INSTRUCTION SET CONT.

55 BNP BRANCH NEGATIVE OR POSITIVE

56 BNZ BRANCH NEGATIVE OR ZERO

57 B BRANCH

60 BF BRANCH FLAG
BRANCH TO THE INSTRUCTION AMLSB P ADDRESS IF THERE IS A FLAG AMLSB Q ADDRESS

61 BNF BRANCH NO FLAG

70 BAL BRANCH AND LINK
THE CONTENT OF THE INSTRUCTION ADDRESS COUNTER (ADDRESS OF NEXT SEQUENTIAL INSTRUCTION) IS STORED AMLSB Q ADDRESS. THEN A BRANCH IS MADE TO THE P ADDRESS. USED TO LINK TO SUBROUTINES.

00 IND INPUT DIGITS

01 INC INPUT CHARACTERS

30 OUD OUTPUT DIGITS

31 OUC OUTPUT CHARACTERS

35 DMP DUMP MEMORY AS DIGITS

OPERATION OF INPUT/OUTPUT INSTRUCTIONS DEPENDS ON USER PREFERENCE AND EQUIPMENT.

IN MOVE DIGIT AND BRANCH AND LINK, ONLY THE OCTAL DIGITS ARE MOVED OR STORED, FLAGS IN DESTINITION ARE UNALTERED.

THE PROGRAMMING EXAMPLES TO THE RIGHT ARE GIVEN TO ILLUSTRATE THE USE OF THE MINI-MINI COMPUTER'S INSTRUCTION SET. THE PROGRAMS HAVE NOT BEEN TESTED AND PROBABLY HAVE ONE OR MORE ERRORS EACH.

POTENTIAL BUILDERS WHO HAVE ACCESS TO LARGER COMPUTERS ARE PROBABLY USED TO AN ASSEMBLY OR COMPILER LANGUAGE INSTEAD OF STRAIGHT MACHINE LANGUAGE AS USED IN THESE EXAMPLES. THERE IS PROBABLY NOT ENOUGH MEMORY TO FIT AN ASSEMBLER PROGRAM INTO MINI-MINI. ASSEMBLERS HAVE HOWEVER BEEN WRITTEN FOR OTHER SMALL COMPUTERS IN FORTRAN AND RUN ON LARGER SCALE HARDWARE. IF SOMEBODY WOULD LIKE TO WRITE AN ASSEMBLER AND SHARE IT WITH OTHER BUILDERS, THEY WOULD GREATLY APPRECIATE IT. ASA STANDARD BASIC FORTRAN IV IS PROBABLY THE MOST UNIVERSAL LANGUAGE TODAY. EVERY MEDIUM AND LARGE SCALE COMPUTER CAN ACCEPT IT.

SUBROUTINE LINKAGE - AN EXAMPLE

THE EXAMPLE SHOWS LINKAGE TO AND FROM A SUBROUTINE FOR FINDING THE ABSOLUTE VALUE OF AN ARGUMENT FIELD. SUBROUTINES WILL BE USED FREQUENTLY SINCE THEY HELP A PROGRAMMER CONSERVE MEMORY SPACE. NOTE THAT THE USE OF INDIRECT ADDRESSING MAKES ACCESSING THE ARGUMENT FIELD MUCH EASIER.

LOCATION	CONTENTS	MNEMONIC	P	Q	COMMENTS
					MAIN PROGRAM
1000	1115000̄010	ADDI	1500	0̄010	CALLING PROGRAM INSTRUCTION
1012	702002110	BAL	2000	2110	CALL ABSOLUTE VALUE SUBROUTINE
1024	1500				ADDRESS OF ARGUMENT
1030	2415001300	C	1500	1300	CALLING PROGRAM INSTRUCTION
		⋮			
					SUBROUTINE TO FIND ABSOLUTE VALUE OF INPUT ARG.
2000	3221130000	SF	2113	0000	SET INDIRECT FLAG IN RETURN ADDRESS = ADDRESS OF ADDRESS OF ARGUMENT
2012	5320502110̄	BPZ	2062	2110̄	TEST SIGN OF ARGUMENT - 2LEVEL INDIRECT ADDR.
2024	2621272110̄	MVF	2127	2110̄	IF NEGATIVE, SAVE ARGUMENT
2036	2221102110̄	SUB	2110̄	2110̄	ZERO ORIGIONAL ARGUMENT
2050	2221102127	SUB	2110̄	2127	PLACE COMPLEMENT IN ARGUMENT LOCATION
2062	1121130̄004	ADDI	2113	0̄004	ADD 4 TO RETURN ADDRESS
2074	3321130000	CF	2113	0000	CLEAR INDIRECT ADDRESS FLAG
2106	470̄000	B	0̄000		P ADDRESS FILLED IN BY CALLING PROGRAM, RETURN TO MAIN
2114	0̄00000000000				ROOM FOR UP TO 12 DIGIT ARGUMENT

PROGRAMMING EXAMPLE — OCTAL TO DECIMAL CONVERSION SUBROUTINE

LOCATION	CONTENTS	MNEMONIC	P	Q	COMMENTS
					SUBROUTINE TO CONVERT OCTAL TO DECIMAL MOVE 10 DIGIT FIELD TO BE CONVERTED TO 1404 8 CHARACTER OUTPUT FIELD IS AT 1405 ADDRESSED ON LEFT. LEADING ZEROES REPLACED BY BLANKS. CHARACTER CODE FOR DIGITS 0-9 IS 20-31 BLANK IS 40 Q ADDRESS ON BAL SHOULD BE 1160
1000	1610470040̄	MVFI	1047	0040̄	SET LEAD ZERO SUPPRESS
1012	1611671333̄	MVFI	1167	1333̄	INITIALIZE POWER OF 10 POINTER
1024	1611131406̄	MVFI	1113	1406̄	INITIALIZE OUTPUT FIELD POINTER
1036	1611170040̄	MVFI	1117	0040̄	SET DIGIT BEING GENERATED TO ZERO OR BLANK
1050	2214041167̄	SUB	1404	1167̄	SUBTRACT A POWER OF 10 FROM OCTAL FIELD
1062	4311760000̄	BPZI	1176	0000̄	BRANCH IF NOT NEGATIVE RESULT
1074	2114041167̄	ADD	1404	1167̄	ADD POWER OF 10 BACK
1106	1600000000̄	MVFI	0000̄	0000̄	MOVE DECIMAL DIGIT TO OUTPUT FIELD
1120	1111130002̄	ADDI	1113	0002̄	OUTPUT FIELD POINTER TO NEXT CHARACTER POSITION
1132	1111670004̄	ADDI	1167	0004̄	TO NEXT LOWER POWER OF 10
1144	1411131426̄	CI	1113	1426̄	TEST IF AT END OF OUTPUT FIELD
1156	4200000000̄	BZI	0000	0000̄	RETURN IF FINISHED; Q ADDR IS POWER 10 POINTER
1170	471036	B	1036		CONTINUE IF NOT FINISHED
1176	1410470040̄	CI	1047	0040̄	TEST IF DIGIT BEING GENERATED IS A BLANK
1210	4512460000̄	BNPI	1246	0000̄	BRANCH IF NOT
1222	1611170020̄	MVFI	1117	0020̄	IF SO, CHANGE TO THE CHARACTER ZERO
1234	1610470020̄	MVFI	1047	0020̄	CLEAR LEAD ZERO SUPPRESS
1246	1111170001̄	ADDI	1117	0001̄	INCREMENT DIGIT BEING GENERATED
1260	471050	B	1050		CONTINUE
1266	4611̄3200				10**7 POWER OF 10 TABLE
1276	364̄1100				10**6
1305	30̄3240				10**5
1313	2̄3420				10**4
1320	1̄750				10**3
1324	1̄44				10**2
1327	1̄2				10**1
1331	0̄1				10**0
1333	1267				ADDRESS POINTERS TO POWERS OF 10
1337	1276				
1343	1304				
1347	1311				
1353	1315				
1357	1320				
1363	1322				
1367	1324				
1373	0̄000000000				OCTAL INPUT FIELD
1405	0000000000000000				8 CHARACTER OUTPUT FIELD

THE ABOVE SUBROUTINE USES THE SUCCESSIVE SUBTRACTION OF POWERS OF 10 METHOD FOR CONVERSION. IN AN ACTUAL DESK CALCULATOR PROGRAM, IT WOULD PROBABLY BE DESIRABLE TO USE THE SUCCESSIVE DIVISION BY 10 METHOD SINCE A DIVISION SUBROUTINE WOULD BE ALREADY IN CORE. THE SPACE OCCUPIED BY THE TABLES WOULD THEN BE FREE FOR OTHER USES.

