lower bits (the address) are copied to the S register in (MEM). Output to the bus labeled “A” on the diagram.

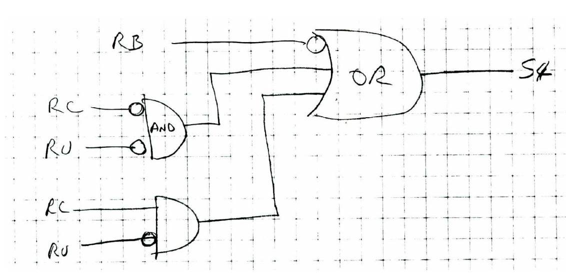
C: not a separate register, but the 1's complement of B.

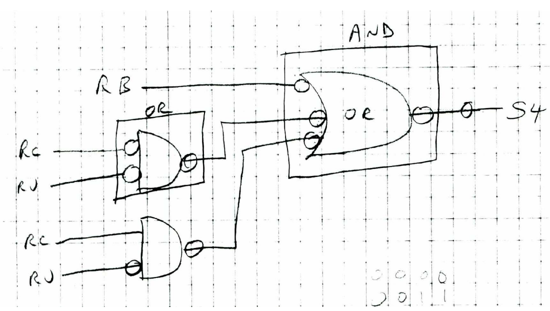


The ALU contains logic (using 74LS181 ALU chips, shown in green) to do any of the following: select the B register; select the complement of the B register (the “C” register); select the U register; select the C register OR’ed with U, or select logical zero. Those logic functions, needed for AGC operation, are shown in the upper right corner of the diagram. The outputs of the 74LS181 selector are gated through a buffer to the read bus.

The original AGC could inclusive OR the outputs of any combination of registers onto the bus, but the control module only used this feature for the B/C and U registers. One of the uses involved the MASK instruction, which is a logical AND: DeMorgan's theorem was used to implement the equivalent of a logical AND by inverting operands through the C register, performing a logical OR through the bus, and then inverting the result.

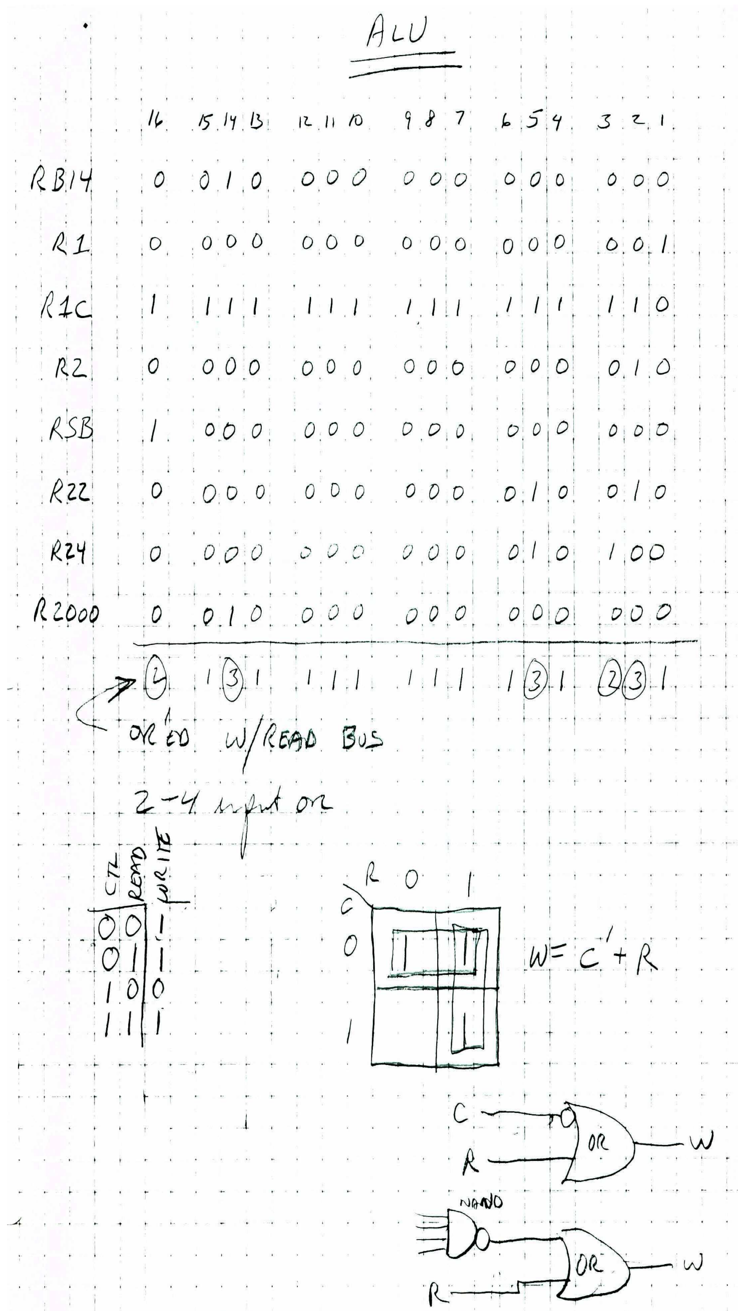
The 74LS181 logic functions are gated to the read bus through the data selector logic, shown in the adjacent diagram.





The ALU also transfers data from the READ bus to the WRITE bus for register-to-register data moves: Data is output from the source register onto the READ bus, then transferred from the READ bus to the WRITE bus through the ALU, and finally loaded from the WRITE bus into the destination register.

The logic that translates the READ bus to the WRITE bus can also force the data lines of the WRITE bus to specific states to gate various arithmetic constants onto the bus. The accompanying diagram shows the the control signal on the left (RB14, R1, etc) and to the right is the bit pattern that's OR'ed onto the bus when the signal is asserted. The default state of the read bus is logical zero, so if no other read signal is asserted, the number that appears on the bus is the constant; otherwise, it's the constant inclusive OR'ed with the contents of the READ bus.

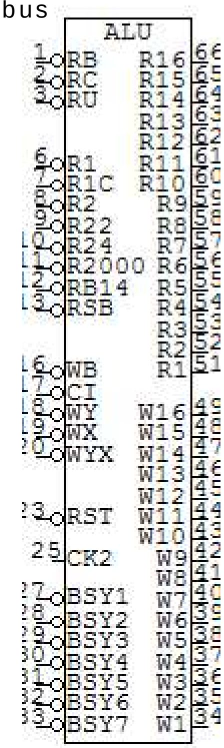


The ALU contains READ bus control logic. Registers in the MEM module, the central registers in the PROC module, and the ALU all interface to the READ bus through tri-state buffers. These buffers are normally in the high-impedance state, but the control module (CTL) issues READ control signals to output the contents of specific registers to the READ bus at certain times. Only one register should be gated onto the READ bus at any given time.

When no READ control signals are asserted, the ALU gates its output onto the READ bus by default. When the control module gates a register onto the READ bus, a BUSY signal is sent to the ALU module, which causes the ALU to inhibit its output. Because of propagation delays, the ALU might continue to output to the READ bus for a brief time while another register is also gated to the bus. To prevent this, all output to the read bus is inhibited during CLK1. This gives the control signals, which transition on the leading edge of CLK1, enough setup time to resolve the conflict.

ALU INPUTS:

I/F	signal	full name	state definition
CLK:			
	CLK1	CLOCK 1	1=read bus setup; inhibit read bus out
	CLK2	CLOCK 2	data transfer occurs on falling edge
CPM:			
	RB	READ B	0=output B register to write bus
	RC	READ C	0=output comp of reg B (C) to write bus
	RU	READ G	0=output U register to write bus
	R1	READ OCTAL 1	0=incl OR 000001 w/write bus
	R1C	READ OCTAL -1	0=incl OR 177776 w/write bus
	R2	READ OCTAL 2	0=incl OR 000002 w/write bus
	R22	READ OCTAL 22	0=incl OR 000022 w/write bus
	R24	READ OCTAL 24	0=incl OR 000024 w/write bus
	R2000	READ OCTAL 2000	0=incl OR 002000 w/write bus
	RB14	READ BIT14	0=incl OR 020000 w/write bus
	RSB	READ SIGN BIT	0=incl OR 100000 w/write bus
	WB	WRITE B	0=write into B from write bus
	CI	WRITE CI	0=set carry register to 1
	WY	WRITE Y	0=write Y
	WX	WRITE X	0=write into X from write bus
	WYX	WRITE Y	0=write into Y from write bus
RBUS:			
	RB_01	READ BUS 01	
	...		
	RB_14	READ BUS 14	
	RB_15	READ BUS 15	US (overflow) bit for read bus
	RB_16	READ BUS 16	SG (sign) bit for read bus
INP:	BUSY1	READ BUS BUSY	0=valid data from INP on read bus
OUT:	BUSY2	READ BUS BUSY	0=valid data from OUT on read bus
CTR:	BUSY3	READ BUS BUSY	0=valid data from CTR on read bus
INT:	BUSY4	READ BUS BUSY	0=valid data from INT on read bus
MBF:	BUSY5	READ BUS BUSY	0=valid data from MBF on read bus
CRG:	BUSY6	READ BUS BUSY	0=valid data from CRG on read bus
ADR:	BUSY7	READ BUS BUSY	0=valid data from ADR on read bus

