bcc	title	prefix/class-number.revision
	MICROPROCESSOR TEST PROGRAM	MPT/W- 44

checked	authors	approval date	revision date
checked	Dieter A. Susset	2-24-70	
approved		classification: Working Paper	
		distribution: Company Private	pages: 60

ABSTRACT and CONTENTS

This document tries to give a guideline through the basic test procedure of the Microprocessor. It is written in a way that someone with little knowledge in the operations of computers and of programming will be able to understand what is going on. Tables of signals and pin locations are intended to help enable someone to perform the test without having to collect the necessary information from different documents.

This document is organized in the order of the instruction addresses of the test program itself.

GENERAL INFORMATION

a) Sequence of Execution in the MICROPROCESSOR:

At one time the outputs of the O-Register (OR), the Read Only Memory (ROM) and the I-Register (IR) will make the following signals available.

OR :ADDRESS of next instruction to be executed

ROM:ROM-word addressed by OR

IR :ROM-word presently executed

1) In a "normal instruction" OR is incremented by 1.

2) An "unsuccessful branch" (GOTO with the branch condition not existing) acts like a normal instruction.

3) A "successful branch" (GOTO with the branch condition existing) means the OR is reloaded with the address of the instruction to be branched to, during the present instruction, at the end of the first machine cycle and the loading of IR is inhibited.

4) A "deferred branch" (DGOTO) means that the address of the instruction to be branched is loaded into OR at the end of the present instruction. This means the next instruction will still be executed and determines fully the continuation of the program.

5) A TAX causes State A to be 2 cycles long by making SC true for 100ns.

b) Program Loop Diagram:

Shows the sequence in which instructions are executed or fetched during a certain program loop.

In the following, these diagrams always have a common format and common symbolism which is used.

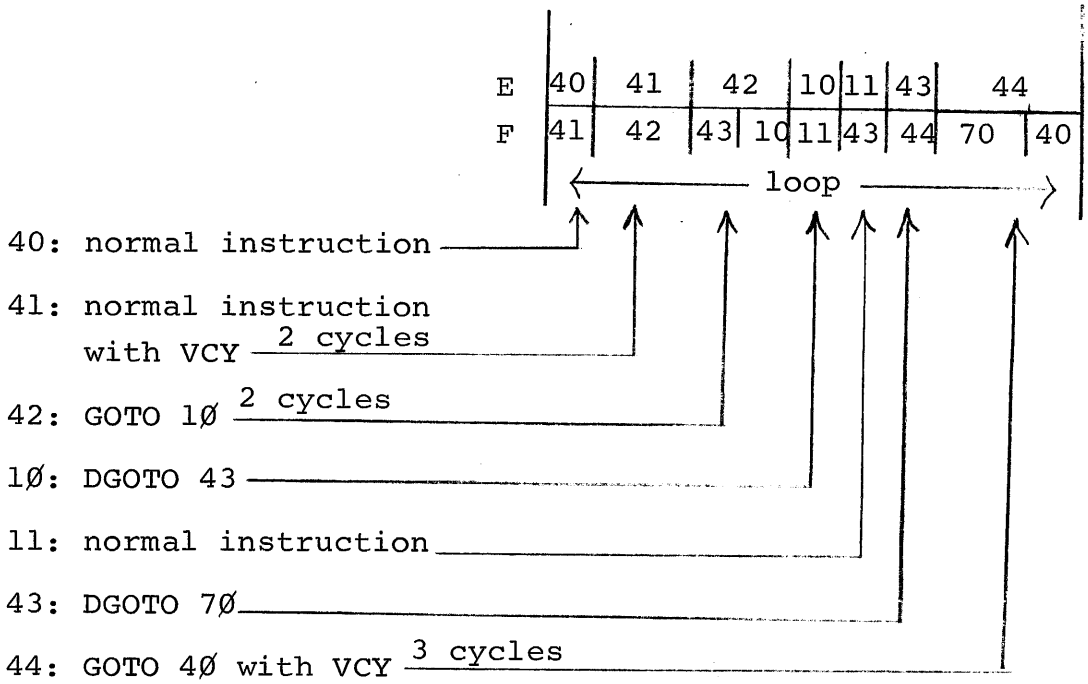
Format: 1 square = 1 machine cycle |↔|

Symbols: E = EXECUTE = time during which the instruction is executed.