IF THE INSTRUCTION 2600060007 OR 1600100000 IS LOADED INTO LOCATION ZERO AND EXECUTED, THE ENTIRE MEMORY WILL BE CLEARED TO ZEROES. TRY TO FIGURE OUT HOW IT WORKS.

CENTRAL PROCESSING UNIT DESIGN

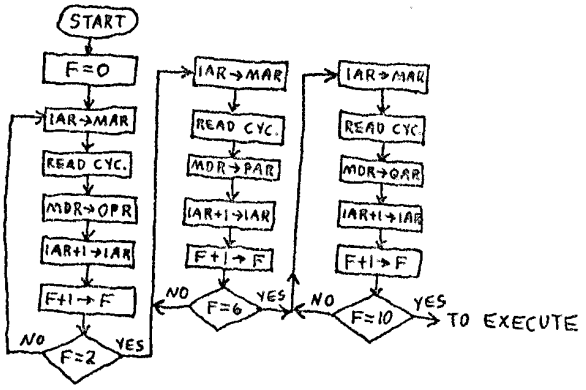
COMPLETE DETAILED PLANS FOR THE CENTRAL PROCESSING UNIT CANNOT BE GIVEN FOR REASONS MENTIONED EARLIER. HOWEVER EACH PHASE OF THE DESIGN PROCEDURE WILL BE DISCUSSED AND A PART OF THAT PHASE WORKED OUT TO THE EQUATION OR CIRCUIT LEVEL. IT WILL BE LEFT TO THE READER TO APPLY THE SAME TECHNIQUE TO COMPLETE EACH PHASE.

REGISTERS

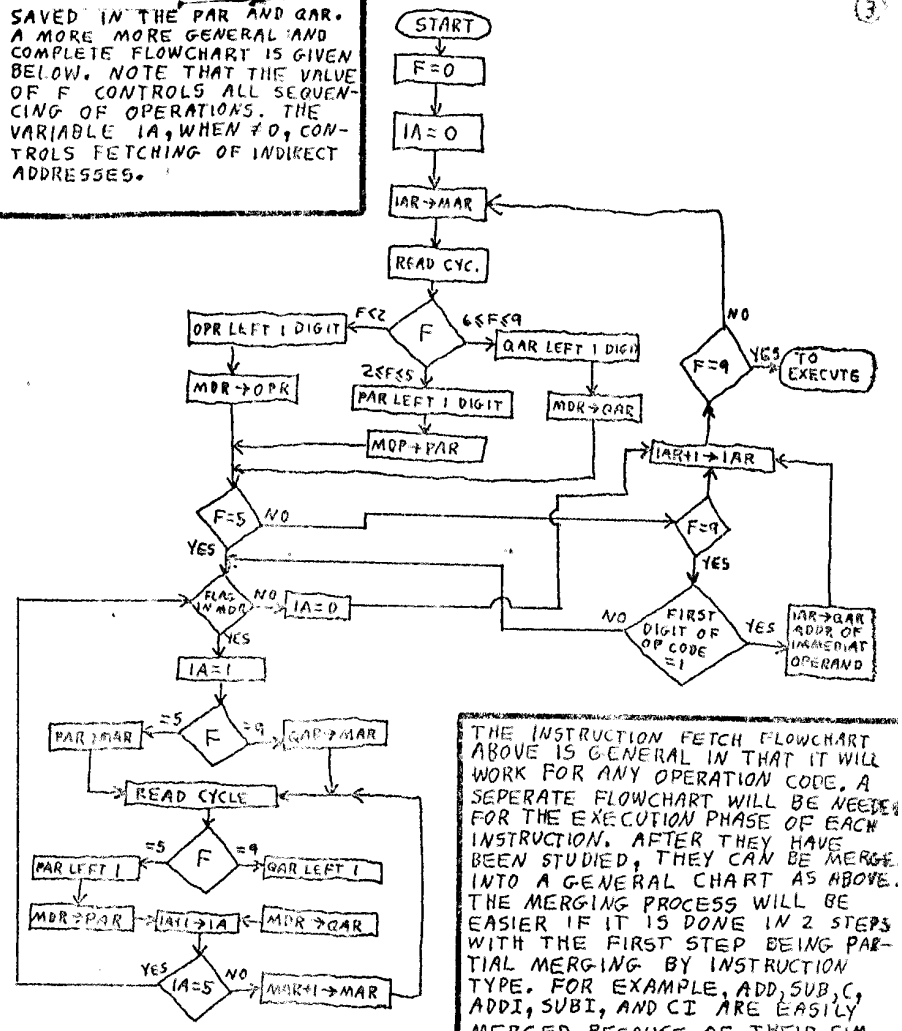
EVEN THOUGH NO ACCUMULATOR REGISTER IS NEEDED, OTHER REGISTERS ARE REQUIRED TO HOLD THE VARIOUS ADDRESSES. THERE IS A MEMORY ADDRESS REGISTER (MAR) OF 12 BITS (4 DIGITS) AND A MEMORY DATA REGISTER (MDR) OF 4 BITS (1 DIGIT WITH FLAG) ASSOCIATED WITH THE MEMORY. 2 REGISTERS OF 12 BITS EACH WILL BE REQUIRED TO HOLD THE P ADDRESS (PAR) AND THE Q ADDRESS (QAR). WE WILL NEED AN INSTRUCTION ADDRESS REGISTER (IAR) OF 12 BITS TO POINT TO THE NEXT INSTRUCTION. AN OPERATION REGISTER (OPR) OF 6 BITS WILL BE NEEDED TO HOLD THE OP CODE OF THE CURRENT INSTRUCTION. FINALLY A DIGIT BUFFER REGISTER OF ONE DIGIT + FLAG WILL BE NEEDED TO SAVE A DIGIT BETWEEN 2 MEMORY CYCLES. THE ENTIRE OPERATION OF THE COMPUTER INVOLVES THE CONTROLLED TRANSFER OF DIGITS AMONG THESE REGISTERS AND THROUGH A 3 BIT PARALLEL ADDER AS WELL AS CAUSING SOME OF THE REGISTERS TO COUNT. WHAT COULD BE SIMPLER?