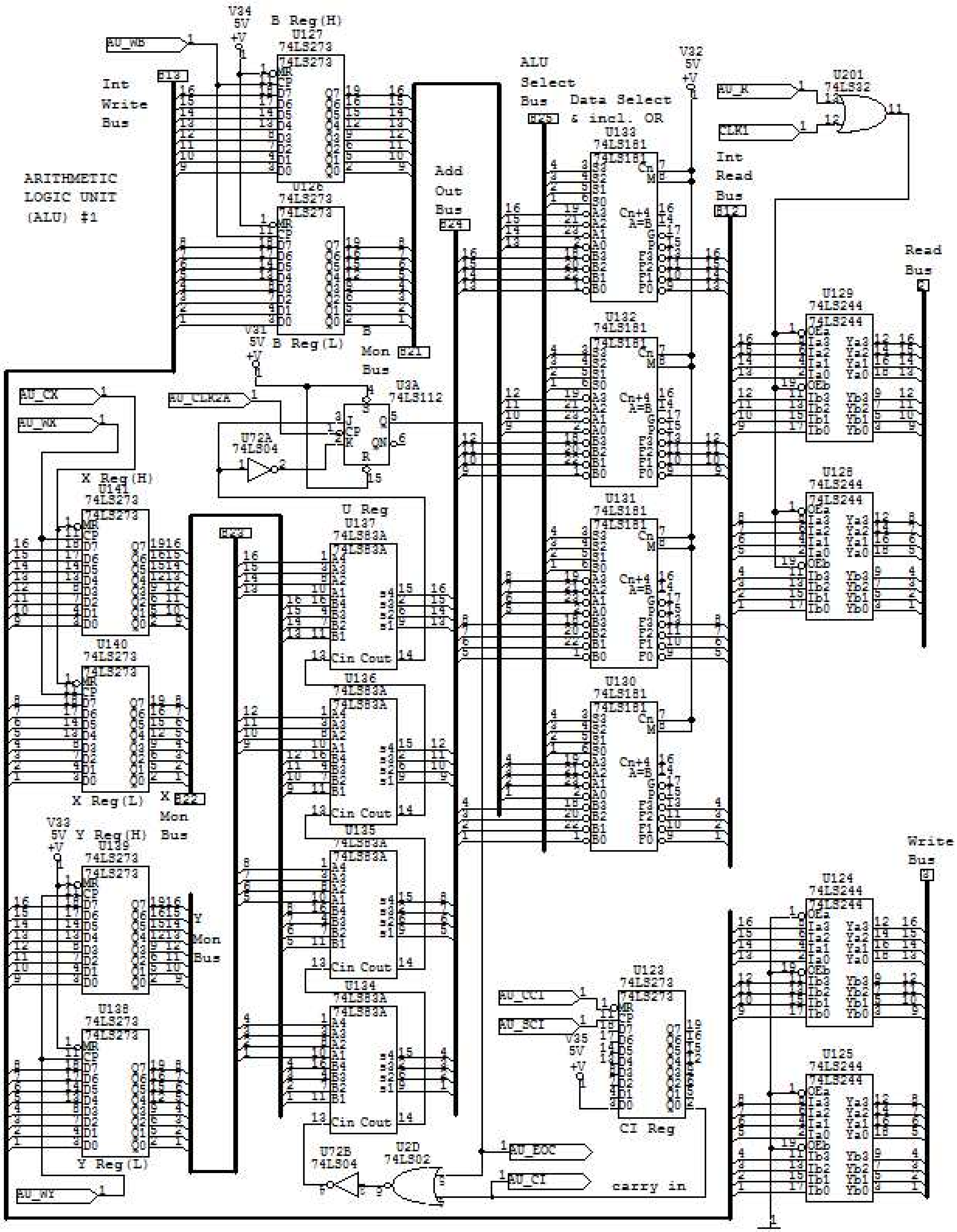


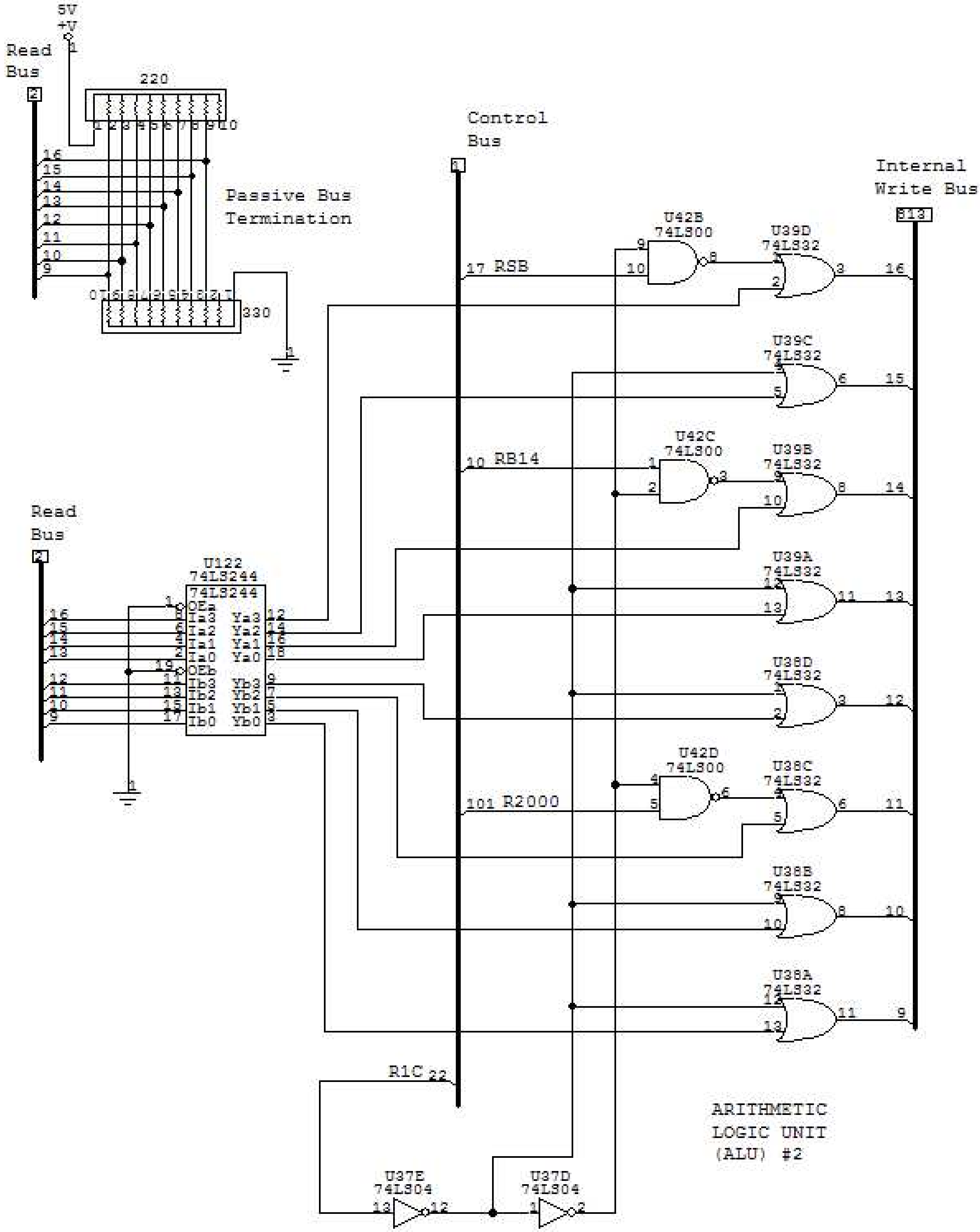
MBF OUTPUTS:

I/F	signal	full name	state definition

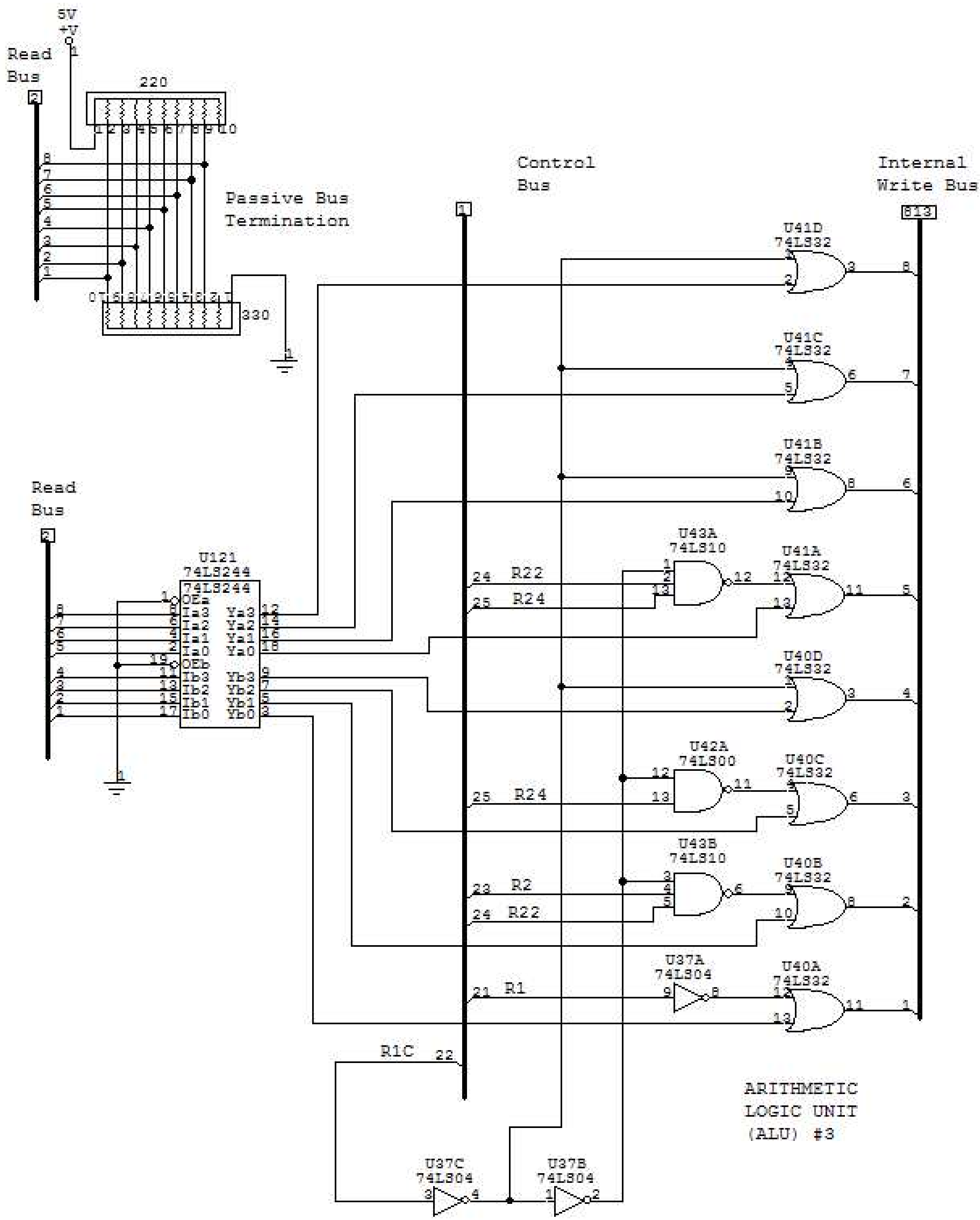
WBUS:

WB_01	WRITE BUS 01	
...		
WB_14	WRITE BUS 14	
WB_15	WRITE BUS 15	US (overflow) bit for write bus
WB_16	WRITE BUS 16	SG (sign) bit for write bus

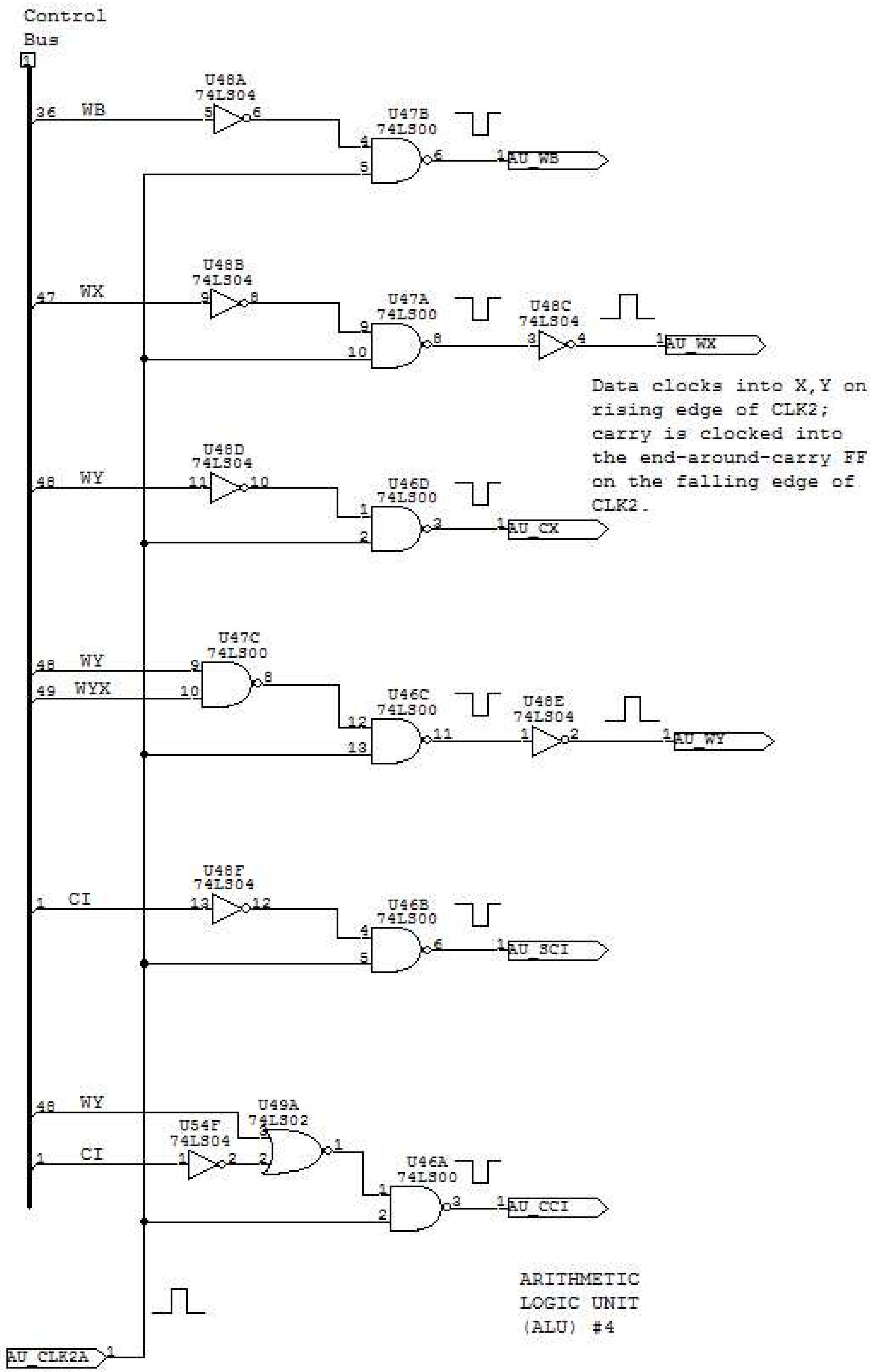

ARITHMETIC
LOGIC UNIT
(ALU) #1
AU_WB
V34
5V
+V
B Reg(H)
U127
74LS273
U126
74LS273
B Reg(L)
Int
Write
Bus
E13
B
Mon
Bus
E21
Add
Out
Bus
E24
ALU
Select
Bus
E25
Data Select
& incl. OR
U133
74LS181
U132
74LS181
U131
74LS181
U130
74LS181
V32
5V
+V
AU_R
CLK1
U201
74LS32
Int
Read
Bus
E12
Read
Bus
U129
74LS244
U128
74LS244
AU_CX
AU_WX
X Reg(H)
U141
74LS273
U140
74LS273
X Reg(L)
X
Mon
Bus
E22
V31
5V
+V
AU_CLR2A
U3A
74LS112
U72A
74LS04
U Reg
U137
74LS83A
U136
74LS83A
U135
74LS83A
U134
74LS83A
E23
V33
5V
Y Reg(H)
U139
74LS273
U138
74LS273
Y Reg(L)
Y
Mon
Bus
AU_WY
U72B
74LS04
U2D
74LS02
AU_EOC
AU_CI
carry in
AU_CC1
AU_SCI
V35
5V
+V
U123
74LS273
CI Reg
Write
Bus
U124
74LS244
U125
74LS244