F = FETCH = time during which the instruction is fetched to be loaded into the I-REGISTER.

Number: ADDRESS of instruction

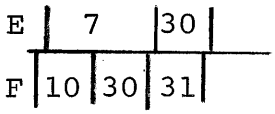
EXAMPLE:



NOTE: See a) also!

Actually, EXECUTE and FETCH is slightly shifted, since the O-Register is loaded at the END of the instruction.

It would look like this:



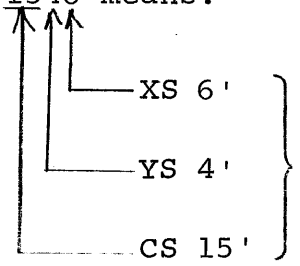
NOTE

Fetch is used to trigger the scope!

c) Address (output from O-Register)

Trigger on "1546" or Trigger on fetch of address

1546 means:



For signal pins see table on page 6

also, Trigger on "Ø" means:

XSØ' - YSØ' - CSØ'

NOTE: To trigger use 4-input trigger box in "NOR" position (light off) - TRIGGER on "+"!

d) Check of Address:

E	1	2	3
F	2	3	1

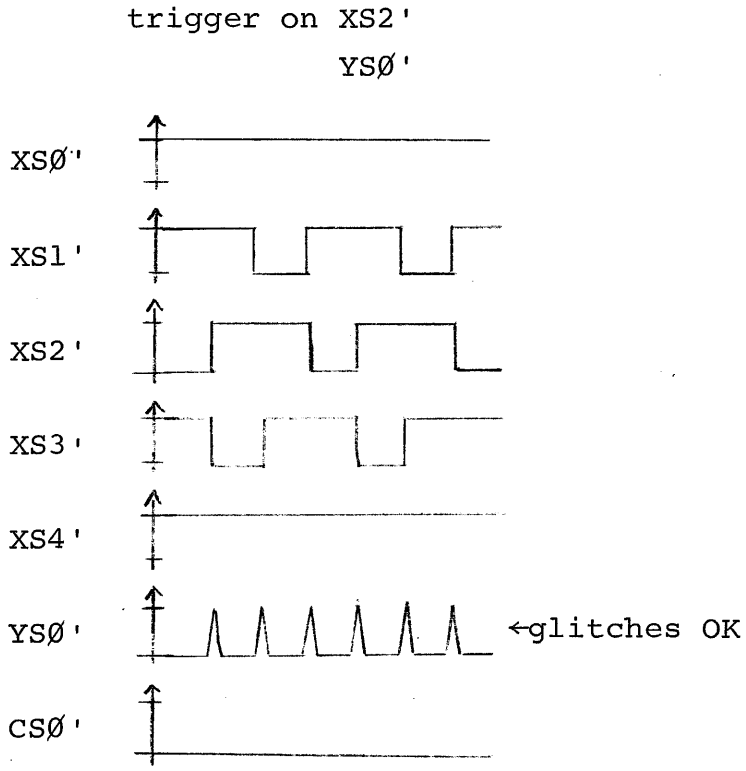
EXAMPLE

for check of

instruction loop

(see page #6 for

pin locations)



	← loop →			← loop →			← loop
E	1	2	3	1	2	3	1
F	2	3	1	2	3	1	2

TABLE OF PIN LOCATIONS

Signal	Card #	Pin #	Signal	Card #	Pin #
I CLOCK A					
CS Ø'	B25	2	XS Ø'	B25	15
CS 1'	B24	2	XS 1'	B25	13
CS 2'	B23	2	XS 2'	B25	11
CS 3'	B22	2	XS 3'	B25	9
CS 4'	B21	2	XS 4'	B25	7
CS 5'	B2Ø	2	XS 5'	B25	5
CS 6'	B19	2	XS 6'	B25	3
CS 7'	B18	2	XS 7'	B25	1
CS1Ø'	B11	2	YS Ø'	B25	18
CS11'	B1Ø	2	YS 1'	B25	16
CS12'	B 9	2	YS 2'	B25	14
CS13'	B 8	2	YS 3'	B25	12
CS14'	B 7	2	YS 4'	B25	1Ø
CS15'	B 6	2	YS 5'	B25	8
CS16'	B 5	2	YS 6'	B25	6
CS17'	B 4	2	YS 7'	B25	4
I CLOCK B					
CS Ø'	B33	2			
CS 1'	B32	2			
CS 2'	B31	2			
CS 3'	B3Ø	2			
CS 4'	B29	2			
CS 5'	B28	2	CS1Ø'	B3	2
CS 6'	B27	2	CS11'	B2	2
CS 7'	B26	2	CS12'	B1	2