INSTRUCTION DIAGRAMMING

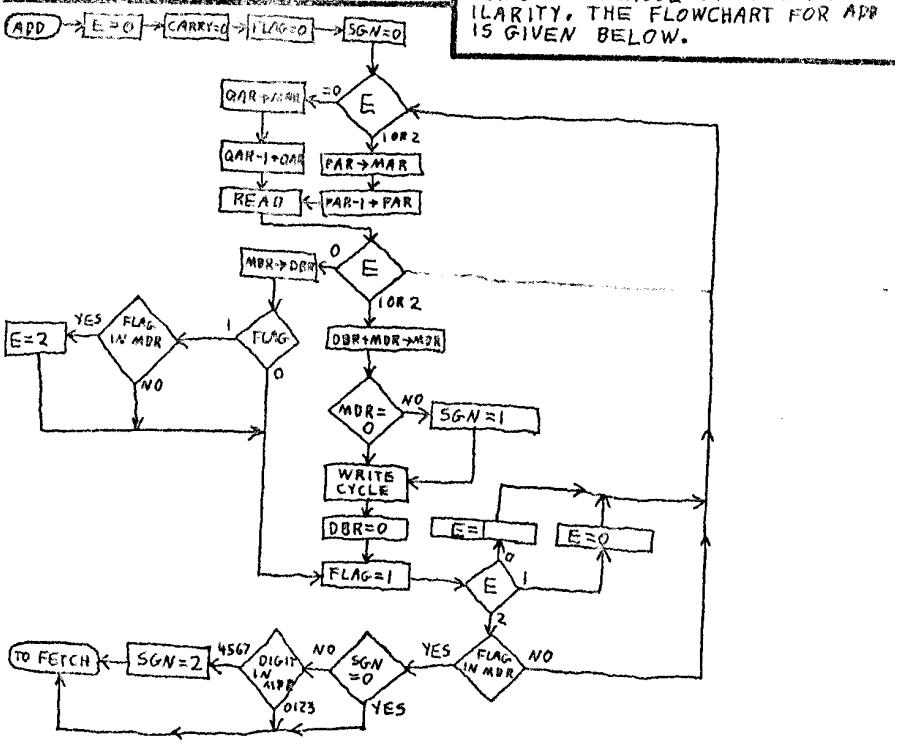
BEFORE WE CAN DECIDE WHAT OPERATIONS EACH REGISTER SHOULD BE CAPABLE OF PERFORMING AND WHAT KIND OF TRANSFER CIRCUITRY IS REQUIRED, WE NEED A DIAGRAM OF THE FETCH AND EXECUTION PHASE OF EACH INSTRUCTION. FOR EXAMPLE, CONSIDER THE FETCH OF THE INSTRUCTION 2101000200. WE NEED TO GET THE 21 INTO OPR, THE 0100 INTO PAR, AND THE 0200 INTO QAR. THE IAR SHOULD BE INCREMENTED BY 10 TO POINT TO THE NEXT INSTRUCTION. BELOW IS A FLOWCHART OF THE FETCH OF THIS INSTRUCTION.



THE VARIABLE F IS A COUNTER THAT COUNTS FETCH CYCLES AND CONTROLS DATA TRANSFERS. THE OPR, PAR, AND QAR ARE SHIFT REGISTERS THAT SHIFT 3 BITS (ONE DIGIT) LEFT BEFORE THEY ARE LOADED WITH A NEW DIGIT ON THE RIGHT END. AT THE END OF THE PROPER NUMBER OF FETCH CYCLES, EACH REGISTER HAS ITS DIGITS IN THE RIGHT PLACE. IF THE INSTRUCTION BEING FETCHED IS AN IMMEDIATE INSTRUCTION, THE QAR SHOULD BE LOADED INSTEAD WITH THE ADDRESS OF THE RIGHTMOST DIGIT OF THE Q ADDRESS WHICH IS AVAILABLE FROM THE IAR WHEN F=9. INDIRECT ADDRESSES WILL ALSO HAVE TO BE RESOLVED DURING INSTRUCTION FETCH SINCE THE INDIRECT FLAG IS NOT SAVED IN THE PAR AND QAR. A MORE MORE GENERAL AND COMPLETE FLOWCHART IS GIVEN BELOW. NOTE THAT THE VALUE OF F CONTROLS ALL SEQUENCING OF OPERATIONS. THE VARIABLE IA, WHEN ≠0, CONTROLS FETCHING OF INDIRECT ADDRESSES.

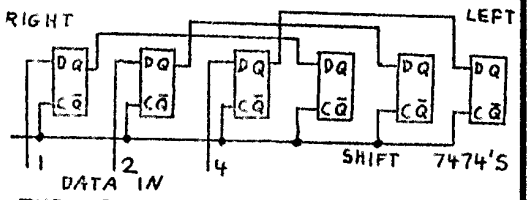


THE INSTRUCTION FETCH FLOWCHART ABOVE IS GENERAL IN THAT IT WILL WORK FOR ANY OPERATION CODE. A SEPERATE FLOWCHART WILL BE NEEDED FOR THE EXECUTION PHASE OF EACH INSTRUCTION. AFTER THEY HAVE BEEN STUDIED, THEY CAN BE MERGED INTO A GENERAL CHART AS ABOVE. THE MERGING PROCESS WILL BE EASIER IF IT IS DONE IN 2 STEPS WITH THE FIRST STEP BEING PARTIAL MERGING BY INSTRUCTION TYPE. FOR EXAMPLE, ADD, SUB, C, ADDI, SUBI, AND CI ARE EASILY MERGED BECAUSE OF THEIR SIMILARITY. THE FLOWCHART FOR ADD IS GIVEN BELOW.

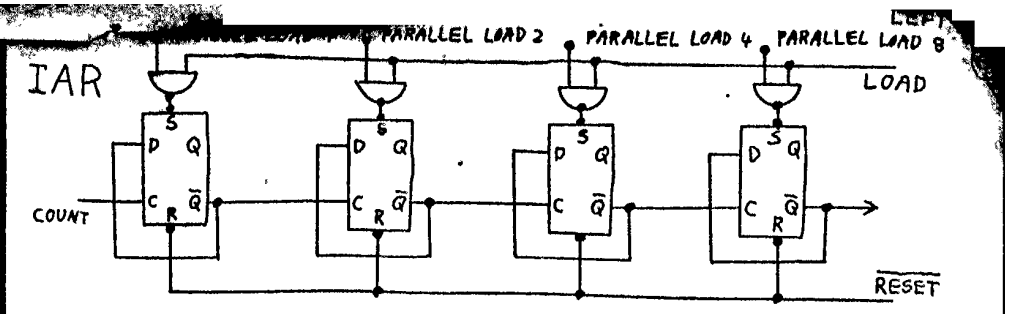


IN THE ABOVE FLOWCHART, E CONTROLS THE REGISTER TRANSFERS. FLAG PREVENTS THE Q FIELD FROM BEING TERMINATED BY A FLAG IN THE RIGHTMOST DIGIT. SGN IS THE MACHINE INDICATOR THAT SHOWS THE SIGN OF THE RESULT. 0=0, 1=+, 2=−. NOTE THAT THE SAME FLOWCHART COULD BE USED FOR SUBTRACT IF THE BLOCK THAT SETS CARRY=0 IS CHANGED TO SET CARRY=1; ~~THE BLOCK THAT SETS DBR=0 IS CHANGED TO SET DBR=7~~; AND THE BLOCK DBR+MDR→MDR INSTEAD USES THE 7'S COMPLEMENT OF DBR

HAVING DIAGRAMMED THE INSTRUCTIONS, WE ARE NOW IN A POSITION TO SPECIFY THE OPERATIONS PERFORMED BY EACH REGISTER AND THE TRANSFER PATHS BETWEEN REGISTERS. THE MAR WILL HAVE TO BE ABLE TO COUNT UP AND ACCEPT A PARALLEL 12 BIT ADDRESS. THE MDR HAS TO ACCEPT A 3 BIT DIGIT AND FLAG INDEPENDENTLY, THAT IS, WHEN A DIGIT IS STORED AS IN MOVE DIGIT, THE FLAG IN THAT MEMORY LOCATION SHOULD NOT BE ALTERED. THE PAR WILL HAVE TO BE ABLE TO SHIFT LEFT 3 BITS AT A TIME, ACCEPT 3 BITS IN PARALLEL AT ITS RIGHT END, AND COUNT DOWN. THE QAR IS THE SAME EXCEPT THAT IT WILL ALSO BE ABLE TO ACCEPT A 12 BIT ADDRESS IN PARALLEL IN THE CASE OF IMMEDIATE INSTRUCTIONS. THE OPR IS PROBABLY THE SIMPLEST REGISTER, IT ONLY HAS TO ACCEPT 3 BITS IN PARALLEL AND SHIFT LEFT. THE IAR MUST COUNT UP AND ACCEPT A 12 BIT ADDRESS FOR THE BRANCH INSTRUCTIONS. THE REGISTERS ARE BEST IMPLEMENTED WITH THE TYPE 7474 DUAL D FLIP-FLOP, OR WHERE A SIGNIFICANT REDUCTION IN EXTERNAL GATES CAN BE REALIZED, THE 7473 OR 7476 DUAL J-K FLIP-FLOPS. THE LATTER ALSO HAS DIRECT SET INPUTS AND COSTS MORE. AS AN EXAMPLE OF FLIP-FLOP USAGE IN REGISTER DESIGN, CONSIDER THE DIAGRAM BELOW WHICH CAN BE USED AS THE OPR.