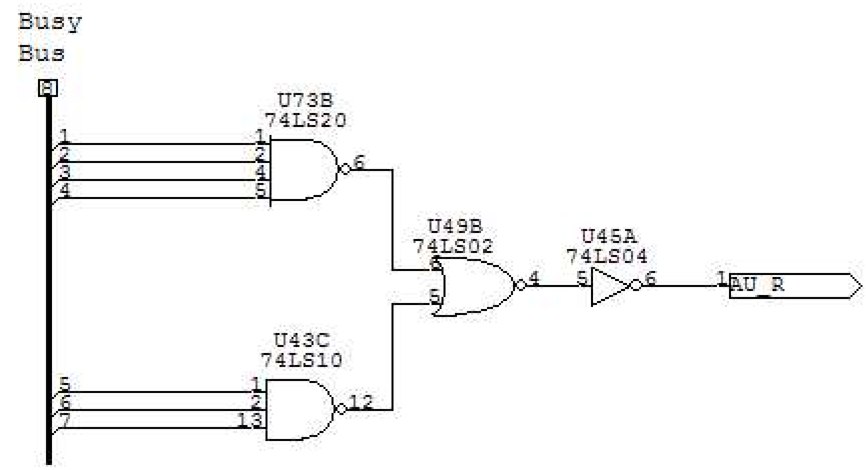
ARITHMETIC
LOGIC UNIT
(ALU) #2

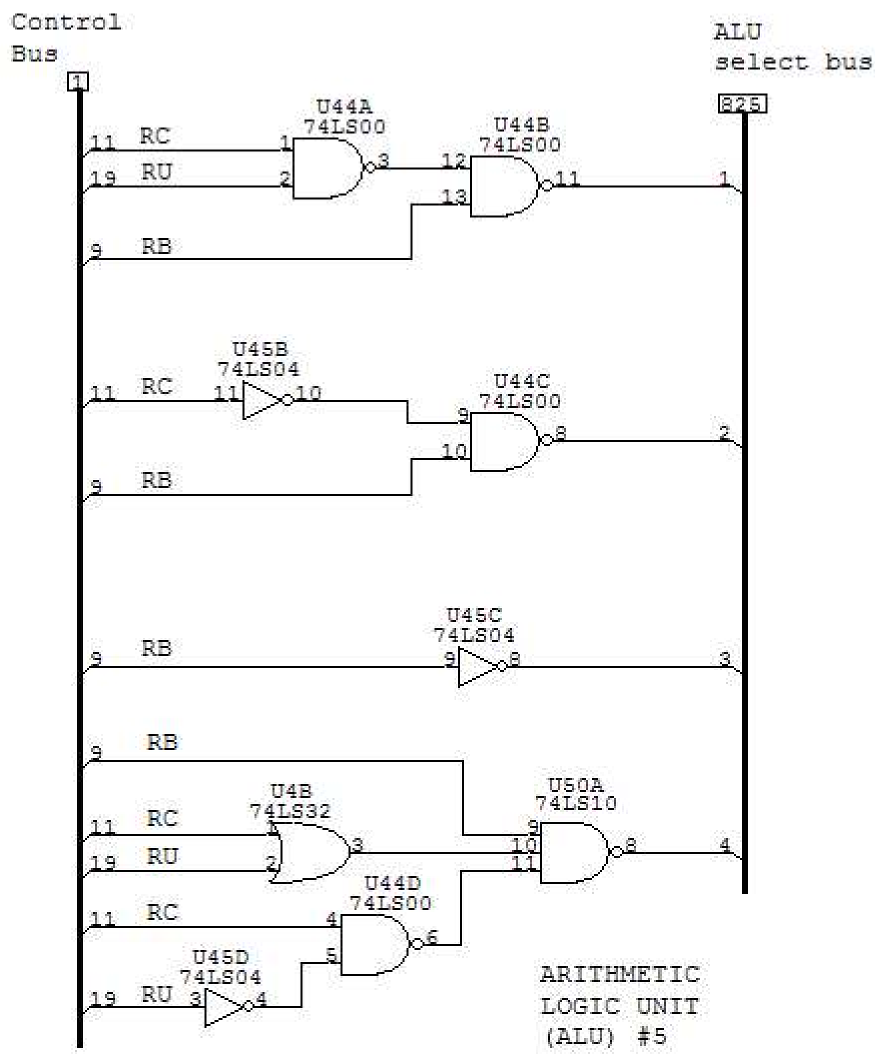


ARITHMETIC LOGIC UNIT (ALU) #3



ARITHMETIC
LOGIC UNIT
(ALU) #4





ARITHMETIC
LOGIC UNIT
(ALU) #5

CRG (Central Register)

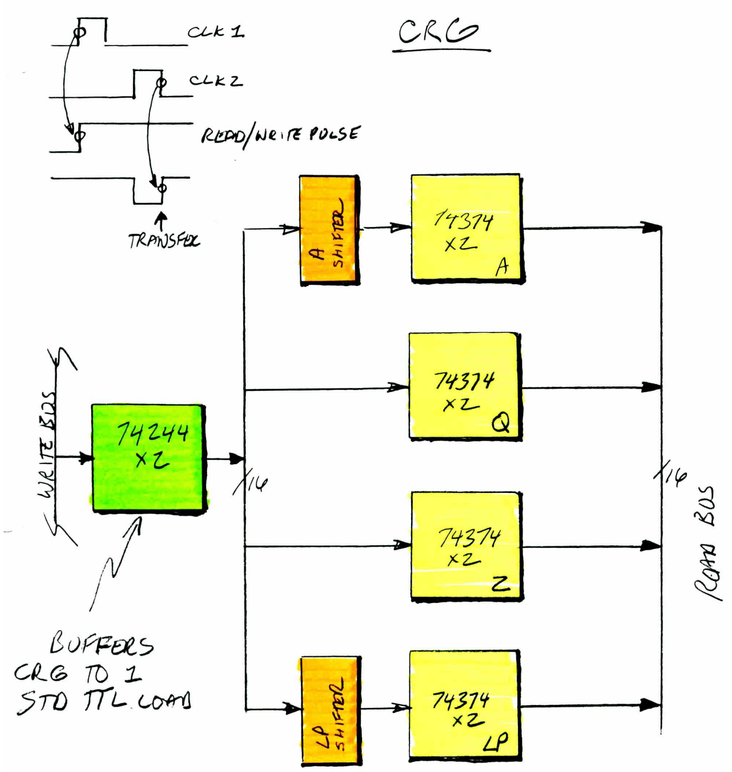
The AGC has four 16-bit registers for general computational use. These are called the "central registers":

A: the 16-bit accumulator, used for general computation.

Z: the 16-bit program counter, which contains the address of the next instruction to be executed.

Q: the 16-bit register used to hold the remainder in the DV instruction, and to hold the return address after TC instructions.

LP: the 16-bit register used to hold the lower product after MP instructions.



Register A and LP shifters

In addition to “normal” control pulses that write each line of the write bus into the corresponding bit of the registers, the A and LP registers have special write control pulses that shift bits:

The WALP control pulse bit-shifts into the A and LP register. The table below shows how the shifter works. The row of 16 comma-separated entries represent bits in the register. The leftmost position is the register MSB, rightmost position is the LSB. The entry shows the bit of the WRITE bus that’s mapped onto that register bit by the shifter.

For the WALP pulse, bit 1 of the WRITE bus (B1) is written into bit 14 of the LP register. “BX” means leave that bit of the register alone (don’t change it).

The same WALP pulse causes bit 2 of the WRITE bus (B2) to be written into the lowest bit of the A register, bit 3 of the WRITE bus (B3) to be written into the next bit, and so forth. “US” (uncorrected sign) is the overflow bit (bit 15) of the WRITE bus. “SG” is the 1's complement sign (bit 16) from the WRITE bus.

WALP for register LP:

BX, BX, B1, BX, BX, BX, BX, BX, BX, BX, BX, BX, BX, BX, BX, BX

WALP for register A:

SG, SG, US, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2

Similarly, the WLP control pulse bit-shifts the WRITE bus into the LP register as follows (D0 on bit 14 of the LP register means force the bit to zero):

WLP

B1, B1, D0, B14, B13, B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2

The logic design for handling bit 14 of the LP register, which takes control inputs from WALP, WLP, and WA3 is shown here. WA3 is identical to WLP.

Depending upon the WALP, WLP, or WA3 inputs, bit 14 of LP will either be set to B1 of the WRITE bus, or forced to zero.

A WA3	B WLP	C WALP	D B1	B14	
0	0	0	0	X	ILLEGAL
0	0	0	1	X	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	X	ILLEGAL
0	1	0	1	X	
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	X	ILLEGAL
1	0	0	1	X	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	
1	1	1	0	X	INACTIVE
1	1	1	1	X	

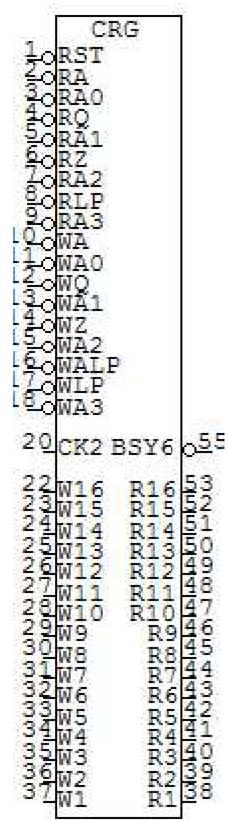
AB \ CD	00	01	11	10
00	X	X		
01	X	X		
11		1	X	X
10	X	X		

$B14 = C' \cdot D$

LOGIC DESIGN FOR BIT 14 INPUT TO LP REGISTER

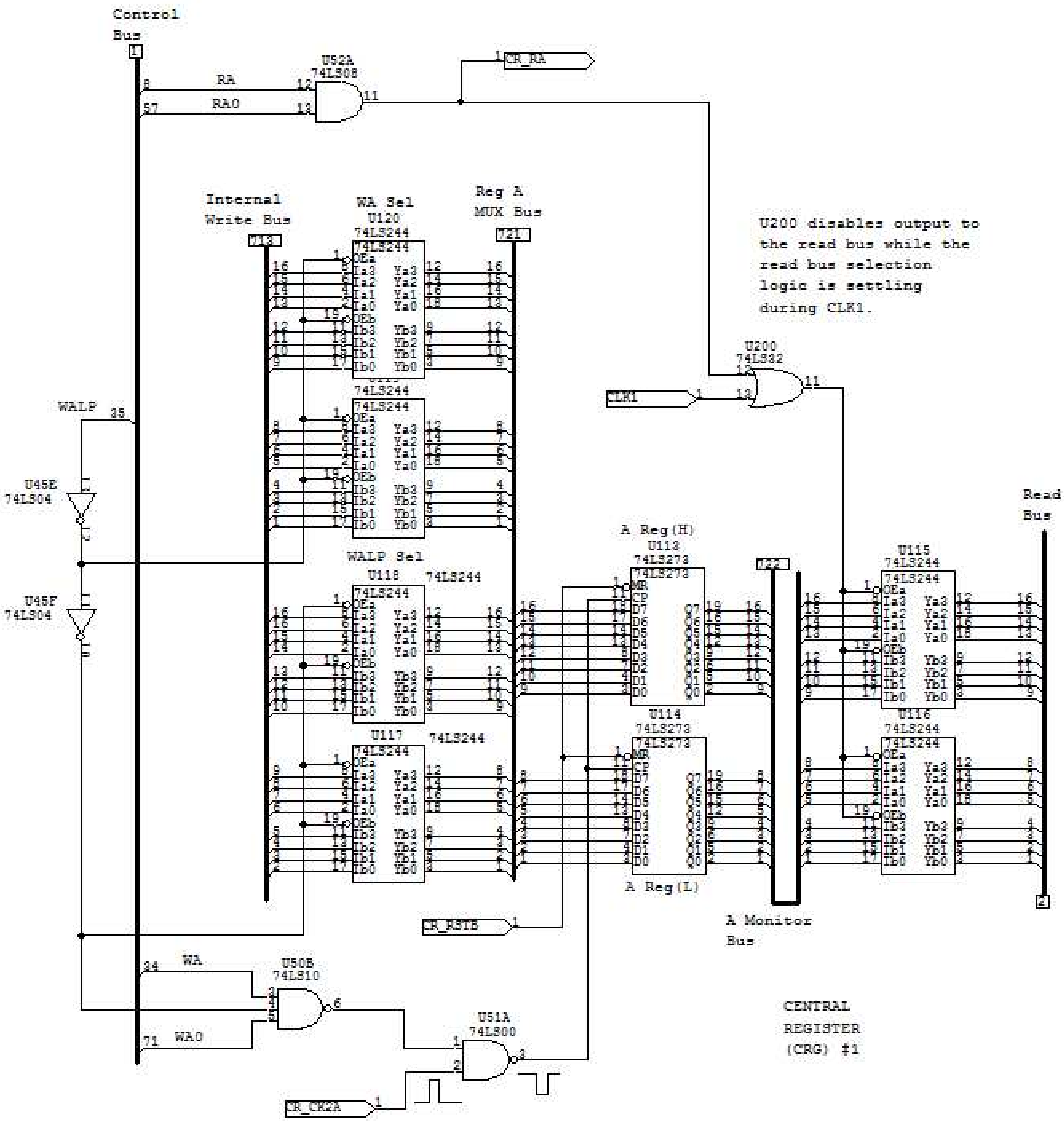
CRG INPUTS:

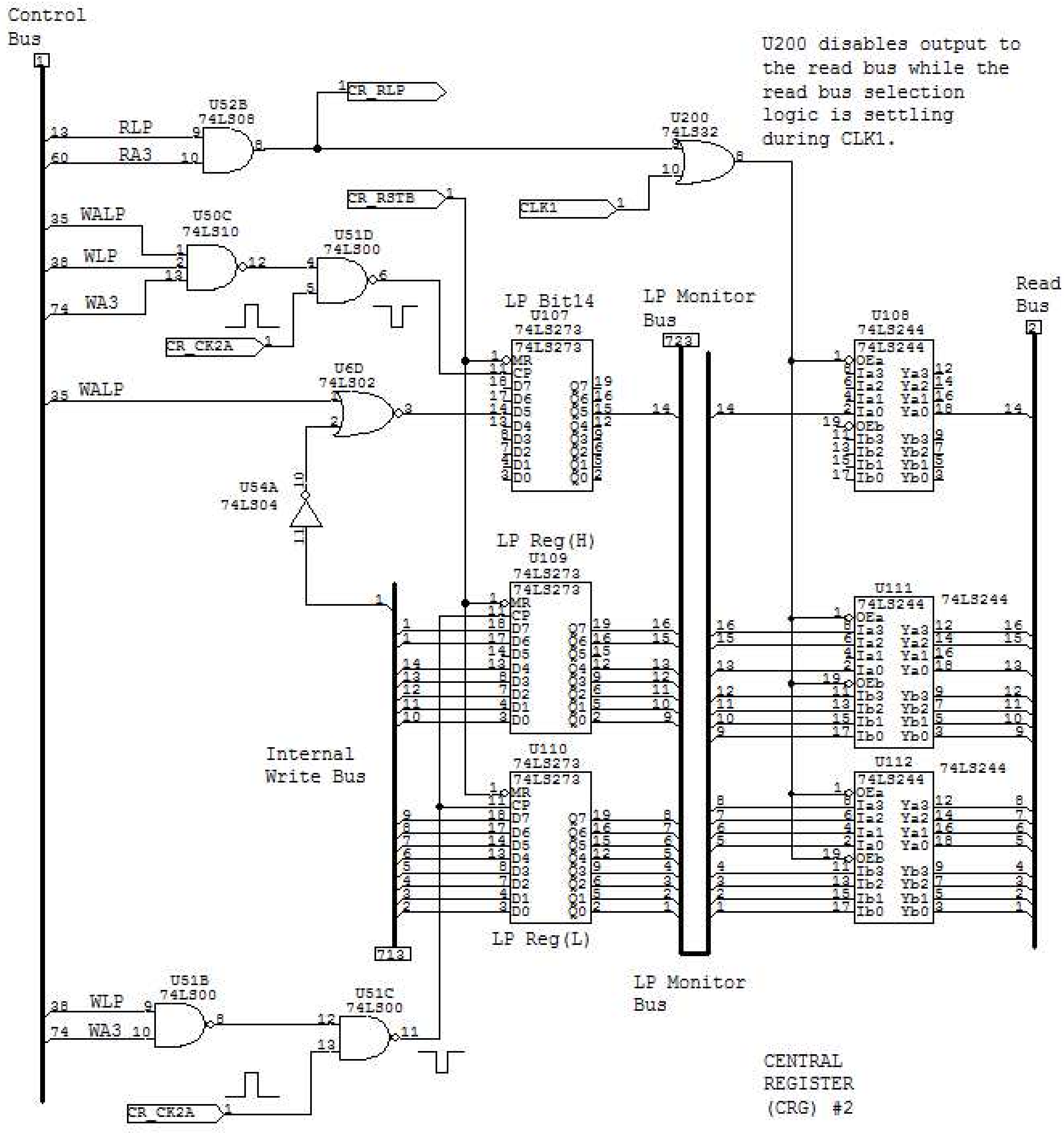
I/F	signal	full name	state definition
CLK:			
	CLK1	CLOCK 1	1=read bus setup; inhibit read bus output
	CLK2	CLOCK 2	data transfer occurs on falling edge
CPM:			
	RA	READ A	0=output A to read bus
	RA0	READ A	0=output A to read bus
	RQ	READ Q	0=output Q to read bus
	RA1	READ Q	0=output Q to read bus
	RZ	READ Z	0=output Z to read bus
	RA2	READ Z	0=output Zto read bus
	RLP	READ LP	0=output LPto read bus
	RA3	READ LP	0=output LP to read bus
	WA	WRITE A	0=load A from write bus
	WA0	WRITE A	0=load A from write bus
	WQ	WRITE Q	0=load Q from write bus
	WA1	WRITE Q	0=load Q from write bus
	WZ	WRITE Z	0=load Z from write bus
	WA2	WRITE Z	0=load Z from write bus
	WALP	WRITE A,LP	0=load A,LP from write bus
	WLP	WRITE LP	0=load LP from write bus
	GENRST	GENERAL RESET	0=General Reset
WBUS:			
	WB_01	WRITE BUS 01	
	...		
	WB_14	WRITE BUS 14	
	WB_15	WRITE BUS 15	US (overflow) bit for write bus
	WB_16	WRITE BUS 16	SG (sign) bit for write bus



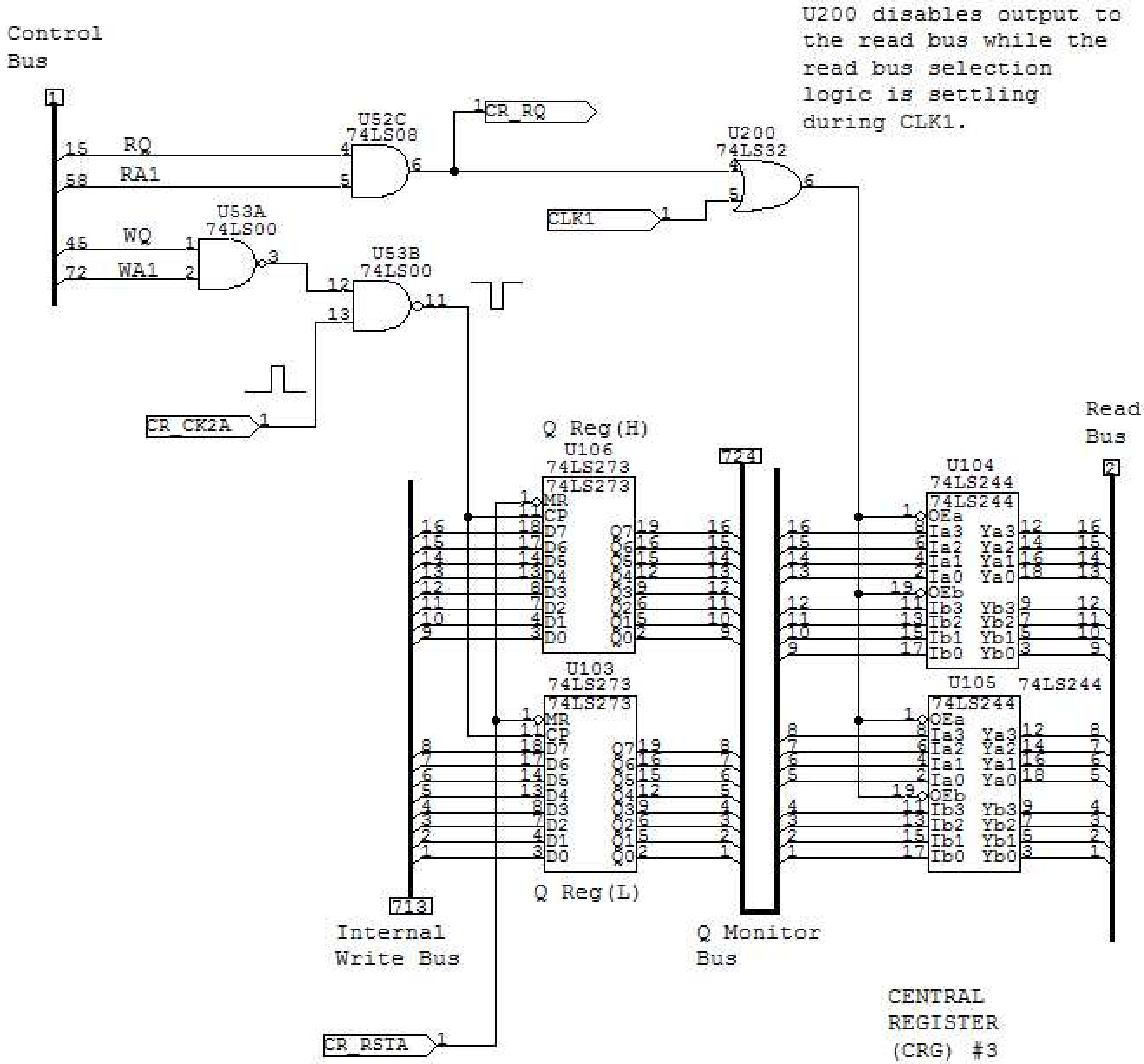
MBF OUTPUTS:

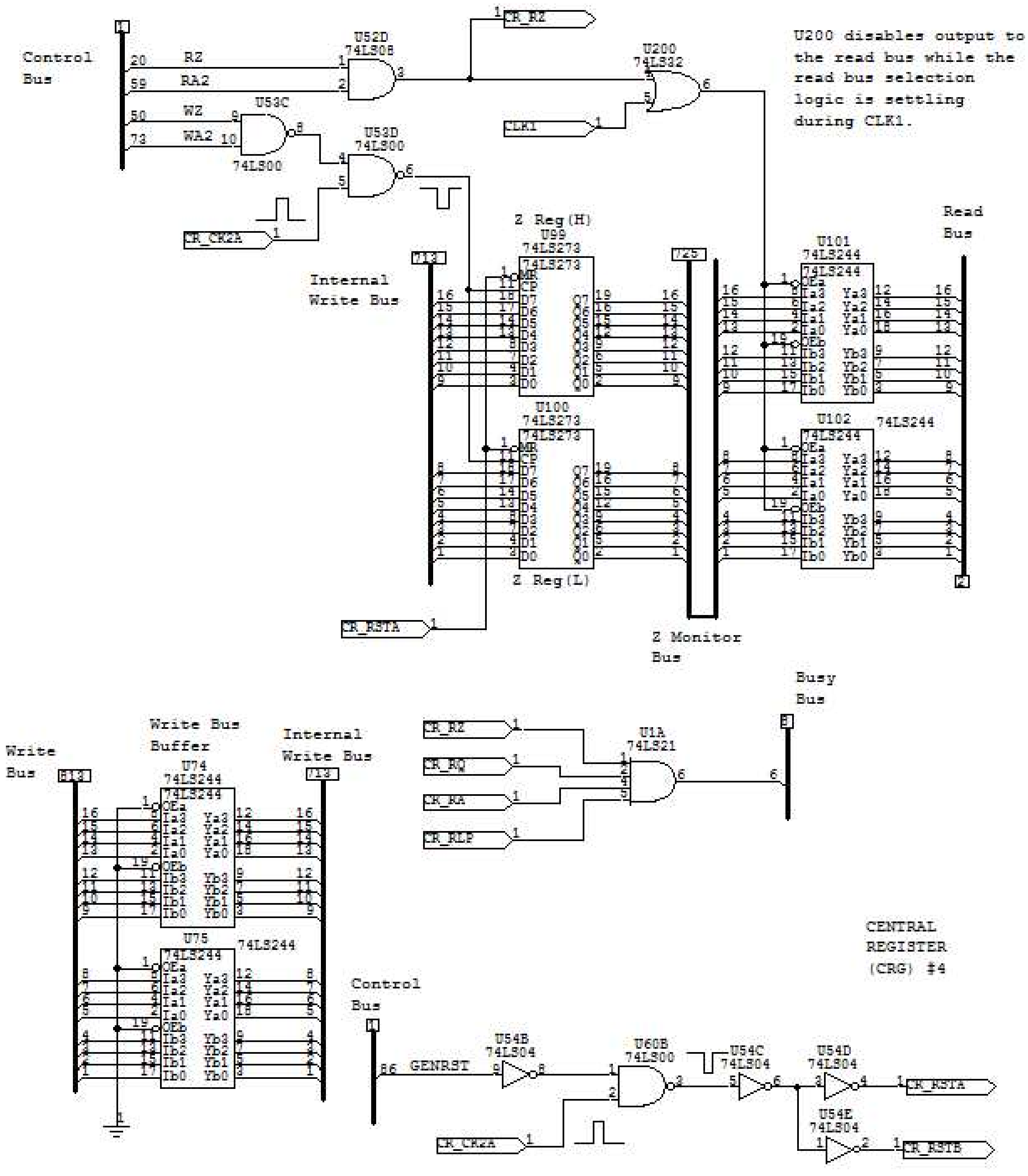
I/F	signal	full name	state definition
RBUS:			
	RB_01	READ BUS 01	
	...		
	RB_14	READ BUS 14	
	RB_15	READ BUS 15	US (overflow) bit for read/write bus
	RB_16	READ BUS 16	SG (sign) bit for read/write bus
	BUSY	READ BUS BUSY	0=output enabled to read bus


Control Bus
RA
RA0
U52A
74LS08
CR_RA
Internal Write Bus
713
WA Sel
U120
74LS244
Reg A MUX Bus
721
U200 disables output to the read bus while the read bus selection logic is settling during CLK1.
U200
74LS32
CLK1
WALP
U45E
74LS04
U45F
74LS04
WALP Sel
U118
74LS244
U117
74LS244
A Reg(H)
U113
74LS273
A Reg(L)
U114
74LS273
722
A Monitor Bus
Read Bus
U115
74LS244
U116
74LS244
CR_RSTB
WA
WA0
U50B
74LS10
U51A
74LS00
CR_CK2A
CENTRAL REGISTER (CRG) #1