NOTE: I CLOCK A: ADDRESS 0B to 1777B

I CLOCK B: ADDRESS 2000B to 3277B

XS-, YS - SIGNALS: Check also cards B3, B11, B26. Pins are the same as on B25.

e) Numbers:

$26B = 26_8 = 10\ 110_2$
octal octal binary

$2B6 = 2\ \underbrace{000000}_{6}\ B = 2000000_8$

24 BIT REGISTERS: Most significant bit = BIT(Ø) = Sign BIT

Bit(Ø) = Ø → positive number

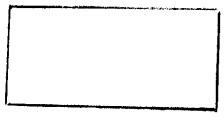
Bit(Ø) = 1 → negative number

OB to 37777777B → positive numbers

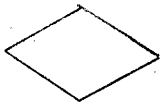
77777777B to 40000000B → negative numbers
(-1B to -40000000B)

f) CONVENTIONS FOR LOGIC BLOCK DIAGRAMMING

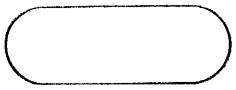
PROCESSING FUNCTION
(or assignment)



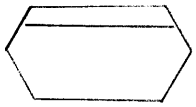
DECISION FUNCTION
(according to a specified condition)



INPUT OR OUTPUT FUNCTION
(or initialization)



SUBROUTINE FUNCTION
(Call)



ENTRY OR EXIT
(to or from other pages or parts of the program)

g) Symbols in MICRO CODE

A ✓ B = A ← B = Load "B" into "A"

VCY = VCYP

h) Two cycle instructions

TAX makes STATEA two cycles long.

VCY (no diode in ROM bit 86)

causes inhibit of the register clocks during STATEA and a transition into STATEB.

Note: All "VCY" notations in MICRO CODE mean really "VCYP", one diode in ROM bit 86.

i.) Signal Levels:

On all signals, high $\geq$ 2.4V
low $\leq$ 0.4V
} trouble if different!

All timing is measured at the 1.5V threshold point!