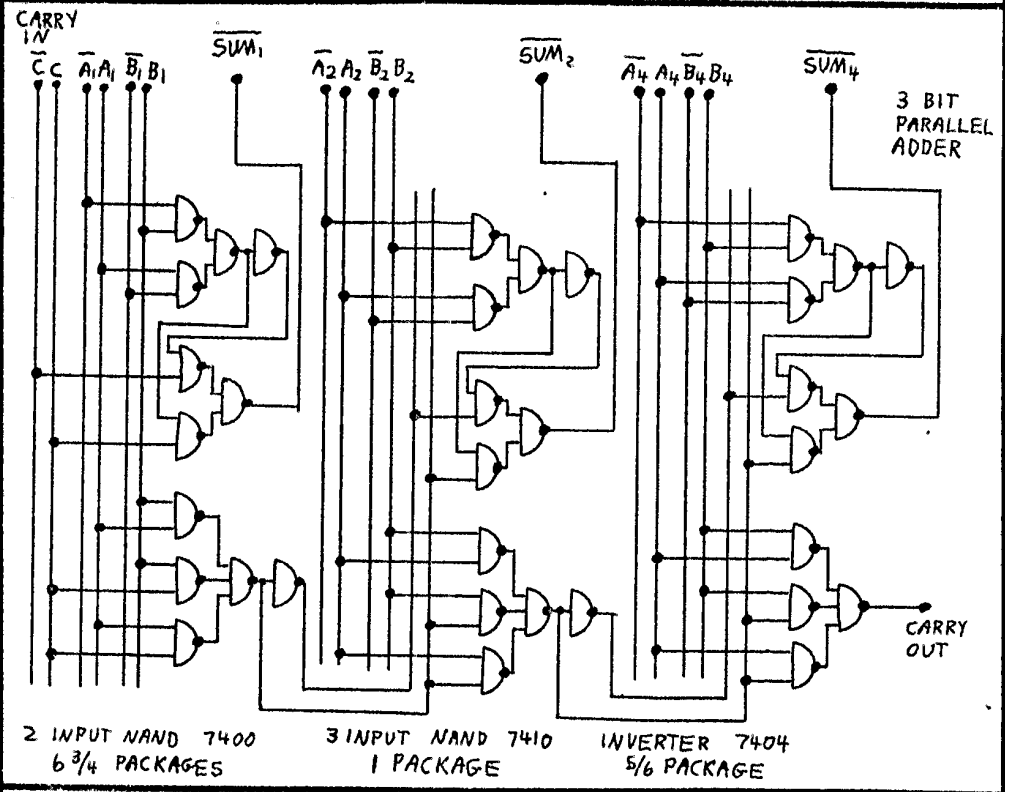


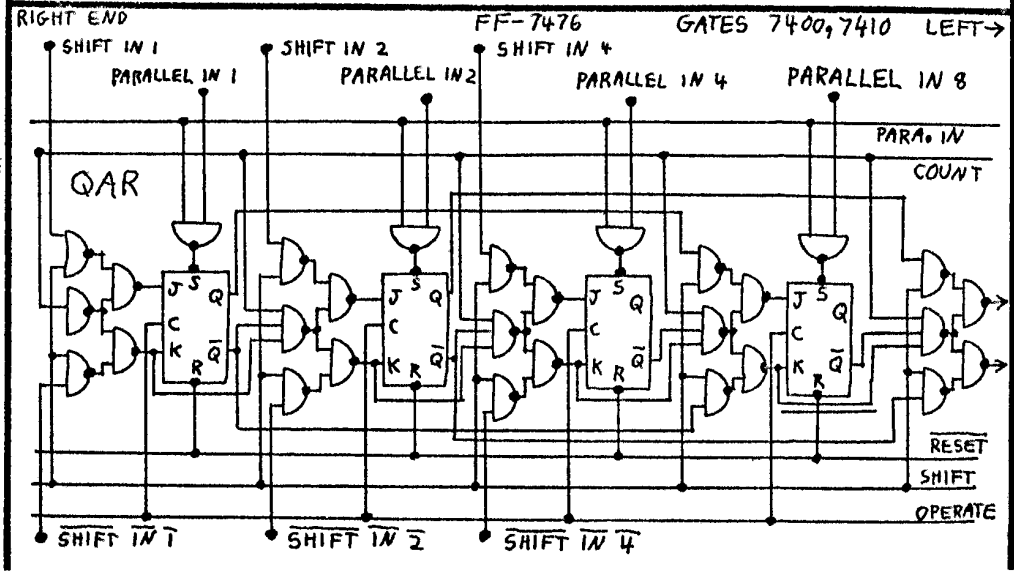
TYPE 7474 DUAL D FLIP-FLOPS ARE USED. TO OPERATE THE REGISTER, THE DIGIT TO BE INSERTED INTO THE REGISTER IS PRESENTED TO THE 3 DATA IN LINES. AFTER A CONDITIONING DELAY OF 100 NSEC OR MORE, A 0 TO 1 TRANSITION ON THE SHIFT LINE WILL CAUSE THE INFORMATION CURRENTLY IN THE REGISTER TO BE SHIFTED LEFT, LOOSING THE LEFT DIGIT AND INSERTING THE DATA INTO THE RIGHT DIGIT. A MORE COMPLEX REGISTER IS THE QAR. EACH CELL OF THE REGISTER HAS TO ACCEPT INFORMATION FROM 3 "SOURCES": 12 BIT PARALLEL INPUT, THE NEXT-3 CELL FOR SHIFTING, AND A LOGIC NETWORK FOR COUNTING. THE SHIFTING AND COUNTING MUST BE DONE WITH THE SYNCHRONOUS INPUTS BUT THE 12 BIT PARALLEL LOAD MAY BE DONE WITH THE DIRECT SET AND CLEAR INPUTS. TO MAKE THE REGISTER SHIFT IN THE DIGIT FROM THE DIGIT IN LINES, ACTIVATE THE SHIFT CONTROL LINE AND ABOUT 100 NSEC OR MORE LATER, PROVIDE A 0 TO 1 TRANSITION ON THE OPERATE LINE. TO COUNT DOWN, ACTIVATE THE COUNT LINE, WAIT 500 NSEC, AND HIT THE OPERATE LINE. TO DO A PARALLEL LOAD, APPLY A 0 TO THE RESET LINE FOR 50 NSEC OR MORE AND THEN ACTIVATE THE LOAD LINE AFTER REMOVING THE RESET SIGNAL. DO NOTHING WITH THE OPERATE LINE. 7473 FLIP-FLOPS CAN BE USED IN THE PAR SINCE PARALLEL LOADING CAPABILITY IS NOT NECESSARY.



THE 4 LOW ORDER BITS OF THE IAR ARE SHOWN ABOVE. TO MAKE THE IAR COUNT, IT IS ONLY NECESSARY TO APPLY A PULSE TO THE COUNT INPUT. THE PARALLEL LOADING OF 12 BITS IS ACCOMPLISHED AS WITH THE QAR EXCEPT THAT THE RESET SIGNAL AND LOAD SIGNAL SHOULD BE OF AT LEAST 200 NSEC DURATION SO THAT TRANSIENTS FROM THE COUNTING HOOKUP CAN DIE OUT.