Control Bus
U200 disables output to the read bus while the read bus selection logic is settling during CLK1.
CR_RLP
U52B
74LS08
RLP
RA3
U200
74LS32
CR_RSTB
CLK1
WALP
WLP
WA3
U50C
74LS10
U51D
74LS00
CR_CK2A
LP Bit14
U107
74LS273
LP Monitor Bus
723
Read Bus
U108
74LS244
U6D
74LS02
U54A
74LS04
LP Reg(H)
U109
74LS273
U111
74LS244
Internal Write Bus
U110
74LS273
U112
74LS244
713
LP Reg(L)
U51B
74LS00
U51C
74LS00
CENTRAL REGISTER (CRG) #2


U200 disables output to the read bus while the read bus selection logic is settling during CLK1.
Control Bus
RQ
RA1
WQ
WA1
U52C 74LS08
U53A 74LS00
U53B 74LS00
U200 74LS32
CR_RQ
CLK1
CR_CK2A
CR_RSTA
Q Reg(H)
U106 74LS273
Q Reg(L)
U103 74LS273
U104 74LS244
U105 74LS244
Read Bus
Internal Write Bus
Q Monitor Bus
CENTRAL REGISTER (CRG) #3



U200 disables output to the read bus while the read bus selection logic is settling during CLK1.
Control Bus
RZ
RA2
WZ
WA2
U52D 74LS08
U53C 74LS00
U53D 74LS00
U200 74LS32
CR_RZ
CLK1
CR_CK2A
Internal Write Bus
Z Reg(H)
U99 74LS273
U100 74LS273
Z Reg(L)
CR_RSTA
Z Monitor Bus
Read Bus
U101 74LS244
U102 74LS244
Busy Bus
CR_RZ
CR_RQ
CR_RA
CR_RLP
U1A 74LS21
Write Bus
Write Bus Buffer
U74 74LS244
U75 74LS244
Internal Write Bus
CENTRAL REGISTER (CRG) #4
Control Bus
GENRST
U54B 74LS04
U60B 74LS00
U54C 74LS04
U54D 74LS04
U54E 74LS04
CR_CK2A
CR_RSTA
CR_RSTB

INT (Interrupt Priority)

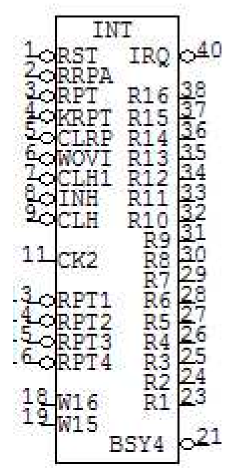
The original AGC had five vectored interrupts. This recreation implements the following 3:

RUPT1	Also called T3RUPT because it's triggered by overflow of the TIME3 priority counter.
RUPT3	Also called T4RUPT because it's triggered by overflow of the TIME4 priority counter. Because the interrupt is used by software to update the DSKY display at regular intervals, it's sometimes called DSRUPT.
RUPT4	Triggered by a key press from the user's keyboard. Also called KEYRUPT.

The AGC software responds to each interrupt by temporarily suspending the current program, executing a short interrupt service routine, and then resuming the interrupted program.

INT INPUTS:

I/F	signal	full name	state definition
CLK:			
	CLK1	CLOCK 1	1=read bus setup; inhibit read bus out
	CLK2	CLOCK 2	data transfer occurs on falling edge
CPM:			
	GENRST	GENERAL RESET	0=reset INT registers
	RRPA	READ RUPT ADDRESS	0=output RPCELL address (2004,2010,2014,2020,2024) to read bus
	RPT	READ RUPT OPCODE	0=load RUPT opcode into RPCELL register
	KRPT	KNOCK DOWN RUPT PRIO	0=reset RUPT latch currently selected by RPCELL register
	CLRP	CLEAR RPCELL	0=clear RPCELL register
	WOVI	WRITE OVF RUPT INH	0=test overflow; if overflow, inhibit interrupt (set INHINT1)
	CLINH1	CLEAR INHINT1	0=clear INHINT1 register
	INH	SET INHINT	0=set INHINT register
	CLINH	CLEAR INHINT	0=clear INHINT register
WBUS:			
	WB_15	WRITE BUS 15	US (overflow) bit for write bus
	WB_16	WRITE BUS 16	SG (sign) bit for write bus
CTR/KBD:			
	RUPT1	INTERRUPT 1	0=trigger interrupt 1 (2004 octal; TIME3 overflow)
	RUPT3	INTERRUPT 3	0=trigger interrupt 3 (2014 octal; TIME4 overflow)
	RUPT4	INTERRUPT 4	0=trigger interrupt 4 (negative edge) (2020 octal; keyboard activity)



Note: interrupt cells above RUPT4 not implemented.

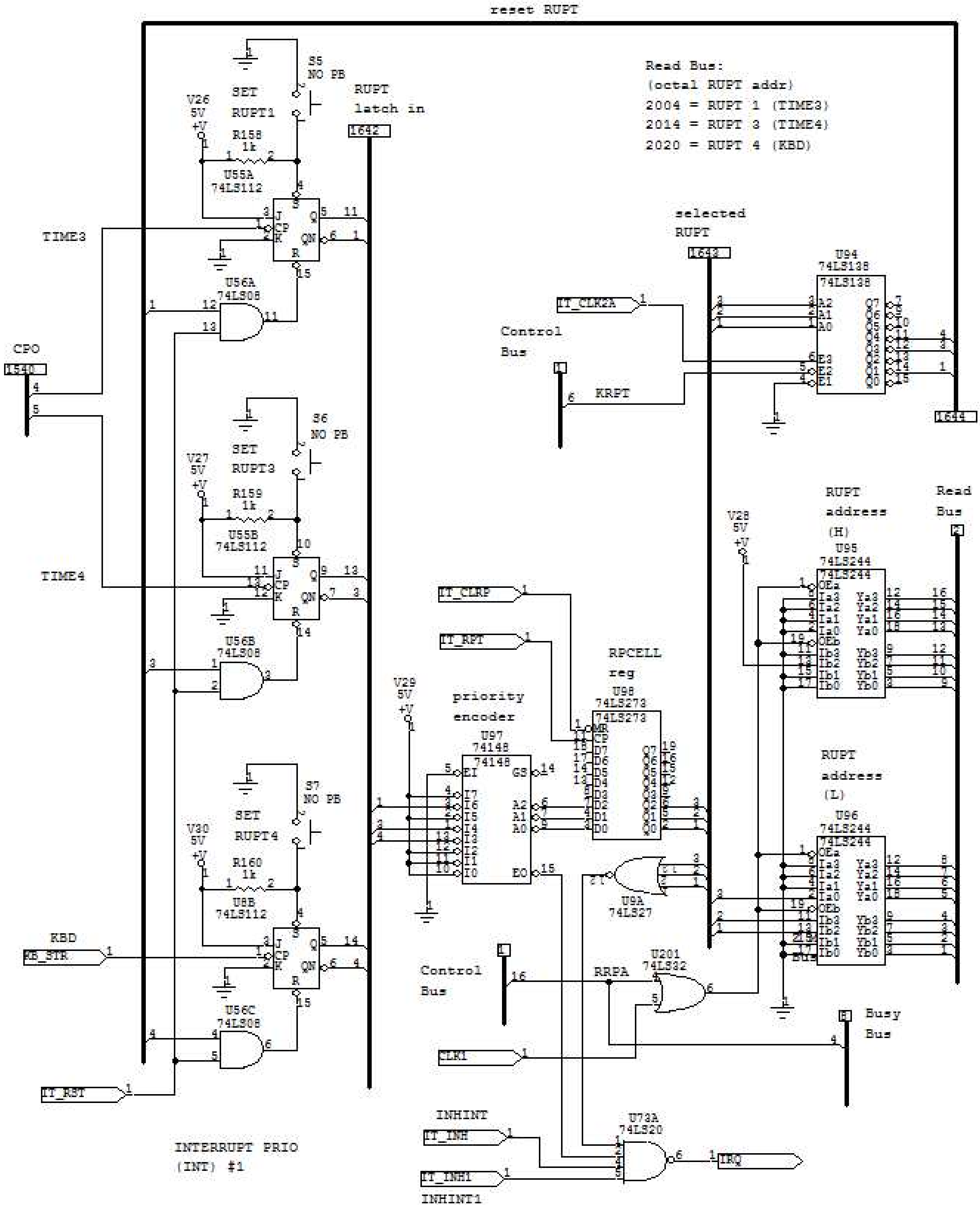
INT OUTPUTS:

I/F	signal	full name	state definition
SEQ:			
	IRQ	INT RQST	0=interrupt requested. Active if allof the following are true: a) one or more RUPT FF's are set b) interrupt is not currently being serviced

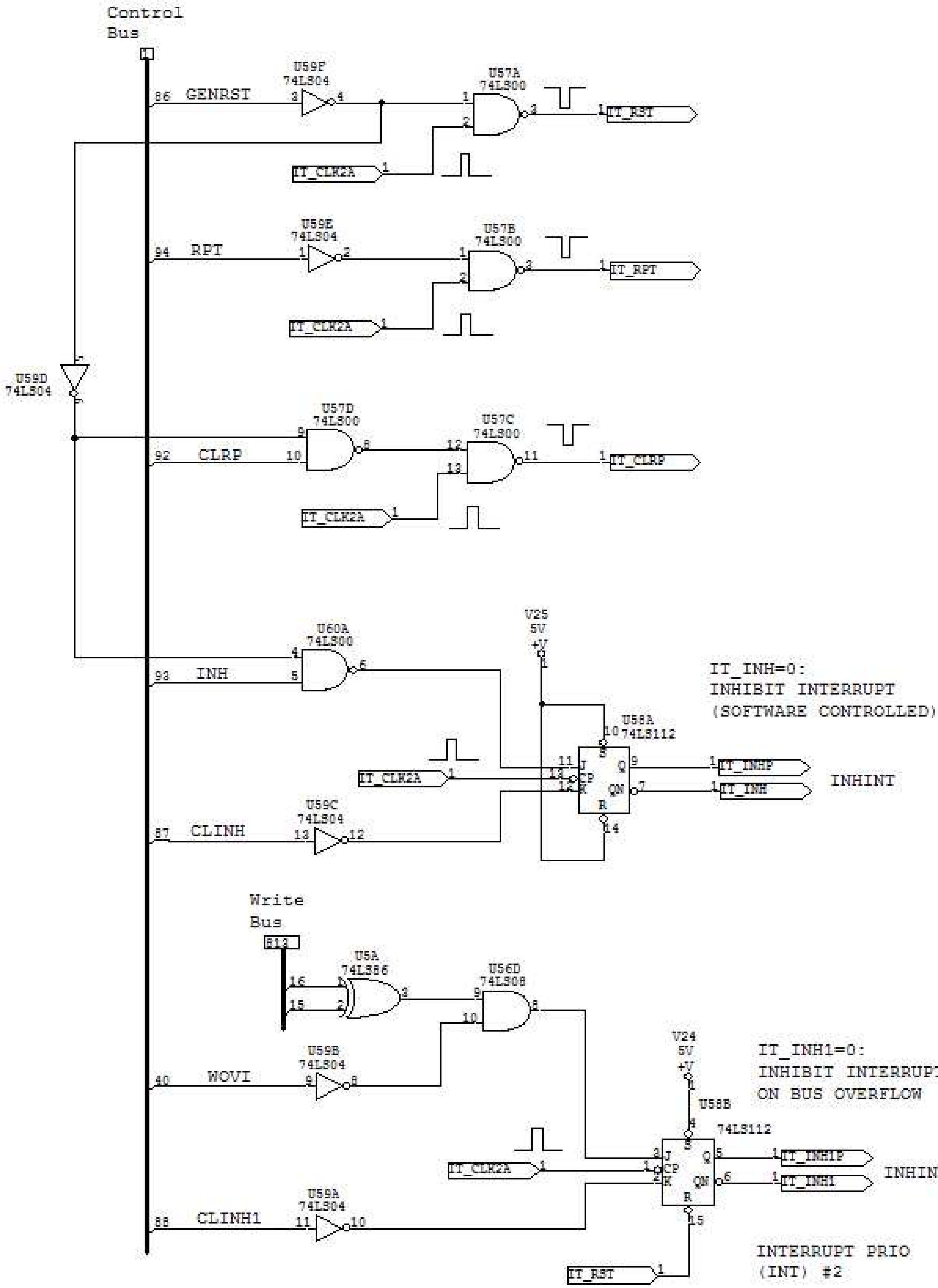
c) interrupts are not inhibited

RBUS:

RB_01	READ BUS 01	
...		
RB_14	READ BUS 14	
RB_15	READ BUS 15	US (overflow) bit for read/write bus
RB_16	READ BUS 16	SG (sign) bit for read/write bus
BUSY	READ BUS BUSY	0=output enabled to read bus



INTERRUPT PRIO (INT) #1


Control
Bus
GENRST
U59F
74LS04
U57A
74LS00
IT_RST
IT_CLK2A
RPT
U59E
74LS04
U57B
74LS00
IT_RPT
IT_CLK2A
U59D
74LS04
U57D
74LS00
U57C
74LS00
CLRP
IT_CLRP
IT_CLK2A
U60A
74LS00
INH
V25
5V
+V
IT_INH=0:
INHIBIT INTERRUPT
(SOFTWARE CONTROLLED)
U58A
74LS112
IT_CLK2A
IT_INHP
IT_INH
INHINT
CLINH
U59C
74LS04
Write
Bus
U5A
74LS86
U56D
74LS08
WOVI
U59B
74LS04
V24
5V
+V
IT_INH1=0:
INHIBIT INTERRUPT
ON BUS OVERFLOW
U58B
74LS112
IT_CLK2A
IT_INH1P
IT_INH1
INHINT1
CLINH1
U59A
74LS04
INTERRUPT PRIO
(INT) #2
IT_RST

CTR (Priority Counter)

The Block I AGC had 20 memory locations dedicated as up/down counters (involuntary counters). The counters would increment or decrement in response to external plus or minus logic signals. Increment (PINC) or decrement (MINC) was handled by one subsequence of microinstructions inserted between any two regular instruction subsequences when counter inputs occurred.