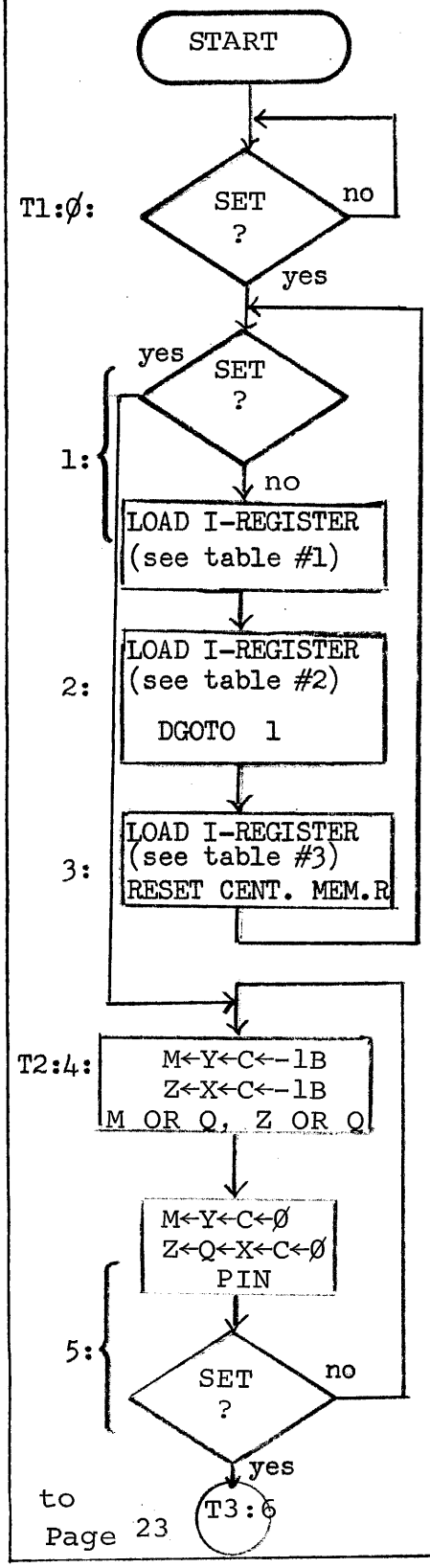
Preliminary Tests

a) Before starting the Microprocessor Test Program, make sure the test ROM board was tested and all bits switch within the specified limits. In case of an untested test ROM board the board has to be tested according to the "Microprocessor ROM Acceptance Procedure". Memo from Chuck Thacker, January 12, 1970.

b) Without a ROM board - check whether the O-Register is properly incrementing in two-cycle steps. Check XS, YS, and CS pins. See page 6 for pin locations.

TEST OF ATTENTION LATCH #1, ROM BITS & I-REGISTER BITS

FLOW CHART



to Page 23

REMARKS

START

Switch power on or push RESET.

INITIAL WAIT LOOP

GOTO Ø IF NOT SET:

Refers to ATTENTION LATCH #1.

Pushing ADVANCE BUTTON will set the latch and step program to next instruction.

TEST OF #1 BITS

GOTO 4 IF SET:

(See instruction "Ø") Goes to test 2 if set.

Bits according to table #1 are loaded into I-REGISTER.

TEST OF #2 BITS

Bits according to table #2 are loaded into I-REGISTER. DGOTO 1 loads 1 into O-REGISTER at the end of the instruction.

TEST OF #3 BITS

Bits according to table #3 are loaded into I-REGISTER. RESET CENTRAL MEMORY REQUEST is set.

TEST OF #4 BITS (C-FIELD)

Load M with "-1B" from Y and C.

Load Z with "-1B" from X and C. Bool boxes are set to OR

BL', BR' show "Ø"

Load M with "Ø" from Y and C. Load Z and Q with "Ø" from X and C. PIN is set.

BL', BR' show "-1B"

Test finished ?

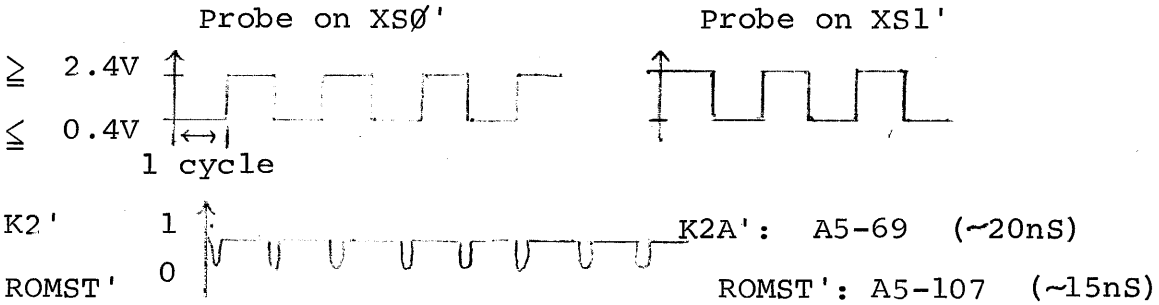
(push ADVANCE button)

T1:Ø: Tests whether ATTENTION LATCH #1 is reset: *)

After switching power on or pushing the RESET button on the Microprocessor, the latch should be reset and the program should loop in instruction Ø.

EXECUTE: E | Ø | trigger on XSØ'

FETCH: F | 1 | Ø | YSØ'



Check levels of RESET: A1-78 & RESET':A1-76 and ADVNO:A9-62 & ADVNC:A9-61 while pushing RESET and ADVANCE buttons.

If OK, push the ADVANCE-button on the Microprocessor to step to the next instruction (the trigger should disappear now).

NOTE: In the following tests always first check the ADDRESS according to the PROGRAM LOOP DIAGRAM to make sure the program is in the right loop. (See page 5).