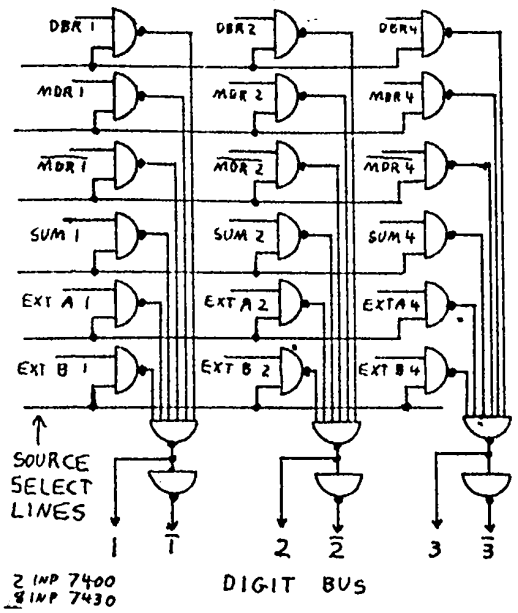
BELOW IS A DIAGRAM OF THE 3 BIT PARALLEL ADDER THAT IS NEEDED FOR THE ARITHMETIC INSTRUCTIONS. TO USE THE ADDER, IT IS NECESSARY ONLY TO APPLY THE 2 DIGITS TO BE ADDED, A AND B, ALONG WITH THE CARRY IN BIT; WAIT 200 NSEC OR MORE; AND LOAD THE SUM OUTPUT AND CARRY OUT BIT INTO SOME REGISTER. NO CLOCKS OR TIMING IS NEEDED FOR THE ADDER.



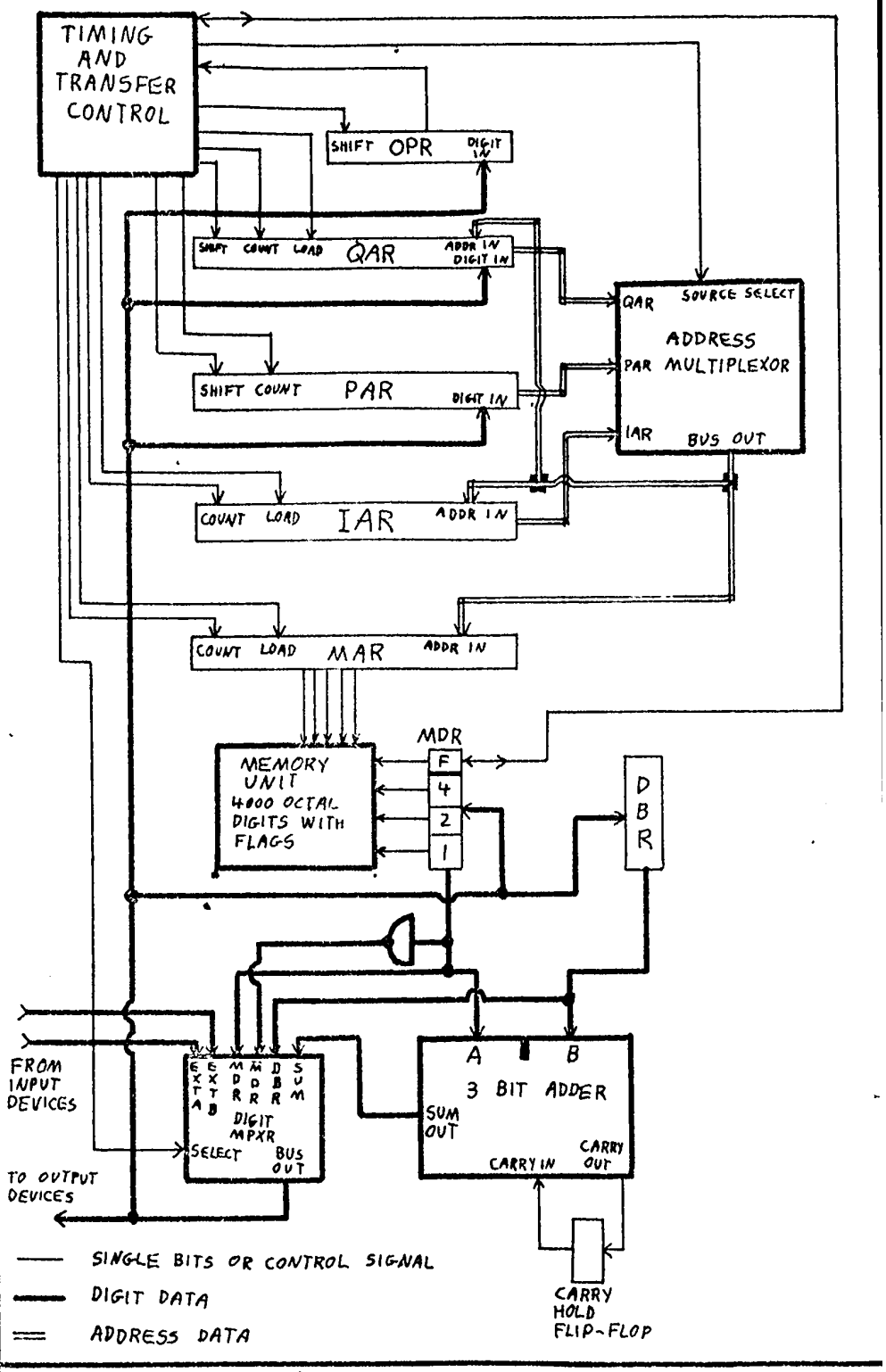


DATA MOVEMENT AND REGISTER MULTIPLEXORS

AS MENTIONED EARLIER, THE ENTIRE OPERATION OF THE COMPUTER IS BASED ON DATA TRANSFERS BETWEEN REGISTERS. IN THE MINI-MINI COMPUTER THERE ARE TWO TYPES OF DATA, THE 3 BIT DIGIT AND THE 12 BIT ADDRESS. THE PAR AND QAR HANDLE BOTH TYPES, THE OTHER REGISTERS HANDLE ONLY 1 TYPE. A LOOK AT THE INSTRUCTION FLOWCHARTS WILL REVEAL ALL THE VARIOUS TRANSFERS THAT ARE NEEDED. A DATA TRANSFER IS DEFINED BY SPECIFYING A SOURCE AND A DESTINATION REGISTER. IN ORDER TO REDUCE THE NUMBER OF GATES REQUIRED, TWO DATA BUSSES WILL BE USED, ONE FOR DIGITS AND ONE FOR ADDRESSES. IN A DATA BUS SYSTEM ALL DESTINATION REGISTER'S INPUTS ARE CONNECTED TO THE DATA BUS. A PARTICULAR REGISTER RECEIVES DATA FROM THE BUS WHEN ITS "LOAD" OR "OPERATE" LINE IS ACTIVATED. THE BUS IS DRIVEN BY A GATING NETWORK CALLED A MULTIPLEXOR. THE OUTPUTS OF ALL POSSIBLE SOURCE REGISTERS ARE CONNECTED TO THE MULTIPLEXOR INPUTS. OTHER MULTIPLEXOR INPUTS ALLOW THE DATA FROM A PARTICULAR SOURCE REGISTER TO BE PLACED ON THE DATA BUS. AS A RESULT, A DATA TRANSFER IS ACCOMPLISHED BY SIMPLY SELECTING A SOURCE REGISTER WITH THE MULTIPLEXOR, WAITING A SHORT TIME, AND ACTIVATING THE DESIRED DESTINATION REGISTER. IN THE DIGIT BUS SYSTEM THERE ARE 6 DISTINCT SOURCES: DBR, MDR, THE 7'S COMPLEMENT OF MDR, ADDER SUM OUTPUT, AND 2 EXTERNAL INPUT SOURCES; AND 7 DESTINATIONS: DBR, MDR, DIGIT INPUT OF QAR, PAR, AND OPR, AND 2 EXTERNAL OUTPUT REGISTERS. THE 7'S COMPLEMENT OF MDR SOURCE WILL BE NEEDED FOR THE SUBTRACT INSTRUCTIONS. THE ADDRESS BUS SYSTEM HAS 3 SOURCES: PAR, QAR, IAR; AND 3 DESTINATIONS: QAR, IAR, MAR. THE DIGIT MULTIPLEXOR IS SHOWN BELOW, THE ADDRESS MULTIPLEXOR IS SIMILIAR BUT NEEDS ONLY 3 SOURCE INPUTS AND ONLY THE TRUE OUTPUT FOR THE BUS.



