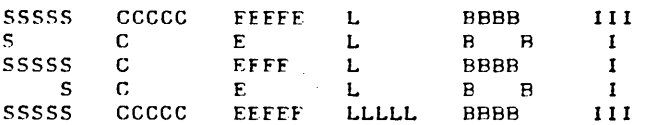


COMPUTER DIGEST

AND

USER'S BULLETIN

VOLUME I - ISSUE II

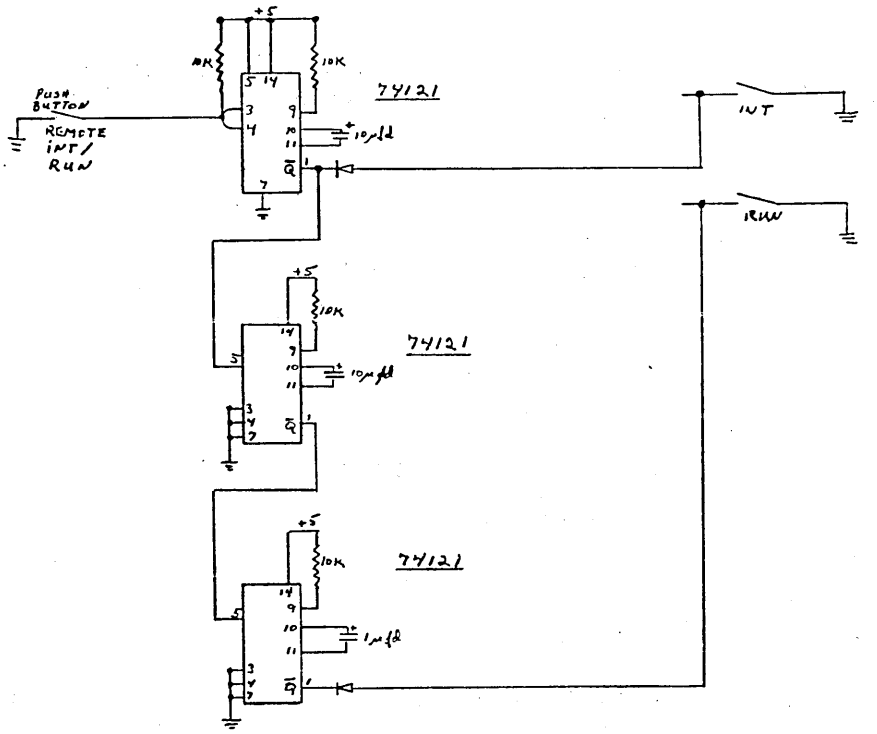
APRIL 1975

EDITOR: N. WADSWORTH

ASSISTANT EDITOR: R. FINDLEY

READERS WRITE

FROM THE APPEARANCE OF THE MAIL IN RESPONSE TO THE JANUARY, 1975, ISSUE OF "THE SCELBI COMPUTER DIGEST" IT SEEMS THE MOST POPULAR ARTICLE WAS "THE FAST MANUAL PROGRAM LOADER PERIPHERAL." A NUMBER OF READERS WROTE TO EXPRESS THEIR PLEASURE WITH THE IDEA AND SEVERAL PEOPLE INCLUDED IDEAS ON IMPROVING THE "FAST MANUAL LOADER."



MR. SOUTHARD'S "AUTO LOAD SYSTEM" CIRCUIT

ONE CONTRIBUTOR, MR. C. A. SOUTHARD, 2519 MEADOWBROOK DRIVE S.E., CEDAR RAPIDS, IOWA, 52403 SENT IN A VERY USEFUL CIRCUIT THAT HE TERMS AN "AUTO LOAD SYSTEM." THE CIRCUIT WHICH IS SHOWN ABOVE, ALLOWS ALL LOADING TO BE DONE FROM THE "REMOTE LOADER BOX." READERS WILL NOTE THAT THE CIRCUIT ELIMINATES THE REQUIREMENT OF HAVING TO PRESS THE "INT" AND THEN THE "RUN" SWITCH ON THE SCELBI-8H WHEN USING THE REMOTE LOADER SWITCHES THROUGH THE USE OF A "REMOTE INT/RUN" SWITCH. THE CIRCUIT AUTOMATICALLY PULSES THE "INT" AND "RUN" CONTROLS ON THE SCELBI-8H SO THAT WHATEVER INSTRUCTION SET UP ON THE CHASSIS TOGGLE SWITCHES - SUCH AS A "RESTART" (RST) WILL BE EXECUTED. THUS, IF THE LOADER PROGRAM GIVEN IN THE JANUARY ISSUE IS PLACED BEGINNING AT PG 00 LOCATION 000 (THE ADDRESS FOR AN RST 0 COMMAND) AND THE CHASSIS SWITCHES ARE SET TO 005 (THE MACHINE CODE FOR AN RST 0 INSTRUCTION) ONE NEEDS ONLY TO SET THE SWITCHES ON THE REMOTE BOX TO THE DESIRED "DATA" TO BE PLACED IN MEMORY AND HIT THE "INT/RUN" SWITCH CONNECTED TO MR. SOUTHARD'S CIRCUIT. (OF COURSE, REGISTERS

"H & L" MUST BE INITIALIZED BY THE USER.) NATURALLY, DIFFERENT TYPES OF ROUTINES COULD BE PLACED AT VARIOUS "RESTART" LOCATIONS AND THOSE ROUTINES EXECUTED BY SETTING THE APPROPRIATE "RST" CODE ON THE SCELBI-8H CHASSIS SWITCHES THEN OPERATING THE REMOTE "INT/RUN" SWITCH. USER'S MAY ESPECIALLY NOTE THAT THE CIRCUIT SUPPLIED BY MR. SOUTHARD COULD HAVE MANY APPLICATIONS BESIDES THE REMOTE PROGRAM LOADER. THE "REMOTE INT/RUN" SWITCH COULD BE REPLACED BY AN ELECTRONIC SWITCH AND THE BASIC CIRCUIT USED TO ALLOW ANY EXTERNAL DEVICE TO "INTERRUPT" THE COMPUTER AND EXECUTE THE COMMAND SET UP ON THE CHASSIS TOGGLE SWITCHES - "RST" TYPE INSTRUCTIONS ARE PARTICULARLY USEFUL IN SUCH APPLICATIONS AS ONE CAN PUT AN "INTERRUPT SERVICE" OR OTHER SUITABLE ROUTINE BEGINNING AT SUCH A LOCATION TO "PROCESS" THE DEVICE INITIATING THE INTERRUPT!

DR. GEORGE L. HALLER, (SUMMER ADDRESS STARTING MAY 15'TH: HOUND EARS CLUB, BLOWING ROCK, N.C. 28605) HAS DEVELOPED AN OCTAL READOUT DEVICE THAT CAN BE USED WITH "FAST LOADER" OR USED AS A MEMORY DUMP DISPLAY. THE CIRCUIT IS SHOWN ON THE NEXT PAGE AND THE COMMENTS FROM DR. HALLER AR AS FOLLOWS.

"AN OCTAL READOUT MAY BE VERY USEFUL IN SETTING UP PROGRAMS AND CHECKING THEM FOR POSSIBLE MISTAKES. THIS READOUT IS PLUGGED INTO ONE OF THE OUTPUTS OF THE COMPUTER AND WHEN GIVEN THE PROPER INSTRUCTION WILL DISPLAY IN THREE DIGITS, THE OCTAL VALUE OF THE "A" REGISTER. THE CONSTRUCTION IS VERY SIMPLE AS SHOWN IN THE DIAGRAM WHICH IS A MODIFICATION AND CORRECTION OF THE CIRCUIT SHOWN IN "POPULAR ELECTRONICS," DECEMBER 1974 PAGE 42. THE READOUT HAS BEEN BUILT ON A THREE BY THREE INCH CIRCUIT BOARD WHICH INCLUDES THE THREE DIGITS AND ENCLOSED IN A SMALL "BUD" METER TYPE BOX. THE IC'S INCLUDING THE "DATA LITES" WILL COST LESS THAN $7.50. THE ONLY OTHER ADDITION IS TO WIRE PLUS 5 VOLTS FROM THE SCELBI-8H TO PIN 10 ON THE OUTPUT SOCKET TO BE USED. ABOUT 300 MA. IS REQUIRED.

TO USE THE READOUT IN CONJUNCTION WITH THE FAST LOADER DESCRIBED IN THE FIRST COPY OF "THE SCELBI COMPUTER DIGEST" THE INSTRUCTION 131 IS ADDED TO THE SHORT LOADING PROGRAM (ASSUMING OUTPUT #14 IS USED FOR THE READOUT). THIS PROGRAM IS SHOWN IN TABLE A.

TABLE A

000-000	101
000-001	131
000-002	370
000-003	060
000-004	000

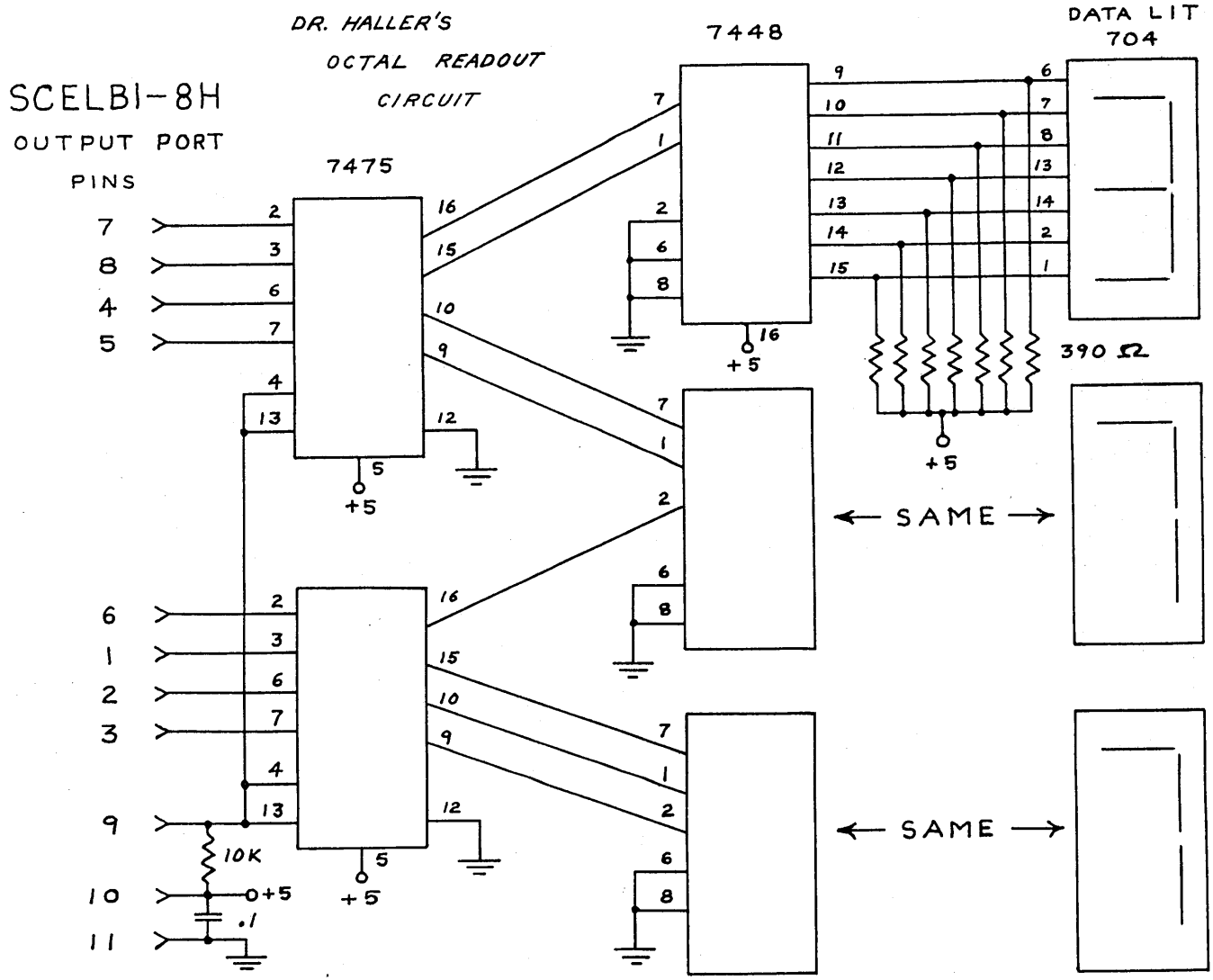
TABLE B

000-000	307
000-001	131
000-002	060
000-003	000

ANOTHER USE OF THE READOUT IS TO CHECK A PROGRAM WITHOUT STEPPING OR RUNNING THE ACTUAL PROGRAM. THIS IS QUITE USEFUL WHEN A PROGRAM HAS JUMPS OR CALLS OR TIMING LOOPS. FOR THIS PURPOSE USE THE PROGRAM SHOWN IN TABLE B. THE STARTING ADDRESS OF THE SEQUENCE TO BE EXAMINED, "H" AND "L," IS PUT INTO THE COMPUTER AND THE RESTART 005 IS USED IN THE REGULAR WAY OF INTERRUPT AND RUN.