This replica implements 5 counters used by the AGC operating system and user interface:

Counter	Addr	Description
OVCTR	34	An overflow counter incremented (PINC) or decremented (MINC) when overflow conditions occur during certain instructions.
TIME2	35	The high-order bits of the AGC clock; incremented (PINC) by overflow of TIME1.
TIME1	36	The low-order bits of the AGC clock; incremented (PINC) by a 100Hz signal from the SCALER (SCL) in the control module (CTL).
TIME3	37	A general purpose timer incremented by a 100Hz signal from the SCALER (SCL) in the control module (CTL).
TIME4	40	A special purpose timer used for software update of the DSKY display. Incremented by a 100Hz signal from the SCALER (SCL) in the control module (CTL).

INT/CTR interface

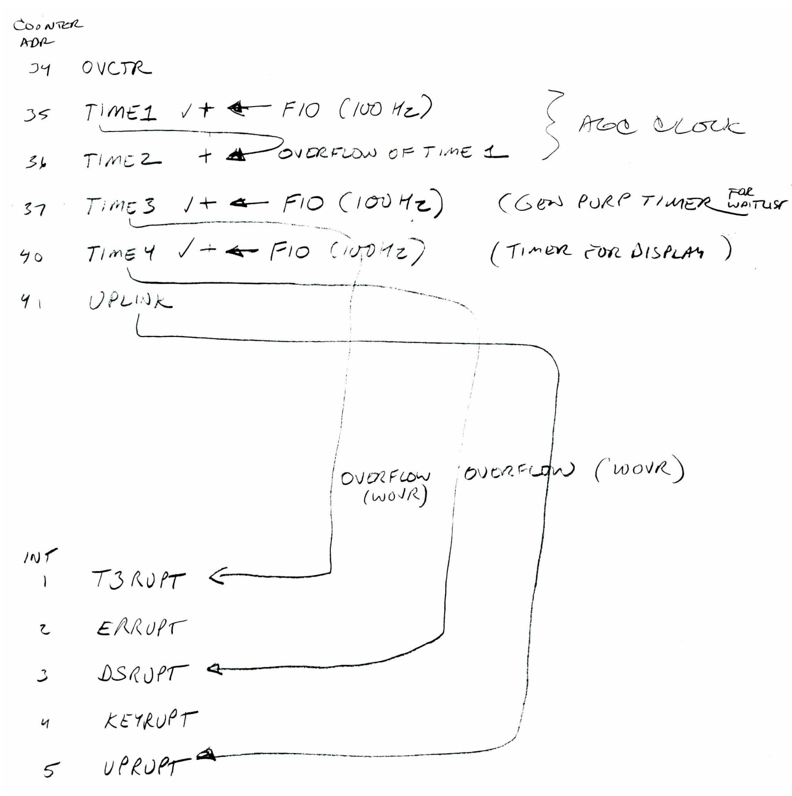
This chart shows how counter overflows are handled. F10 (from the scaler) increments TIME1, TIME3, and TIME4.

A positive overflow of TIME1 causes an increment of TIME2.

Positive overflow of TIME3 triggers a T3RUPT interrupt.

Positive overflow of TIME4 triggers a DSRUPT (T4RUPT) interrupt.

The addresses of TIME1 and TIME2 are reversed for Block II. This chart shows the Block I order, but my replica used the Block II order for compatibility with the COLOSSUS flight software.



CTR Design Problem

During unit testing, I uncovered a bug in my implementation of CTR: The plus inputs set the P-cell which eventually triggers a PINC subsequence and increments the counter. Minus inputs set the M-cell, which triggers a MINC subsequence and decrements the counter. Near-simultaneous plus and minus inputs should cancel out, producing no change in count.

As a consequence, the counter logic was designed so that, if the P- and M-cells are both set, no counter sequence (PINC or MINC) is selected. The problem is, the P- and M-cells are reset by WOVR, which only occurs in PINC or MINC. So, if both cells are set--and therefore, no subsequence is selected--WOVR is never issued, the cells never reset, and all counting activity is disabled for that counter.

Here are the options I developed fixing the design:

a) Create a new subsequence similar to PINC or MINC that issues WOVR, but does not change the counter. Select this new subsequence when the P- and M-cells are simultaneously set.

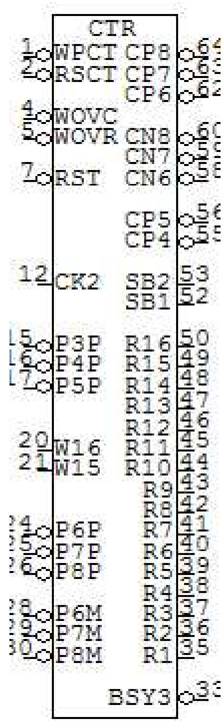
b) Allow the P- and M-cells to trigger PINC and MINC sequences, resulting in a net change of zero to the counter.

c) Leave the design as-is, because the M-cell is only used for OVCTR in my implementation, and therefore, the problem can never occur.

I choose option C to avoid redesign and re-unit-testing of CTR.

CTR INPUTS:

I/F	signal	full name	state definition
CLK:			
	CLK1	CLOCK 1	1=read bus setup; inhibit read bus out
	CLK2	CLOCK 2	data transfer occurs on falling edge
CPM:			
	GENRST	GENERAL RESET	0=reset CTR registers
	WPCTR	WRITE PSEQ	0=write sequence into PSEQ
	RSCT	READ PCELL ADDRESS	0=output PCELL address (034-043) to read bus
	WOVC	WRITE OVRFLOW CNTR	0=test overflow and inc/dec OVCTR
	WOVR	WRITE OVERFLOW	0=clear selected PCELL and handle counter overflow (if any)
WBUS:			
	WB_15	WRITE BUS 15	US (overflow) bit for write bus
	WB_16	WRITE BUS 16	SG (sign) bit for write bus
EXTERNAL:			
	P3P	P3 CELL + COUNT	0=count up P3 counter (036 octal; TIME1)
	P4P	P4 CELL + COUNT	0=count up P4 counter (037 octal; TIME3)
	P5P	P5 CELL + COUNT	0=count up P5 counter (040 octal, TIME4)
			Note: priority cells 6-20 not implemented.



CTR OUTPUTS:

I/F	signal	full name	state definition
COUNTER OVERFLOW:			
	CPO_04	P4 + OVERFLOW	0=P4 cell pos ovf (during WOVR) note: TIME3 + overflow; connect to INT subsystem to trigger T3RUPT interrupt.
	CPO_05	P5 + OVERFLOW	0=P5 cell pos ovf (during WOVR)

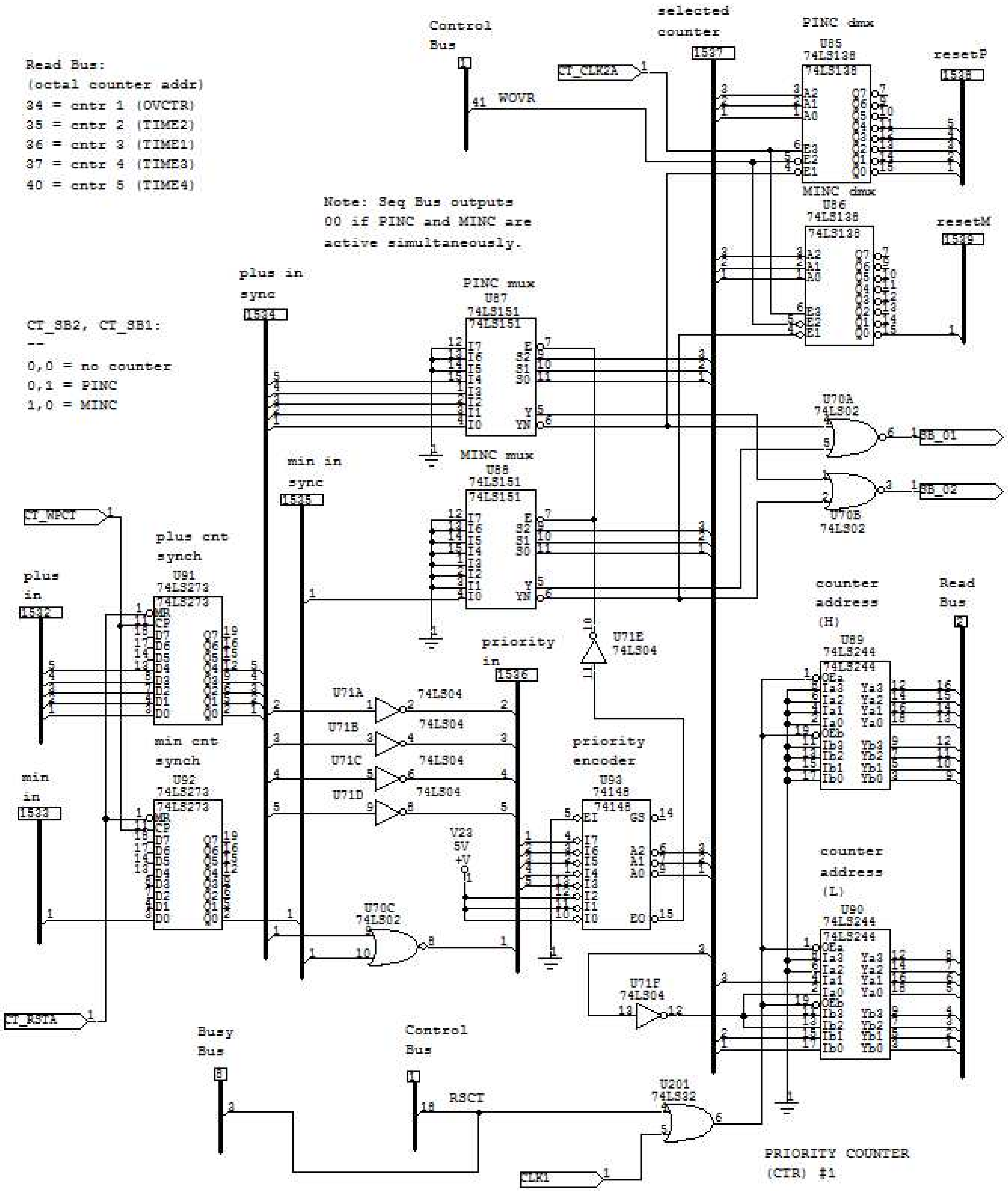
		note: TIME4 + overflow; connect to INT subsystem to trigger T4RUPT (DSRUPT) interrupt.