MINI-MINI COMPUTER FUNCTIONAL BLOCK DIAGRAM



THE TIMING AND TRANSFER CONTROL BLOCK ABOVE IS RESPONSIBLE FOR IMPLEMENTING THE INSTRUCTION FLOWCHARTS DRAWN EARLIER. THE VARIOUS CONTROL SIGNALS IT GENERATES ARE ALL A FUNCTION OF THE OPR CONTENTS, THE FLAG BIT, AND THE INTERNAL MACHINE CYCLE COUNTERS SUCH AS THE F, I, AND E COUNTERS IN THE FLOWCHARTS. EVENTS OCCURRING DURING A SINGLE MEMORY CYCLE CAN BE COORDINATED BY THE VARIOUS MEMORY TIMING SIGNALS THAT WILL BE AVAILABLE. UNFORTUNATELY, NO SPECIFIC DETAILS CONCERNING THIS BOX CAN BE GIVEN WITHOUT HAVING ACTUALLY CONSTRUCTED ONE. THIS IS AN AREA OF THE MACHINE WHERE THE BUILDER CAN DEMONSTRATE HIS INGENUITY! SO FAR MOST OF THE HARD-TO-FIND SPECIFICS HAVE BEEN GIVEN, THE REST IS RELATIVELY STRAIGHTFORWARD.

GO TO IT!

MINI-MINI COMPUTER

THE INFORMATION ON THIS AND THE NEXT FEW SHEETS IS A SKETCHY DESCRIPTION OF A SMALL, GENERAL PURPOSE COMPUTER BUILT AROUND A STACK OF 4 MEMORY PLANES, ORIGIONALLY FROM A 1401 COMPUTER, AVAILABLE QUITE CHEAPLY FROM MIKE QUINN ELECTRONICS. WHERE SPECIFIC LOGIC DIAGRAMS OR BOOLEAN EQUATIONS ARE GIVEN, THEY ARE DESIGNED AROUND THE 7400 SERIES OF TTL INTEGRATED CIRCUITS, ALSO AVAILABLE FROM MIKE QUINN AT SURPLUS PRICES. THESE SHEETS ARE INTENDED TO START THE FLOW OF IDEAS IN AM AMBITIOUS EXPERIMENTER AND SUPPLY HARD-TO-FIND INFORMATION, PARTICULARLY ON MEMORY DRIVE CIRCUITS AND INSTRUCTION SET DESIGN. CONSTRUCTIONAL INFORMATION AS FOUND IN POPULAR ELECTRONICS WOULD 1. REQUIRE A FULL LENGTH BOOK, 2. REQUIRE THAT THE AUTHOR HAVE BUILT THE MACHINE EXACTLY AS DESCRIBED. (THE AUTHOR HAS BUILT A COMPUTER BUT IT IS SOMEWHAT LARGER THAN THE ONE DESCRIBED HERE.) DETAILS CONCERNING COMBINATIONAL LOGIC DESIGN AND SEQUENCING OF OPERATIONS ARE AVAILABLE IN ANY GOOD TEXT BOOK OR MANUFACTURER'S LITERATURE. IT IS HOPED THAT THE MACHINE DESCRIBED CAN BE BUILT FOR $300 NOT INCLUDING POWER SUPPLIES OR I/O EQUIPMENT. A PENETRATING SEARCH OF THE SURPLUS MARKET SHOULD PRODUCE A TELETYPE MACHINE WITH PAPER TAPE, AN IDEAL I/O DEVICE. WITH A GOOD DESK CALCULATOR PROGRAM, THIS SMALL COMPUTER SHOULD OUTPERFORM ELECTRONIC DESK CALCULATORS IN THE $3K TO $5K RANGE. HOWEVER, BEING A GENERAL PURPOSE COMPUTER, ITS APPLICATIONS ARE LIMITED ONLY BY ONES IMAGINATION, PROGRAMMING SKILL, AND THE MACHINE'S MEMORY CAPACITY (IN THAT ORDER).

ADDRESS ANY COMMENTS OR CRITICISMS ABOUT THESE SHEETS TO: HAL CHAMBERLIN 123 ASHE AVE. RALEIGH NORTH CAROLINA 27605. IF ANYBODY BUILDS ONE OF THESE AND WOULD LIKE TO HELP FORM A USER'S GROUP FOR PROGRAM SHARING, WRITE ME!

INSTRUCTION FORMAT

ALL OPERATIONS ARE MEMORY-TO-MEMORY ON A SERIAL BY OCTAL DIGIT BASIS. THEREFORE NO COMPLICATED ACCUMULATORS ARE NEEDED, REDUCING HARDWARE TO A MINIMUM. NUMBERS MAY BE AS SHORT AS 2 DIGITS OR AS LONG AS NEEDED. EACH INSTRUCTION CONSISTS OF 10 OCTAL DIGITS.

THE OP CODE IS 2 DIGITS IN LENGTH THE Q ADDRESS SPECIFIES THE MEMORY ADDRESS OF A SOURCE FIELD, AND THE P ADDRESS, A DESTINATION FIELD. IF THESE ADDRESSES ARE MADE FIELDS BY PUTTING A FLAG IN THE LEFTMOST DIGIT, THEY MAY BE MODIFIED BY OTHER INSTRUCTIONS. A FLAG IN THE RIGHTMOST DIGIT OF AN ADDRESS SPECIFIES INDIRECT ADDRESSING IN WHICH THE STRING OF 4 DIGITS ADDRESSED ON THE LEFT BY THE INDIRECT ADDRESS ARE USED INSTEAD FOR THE ADDRESS. AS AN EXAMPLE, CONSIDER THE INSTRUCTION "ADD". THE FIELD AT THE MEMORY LOCATION SPECIFIED BY THE Q ADDRESS IS ADDED DIGIT-BY-DIGIT TO THE FIELD AT THE MEMORY LOCATION SPECIFIED BY THE P ADDRESS UNTIL THE LEFTMOST DIGIT OF THE Q FIELD HAS BEEN ADDED. IF THE CORRESPONDING P DIGIT DID NOT HAVE A FLAG, 7'S OR 0'S ARE ADDED TO P DIGITS DEPENDING ON Q'S SIGN UNTIL A FLAG IS FOUND IN THE P FIELD. THE SUM IS STORED DIGIT-BY-DIGIT AS IT IS FORMED IN THE P FIELD. NOTE THAT THE LENGTH OF THE P FIELD MUST BE EQUAL TO OR GREATER THAN THE LENGTH OF THE Q FIELD.

THE RESULTS OF AN ADD, SUBTRACT, OR COMPARE INSTRUCTION WILL SET THE +, -, AND ZERO INDICATORS WHICH CAN BE TESTED

ALPHAMERIC CHARACTERS CAN BE REPRESENTED AS A PAIR OF DIGITS.

SOME INSTRUCTIONS WILL NOT USE THE Q ADDRESS FOR ANYTHING BUT THEY LOAD IT INTO THE Q ADDRESS REGISTER ANYWAY. THESE UNUSED Q FIELDS CAN HOWEVER BE USED TO STORE CONSTANTS, ETC.

AMLSB = AT THE MEMORY LOCATION SPECIFIED BY THE....