*) BOARDS REQUIRED:
OOS-REGISTER, LOCAL CONTROL, CONTROL LOGIC, BRANCH CONDITION, I-REGISTERS.
JUMPER from A10-75 to A10-86.
Tie A10-93 with 150Ω RESISTOR to +5V

Test Speed 10.5 MHZ

1, 2, 3 : & T2 :4, 5: Test of I-Registers and X- and Y- Bus.

(See also pages 14 to 19).

1) With test ROM board in slot B25: Check I REGISTER CLOCKS (BANKA)

Perform test a) on RAØ-pins and I3 CLOCKA'

Perform test b)

push ADVANCE

Perform test c) on RAØ-pins and I3 CLOCKA'

Perform test d)

2) With test ROM-board in slot B33 and B-SELECT (A7-38) grounded: Check I REGISTER CLOCKS (BANKB)

Perform test a) on RBØ-pins and I3 CLOCKB'

Perform test b)

push ADVANCE

Perform test c) on RBØ-pins and I3 CLOCKB'

Perform test d)

3) With test ROM board in slot B11 and CS1Ø' (A11-116) taped and (B11-2) grounded:

Perform test a) on RA1-pins and I3 CLOCKA'

Perform test b)

push ADVANCE

Perform test c) on RA1-pins and I3 CLOCKA'

Perform test d)

4) (See page 19)

BIT # → X 4 ← BIT #4
CLOCK → X 3 ← K3 } C-FIELD

	INPUT PIN #									OUTPUT PIN #	
RAØ	RA1	RBØ	RB1	B12	B13	B14	B15	B16	B17	True	False
5	6	7	8	1 2	1 2	2 3	2 3	X	1 3	4	3
11	12	13	14	1 2	1 2	1 2	1 2	1 2	1 2	10	9
19	20	21	22	3 3	3 3	3 3	3 3	3 3	3 3	18	17
25	26	27	28	3 2	3 2	3 2	3 3	3 3	3 3	24	23
33	34	35	36	3 3	3 2	3 2	2 2	2 2	2 2	32	31
39	40	41	42	3 2	3 2	3 2	3 3	3 3	2 2	38	37
47	48	49	50	1 3	X	3 2	3 2	1 2	3 2	46	45
75	76	77	78	1 3	1 3	1 2	3 3	1 3	1 3	73	74
81	82	83	84	4 3	4 3	4 3	4 3	4 3	4 3	79	80
89	90	91	92	4 3	4 3	4 3	4 3	4 3	4 3	87	88
95	96	97	98	4 3	4 3	4 3	4 3	4 3	4 3	93	94
103	104	105	106	4 3	4 3	4 3	4 3	4 3	4 3	101	102
109	110	111	112	3 3	3 3	1 3	3 3	3 3	2 3	107	108
117	118	119	120	1 3	3 3	3 3	3 3	3 3	3 3	115	116
123	124	125	126	1 2	3 2	1 2	1 2	1 2	2 2	121	122