IN ADDITION TO THESE TWO EXAMPLES THE EXAMINATION OF ANY REGISTER OR MEMORY CAN BE MADE BY LOADING INTO REGISTER "A" AND THEN OUTPUTTING IT."

DR. HALLER HAS ALSO DEVELOPED A "GOLF HANDICAP" PROGRAM WHICH HE DESCRIBES AS FOLLOWS.


DR. HALLER'S
OCTAL READOUT
CIRCUIT
SCELBI-8H
OUTPUT PORT
PINS
7475
7448
DATA LIT
704
390 Ω
+5
10K
.1
SAME
SAME

"THE SCORES ARE DEPOSITED BY MEANS OF THE ASCII KEYBOARD. THE READ-OUT IS ON A SMALL THREE DIGIT LED INDICATOR WHICH I BUILT FOR LESS THAN $10.00 IN PARTS. EACH SCORE AS INPUTTED IS INDICATED AND IF IT CHECKS IS DEPOSITED BY DEPRESSING THE "D" KEY. IF AN ERROR HAS BEEN MADE DEPRESSING THE "C" KEY WILL CANCEL THE ENTRY. AFTER TWENTY SCORES ARE ENTERED, THE "P" KEY IS DEPRESSED WHICH CAUSES THE COMPUTER TO PROCEED AND CALCULATE THE HANDICAP AND INDICATE IT ON THE INDICATOR. THE PROGRAM AVERAGES THE TEN LOWEST SCORES, SUBTRACTS A CONSTANT FOR THE GOLF COURSE RATING, MULTIPLIES THE REMAINDER BY 0.85 ROUNDS OFF THE NUMBER AND OUTPUTS IT."

ACCORDING TO DR. HALLER THE PROGRAM CONTAINS DOUBLE PRECISION ADDING, SUBTRACTING, MULTIPLICATION AND DIVISION, ROUND OFF, SORTING AND CHARACTER RECOGNITION. HE HAS INDICATED THAT HE WILL MAKE THE PROGRAM AVAILABLE TO INTERESTED "SCELBI COMPUTER DIGEST" READERS IF THEY WILL RE-IMBURSE HIM THE $2.00 REQUIRED TO DUPLICATE AND MAIL THE PROGRAM.

THANK YOU DR. HALLER, SOUNDS LIKE A NICE LITTLE PACKAGE FOR THOSE INTERESTED IN WORKING WITH MATHEMATICAL OPERATIONS!

A NUMBER OF READERS HAVE EXPRESSED AN INTEREST IN SEEING ARTICLES ON INTERFACING THE "TV-TYPEWRITER" WHICH HAS APPEARED IN "RADIO ELECTRONICS" TO THE SCELBI-8H. THERE ALSO APPEARS TO BE CONSIDERABLE INTEREST IN INTERFACING A CALCULATOR CHIP AS AN EXTERNAL "HARDWARE" MATHEMATICAL DEVICE. WE KNOW SEVERAL SCELBI-8H USER'S ARE WORKING ON SUCH PROJECTS AND WE HOPE THIS ANNOUNCEMENT THAT AN "INSTANT AUDIENCE" AWAITS THEIR RESULTS WILL SPUR THEM ON TO COMPLETE AND DESCRIBE THEIR WORKS.

WE ALSO KNOW THAT A NUMBER OF SCIENTIFIC USER'S HAVE INTERFACED D/A (DIGITAL TO ANALOG) AND A/D (ANALOG TO DIGITAL) CONVERTERS TO THE SCELBI-8H AND WE WANT TO LET THEM KNOW, SIMILARLY, THAT A GOOD NUMBER OF THEIR FELLOW COMPUTER USER'S WOULD LIKE TO SEE SOME ARTICLES ON SPECIFIC CIRCUITS AND APPLICATIONS. HERE'S A CHANCE FOR SOME OF YOU D/A AND A/D SPECIALISTS TO SHOW YOUR STUFF!

* *

- IN THIS ISSUE -

A DIGITAL (TTL) INTEGRATED CIRCUIT TESTER PERIPHERAL

A DIGITAL INTEGRATED CIRCUIT TESTER FOR TESTING TTL (TRANSISTOR-TRANSISTOR-LOGIC) DEVICES IS A USEFUL INSTRUMENT TO HAVE AROUND WHETHER YOUR A HOBBYIST WHO LIKES TO TAKE ADVANTAGE OF "SURPLUS" SALES AND NEEDS TO "WEED OUT" DEFECTIVE DEVICES OR A MANUFACTURER WHO WANTS TO PERFORM A QUICK FUNCTIONAL TEST ON INCOMING DEVICES BEFORE THEY ARE INSTALLED ON PRINTED CIRCUIT CARDS.

IF ONE LOOKS THROUGH THE ADVERTISEMENTS IN THE ELECTRONIC TRADE MAGAZINES TODAY ONE CAN SEE I.C. TESTERS SELLING FROM $500.00 TO SEVERAL THOUSANDS OF DOLLARS. HOWEVER, SCELBI-8H OWNERS CAN BUILD THE LITTLE I.C. TESTER PERIPHERAL DESCRIBED HERE FOR ABOUT $20.00 IF ALL THE PARTS ARE PURCHASED NEW. THE TESTER UTILIZES THE COMPUTER TO PROVIDE INPUTS TO THE DEVICE UNDER TEST AND SIMILARLY USES THE COMPUTER TO ASCERTAIN WHETHER THE DEVICE IS FUNCTIONING PROPERLY UNDER THE GIVEN INPUT CONDITIONS.

THE UNIT IS BASICALLY A SIMPLE "STATIC" TESTER. THAT IS, THE DEVICE DOES NOT TEST SUCH PARAMETERS AS RISE/FALL TIME OR PROPAGATION DELAYS. IT DOES, THOUGH, TEST FOR THE PROPER "LOGIC" OPERATION OF THE I.C. BEING TESTED FOR A SPECIFIC SET OF INPUT CONDITIONS. THIS TYPE OF TEST IS ADEQUATE FOR MANY APPLICATIONS AS IT DOES PROVIDE INFORMATION ABOUT THE GENERAL FUNCTIONAL CONDITION OF THE DEVICE.

THE TESTER, DRIVEN BY A SCELBI-8H COMPUTER, IS QUITE FAST. A TYPICAL INTEGRATED CIRCUIT CAN BE THOROUGHLY TESTED - AND CYCLED THROUGH THE TEST - SEVERAL HUNDRED TIMES IN WELL UNDER ONE SECOND. THUS ONE CAN TEST I.C.'S ABOUT AS FAST AS ONE CAN PLUG THEM INTO THE TEST SOCKET AND REMOVE THEM - WHICH CAN BE DONE AT A RATE OF 5 TO 6 PER MINUTE OR 300 TO 400 DEVICES PER HOUR.

A SCHEMATIC OF THE SCELBI I.C. TESTER IS SHOWN ON THE CENTER PAGES OF THIS ISSUE OF "THE SCELBI COMPUTER DIGEST." ONLY 10 7400 SERIES I.C. DEVICES ARE USED IN THE INSTRUMENT. FOUR 7475 "LATCHES" AND HALF A DOZEN TYPE 7417 "OPEN COLLECTOR" NON-INVERTING BUFFERS.

SIXTEEN SWITCHES, S1 - S16 ARE REQUIRED. THESE MAY BE SUBMINIATURE SLIDE OR TOGGLE SWITCHES. OR, SINCE THE SWITCHES ARE SPST, THOSE THAT CAN PROCURE THEM, MAY DESIRE TO USE THE NICE, SUPER-MINIATURE" SPST TYPE OF SWITCHES THAT ARE PROVIDED IN A PACKAGE RESEMBLING A 16 PIN INTEGRATED CIRCUIT, WHICH PROVIDE EIGHT SWITCHES TO A PACKAGE.

THE READER SHOULD CAREFULLY NOTE THE 16 TEST POINTS ON THE SCHEMATIC LABELED "TP1 - TP16." THESE TEST POINTS CAN BE MADE FROM "VECTOR" PINS OR STAKES, OR EVEN SHORT PIECES OF WIRE TO WHICH ONE MAY ATTACH AN ALLIGATOR CLIP OR TEST PROBE. THE TEST POINTS SERVE SEVERAL FUNCTIONS. FIRST, THEY ALLOW THE CONNECTION OF POWER TO THE PROPER PINS OF THE DEVICE IN THE TEST SOCKET. THIS ARRANGEMENT IS NECESSARY TO GIVE FLEXIBILITY AS DIFFERENT TYPES OF TTL I.C.'S REQUIRE POWER AT DIFFERENT PIN NUMBERS. THE READER MAY NOTE THE ALLIGATOR CLIP LEADS SHOWN ON THE SCHEMATIC COMING FROM "+5V" AND "GND." THOSE CLIP LEADS ARE USED TO CONNECT POWER TO THE PROPER "TEST POINTS" FOR THE DEVICE BEING TESTED.

THE SECOND REASON FOR PROVIDING THE TEST POINTS IS SO ONE CAN READILY CONNECT AN OSCILLOSCOPE OR METER TO VARIOUS PINS OF THE DEVICE UNDER TEST TO OBSERVE OPERATION OR MAKE ANALYSIS OF THE TYPE(S) OF FAILURE THAT MIGHT BE FOUND USING THE INSTRUMENT.

THE 16 PIN TEST SOCKET FOR THE I.C. TESTER SHOULD BE A GOOD QUALITY

SOCKET DESIGNED FOR MULTIPLE-INSERTION USE. D O N O T -- REPEAT -- D O N O T USE A REGULAR "DIP" SOCKET AS IT WILL BE RENDERED USELESS AFTER A FEW HUNDRED I.C. INSERTIONS AND THUS HAVE TO BE REPEATEDLY REPLACED.

LAYOUT OF THE CIRCUIT IS NOT CRITICAL. THE ENTIRE CIRCUITRY, INCLUDING THE SWITCHES AND THE I.C. TEST SOCKET, CAN BE MADE IN "BREADBOARD" FASHION ON A PIECE OF "VECTOR" BOARD OR SIMILAR MATERIAL. OR, IF ONE WANTS TO GET FANCY, A NICE ARRANGEMENT WITH THE SWITCHES AND TEST POINTS (INSULATED! FROM THE CHASSIS!) PLUS THE I.C. TEST SOCKET ON THE OUTSIDE OF A "MINI-BOX" AND THE CIRCUITRY INSIDE THE BOX CAN BE DEVISED. IT IS ADVISABLE TO PROVIDE SOCKETS FOR THE 7417 BUFFERS SO THAT THEY MAY BE REPLACED IN THE EVENT A "CATASTROPHIC" FAILURE OF A DEVICE UNDER TEST SHOULD DAMAGE A BUFFER.