SEQ:

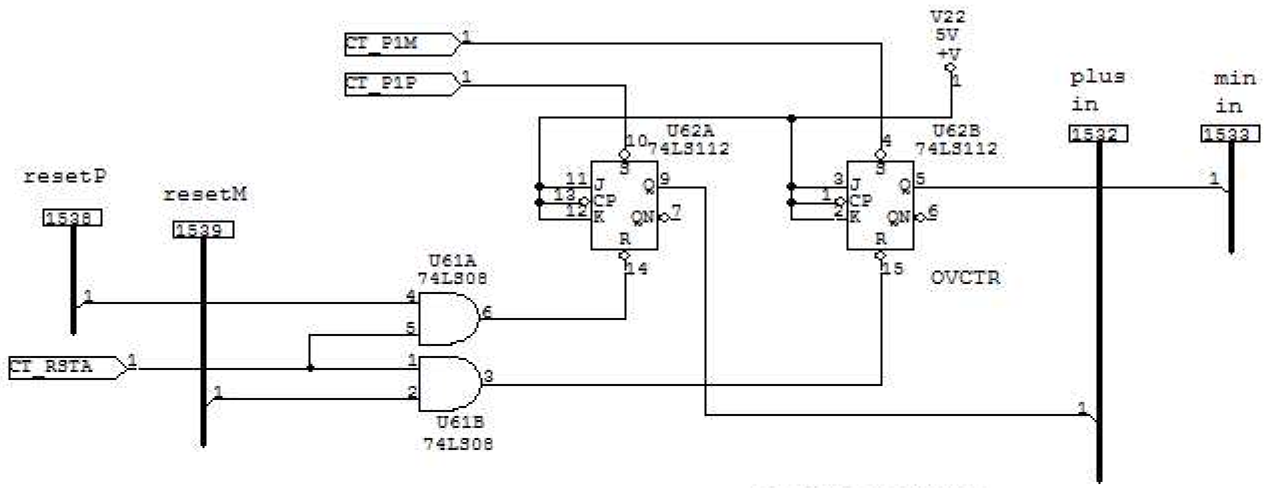
SB_01	SUB SEL 01	SB_01 is LSB; SB_02 is MSB
SB_02	SUB SEL 02	00=no counter; 01=PINC; 10=MINC

RBUS:

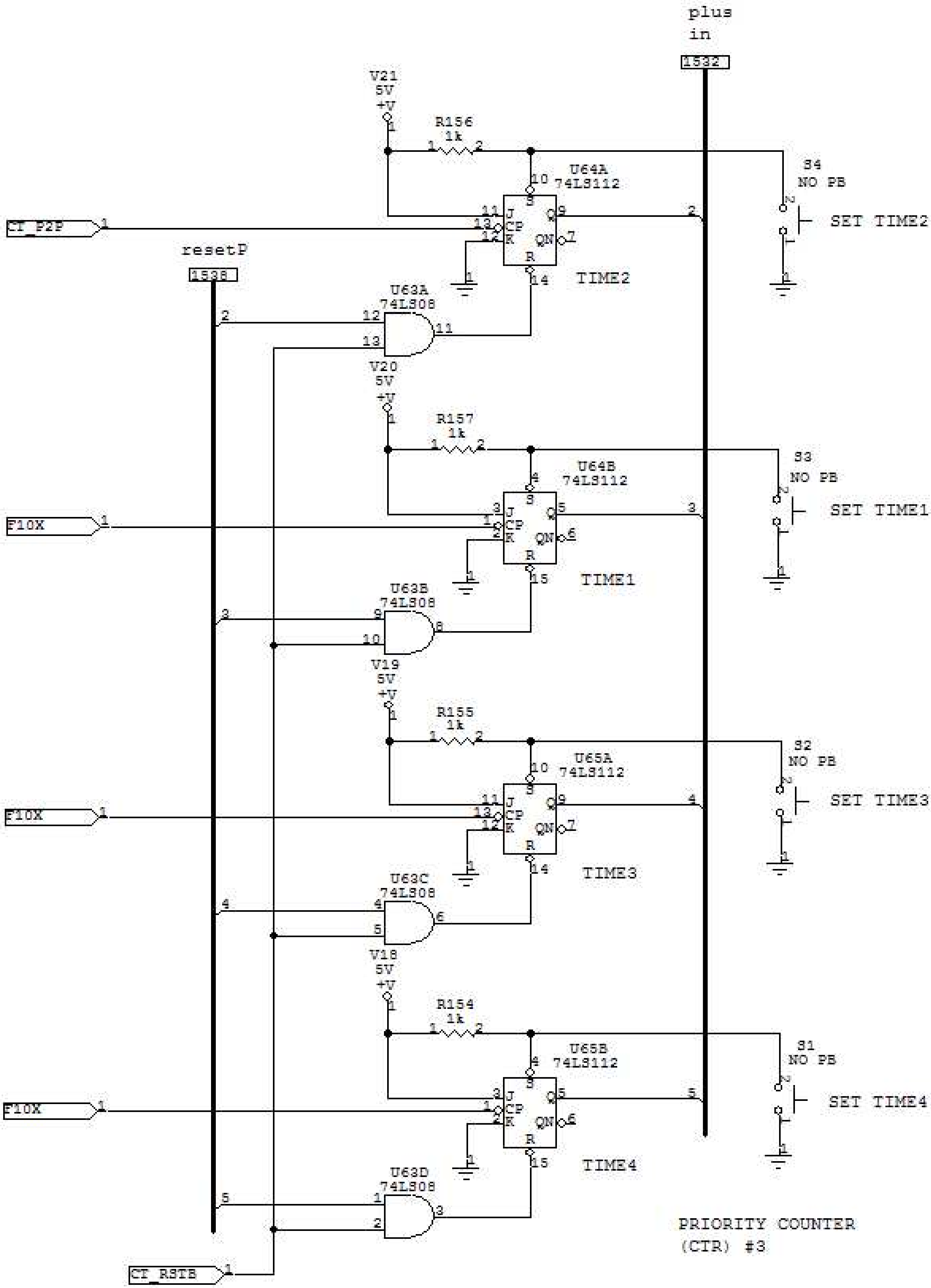
RB_01	READ BUS 01	
...		
RB_14	READ BUS 14	
RB_15	READ BUS 15	US (overflow) bit for read/write bus
RB_16	READ BUS 16	SG (sign) bit for read/write bus
BUSY	READ BUS BUSY	0=output enabled to read bus

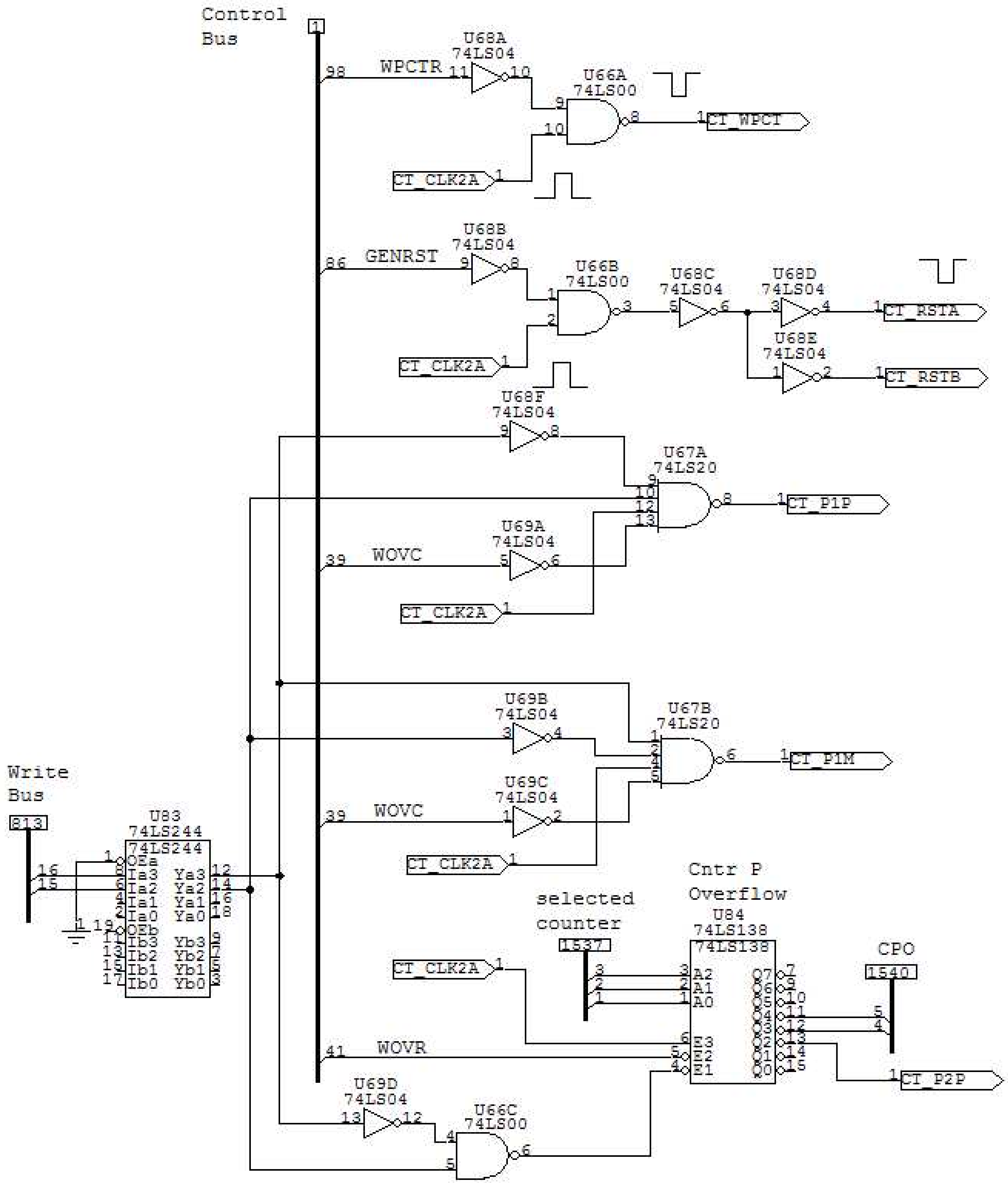


Read Bus:
(octal counter addr)
34 = cntr 1 (OVCTR)
35 = cntr 2 (TIME2)
36 = cntr 3 (TIME1)
37 = cntr 4 (TIME3)
40 = cntr 5 (TIME4)
Note: Seq Bus outputs 00 if PINC and MINC are active simultaneously.
CT_SB2, CT_SB1:
--
0,0 = no counter
0,1 = PINC
1,0 = MINC
Control Bus
WOVR
CT_CLK2A
selected counter
1537
PINC dmx
U85
74LS138
resetP
1538
MINC dmx
U86
74LS138
resetM
1539
plus in sync
1534
PINC mux
U87
74LS151
min in sync
1535
MINC mux
U88
74LS151
U70A
74LS02
SB_01
SB_02
U70B
74LS02
CT_WPCT
plus cnt synch
U91
74LS273
plus in
1532
min cnt synch
U92
74LS273
min in
1533
U71A
U71B
U71C
U71D
74LS04
priority in
1536
U71E
74LS04
priority encoder
U93
74148
V23
5V
U70C
74LS02
counter address (H)
U89
74LS244
Read Bus
counter address (L)
U90
74LS244
U71F
74LS04
CT_RSTA
Busy Bus
Control Bus
RSCT
U201
74LS32
CLK1
PRIORITY COUNTER (CTR) #1



PRIORITY COUNTER
(CTR) #2


plus
in
1532
V21
5V
+V
R156
1k
U64A
74LS112
S4
NO PB
SET TIME2
CT_P2P
resetP
1538
TIME2
U63A
74LS08
V20
5V
+V
R157
1k
U64B
74LS112
S3
NO PB
SET TIME1
F10X
TIME1
U63B
74LS08
V19
5V
+V
R155
1k
U65A
74LS112
S2
NO PB
SET TIME3
F10X
TIME3
U63C
74LS08
V18
5V
+V
R154
1k
U65B
74LS112
S1
NO PB
SET TIME4
F10X
TIME4
U63D
74LS08
CT_RSTB
PRIORITY COUNTER
(CTR) #3

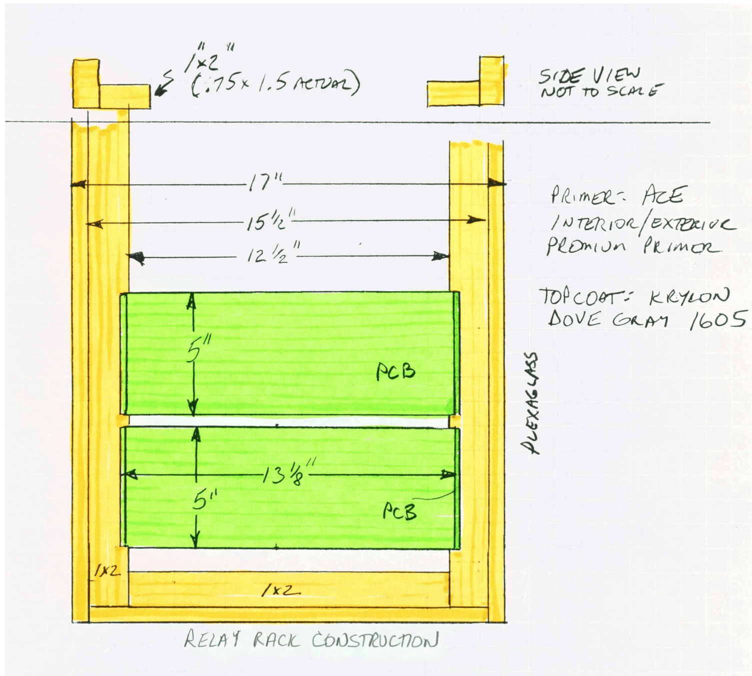


PRIORITY COUNTER (CTR) #4

Fabrication

The PROC module is (4) 13"x5" circuit boards, and 1 control panel.

Module Rack



The module framework is designed to resemble a relay rack, but scaled to fit the circuit board dimensions. It is constructed out of 1"x2" pine and spray-painted semi-gloss gray.

Circuit boards are mounted to the rack by 2 phillips screws at either end. Nylon spacers (1/4") are used as standoffs to hold the board edges above the rack. The boards are mounted so the chips are in the back and the pins are wiring are visible from the front.

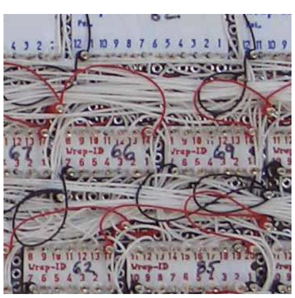
Power is distributed by 2 heavy aluminum bus bars mounted vertically, one per side, on the back of the module. Machine screws are mounted through the bus bars at evenly-spaced intervals to provide connection points for the boards.

Solid copper wire (24 gauge) connects the boards to the bus bars. Ring terminals are used on the bus bar side of the connection. On the circuit board size, the wires are soldered directly to the supply rails.