INSTRUCTION SET

OCTAL OP CODE	MNEMONIC	OPERATION
26	MVF	MOVE FIELD THE FIELD AMLSB Q ADDRESS IS MOVED TO MLSB P ADDRESS
25	MVD	MOVE DIGIT THE DIGIT AMLSB Q ADDRESS IS MOVED WITH FLAG TO THE DIGIT POSITION AMLSB P ADDRESS
16	MVFI	MOVE FIELD IMMEDIATE THE Q ADDRESS IS MOVED TO THE FIELD AMLSB P ADDRESS.
15	MVDI	MOVE DIGIT IMMEDIATE THE RIGHTMOST DIGIT OF THE Q ADDRESS IS TO THE DIGIT POSITION AMLSB P ADDR.
21	ADD	ADD THE FIELD AMLSB Q ADDRESS IS ADDED TO THE FIELD AMLSB P ADDRESS. RESULT AMLSB P ADDR.
22	SUB	SUBTRACT
11	ADDI	ADD IMMEDIATE
12	SUBI	SUBTRACT IMMEDIATE
32	SF	SET FLAG A FLAG IS SET AMLSB P ADDR.
33	CF	CLEAR FLAG
24	C	COMPARE THE FIELD AMLSB Q ADDRESS IS SUBTRACTED FROM THE FIELD AMLSB P ADDRESS. THE DIFFERENCE IS DISCARDED.
14	CI	COMPARE IMMEDIA
40	NOP	NO OPERATION
41	BPI	BRANCH POSITIVE INDICATOR
42	BZI	BRANCH ZERO INDICATOR
43	BPZI	+ OR Ø
44	BNI	BRANCH NEGATIVE INDICATOR
45	BNPI	+ OR -
46	BNZI	- OR Ø
47	B	BRANCH IF THE TESTED INDICATORS ARE ON, BRANCH TO INSTRUCTI AMLSB P ADDRESS.
50	NOP	NO OPERATION
51	BP	BRANCH POSITIVE BRANCH TO THE INSTRUCTION AMLSB P ADDRES IF THE FIELD AMLSB Q ADDRES IS POSITIVE.
52	BZ	BRANCH ZERO
53	BPZ	BRANCH + OR Ø
54	BN	BRANCH NEGATIV

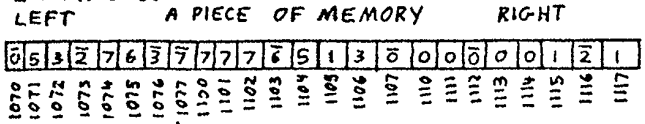
CONTINUED....

MACHINE ORGANIZATION

THE MEMORY HOLDS 4000 OCTAL DIGITS. AN ADDITIONAL BIT IN EACH DIGIT LOCATION IS THE FLAG, WHICH DEFINES BOUNDARIES BETWEEN GROUPS OF DIGITS CALLED FIELDS. THE FLAG CAN ALSO BE USED TO SPECIFY AN INDIRECT ADDRESS IN AN INSTRUCTION. GIVEN A MEMORY ADDRESS, THE FIELD AT THAT ADDRESS IS FOUND BY MOVING LEFT (TO LOWER ADDRESSES) UNTIL A DIGIT WITH THE FLAG SET IS FOUND. FIELDS ARE TREATED AS OCTAL NUMBERS. NEGATIVE OCTAL NUMBERS ARE STORED IN 8'S COMPLEMENT FORM. A FIELD IS NEGATIVE IF THE LEFTMOST DIGIT IS A 4, 5, 6, OR 7. TO COMPLEMENT A NUMBER, SUBTRACT EACH DIGIT FROM 7 AND ADD 1 TO THE RESULTING FIELD.

EXAMPLES:

LEFT A PIECE OF MEMORY RIGHT



THE FIELD $^{15}1072_8$ IS $053_8 = 40_{10}$

THE FIELD AT 1071_8 IS $05_8 = 5_{10}$

THE FIELD AT 1073_8 IS $0532_8 = 346_{10}$

THE FIELD AT 1075_8 IS $276_8 = 190_{10}$

THE FIELD AT 1106_8 IS $6513_8 = -1265_8 = -693_{10}$

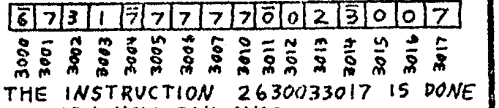
THE FIELD AT 1102_8 IS $7777_8 = -0001_8 = -1_{10}$

THE FIELD AT 1115_8 IS $0001_8 = 1_{10}$

THE FIELD AT 1077_8 IS $37_8 = 31_{10}$

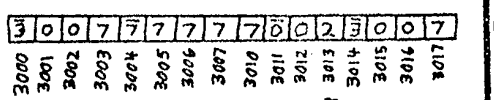
INSTRUCTION EXAMPLES

ASSUME MEMORY CONTAINS:



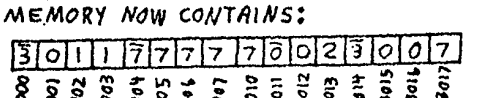
THE INSTRUCTION 2630033017 IS DONE

MEMORY NOW CONTAINS:



THE INSTRUCTION 1130030002 IS DONE

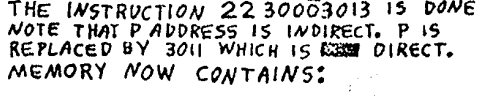
MEMORY NOW CONTAINS:



THE INSTRUCTION 2230003013 IS DONE

NOTE THAT P ADDRESS IS INDIRECT. P IS REPLACED BY 3011 WHICH IS DIRECT.

MEMORY NOW CONTAINS:



THE INSTRUCTION 4546500000 WILL RESULT IN A BRANCH TO 4650 BECAUSE THE RESULT OF THE LAST ARITHMETIC OPERATION WAS NOT ZERO.

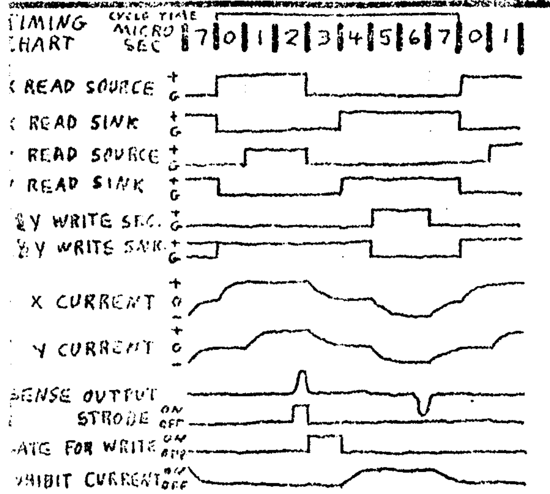
MEMORY UNIT - 4000 WORDS OF 4 BITS EACH

GENERAL

THE MEMORY UNIT CAN BE BUILT FROM A STACK OF 4 1401 MEMORY PLANES AVAILABLE FOR [illegible] THESE PLANES HAVE 4000 CORES ON A 50 X 80 MATRIX. 2 SENSE LINES THREAD 2000 CORES EACH ON A 50 BY 40 MATRIX. 2 INHIBIT LINES PARALLEL THE 2 SENSE LINES. THE 2 INHIBIT LINES MAY B CONNECTED IN SERIES BUT EACH SENSE LINE SHOULD HAVE ITS OWN SENSE AMPLIFIER.

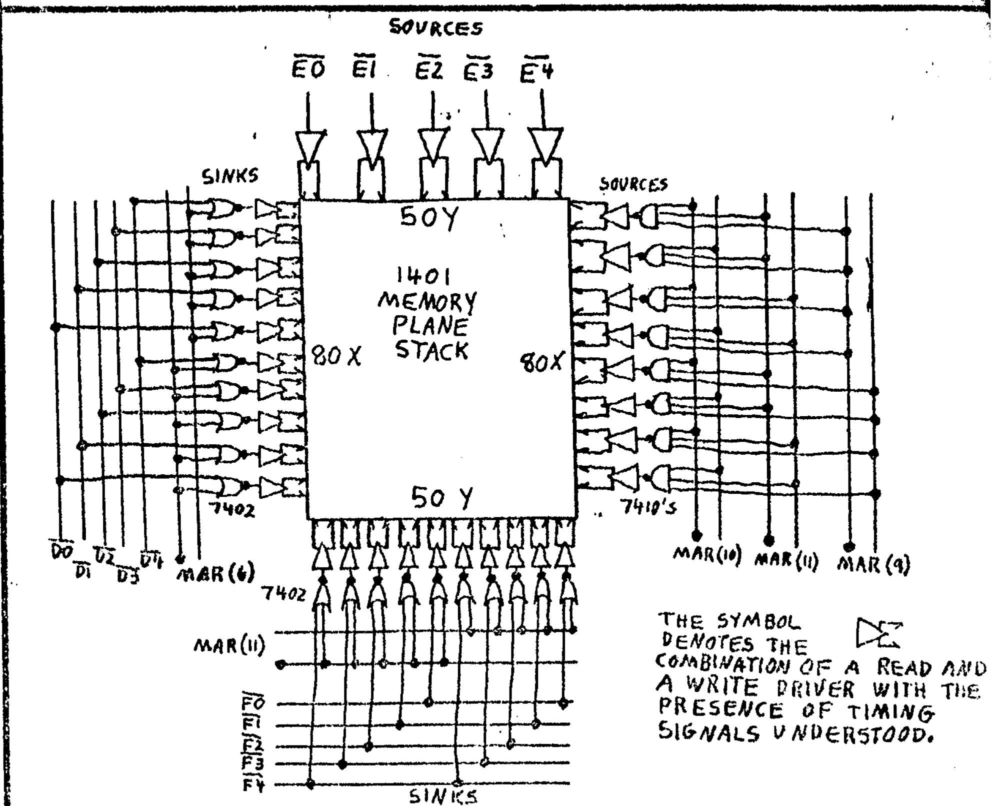
HALF-SELECT CURRENT IS 250-300 MA. THE 50 MIL O.D. CORES SWITCH WITHIN 1 μSEC OF APPLICATION OF FULL SELECT CURRENT. SENSE SIGNAL IS ABOUT 30 MV.