IF ONE IS USING STANDARD SCELBI INTERFACE PROTOCOL, THE I/O CONNECTORS ON THE I.C. TESTER SHOULD BE 11 PIN MALE AMPHENOL TYPE 86CP11 UNITS TO MATE WITH STANDARD SCELBI I/O CABLES. POWER TO THE UNIT MAY BE RUN THROUGH SPARE CONNECTOR PINS, OR ONE MAY WISH TO PROVIDE SEPARATE CONNECTION POINTS BETWEEN THE TESTER AND THE POWER SUPPLY. NOTE, HOWEVER, THAT THE TESTER SHOULD BE SUPPLIED FROM THE SAME POWER SOURCE AS THE COMPUTER!

SET UP AND OPERATION OF THE TESTER

THE SET UP AND OPERATION OF THE TESTER IS STRAIGHT-FORWARD. THE INSTRUMENT IS CONNECTED TO TWO OUTPUT PORTS AND TWO INPUT PORTS ON THE SCELBI-8H. THE PORTS TO BE USED ARE SELECTED ACCORDING TO THOSE ASSIGNED IN THE "TEST PROGRAM." OUTPUT PORTS 10 & 11 AND INPUT PORTS 0 & 1 WERE USED IN THE DEMONSTRATION PROGRAMS THAT FOLLOW. POWER IS CONNECTED TO THE TESTER IF NOT PROVIDED FOR IN THE I/O CABLING.

NEXT, THE TYPE OF DEVICE TO BE TESTED IS SELECTED, AND AN APPROPRIATE PROGRAM (AS WILL BE DISCUSSED SHORTLY) IS LOADED INTO THE COMPUTER.

THEN, THE SWITCHES ON THE I.C. TESTER ARE SET ACCORDING TO THE FOLLOWING SIMPLE SCHEME: ALL SWITCHES GOING TO "INPUTS" OF THE DEVICE TO BE TESTED ARE SET TO THE "CLOSED" OR "ON" CONDITION. ALL OTHER SWITCHES ARE LEFT IN THE "OPEN" OR "OFF" CONDITION!

NEXT, THE CLIP LEADS THAT SUPPLY POWER TO THE DEVICE UNDER TEST ARE CONNECTED TO THE APPROPRIATE "TEST POINTS" TO PROVIDE +5 VOLTS AND GROUND TO THE DEVICE.

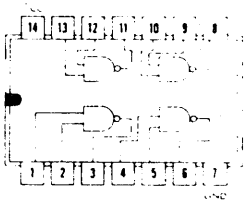
I M P O R T A N T N O T I C E

> IT IS EXTREMELY IMPORTANT TO ENSURE THAT THE SWITCHES GOING TO THE PINS OF THE TEST SOCKET CORRESPONDING TO THE POWER CONNECTIONS OF THE DEVICE UNDER TEST ARE IN THE "OPEN" OR "OFF" SETTING! FAILURE TO DO SO CAN RESULT IN A DRIVING BUFFER BEING DAMAGED BY APPLICATION OF POWER TO AN "OUTPUT" CONNECTION OF A BUFFER (PARTICULARLY IN REGARDS TO THE +5 VOLT LEAD)!

BE PARTICULARLY CAREFUL WHEN SETTING UP FOR TESTING A 14 PIN DEVICE IN THE 16 PIN TEST SOCKET AS PINS 8 THROUGH 14 OF THE DEVICE WILL NOW

BE INSERTED IN PINS 10 THROUGH 16 OF THE TEST SOCKET.

AS AN EXAMPLE OF SETTING UP THE SWITCHES, CONSIDER A DEVICE SUCH AS A TYPE 7400 QUAD "NAND" GATE DEVICE SHOWN IN THE DIAGRAM BELOW. PINS 1 & 2, 4 & 5, 9 & 10, AND 12 & 13 ARE INPUTS TO SUCH A PACKAGE. THUS, SWITCHES 1 & 2, 4 & 5, 11 & 12, AND 14 & 15 WOULD BE PLACED IN THE "ON" OR "CLOSED" STATE ON THE I.C. TESTER AND ALL OTHER SWITCHES WOULD BE LEFT IN THE "OPEN" OR "OFF" CONDITION.



POWER FOR THE 7400 DEVICE WOULD BE CONNECTED TO TEST POINT TERMINALS "TP7" (GND) AND "TP16" (+5 VOLTS). MAKE SURE SWITCHES "S7" AND "S16" ARE "OPEN!"

FINALLY, INSTALL A DEVICE TO BE TESTED AND START THE "TEST PROGRAM" ON THE COMPUTER. IN THE DEMONSTRATION PROGRAMS THAT FOLLOW, THE DEVICE THAT "PASSES" A TEST SERIES WILL RESULT IN THE SCELBI-8H COMING TO A HALT CONDITION WITH ALL THE MEMORY CONTENTS LAMPS OFF. A DEVICE THAT "FAILS" WILL RESULT IN THE COMPUTER HALTING WITH THE MEMORY CONTENTS LAMPS ALL TURNED "ON."

SAMPLE I.C. TESTER PROGRAMS

THE PRINCIPAL BEHIND PROGRAMS TO OPERATE THE I.C. PROGRAM ARE QUITE SIMPLE. SIGNALS ARE SENT FROM THE COMPUTER (VIA OUTPUT PORTS) TO DRIVE THE INPUTS OF THE DEVICE UNDER TEST TO DESIRED STATES, AND THEN OUTPUTS FROM THE DEVICE ARE SAMPLED BY THE COMPUTER (VIA INPUT PORTS) AND ANALYZED TO ASCERTAIN IF THEY ARE "CORRECT."

BY OBSERVING THE SCHEMATIC DIAGRAM OF THE I.C. TESTER ONE CAN SEE THAT THE OUTPUTS FROM COMPUTER PORT "A" DRIVE TEST SOCKETS PINS 1 - 8 WITH THE LINE CORRESPONDING TO THE MOST SIGNIFICANT BIT IN THE ACCUMULATOR DRIVING PIN 1 OF THE TEST SOCKET, AND THE LINE CORRESPONDING TO THE LEAST SIGNIFICANT BIT DRIVING PIN 8. SIMILARLY, PINS 9 - 16 OF THE TEST SOCKET ARE DRIVEN BY THE LINES OF OUTPUT PORT "B" WITH PIN 9 CORRESPONDING TO THE "MSB" OF THE ACCUMULATOR AND PIN 16 TO THE "LSB" OF THE ACCUMULATOR WHEN AN OUTPUT INSTRUCTION IS EXECUTED.

LIKEWISE, OUTPUTS FROM THE I.C. TEST SOCKET (AS WELL AS INPUTS!) GO TO INPUT PORTS IN THE SAME PATTERN. SINCE ALL PINS FROM THE I.C. TEST SOCKET GO TO THE COMPUTER, IT IS NECESSARY TO USE SOFTWARE TO ISOLATE THE PARTICULAR PINS OF INTEREST WHEN A DEVICE IS BEING TESTED. THIS CAN BE ACCOMPLISHED QUITE READILY FOR EITHER POSSIBLE SIGNAL STATE BY USING A "BOOLEAN AND" OPERATION OR A COMBINATION OF A "BOOLEAN AND" AND A "BOOLEAN EXCLUSIVE-OR" OPERATION AS CAN BE VERIFIED BY EXAMINING THE SAMPLE PROGRAM ILLUSTRATED ON THE NEXT SEVERAL PAGES. A COMPLETE LISTING OF THE PROGRAM WITH OBJECT CODE IS PROVIDED AS A READY-MADE PROGRAM WITH WHICH BUILDERS MAY INITIALLY TEST THEIR "I.C. TESTER."

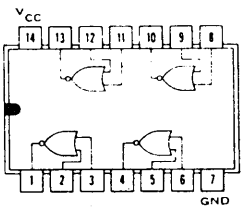
```
                    /7400 I.C. TEST

000 000  016   B7400, LBI 000  /SET TEST COUNTER
000 001  000
000 002  006   D7400, LAI 330  /SEND 11 TO GATES 1 AND 2
000 003  330
000 004  121   OUT 10
000 005  101   INP 0           /GET RESULTS
000 006  044   NDI 044         /TEST
000 007  044
000 010  110   JFZ BAD
000 011  134
000 012  000
000 013  006   LAI 066         /SEND 11 TO GATES 3 AND 4
000 014  066
000 015  123   OUT 11
000 016  103   INP 1           /GET RESULTS
000 017  044   NDI 110         /TEST
000 020  110
000 021  110   JFZ BAD
000 022  134
000 023  000
000 024  006   LAI 220         /SEND 10 TO GATES 1 AND 2
000 025  220
000 026  121   OUT 10
000 027  101   INP 0           /GET RESULTS
000 030  044   NDI 044
000 031  044
000 032  054   XRI 044         /TEST
000 033  044
000 034  110   JFZ BAD
000 035  134
000 036  000
000 037  006   LAI 044         /SEND 10 TO GATES 3 AND 4
000 040  044
000 041  123   OUT 11
000 042  103   INP 1           /GET RESULTS
000 043  044   NDI 110
000 044  110
000 045  054   XRI 110         /TEST
000 046  110
000 047  110   JFZ BAD
000 050  134
000 051  000
000 052  006   LAI 110         /SEND 01 TO GATES 1 AND 2
000 053  110
000 054  121   OUT 10
000 055  101   INP 0           /GET RESULTS
000 056  044   NDI 044
000 057  044
000 060  054   XRI 044         /TEST
000 061  044
000 062  110   JFZ BAD
000 063  134
000 064  000
000 065  006   LAI 022         /SEND 01 TO GATES 3 AND 4
000 066  022
000 067  123   OUT 11
000 070  103   INP 1           /GET RESULTS
000 071  044   NDI 110
000 072  110
```

```
000 073  054   XRI 110          /TEST
000 074  110
000 075  110   JFZ BAD
000 076  134
000 077  000
000 100  006   LAI 000          /SEND 00 TO GATES 1 AND 2
000 101  000
000 102  121   OUT 10
000 103  101   INP 0            /GET RESULTS
000 104  044   NDI 044
000 105  044
000 106  054   XRI 044          /TEST
000 107  044
000 110  110   JFZ BAD
000 111  134
000 112  000
000 113  123   OUT 11           /SEND 0 0 TO GATES 3 & 4
000 114  103   INP 1            /GET RESULTS (1)
000 115  044   NDI 110
000 116  110
000 117  054   XRI 110          /TEST
000 120  110
000 121  110   JFZ BAD
000 122  134
000 123  000
000 124  010   INB              /SEE IF TEST IS FINISHED
000 125  110   JFZ D7400
000 126  002
000 127  000
000 130  000   ALDONE, 000      /DUT PASSED - LIGHTS OUT
000 131  104   JMP B7400
000 132  000
000 133  000
000 134  377   BAD, 377         /DUT FAILED - LIGHTS ON
000 135  104   JMP B7400
000 136  000
000 137  000
```