* { RA0 RA1 RB0 RB1 B12 B13 B14 B15 B16 B17

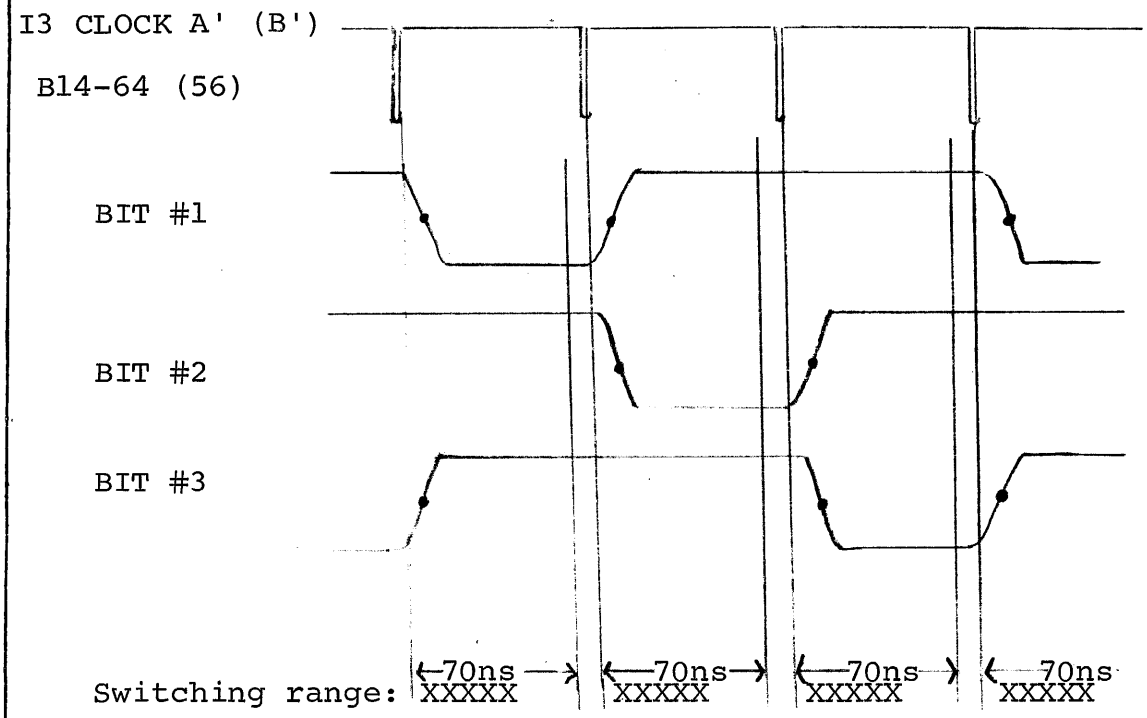
*C-FIELD: tested after pushing of ADVANCE

Note: TAX and VCYP cause a one cycle delay and are tested later.

a) Test of I-Register inputs

E	1	2	3
F	2	3	1

trigger on XS2
YSØ



Check all bits for right timing according to picture (up and down going edges must switch within 70ns period).

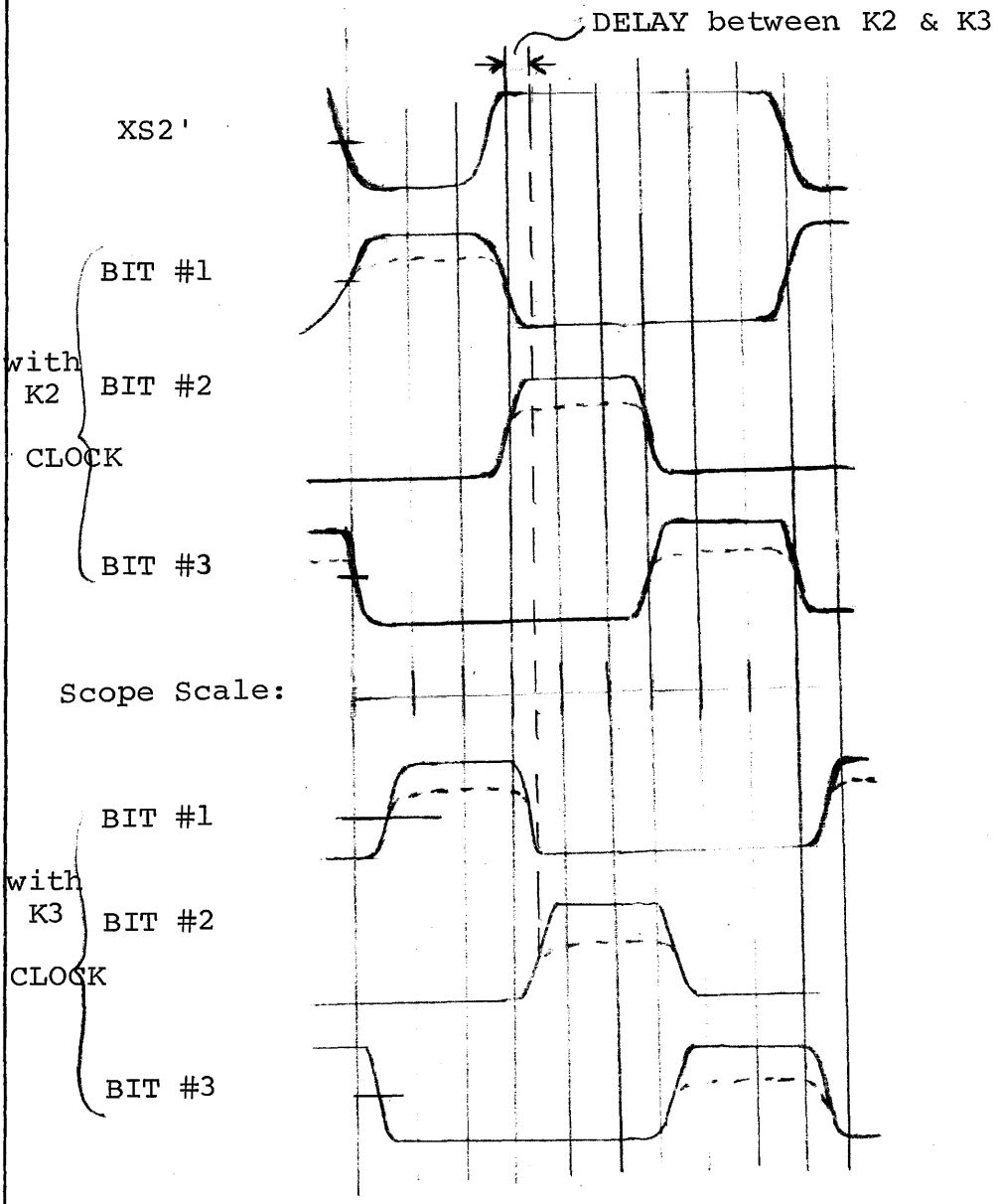
Check levels (0.4V, 2.4V)