Perfect Surplus
Frames sell at $8.50 to 8.95 each or $25 to $34-4 frame



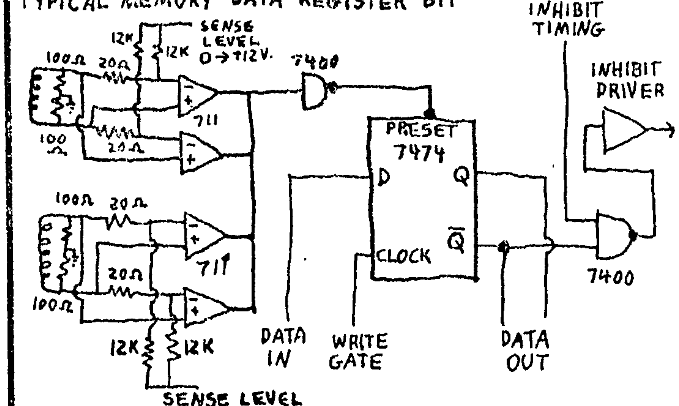
G = GROUND POSITIVE CURRENT IS READ.

THE SENSE WINDING RUNS PARALLEL TO THE 80 DRIVE LINES. THE Y READ CURRENT IS STAGGERED 1μS SO THAT NOISE COUPLED TO THE SENSE LINE FROM X SWITCHING WILL DIE OUT BEFORE THE CORE IS SWITCHED BY THE FULL SELECT CURRENT. THE READ AND WRITE SINKS ARE LEFT ON 1μS AFTER THE CORRESPONDING SOURCES ARE SWITCHED OFF SO THAT THE INDUCTIVE ENERGY STORED IN THE DRIVE LINE CAN DISSIPATE ITSELF IN THE SOURCE LOAD RESISTOR.
THE CYCLE TIME CAN BE REDUCED TO 4μS IF THE DRIVE VOLTAGES ARE INCREASED AND GREAT CARE IS TAKEN.

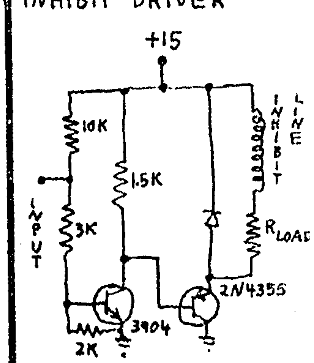


SOURCES
E0 E1 E2 E3 E4
SINKS
SOURCES
50Y
1401 MEMORY PLANE STACK
80X
80X
50 Y
7402
7410's
D0 D1 D2 D3 D4
MAR(6)
MAR(10) MAR(11) MAR(9)
7402
MAR(11)
F0 F1 F2 F3 F4
SINKS
THE SYMBOL DENOTES THE COMBINATION OF A READ AND A WRITE DRIVER WITH THE PRESENCE OF TIMING SIGNALS UNDERSTOOD.

TYPICAL MEMORY DATA REGISTER BIT



INHIBIT DRIVER



ADDRESSING LOGIC

SINCE 1401 PLANES ARE NOT 64 BY 64 WITH 4096 CORES, SOME COMBINATIONAL LOGIC IS NECESSARY TO CONVERT THE 12 BIT ADDRESS IN THE MEMORY ADDRESS REGISTER (MAR) TO AN X AND Y LINE SELECTION. THE LOGIC SHOULD BE SUCH THAT ADDRESSES 0 TO 3999_{10} (0000 TO 7637_8) ARE UNIQUE AND CONTIGUOUS. ADDRESSES 4000 TO 4095 SHOULD PRODUCE ZEROES.

BY FACTORING THE 2'S FROM THE NUMBER OF X AND Y DRIVE LINES AND ASSIGNING THEM TO MAR BITS 7-11, THE COMBINATIONAL LOGIC IS REDUCED TO 7 VARIABLES. LET T=MAR(0), U=MAR(1), V=MAR(2), W=MAR(3), X=MAR(4), Y=MAR(5), AND Z=MAR(6). THE TOTAL LOGIC NETWORK IS AND-OR-NAND WHICH IS REALIZABLE WITH ALL NAND GATES. DEFINE THE VARIABLES A1, A2, A4; B1, B2, B4; C1, C2, C4 WHICH ARE GENERATED BY THE FIRST 2 LEVELS AND DEFINE D0, D1, D2, D3, D4; E0, E1, E2, E3, E4; F0, F1, F2, F3, F4 WHICH ARE GENERATED BY THE THIRD LEVEL.

$A1 = \bar{T}V + \bar{U}VX + TU\bar{V}X + VXZ.$
$A2 = \bar{T}U + U\bar{U}W + T\bar{U}VW + UVXY$
$\overline{A4} = \bar{T} + \bar{U}V + U\bar{V} + U\bar{W}X$
$B1 = \bar{T}X + \bar{U}\bar{V}X + TU\bar{V}Z + TUWZ$
$B2 = \bar{T}W + \bar{U}\bar{V}W + TU\bar{V}Y + UWY$
$\overline{B4} = \bar{T} + \bar{V} + UW\bar{X}$
$C1 = \bar{T}Z + \bar{U}Z + V\bar{W}\bar{X}Z$
$C2 = \bar{T}Y + \bar{U}Y + V\bar{W}\bar{X}Y$
$\overline{C4} = \bar{T} + \bar{U} + V\bar{W}\bar{X}$

$D0 = \overline{A1}\ \overline{A2}\ \overline{A4}$
$D1 = A1\ \overline{A2}\ \overline{A4}$
$D2 = \overline{A1}\ A2\ \overline{A4}$
$D3 = A1\ A2\ \overline{A4}$
$D4 = \overline{A1}\ \overline{A2}\ A4$

$E0 = \overline{B1}\ \overline{B2}\ \overline{B4}$
$E1 = B1\ \overline{B2}\ \overline{B4}$
$E2 = \overline{B1}\ B2\ \overline{B4}$
$E3 = B1\ B2\ \overline{B4}$
$E4 = \overline{B1}\ \overline{B2}\ B4$

$F0 = \overline{C1}\ \overline{C2}\ \overline{C4}$
$F1 = C1\ \overline{C2}\ \overline{C4}$
$F2 = \overline{C1}\ C2\ \overline{C4}$
$F3 = C1\ C2\ \overline{C4}$
$F4 = \overline{C1}\ \overline{C2}\ C4$

THE SOURCES AND SINKS ARE ARRANGED IN A RECTANGULAR MATRIX AS SHOWN TO THE RIGHT. THE -ww- ARE MAIN DRIVE LINES. ONE DRIVE LINE IN THE MATRIX IS SELECTED FOR READ BY DRIVING ONE READ SOURCE TO +15V. AND DRIVING ONE READ SINK TO GROUND. DRIVING THE CORRESPONDING WRITE SOURCES AND SINKS SELECTS THE SAME DRIVE LINE BUT DRIVES CURRENT IN THE REVERSE DIRECTION. THERE ARE 2 SUCH MATRICES, ONE FOR X LINES AND ONE FOR Y. THE SOURCES AND SINKS ARE REPRESENTED AS SPDT SWITCHES.