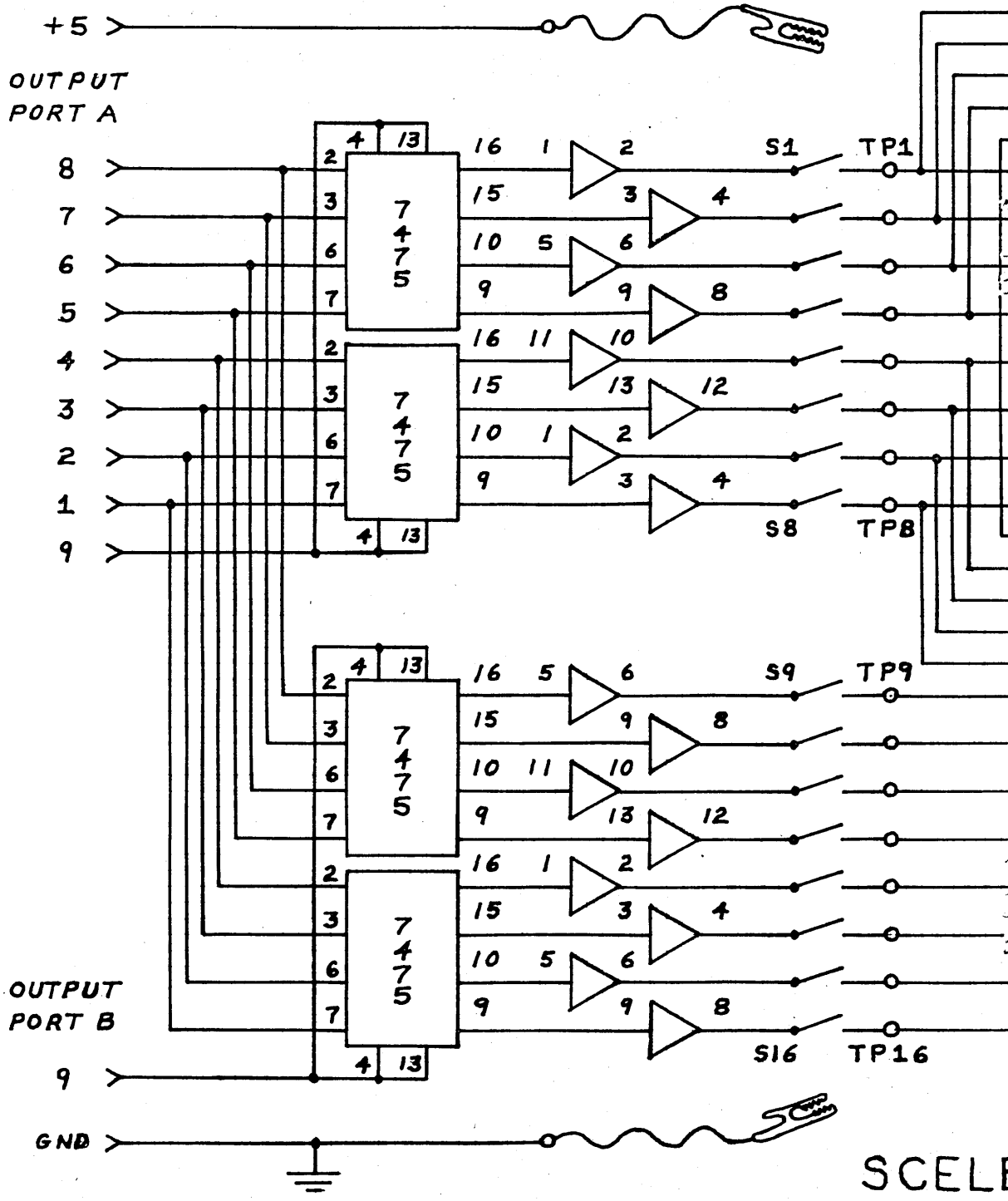
THE READER MIGHT NOTE THAT THE VERY SAME PROGRAM COULD BE USED TO TEST AN "OPEN COLLECTOR" EQUIVALENT OF THE 7400 I.C., SUCH AS A TYPE 7403, AND THAT THE DESIGN OF THE I.C. TESTER ALLOWS ONE TO TEST BOTH REGULAR TTL AND "OPEN COLLECTOR" TTL DEVICES.

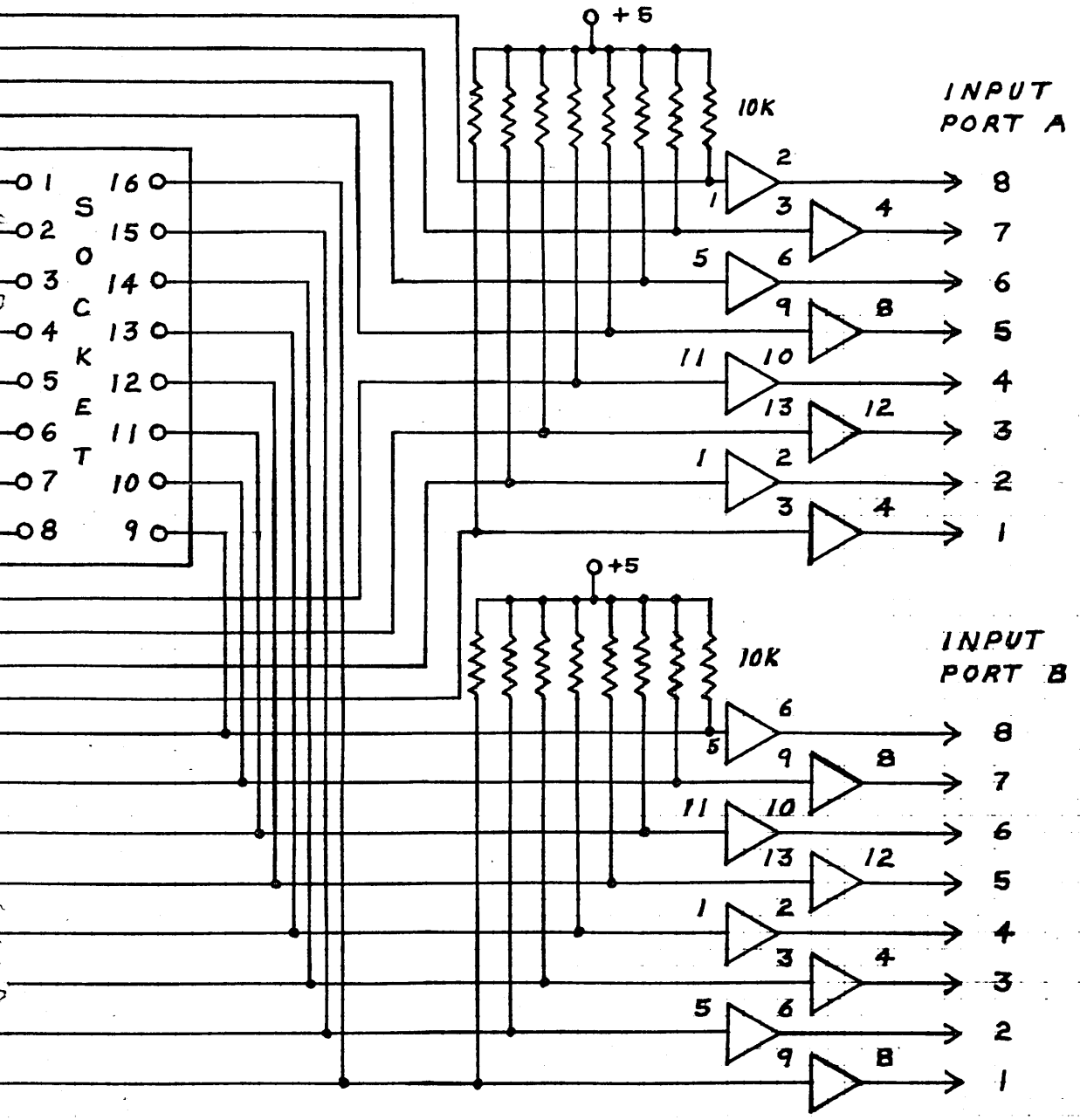
A PROGRAM FOR ANOTHER POPULAR TTL DEVICE, THE 7402 QUAD "NOR" PACKAGE IS ILLUSTRATED NEXT. THE DEVICE PIN ASSIGNMENT IS SHOWN BELOW.



THUS, SWITCHES 2 & 3, 5 & 6, 10 & 11 AND 13 & 14 WOULD BE PLACED IN THE "ON" CONDITION (CLOSED) ON THE TESTER AND ALL OTHER SWITCHES SET TO THE "OFF" (OPEN) CONDITION. THE POWER CONNECTIONS WOULD BE THE SAME AS FOR THE 7400 DEVICE. WITH A 7402 DEVICE IN THE TEST SOCKET THE FOLLOWING PROGRAM WOULD BE RUN TO TEST THE DEVICE.

```
                    /7402 I.C. TEST
                    /
                    /
                    ORG 000 140
000 140  016        B7402, LBI 000   /SET TEST CNTR
000 141  000
000 142  006        D7402, LAI 154   /SEND  1 1  TO GATES 1 & 2
000 143  154
000 144  121        OUT 10
000 145  101        INP 0            /GETS RESULTS  (0)
000 146  044        NDI 220          /TEST
000 147  220
000 150  110        JFZ BAD
000 151  263
000 152  000
000 153  006        LAI 154          /SEND  1 1  TO GATES 3 & 4
000 154  154
000 155  123        OUT 11
000 156  103        INP 1            /GET RESULTS  (0)
000 157  044        NDI 022          /TEST
000 160  022
000 161  110        JFZ BAD
000 162  263
000 163  000
000 164  006        LAI 110          /SEND  1 0  TO GATES 1 & 2
000 165  110
000 166  121        OUT 10
000 167  101        INP 0            /GET RESULTS  (0)
000 170  044        NDI 220          /TEST
000 171  220
000 172  110        JFZ BAD
000 173  263
000 174  000
000 175  006        LAI 110          /SEND  1 0  TO GATES 3 & 4
000 176  110
000 177  123        OUT 11
000 200  103        INP 1            /GET RESULTS  (0)
000 201  044        NDI 022          /TEST
000 202  022
000 203  110        JFZ BAD
000 204  263
000 205  000
000 206  006        LAI 044          /SEND  0 1  TO GATES 1 & 2
000 207  044
000 210  121        OUT 10
000 211  101        INP 0            /GET RESULTS  (0)
000 212  044        NDI 220          /TEST
000 213  220
000 214  110        JFZ BAD
000 215  263
000 216  000
```


+5
OUTPUT PORT A
8
7
6
5
4
3
2
1
9
7475
7475
7475
7475
S1
TP1
S8
TP8
S9
TP9
S16
TP16
OUTPUT PORT B
9
GND



I.C. POWER: 7475: +5V PIN 5, GND PIN 12
7417: +5V PIN 14, GND PIN 7

BI I.C. TESTER

```
000 217  006   LAI 044          /SEND  0 1  TO GATES 3 & 4
000 220  044
000 221  103   INP 1            /GET RESULTS  (0)
000 222  044   NDI 022          /TEST
000 223  022
000 224  110   JFZ BAD
000 225  263
000 226  000
000 227  006   LAI 0            /SEND  0 0  TO GATES 1 & 2
000 230  000
000 231  121   OUT 10
000 232  101   INP 0            /GET RESULTS  (1)
000 233  044   NDI 220
000 234  220
000 235  054   XRI 220          /TEST
000 236  220
000 237  110   JFZ BAD
000 240  263
000 241  000
000 242  123   OUT 11           /SEND  0 0  TO GATES 3 & 4
000 243  103   INP 1            /GET RESULTS  (1)
000 244  044   NDI 022
000 245  022
000 246  054   XRI 022          /TEST
000 247  022
000 250  110   JFZ BAD
000 251  263
000 252  000
000 253  010   INB              /SEE IF TEST IS FINISHED
000 254  110   JFZ D7402
000 255  142
000 256  000
000 257  000   ALDONE, 0        /DUT PASSED - LIGHTS OFF
000 260  104   JMP B7402
000 261  140
000 262  000
000 263  377   BAD, 377         /DUT FAILED - LIGHTS ON
000 264  104   JMP B7402
000 265  140
000 266  000
              /
```

ONE CAN DEVELOP PROGRAMS FOR FAR MORE COMPLEX I.C.'S USING THE SAME PRINCIPALS AS FOR THE NAND AND NOR GATES PROGRAMS ILLUSTRATED. THE NEXT ISSUE WILL PROVIDE SEVERAL MORE SAMPLE PROGRAMS INCLUDING AN ILLUSTRATIVE ONE FOR A MORE COMPLEX DEVICE SUCH AS A "DUAL FLIP-FLOP." BY THAT TIME INTERESTED READERS SHOULD HAVE THEIR TESTERS CONSTRUCTED AND READY TO USE. IF USER'S CONTRIBUTE PROGRAMS FOR DEVICES, AND READER INTEREST WARRANTS IT, WE WILL PUBLISH PROGRAMS ON A REGULAR BASIS SO THAT USER'S CAN BUILD UP A LIBRARY OF SUITABLE I.C. TEST PROGRAMS. WHAT SAY?