b) Test of I-Register Outputs

E	1	2	3
F	2	3	1

trigger on XS2
YSØ

picture for true outputs!



Set scope to 9 divisions for full loop of XS2'.

Compare relationship of K2 clocked signals and K3 clocked signals to XS2'. (K3 bits are delayed & switch later.)

Check levels on true and false outputs.

c) Test of I-Register inputs (C-Field)

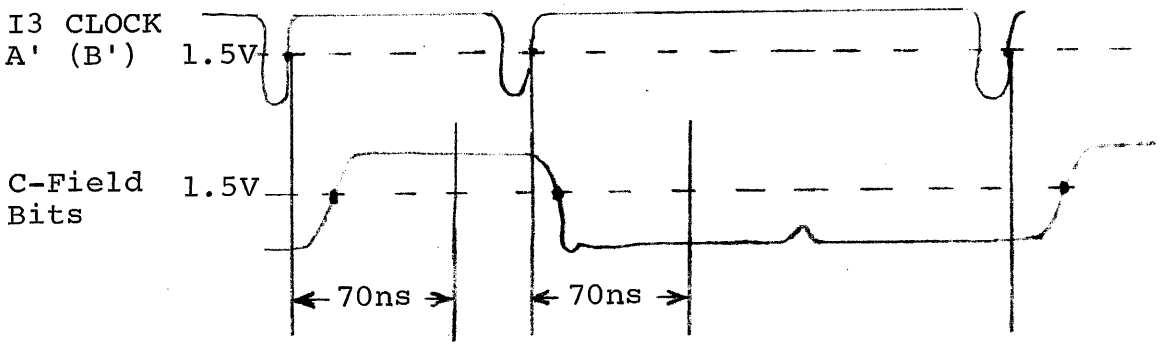
E	4	5	
F	5	6	4

trigger on XS5'
YSØ'

This test loads alternately -1B (all ones) and Ø from the C-field onto the X-bus to the M-Register and onto the Y-bus to the Z-Register. This checks X-Bus, Y-Bus and proper loading of the M- and Z-Registers. Q = Ø at all times. The bool boxes are set to M or Q, Z or Q . PIN is set.

Probe #1 on I3 CLOCK A' (B14-64)
or I3 CLOCK B' (B14-56)

Probe #2 on Input Pin according to test and to table 4.