Materials were purchased from Home Depot, ACE Hardware, and Radio Shack.

Circuit Boards

The circuit boards are 13"x5" general purpose prototyping boards, epoxy glass with double-side plated through pads on 0.1" centers (JAMECO 21477CL).



ICs are mounted in level 3 machine tooled wire-wrap sockets: 8, 14, 16, 20, 24, and 28 pin (JAMECO). Each socket has the pin-out labeled with a wire-wrap socket ID marker, which slips onto the socket before wrapping (JAMECO). The part number is written onto the ID marker.

Sockets are arranged in 4 horizontal rows on each board, with about 10 sockets per row.

Power is distributed on the back-side of each board by bare 24-gauge solid copper wire supply rails soldered at equal intervals to Klipwrap terminals: 3-prong terminals with a square tail for wire-wrapping (JAMECO 34163CL). A +5V rail runs above each row of sockets and a ground rail runs below. Each rail connects directly to the aluminum module power bus using a ring tail connector.

On the pin side of the board, all connections are made with 30 AWG Kynar wire-wrap wire (JAMECO). Red wire is used for direct connections to the +5V supply rail. Black wire is used for direct connections to ground. White wire is used for everything else.

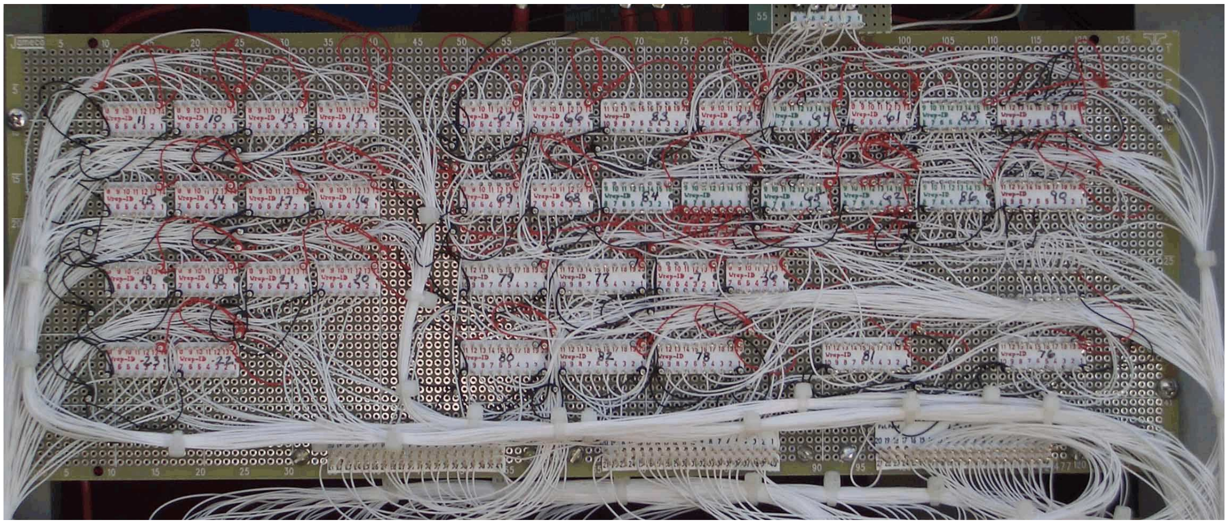
Power connections from the supply rails to each ICs are double-wrapped. Bypassing capacitors (.1 uf disc) are soldered across the supply rails at the Klipwrap terminals; about 1 capacitor for every 2 IC packages.

All connections were stripped and hand-wrapped using a Radio Shack hand-wrap tool. As each connection was made, the corresponding line on the schematic was marked with a colored highlighter.

DIP resistor networks (JAMECO) plugged into 20-pin wire-wrap sockets were used as current limiting resistors for the panel indicators.

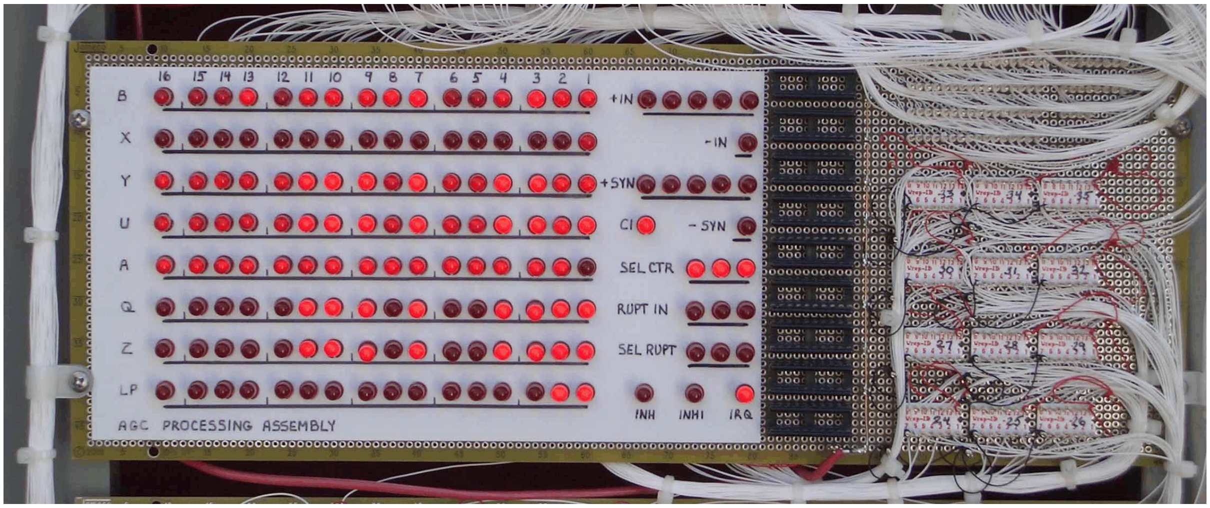
PROC Printed Circuit Board (PCB) A

The A board contains display drivers for the B board (left side), buffers for the interfaces to external modules (bottom right), and priority counter logic (upper right). Sockets for 3 IDE interface cables to external modules are visible at the bottom.



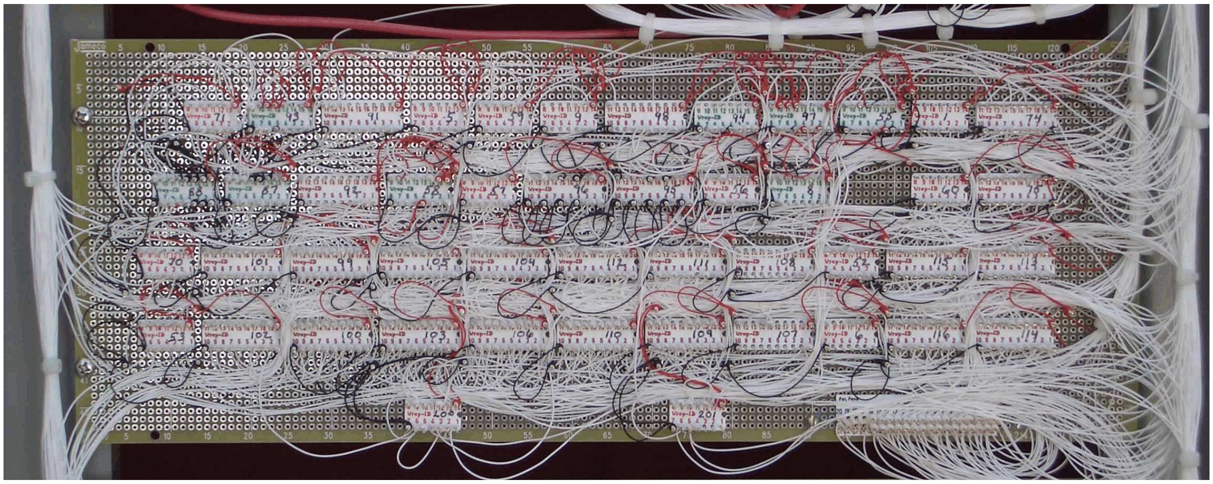
PROC Printed Circuit Board (PCB) B

The B board contains the display indicators, their current limiting resistor networks, and the open collector drivers. The display panel is a sheet of white styrene plastic. A push pin was used to make holes through the plastic and the LEDs were inserted in rows. The panel was hand-lettered with an indelible marker.



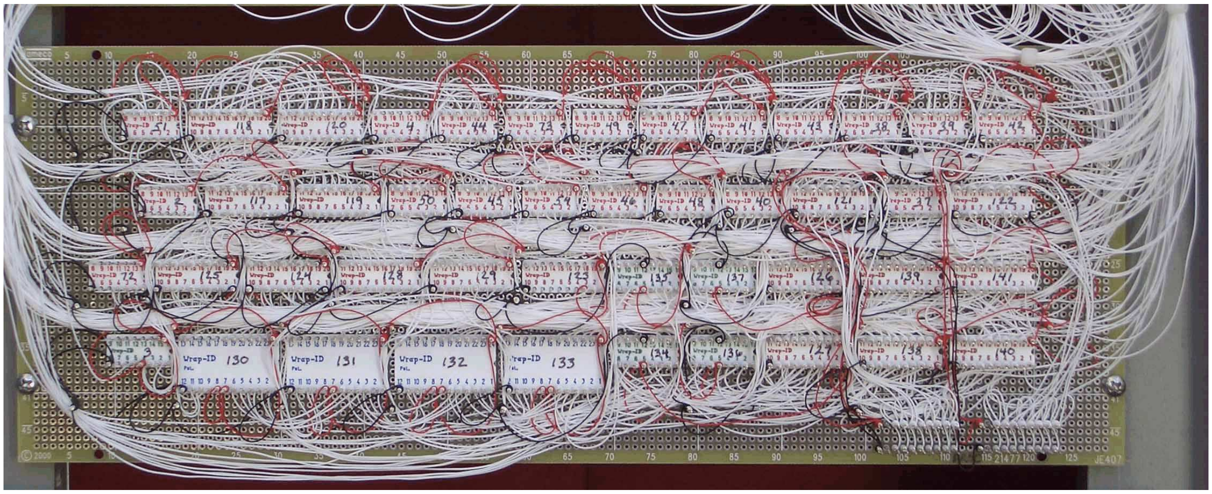
PROC Printed Circuit Board (PCB) C

The C board contains the the logic for the interrupt (INT) subsystem (upper half of the board), and the central registers (CRG; lower half of the board).



PROC Printed Circuit Board (PCB) D

The D board contains the ALU logic. The large 74181 ALU chips are at the bottom right. The four chips that form the ADDER are in the bottom half of the board, slightly to the right of the middle.



Parts (ICs)

IC's, sockets, PCB's, resistors, capacitors, wire-wrap wire were purchased from JAMECO. IDE wire-wrap sockets were from DigiKey. Wire ties, wire-wrap tools, and copper wire were from Radio Shack. IDE ribbon cables were purchased from an online computer supplier.

74LS00	(9)	U60,U66,U57,U53,U51,U44,U46,U47,U42
74LS02	(4)	U70,U6,U49,U2
74LS04	(11)	U54,U45,U37,U68,U7,U36,U69,U71,U59,U48,U72
74LS06	(26)	U28,U29,U30,U31,U27,U26,U23,U24,U25,U22,U21,U20,U18,U19,U17,U16,U15,U12,U13,U14,U11,U10,U35,U34,U33,U32
74LS08	(4)	U63,U61,U56,U52
74LS10	(2)	U50,U43
74LS20	(2)	U67,U73
74LS21	(1)	U1
74LS27	(1)	U9
74LS32	(5)	U4,U41,U40,U39,U38
74LS83	(4)	U134,U135,U136,U137
74LS86	(1)	U5
74LS112	(7)	U65,U64,U62,U58,U55,U8,U3
74LS138	(4)	U84,U85,U86,U94
74LS148	(2)	U93,U97
74LS151	(2)	U87,U88
74LS181	(4)	U130,U131,U132,U133
74LS244	(33)	U74,U75,U76,U77,U78,U79,U80,U81,U82,U83,U89,U90,U95,U96,U101,U102,U104,U105,U108,U111,U112,U115,U116,U117,U118,U119,U120,U121,U122,U124,U125,U128,U129
74LS273	(19)	U91,U92,U98,U99,U100,U103,U106,U107,U109,U110,U113,U114,U123,U126,U127,U138,U139,U140,U141

Power Budget

	qty	mA (ea)	mA (tot)
74LS00	9	2.4	21.6
74LS02	4	2.4	9.6
74LS04	11	3.6	39.6
74LS06	26	3.6	93.6
74LS08	4	4.4	17.6
74LS10	2	1.8	3.6
74LS20	2	1.2	2.4
74LS21	1	2.2	2.2
74LS27	1	3.4	3.4
74LS32	5	4.9	24.5
74LS83	4	22.0	88.0
74LS86	1	6.1	6.1
74LS112	7	4.0	28.0
74LS138	4	6.3	25.2
74LS148	2	12.0	24.0
74LS151	2	6.0	12.0
74LS181	4	21.0	84.0
74LS244	33	32.0	1056.0

74LS273	19	17.0	323.0
LED	153	20.0	3060.0

			4.9 Amps total
			1.9 Amps (excluding LEDs)