AN ASSEMBLER PROGRAM IS A POWERFUL TOOL THAT RELIEVES THE PROGRAMMER OF MUCH OF THE TEDIOUS DETAIL WORK INVOLVED WITH DEVELOPING A MACHINE LANGUAGE PROGRAM. AS USER'S KNOW, BEFORE A COMPUTER CAN PERFORM A TASK, A PROGRAM MUST BE PLACED IN MEMORY - THAT IS, THE MACHINE CODE FOR EACH TYPE OF INSTRUCTION TO BE PERFORMED MUST BE PLACED IN THE PROPER SEQUENCE(S) IN THE PROPER ADDRESS(ES) IN MEMORY. THE MACHINE CODES ARE THE ACTUAL PATTERNS OF ONES AND ZEROES WITHIN A "WORD." MOST SCELBI-8H USER'S SOON LEARN TO THINK OF THE MACHINE CODES IN THEIR OCTAL EQUIVALENT FORM, BUT THEY REALIZE THAT THESE CODES ARE A SHORTHAND FORM OF THE ACTUAL BINARY NUMBERS UTILIZED BY THE COMPUTER.

HOWEVER, AS MOST READERS ALSO KNOW, TRYING TO REMEMBER THE MACHINE CODES, EVEN IN THE SHORTHAND OCTAL REPRESENTATION, FOR ALL OF THE 170 PLUS TYPES OF INSTRUCTIONS THAT A SCELBI-8H CAN EXECUTE, IS A SOMEWHAT FORMIDABLE TASK. SO MUCH SO, THAT PEOPLE GENERALLY REVERT TO USING A SYMBOLIC NAME FOR A PARTICULAR TYPE OF INSTRUCTION WHEN "THINKING" ABOUT A PARTICULAR FUNCTION. THESE SYMBOLIC NAMES FOR INSTRUCTIONS ARE REFERRED TO BY COMPUTER TECHNOLOGIST AS "MNEMONICS." IT IS A GOOD NAME BECAUSE THE WORD "MNEMONIC" ACTUALLY MEANS "TO ASSIST THE MEMORY" WHICH IS PRECISELY WHAT ONE IS DOING WHEN ONE THINKS OF AN INSTRUCTION AS BEING AN EASILY REMEMBERED "NAME" RATHER THAN AN ABSTRACT SERIES OF DIGITS. SCELBI-8H OPERATORS ARE FAMILIAR WITH MOST OF THE MNEMONICS USED TO REPRESENT THE VARIOUS TYPES OF INSTRUCTIONS AS THEY ARE PRESENTED TO USERS IN DETAIL IN THE INSTRUCTION MANUAL FOR THE COMPUTER.

BUT, WHILE MNEMONICS ARE MOST USEFUL FOR A HUMAN TO REMEMBER AND WORK WITH AS A PROGRAM IS "THOUGHT UP" AND LOGICALLY DESIGNED, AT SOME POINT, WHEN THE PROGRAMMER IS READY TO ACTUALLY PLACE A PROGRAM INTO THE COMPUTER AND HAVE IT OPERATE, THE MNEMONIC REPRESENTATIONS MUST BE CONVERTED TO THE ACTUAL MACHINE CODE NUMBERS UTILIZED BY THE COMPUTER. FOR SMALL PROGRAMS, IT IS A RELATIVELY EASY MATTER FOR A HUMAN TO "LOOK UP" THE MACHINE CODE FOR EACH TYPE OF MNEMONIC USED IN THE PROGRAM AND THEREBY PRODUCE A LIST OF THE CODES TO BE PLACED AT SELECTED ADDRESSES IN THE COMPUTER'S MEMORY. OF COURSE, IT IS ALSO NECESSARY TO KEEP TRACK OF WHERE VARIOUS PORTIONS OF THE PROGRAM ARE PLACED IN MEMORY (THE ADDRESSES) BECAUSE SOME TYPES OF INSTRUCTIONS SUCH AS "JUMP" AND "CALL" INSTRUCTIONS USE THESE VALUES WITHIN THE INSTRUCTION. HOWEVER, FOR RELATIVELY SMALL PROGRAMS, THIS TASK CAN BE HANDLED IF ONE KEEPS THINGS (ESPECIALLY ONE'S THOUGHTS!) ORGANIZED AS THE JOB IS DONE.

BY SMALL PROGRAM, WHAT IS MEANT IS ONE LIMITED TO SAY, LESS THAN ABOUT ONE HUNDRED INSTRUCTIONS. FOR, AS A PROGRAM GETS LARGE, SEVERAL FACTORS BEGIN TO COMPLICATE THE PROCESS. FIRST OF ALL, THE ACTUAL TASK OF LOOKING UP THE MACHINE CODE FOR EACH INSTRUCTION BEGINS TO BECOME SOMEWHAT TEDIOUS, IF NOT DOWN-RIGHT BORING. SECONDLY, AS THE SIZE OF THE PROGRAM GROWS, IT BECOMES MORE AND MORE DIFFICULT FOR THE PROGRAMMER TO KEEP TRACK OF THE ADDRESSES OF ROUTINES OR SUBROUTINES THAT JUMP OR CALL INSTRUCTIONS MIGHT REFER TO. AND THIRD, BUT BY NO MEANS LEAST, THE POSSIBILITY FOR THE PROGRAMMER TO MAKE AN ERROR INCREASES - AND ERROR'S IN PROGRAMS ARE NO LITTLE MATTER! PARTICULARLY IF THE ERROR IS ONE THAT EVENTUALLY RESULTS IN A LARGE BLOCK OF MACHINE CODE HAVING TO BE MOVED (EVEN ONE LITTLE ADDRESS LOCATION!) AND THAT IN TURN RESULTS IN MEMORY ADDRESS REFERENCE INSTRUCTIONS (THOSE JUMP AND CALL TYPES) HAVING TO BE MODIFIED TO ACCOUNT FOR THE NEW LOCATION OF THE START OF A ROUTINE OR SUBROUTINE!

THESE TYPES OF PROBLEMS ASSOCIATED WITH MANUALLY CONVERTING A MNEMONIC LISTING (GENERALLY TERMED THE "SOURCE LISTING") TO MACHINE LAN-

GUAGE CODE (TERMED THE "OBJECT CODE") CAN BE AVOIDED BY THE USE OF AN ASSEMBLER PROGRAM!

FOR, AN ASSEMBLER PROGRAM IS A PROGRAM ITSELF THAT DIRECTS THE COMPUTER TO DO THE "DIRTY WORK" OF CONVERTING A SOURCE LISTING (MADE USING MNEMONICS) TO OBJECT (MACHINE LANGUAGE) CODE! THE ASSEMBLER PROGRAM ALSO TAKES CARE OF ASSIGNING THE CODE TO ADDRESS LOCATIONS IN MEMORY (AS ORIGINALLY DIRECTED BY THE PROGRAMMER AS WILL BE EXPLAINED SHORTLY) AND OF AUTOMATICALLY KEEPING TRACK OF THE STARTING ADDRESSES OF SUBROUTINES AND ROUTINES, THROUGH A PROCESS REFERRED TO AS "LABELING" SO THAT INSTRUCTIONS THAT REFER TO SUCH ROUTINES WILL AUTOMATICALLY HAVE THE CORRECT ADDRESS PLACED IN THE "ADDRESS BYTES" OF THE INSTRUCTION.

AN ASSEMBLER PROGRAM CAN BE THOUGHT OF AS CONSISTING OF SEVERAL MAJOR FUNCTIONAL BLOCKS. ONE SECTION OF AN ASSEMBLER PROGRAM CONSIST ESSENTIALLY OF A LARGE "LOOK UP" TABLE THAT CONTAINS ALL THE POSSIBLE MNEMONICS ALONG WITH THEIR MACHINE CODE REPRESENTATION. THUS, WHENEVER THE ASSEMBLER PROGRAM FINDS A "MNEMONIC" IT SIMPLY GOES TO THE MNEMONIC CONVERSION TABLE, AND LOCATES THE MACHINE CODE EQUIVALENT. IT DOES THAT A LOT FASTER AND A LOT MORE ACCURATELY THAN ANY HUMAN CAN DOT IT! NEXT, AN ASSEMBLER PROGRAM IS ABLE TO ACTUALLY "BUILD UP" A TABLE GENERALLY CALLED A "SYMBOL" TABLE, AND THEN USE THE SYMBOL TABLE IT HAS BUILT UP TO ASSIGN ADDRESSES TO ROUTINES AND SUBROUTINES. A "SYMBOL TABLE" IS BUILT UP BY THE PROGRAM LOOKING FOR "LABELS" ASSIGNED BY THE PROGRAMMER TO INDICATE THE START OF A ROUTINE OR SUBROUTINE IN THE SOURCE LISTING. WHENEVER A "LABEL" IS FOUND, THE ASSEMBLER SAVES THE "LABEL" AND AN ACTUAL "ADDRESS" ASSOCIATED WITH THAT LABEL. (THE LABEL-ADDRESS ASSOCIATION WILL BE EXPLAINED SHORTLY.) THEN, WHENEVER A "LABEL" IS USED BY THE PROGRAMMER AS AN "ADDRESS" PORTION OF A JUMP OR CALL INSTRUCTION, THE ASSEMBLER PROGRAM SIMPLY GOES BACK TO THE "SYMBOL TABLE" AND EXTRACTS THE CORRESPONDING ADDRESS. THUS, THE ASSEMBLER RELIEVES THE PROGRAMMER OF THE DUTY OF HAVING TO KEEP TRACK OF THE ACTUAL ADDRESSES OF ROUTINES THAT ARE REFERRED TO BY OTHER INSTRUCTIONS. ADDITIONALLY, AN ASSEMBLER PROGRAM MAINTAINS WHAT IS SOMETIMES TERMED A "PSUEDO-PROGRAM COUNTER." THE "PSUEDO-PROGRAM COUNTER" KEEPS TRACK OF WHERE EACH INSTRUCTION WILL BE STORED IN MEMORY - AND TAKES ACCOUNT OF MULTIPLE-BYTE INSTRUCTIONS. THE "PSUEDO-PROGRAM COUNTER" IS THUS USED IN CONJUNCTION WITH "LABELED INSTRUCTIONS" TO PLACE THE EFFECTIVE ADDRESS OF THE START OF ROUTINES IN THE ADDRESS PORTION OF THE "SYMBOL TABLE" PREVIOUSLY MENTIONED!

THUS, AN ASSEMBLER PROGRAM BASICALLY IS ABLE TO "READ" A "SOURCE LISTING" WRITTEN WITH MNEMONICS AND CONVERT THE LISTING WITH GREAT SPEED AND ACCURACY, TO THE ACTUAL OBJECT CODE USED BY THE COMPUTER. IN ADDITION TO READING THE MNEMONIC LISTING, HOWEVER, AN ASSEMBLER PROGRAM MUST ALSO BE ABLE TO INTERPRET SEVERAL "COMMANDS" GIVEN BY THE PROGRAMMER WHICH ARE CALLED "PSUEDO-OPERATORS." "PSUEDO-OPERATORS" ARE COMMANDS INSERTED IN A SOURCE LISTING THAT DIRECT THE ASSEMBLER PROGRAM TO PERFORM A FUNCTION, BUT THAT ARE NOT A PART OF THE ACTUAL PROGRAM BEING ASSEMBLED. AN ASSEMBLER PROGRAM MAY HAVE A VARIETY OF "PSUEDO-OPERATOR" COMMANDS DEPENDING ON THE SOPHISTICATION OF THE ASSEMBLER. FOR INSTANCE, SOME ASSEMBLERS HAVE "MACRO" OPERATORS THAT ALLOW A PROGRAMMER TO REFER TO A SERIES OF INSTRUCTIONS BY "MAKING UP" A NEW MNEMONIC NAME. SUCH AN ASSEMBLER CAN THEN INSERT THE PREVIOUSLY DEFINED SERIES OF INSTRUCTION WHENEVER IT FINDS A "MACRO" OPERATOR. SOME ASSEMBLERS ALSO HAVE "PSUEDO-OPERATORS" THAT PERMIT THE PROGRAMMER TO DEFINE NEW MNEMONIC NAMES. VIRTUALLY ALL ASSEMBLERS HAVE AT LEAST TWO "PSUEDO-OPERATORS" THAT ARE FUNCTIONALLY EQUIVALENT TO THE FOLLOWING. AN "ORG" PSUEDO-OPERATOR WHICH ENABLES THE PROGRAMMER TO SET THE VALUE OF THE ASSEMBLER'S "PSUEDO-PROGRAM COUNTER" BY ASSIGNING AN ACTUAL ADDRESS AFTER THE "ORG" COMMMAND, SUCH AS, "ORG 001 200" WHICH WOULD DIRECT THE AS-

SEMBLER TO START ASSEMBLING CODE BEGINNING WITH AN ADDRESS LOCATION OF PAGE 01 LOCATION 200 IN A SCELBI-8H SYSTEM. AND, AN "END" PSUEDO-OPERATOR WHICH INDICATES TO THE ASSEMBLER THAT IT IS "FINISHED" ASSEMBLING A PROGRAM.

AN ASSEMBLER, HOWEVER, GENERALLY DOES MORE TO EASE THE TASK OF DEVELOPING PROGRAMS THAN SIMPLY CONVERTING THE MNEMONIC LISTING TO MACHINE CODE. GENERALLY, IT ALLOWS A "SYSTEM" TO BE IMPLEMENTED THAT GREATLY SPEEDS UP THE OVER-ALL PROGRAM DEVELOPMENT PROCESS. FOR INSTANCE, IN THE LAST ISSUE OF "THE SCELBI COMPUTER DIGEST" MENTION WAS MADE OF HOW AN "EDITOR PROGRAM" COULD PRODUCE A PUNCHED PAPER TAPE OR A MAGNETIC TAPE OF A SOURCE LISTING. SUCH A TAPE CAN THEN BE PROCESSED DIRECTLY BY AN ASSEMBLER PROGRAM, THUS ELIMINATING A LOT OF PENCIL AND PAPER WORK. SUCH A SYSTEM GREATLY REDUCES THE AMOUNT OF WORK IN PROGRAM DEVELOPMENT PARTICULARLY BECAUSE, IF THE PROGRAM NEEDS TO BE REVISED OR UPDATED, ONE CAN SIMPLY RELOAD THE TAPE BACK INTO THE COMPUTER AND USE THE EDITOR PROGRAM TO REVISE THE SOURCE LISTING, THEN QUICKLY AND EASILY, RUN THE REVISED SOURCE LISTING BACK THROUGH THE ASSEMBLER PROGRAM TO OBTAIN THE REVISED MACHINE CODE. FURTHERMORE, AN ASSEMBLER PROGRAM WILL GENERALLY HAVE A MEANS OF LOADING THE OBJECT CODE IT PRODUCES DIRECTLY INTO MEMORY, OR SIMILAR TO THE EDITOR PROGRAM, BE ABLE TO PRODUCE A TAPE OF THE OBJECT (MACHINE LANGUAGE) CODE THAT CAN BE SAVED PERMANENTLY AND RELOADED INTO THE COMPUTER WHENEVER IT IS DESIRED TO OPERATE THE PROGRAM.

TYPICAL ASSEMBLER PROGRAM OPERATION

OPERATION OF AN ASSEMBLER PROGRAM IS SIMPLICITY ITSELF. A TYPICAL PROGRAM SUCH AS THE SMALL EXAMPLE PROVIDED BELOW IS READ IN BY THE ASSEMBLER PROGRAM FROM PUNCHED PAPER TAPE OR A MAGNETIC TAPE.

```
        ORG 001 200 /"ORG" PSUEDO-OPERATOR
START,  LHI 003     /SET PAGE ADDRESS POINTER
        LLI 000     /SET LOCATION ON PG POINTER
        XRA         /SET ACCUMULATOR TO 000
AGAIN,  LMA         /LOAD CONTENTS OF ACC INTO MEMORY
        INL         /ADVANCE MEMORY POINTER
        JFZ AGAIN   /JUMP TO LABEL "AGAIN" IF NOT DONE
        HLT         /END OF PROGRAM
        END         /"END" PSUEDO-OPERATOR FOR ASSEMBLER
```

THE ABOVE SAMPLE PROGRAM (THAT SETS ALL LOCATIONS ON PAGE 03 TO CONTAIN 000) ILLUSTRATES THE USE OF "PSUEDO-OPERATORS" AND LABELS SUCH AS "START" AND "AGAIN." IN A TYPICAL ASSEMBLER, THE ABOVE SOURCE LISTING WOULD FIRST BE READ BY THE ASSEMBLER TO BUILD UP A "SYMBOL" TABLE. PROCESSING THE SOURCE LISTING IN SEVERAL "PASSES" IS GENERALLY REQUIRED BECAUSE IF A "LABEL" IS REFERRED TO BY AN INSTRUCTION BEFORE IT HAS BEEN ASSIGNED AN ADDRESS, THE SYMBOL TABLE IN THE ASSEMBLER PROGRAM WOULD NOT CONTAIN THE ADDRESS. THUS, IT IS NECESSARY TO "DEFINE" THE SYMBOL TABLE BEFORE THE ASSEMBLER STARTS PRODUCING THE ACTUAL "OBJECT CODE."

ONCE THE "FIRST PASS" OF THE SOURCE LISTING HAD BEEN PROCESSED AND THE SYMBOL TABLE COMPILED, THE SOURCE TAPE WOULD BE RE-READ AND THE ASSEMBLER COULD PRODUCE THE OBJECT CODE. TYPICALLY, THE OBJECT CODE WOULD BE PUNCHED OUT ON PAPER TAPE OR PLACED ON MAGNETIC TAPE AS IT WAS PRODUCED. THE OBJECT CODE IS GENERALLY PRODUCED IN A FORMAT THAT WILL ENABLE A "LOADER" PROGRAM TO BE USED TO LOAD THE ACTUAL PROGRAM PRODUCED BACK INTO THE COMPUTER'S MEMORY FOR OPERATION. SUCH A FORMAT ALLOWS

THE LOADER PROGRAM TO START PLACING CODE IN THE MEMORY ADDRESS SPECIFIED BY AN "ORG" PSUEDO-OPERATOR.

AN ASSEMBLER PROGRAM THAT OPERATES IN THE ABOVE FASHION IS OFTEN TERMED A "TWO-PASS ASSEMBLER" AS IT REQUIRES A MINIMUM OF TWO PASSES TO PRODUCE COMPLETE OBJECT CODE.

MOST ASSEMBLER PROGRAMS ALSO ALLOW AN OPTIONAL ADDITIONAL "PASS" TO BE MADE THAT WILL PRESENT A COMPLETE LISTING OF BOTH THE ORIGINAL SYMBOLIC SOURCE CODE ALONGSIDE THE COMPLETED OBJECT CODE OF THE PROGRAM. SUCH A LISTING IS EXTREMELY VALUABLE AS A TESTING AND DEBUGGING AID AS IT SHOWS THE PROGRAMMER PRECISELY WHERE EACH INSTRUCTION IS LOCATED AND THE MACHINE CODE FOR EACH INSTRUCTION. IN ADDITION, SOME ASSEMBLERS ALLOW THE PROGRAMMER TO OBTAIN A LISTING OF THE "SYMBOL TABLE" THAT WAS BUILT UP BY THE ASSEMBLER PROGRAM AS A CONVENIENT REFERENCE TO THE STARTING ADDRESSES OF ALL LABELED ROUTINES OR SUBROUTINES. SUCH A LISTING SHOWS THE SYMBOLIC "LABEL" ALONG WITH THE "ADDRESS" TO WHICH THE LABEL REFERS. FOR, IN ESSENCE, ALL A LABEL DOES IS ASSIGN A REFERENCE "ADDRESS" TO A PARTICULAR INSTRUCTION. AN EXAMPLE OF A THIRD PASS LISTING AND A "SYMBOL TABLE" LISTING FOR THE EXAMPLE PROGRAM IS ILLUSTRATED BELOW.

```
                                ORG 001 200 /"ORG" PSUEDO-OPERATOR
001 200   056   START,  LHI 003      /SET PAGE ADDRESS POINTER
001 201   003
001 202   066          LLI 000      /SET LOCATION ON PG POINTER
001 203   000
001 204   250          XRA          /SET ACCUMULATOR TO 000
001 205   370   AGAIN, LMA          /LOAD CONTENTS OF ACC INTO MEMORY
001 206   060          INL          /ADVANCE MEMORY POINTER
001 207   110          JFZ AGAIN    /JUMP TO LABEL "AGAIN" IF NOT DONE
001 210   205
001 211   001
001 212   000          HLT          /END OF PROGRAM
                       END          /"END" PSUEDO-OPERATOR FOR ASSEMBLER

                       START:  001 200
                       AGAIN:  001 205
```

ONE MORE FUNCTION THAT MANY ASSEMBLER PROGRAMS CAN ALSO PERFORM IS TO CHECK THE SYNTAX OF THE SOURCE LISTING PROGRAM FOR ERRORS - AND IF SUCH ERRORS ARE FOUND, TO NOTIFY THE OPERATOR. MOST ASSEMBLER PROGRAMS THAT CONTAIN SUCH CAPABILITY CAN TELL THE PROGRAMMER WHAT "LINE NUMBER" IN THE SOURCE LISTING THE ERROR WAS FOUND IN, AND ALSO WHAT TYPE OF ERROR WAS NOTED. FOR INSTANCE, IF THE PROGRAMMER WERE TO LEAVE OUT THE "IMMEDIATE" PART OF THE "LHI" INSTRUCTION USED IN THE ABOVE SAMPLE PROGRAM, THE ASSEMBLER PROGRAM COULD PRESENT A MESSAGE SUCH AS:

LINE 002: INCOMPLETE INSTRUCTION

THUS ADVISING THE PROGRAMMER OF AN ERROR. THE ASSEMBLER THEN MIGHT IMMEDIATELY STOP PROCESSING THE SOURCE LISTING, OR IT COULD BE DESIGNED TO PUT IN A "DUMMY" CODE IN THE "IMMEDIATE" BYTE PORTION OF THE INSTRUCTION AND CONTINUE PROCESSING THE "PASS." THEN, THE PROGRAMMER COULD CORRECT THE OBJECT CODE AT THAT LOCATION WHEN THE ASSEMBLED PROGRAM WAS LOADED INTO THE COMPUTER FOR CHECK-OUT AND TESTING.

AN ASSEMBLER PROGRAM, AS SHOULD NOW BE APPARENT, CAN SAVE MACHINE LANGUAGE PROGRAMMERS, A LOT OF TEDIOUS, TIME CONSUMING WORK!

MORE MULTIPLE-PRECISION MATHEMATICAL ROUTINES

IN THE JANUARY, 1975 ISSUE OF "THE SCELBI COMPUTER DIGEST" THE ARTICLE ON "MULTIPLE-PRECISION ADDITION" CLOSED WITH THE COMMENT THAT THE "ADD" SUBROUTINE PRESENTED IN THE ARTICLE COULD BE USED TO EFFECTIVELY PERFORM SUBTRACTION IF THE NUMBER TO BE SUBTRACTED WAS PLACED IN IT'S TWO'S COMPLEMENT FORM. IN THIS ARTICLE, A SUITABLE "MULTIPLE-PRECISION" TWO'S COMPLEMENT SUBROUTINE WILL BE PRESENTED. HOWEVER, BEFORE PROCEEDING FURTHER, IT MIGHT BE BENEFICIAL TO REVIEW THE CONCEPT OF "TWO'S COMPLEMENT" NOTATION AS IT APPLIES TO OPERATIONS IN A DIGITAL COMPUTER.

WHEN NUMBERS ARE STORED IN A DIGITAL COMPUTER, IN THEIR BINARY FORM, ONE INVARIABLY HAS TO BEAR WITH THE PROBLEM OF DIFFERENTIATING BETWEEN NUMBERS THAT ARE MEANT TO REPRESENT "POSITIVE" VALUES AND THOSE THAT ARE TO REPRESENT "NEGATIVE" VALUES. GENERALLY SPEAKING, THIS DIFFERENTATION PROCESS MUST BE ARBITRARILY CREATED AS A PART OF THE SOFTWARE THAT "OPERATES" THE COMPUTER. OVER THE YEARS, COMPUTER DESIGNERS HAVE DEVELOPED SEVERAL CONVENTIONS FOR ALLOWING THE MANIPULATION OF "POSITIVE" AND "NEGATIVE" VALUES AND ONE OF THE MOST POPULAR CONVENTIONS HAS BEEN TERMED "TWO'S COMPLEMENT" NOTATION. THE METHOD OF HANDLING NUMBER IN THE "TWO'S COMPLEMENT" MANNER IS ENTWINED WITH THE TYPICAL METHOD IN WHICH DIGITAL COMPUTERS ARE DESIGNED TO PERFORM MATHEMATICAL FUNCTIONS. THAT IS, THE ACTUAL "HARDWARE" OR INTERNAL ELECTRONIC LOGIC IMPLEMENTED TO ENABLE THE MACHINE TO PERFORM MATHEMATICAL FUNCTIONS.

THE CONVENTION CAN BE STATED RATHER SIMPLY. A "NEGATIVE" NUMBER IS REPRESENTED AS THE "COMPLEMENT" OF A "POSITIVE" NUMBER PLUS ONE. THE "COMPLEMENT" OF A BINARY NUMBER IS OBTAINED BY SIMPLY REPLACING EVERY BIT IN THE "1" STATE WITH A "0" AND EVERY BIT IN A "0" STATE WITH A "1." THUS, SINCE, IN AN 8 BIT BINARY REGISTER, THE DECIMAL VALUE "5" WOULD APPEAR AS:

0 0 0 0 0 1 0 1

THEN, IN A SIMILAR 8 BIT REGISTER, THE "COMPLEMENT" OF "5" WOULD BE:

1 1 1 1 1 0 1 0

AND, TO OBTAIN THE "TWO'S COMPLEMENT" ONE WOULD SIMPLY ADD "1" TO THE "COMPLEMENT" (IN THE LEAST SIGNIFICANT BIT POSITION) YIELDING:

1 1 1 1 1 0 1 1

NOTE SEVERAL INTERESTING AND RATHER USEFUL FEATURES ABOUT THE TWO'S COMPLEMENT CONVENTION. FIRST, IT IS PRETTY EASY TO OBTAIN - ONE SIMPLY "COMPLEMENTS" THE ORIGINAL VALUE AND ADDS "1." NEXT, ONE CAN READILY OBSERVE THAT IN THE CASE OF POSITIVE "5," THE MOST SIGNIFICANT BIT(S) IN THE REGISTER ARE "0." IN THE CASE OF NEGATIVE "5" THE MOST SIGNIFICANT BIT(S) IN THE REGISTER ARE "1." HOW CONVENIENT! IF ONE ALWAYS RESTRICTS THE MAGNITUDES OF NUMBERS SO THEY USE ONE LESS NUMBER OF BITS THAN ARE AVAILABLE IN THE REGISTER, ONE CAN USE THE MOST SIGNIFICANT BIT IN A REGISTER TO INDICATE THE "SIGN" OF THE NUMBER! BUT, THAT IS NOT ALL. LOOK WHAT HAPPENS IF ONE ACTUALLY ADDS THE VALUE OF A NUMBER AND ITS "TWO'S COMPLEMENT."

```
          0 0  0 0 0  1 0 1
ADD:      1 1  1 1 1  0 1 1
        ------------------
RESULT:   0 0  0 0 0  0 0 0
```

IN A FIXED LENGTH REGISTER, THE RESULT IS JUST WHAT ONE WOULD EXPECT IF ONE "ADDED" A NUMBER AND IT'S MINUS VALUE - OR, PUT ANOTHER WAY, SUBTRACTED A NUMBER FROM ITSELF. AND HERE IS ONE OF THE IMPORTANT REASONS FOR USING THE "TWO'S COMPLEMENT" METHOD - THE MACHINE IS ABLE TO EFFECTIVELY SUBTRACT EVEN THOUGH IT'S ARCHITECTURE MAY ONLY BE CAPABLE OF PERFORMING AN ACTUAL "ADDITION" OPERATION!

THE TWO'S COMPLEMENT CONVENTION FOR REPRESENTING "MINUS" VALUES THUS IS POPULAR BECAUSE IT IS ONE OF THE SIMPLEST AND EASIEST WAYS FOR A DIGITAL COMPUTER TO MANIPULATE SUCH NUMBERS.

THE FACT THAT THE TWO'S COMPLEMENT CONVENTION IS QUITE POPULAR WAS OBVIOUSLY TAKEN INTO ACCOUNT BY THE DESIGNERS OF THE 8008 AND SIMILAR CPU UNITS BECAUSE OF THE CLASS OF CONDITIONAL TEST INSTRUCTIONS SUCH AS THE "JTS/JFS, CTS/CFS, AND RTS/RFS" WHICH TEST THE STATUS OF THE MOST SIGNIFICANT BIT IN A REGISTER AND CONSIDER IT AS THE "SIGN" INDICATOR FOR SETTING THE APPROPRIATE "SIGN" FLAG. PROGRAMMER'S CAN THUS DEVELOP PROGRAMS THAT "TEST" THE CONDITION OF THE "SIGN" FLAG TO SEE WHEN THE VALUE IN A REGISTER IS "PLUS" OR MINUS" WHEN USING "TWO'S COMPLEMENT" NOTATION.

O.K., SO NOW ONE HAS A LITTLE BACKGROUND IN TWO'S COMPLEMENT NOTATION. AND BY NOW, MOST READERS KNOW, THAT WHAT GOES FOR "ONE REGISTER" CAN BE EXTENDED TO WORK FOR "MULTIPLE-REGISTERS." THUS, IF ONE WANTS TO HANDLE NUMBERS GROUPED AS 16, OR 24, OR 32 BINARY BITS, ONE CAN EASILY DO SO BY USING MULTIPLE-PRECISION "CHAINING" TECHNIQUES. AND IF ONE SIMPLY SETS ASIDE THE MOST SIGNIFICANT BIT IN A GROUP OF BITS AS A "SIGN" BIT, ONE CAN REPRESENT "POSITIVE" AND "NEGATIVE" NUMBERS USING TWO'S COMPLEMENT NOTATION. SO, IF ONE WANTS TO SUBTRACT MULTIPLE-PRECISION BINARY NUMBERS, BY FORMING THE "TWO'S COMPLEMENT" OF ONE OF THE NUMBERS AND USING THE MULTIPLE-PRECISION ADDITION ROUTINE PRESENTED IN THE PREVIOUS ISSUE OF "THE SCELBI COMPUTER DIGEST," ONE NEEDS MERELY TO DEVELOP A NICE "MULTIPLE-PRECISION" TWO'S COMPLEMENT ROUTINE THAT WILL TAKE A VALUE IN A STRING OF MEMORY WORDS, AND REPLACE IT WITH THE TWO'S COMPLEMENT VALUE. A SUITABLE ROUTINE IS PRESENTED HERE.

```
                    /MULTIPLE-PRECISION 2'S COMPLEMENT ROUTINE
                    /USER SETS UP "H" & "L" TO LSW OF # TO BE PROCESSED
                    /AND REGISTER "B" TO THE NUMBER OF WORDS THAT THE #
                    /IS STORED IN PRIOR TO CALLING THE ROUTINE
                    /
TWOCOM, LAM         /FETCH 1'ST WORD (LEAST SIGNIFICANT BITS OF #)
        XRI 377     /EXCLUSIVE "OR" YIELDS PURE COMPLEMENT OF 1'ST WORD
        ADI 001     /NOW ADD "1" TO FORM 2'S COMPLEMENT OF 1'ST WORD
MORCOM, LMA         /RETURN 2'S COMPLEMENT VALUE TO ORIGINAL STORAGE
        RAR         /PUT THE CARRY BIT INTO THE ACCUMULATOR
        LDA         /SAVE THE STATUS OF THE CARRY BIT IN REG "D"
        DCB         /DECREMENT THE # OF WORDS COUNTER
        RTZ         /RETURN TO CALLING RTN WHEN COUNTER = ZERO
        INL         /ADVANCE THE MEMORY POINTER TO NEXT WORD IF NOT DONE
        LAM         /FETCH THE NEXT MOST SIGNIFICANT WORD TO THE ACC
        XRI 377     /AND EXCLUSIVE "OR" TO PRODUCE THE COMPLEMENT
        LEA         /SAVE THE COMPLEMENT IN REG "E" TEMPORARILY
        LAD         /GET PREVIOUS CARRY STATUS BACK INTO ACCUMULATOR
        RAL         /AND POSITION IT INTO THE CARRY BIT
        LAI 000     /SET ACC TO 000 WITHOUT INTERFERING WITH CARRY BIT
        ACE         /ADD COMPLEMENT VALUE WITH ANY PREVIOUS CARRY
        JMP MORCOM  /RESTORE CURRENT WORD TO MEMORY AND CONTINUE
```

THE "TWOCOM" ROUTINE EXPECTS THE MULTIPLE-PRECISION NUMBER TO BE STORED IN THE SAME FORMAT AS WAS SPECIFIED FOR THE "ADD" ROUTINE DISCUSSED IN THE PREVIOUS ARTICLE.

THE PROGRAMMER SHOULD TAKE NOTE OF THE METHOD IN WHICH THE CARRY BIT HAD TO BE SAVED IN THE ABOVE ROUTINE. IT IS SOMETIMES NECESSARY TO SAVE THE CARRY STATUS IN SUCH A MANNER. IN THIS CASE IT WAS NECESSARY BECAUSE THE "XRI 377" IS ONE OF THE CLASS OF INSTRUCTIONS THAT ALWAYS SETS THE CARRY FLAG TO THE "CLEARED" OR "0" CONDITION - IT WOULD THUS DESTROY AN "CARRY" FROM A PREVIOUS OPERATION. THE READER IS ALSO CAUTIONED TO NOTE THAT IT IS ONLY NECESSARY TO ADD "1" TO THE LEAST SIGNIFICANT BIT IN THE MULTIPLE-PRECISION NUMBER AND THEN MAKE SURE ANY "CARRY" IS PROPOGATED ACROSS THE REMAINING HIGHER SIGNIFICANT BITS, RATHER THAN ADDING "1" TO EACH WORD AFTER IT IS COMPLEMENTED IN THE STRING!

THE TWO'S COMPLEMENT ROUTINE IS VALUABLE TO HAVE AROUND IN ONE'S PROGRAM LIBRARY - FOR THE PURPOSE DESCRIBED ABOVE, AND FOR OTHER APPLICATIONS. HOWEVER, AS WE HOPE MOST SCELBI-8H USER'S KNOW, IT IS NOT THE ONLY METHOD WHERE-BY A PROGRAMMER CAN EFFECTIVELY SUBTRACT NUMBERS WITH A SCELBI-8H MINI-COMPUTER. FOR, THE INSTRUCTION SET OF THE 8008 CPU ALREADY CONTAINS A CLASS OF "SUBTRACT" COMMANDS THAT ACTUALLY PERFORM AN EQUIVALENT TWO'S COMPLEMENT ADDITION. THUS, ONE COULD USE A SUBTRACTION ROUTINE SIMILAR TO THE ADDITION ROUTINE THAT UTILIZED THE AVAILABLE SUBTRACTION INSTRUCTIONS. IN THE CASE OF MULTIPLE-PRECISION WORK, ONE WANT TO USE THE "SUBTRACT WITH BORROW" CLASS OF INSTRUCTIONS ORGANIZED IN A ROUTINE SUCH AS THE ONE ILLUSTRATED NEXT.

```
                    /MULTIPLE-PRECISION SUBTRACT ROUTINE
                    /USER SETS "H & L" TO LSW OF 1'ST NUMBER
                    /USER SETS "D & E" TO LSW OF # TO BE SUBTRACTED
                    /AND REG "B" TO "PRECISION" B4 CALLING ROUTINE
                    /RESULT OF SUBTRACTION IS LEFT IN FORMER SUBTRAHEND
                    /
SUBBER, NDA         /CLEAR CARRY FLAG AT START OF ROUTINE
SUBTRA, LAM         /GET FIRST NUMBER INTO ACCUMULATOR
        CAL SWITCH  /SWITCH POINTERS TO SUBTRAHEND
        SBM         /SUBTRACT 2'ND FROM 1'ST WITH BORROW
        LMA         /PLACE RESULT BACK INTO FORMER SUBTRAHEND
        DCB         /DECREMENT # OF WORDS (PRECISION) COUNTER
        RTZ         /BACK TO CALLING ROUTINE WHEN CHAIN COMPLETED
        CAL ADV     /OTHERWISE ADVANCE SUBTRAHEND POINTER
        CAL SWITCH  /CHANGE POINTER BACK TO 1'ST NUMBER
        CAL ADV     /ADVANCE POINTER TO NEXT SIGNIFICANT WORD
        JMP SUBTRA  /REPEAT PROCESS FOR NEXT WORD IN CHAIN
```

THE SUBROUTINES "ADV" AND "SWITCH" REFERRED TO BY THE ABOVE ROUTINE WERE DESCRIBED IN THE INITIAL ARTICLE ON MULTIPLE-PRECISION ARITHMETIC.

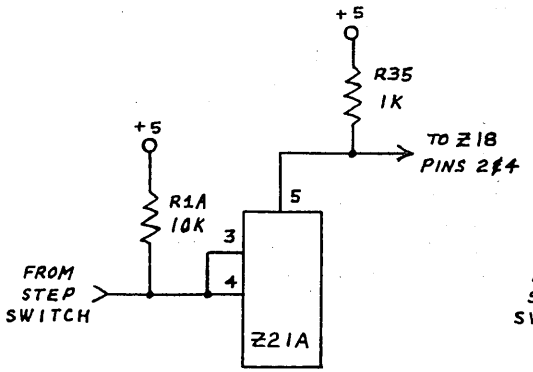
****
****
* *
* INTERESTED IN GETTING INTO MULTIPLE-PRECISION MULTIPLICATION? *
* *
* LET US KNOW - WERE TRYING TO PRODUCE WHAT Y O U WANT! *
* *
****
****

IMPROVING THE OPERATION OF THE SCELBI-8H STEP SWITCH

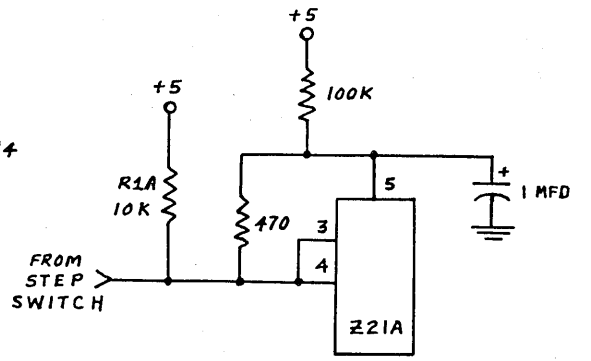
SEVERAL OF OUR CUSTOMERS HAVE EXPERIENCED DIFFICULTY WHEN USING THE STEP SWITCH ON THE SCELBI-8H. THE SYMPTOMS HAVE BEEN EXPRESSED AS "ERRATIC" OPERATION IN THAT SOMETIMES THE COMPUTER WOULD ADVANCE TWO CYCLES AND AT OTHER TIMES WOULD FAIL TO STEP WHEN THE SWITCH WAS OPERATED.

EARLY INDICATIONS WERE THAT SOME STEP SWITCHES THEMSELVES WERE DEFECTIVE, BUT FURTHER EXAMINATION OF THE PROBLEM HAS LED TO MINOR CIRCUIT MODIFICATIONS TO ELIMINATE SUCH "ERRATIC" OPERATION. USER'S WHO ARE EXPERIENCING ANY DIFFICULTY WITH OPERATION OF THE STEP SWITCH, WHO HAVE TYPE #1100 CPU CARDS - REVISION "A" - (WITH I.C. Z21A INSTALLED) SHOULD CONSIDER PERFORMING THE FOLLOWING MODIFICATIONS TO THEIR UNIT TO CORRECT SUCH PROBLEMS.

1.) CIRCUIT MODIFICATION TO THE CPU CARD STEP CIRCUITRY AT I.C. Z21A. CHANGE THE CIRCUITRY AT Z21A TO APPEAR AS SHOWN IN THE SCHEMATIC BELOW.



ORIGINAL CIRCUIT



MODIFIED CIRCUIT

THE CHANGE CONSIST OF ADDING TWO RESISTORS (A 470 OHM AND A 100,000 OHM 1/4 WATT) PLUS A 1 MFD CAPACITOR AT Z21A AS ILLUSTRATED IN THE SCHEMATIC. THE ADDITION OF THE COMPONENTS CAN BEST BE IMPLEMENTED BY MOUNTING THE PARTS ON THE CIRCUIT SIDE OF THE CPU CARD AS SHOWN IN THE DIAGRAM ON THE NEXT PAGE. THE DIAGRAM SHOWS THE FOIL PATTERN ON THE CIRCUIT SIDE OF THE CARD IN THE VICINITY OF I.C. Z21A (TOWARDS THE UPPER RIGHT HAND CORNER NEAR THE TRIMPOTS WHEN VIEWED FROM THE BACK OF THE CARD WITH THE CONNECTOR PINS EDGE OF THE CARD NEAREST THE OBSERVER). BE SURE TO CUT THE P.C. FOIL AT THE POINT NEXT TO THE FOIL PAD MARKED "A" IN THE DIAGRAM.

2.) CHANGE THE WIRING ON THE BACKPLANE CARD AT SLOT XA02 (CPU CARD) BY PERFORMING THE FOLLOWING STEPS.

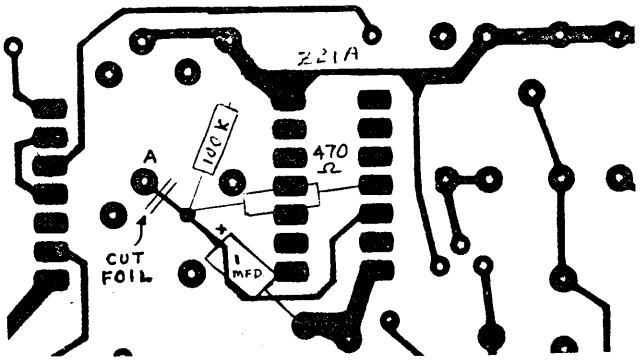
A.) REMOVE THE JUMPER WIRE(S) BETWEEN PINS AX, AY AND AZ (THE "RDYN" SIGNAL) OF SLOT XA02.

B.) INSTALL A JUMPER BETWEEN AX AND AZ OF SLOT XA02 (LEAVING PIN AY "OPEN").

C. RUN A WIRE FROM PIN AY OF XA02 TO PIN BF (T3N SIGNAL) OF XA02.

THE FIRST MODIFICATION ELIMINATES THE POSSIBILITY OF CONTACT BOUNCE UPON RELEASE OF THE STEP SWITCH CAUSING MULTIPLE STEP OPERATION.

THE SECOND MODIFICATION ELIMINATES A POSSIBLE "RACE" CONDITION IN THE CONTROL LOGIC FROM OCCURING THAT CAN SOMETIMES RESULT IN THE STEP CIRCUIT FAILING TO "STEP" THE COMPUTER WHEN IN THE INTERRUPT MODE. IT WILL BE NOTED THAT THE SECOND MODIFICATION WILL RESULT IN MINOR CHANGES OCCURING IN THE SEQUENCE IN WHICH THE INTERRUPT AND STATUS LIGHTS APPEAR WHEN AN INTERRUPT IS RECEIVED AFTER THE COMPUTER HAS BEEN IN THE RUN MODE - DEPENDING ON THE OPERATION BEING PERFORMED AT THE TIME THE INTERUPT SWITCH IS ACTIVATED. HOWEVER, THE BASIC OPERATION REMAINS THE SAME: WHEN THE OPERATOR DEPRESSES THE INTERRUPT SWITCH, THE STEP BUTTON SHOULD BE ADVANCED (IF NECESSARY) UNTIL O N L Y THE "I" LAMP IS LIT. IF, FOR INSTANCE, THE INTERRUPT AND A STATUS LAMP COMES ON WHEN THE "INT" SWITCH IS OPERATED, THEN THE OPERATOR SHOULD OPERATE THE STEP BUTTON UNTIL THE STATUS LAMP(S) GO OFF (WITH THE "I" LAMP REMAINING ON) TO SIGNIFY THE START OF THE INTERRUPT CYCLE. THE OPERATOR THEN PROCEEDS AS IN THE PAST TO USE THE STEP SWITCH TO BRING THE STATUS LAMPS TO THE DESIRED CONDITION(S) (IF REQUIRED) FOR INSERTING "INTERRUPT MODE" INSTRUCTIONS TO THE COMPUTER. (THE CASE OF BOTH THE "I" LAMP AND THE STATUS LAMP(S) BEING LIT WHEN THE INTERRUPT MODE IS INITIALLY ENTERED SIGNIFIES THAT THE COMPUTER HAS NOT FINISHED THE LAST MULTI-BYTE INSTRUCTION BEING EXECUTED WHEN THE "INTERRUPT" SIGNAL WAS RECEIVED - THIS MUST BE DONE IN THE STEP MODE BEFORE THE NEW COMMAND IS GIVEN VIA THE CHASSIS SWITCHES).



PARTS PLACEMENT ON CPU FOIL PATTERN FOR STEP CIRCUITRY MODIFICATION

* *

THE MORE PEOPLE WHO CONTRIBUTE ARTICLES AND PROGRAMS FOR PUBLICATION IN

"THE SCELBI COMPUTER DIGEST & USER'S BULLETIN"

THE BETTER THE ISSUES SHALL BE!

SCELBI COMPUTER CONSULTING, INC.
1322 REAR - BOSTON POST ROAD
MILFORD, CT. 06460