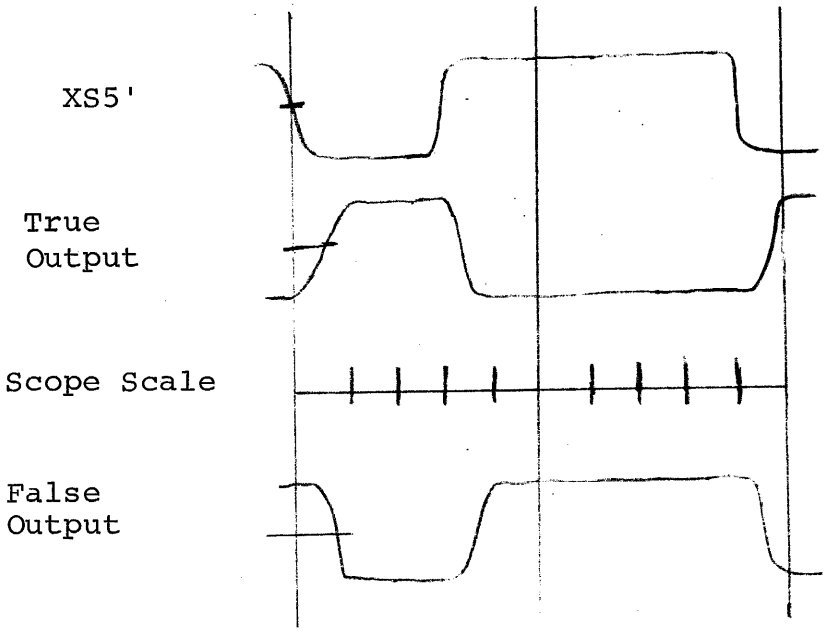


Check all bits for right timing according to picture (up and down going edges must switch within 70ns period).

d) Test of I-Register Outputs (C-Field)

E	4	5	
F	5	4	6

trigger on XS5, YSØ



Set scope to 9 divisions for full loop of XS5.

Check relationship between XS5' signal and output signals according to picture. Check signal levels.

4) With test ROM board in slot B3 and CS1Ø' (A11-116) taped and (B11-2) and B-SELECT (A7-38) grounded:

Perform test a) on RB1-pins and I3 CLOCKB'

Perform test b)

push ADVANCE

Perform test c) on RB1-pins and I3 CLOCKB'

Perform test d)

Note: RAØ... signals connected to boards B18 to B25

RA1... B04 to B11

RBØ... B26 to B33

RB1... B01 to B03

5*) Test of X- and Y- Bus:

With test ROM board in slot B25 and taped and grounded pins put back to normal:

E	4	5	
F	5	6	4

trigger on XS5, YSØ

Check signals and signal level according to picture below.

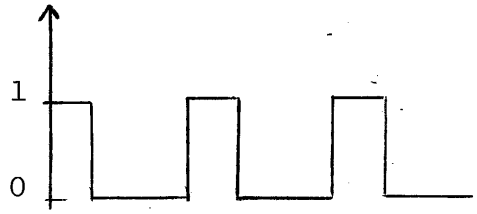
For pin locations see table on page 21.

*) ADDITIONAL BOARDS REQUIRED:

MQZ - REGISTERS, ADDERS, HOLDING REGISTERS, SPECIAL FUNCTION, MEMORY INTERFACE, REQUEST STROBE.

All other boards if available. Remove jumpers from A10 before board is plugged in.

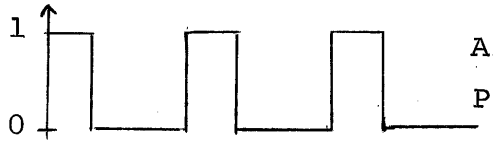
Y, X



1 ≧ 2.4V

Ø ≦ 0.4V

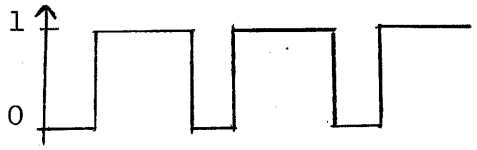
C



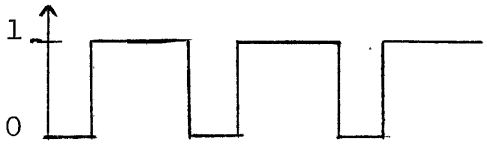
A25-, A20-, A14-

Pins 23,35,47,59,83,95,107,115

M,Z

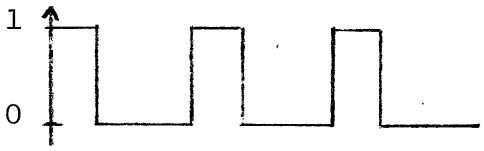


MS2, MS3
MS4,
B(7),
PIN'

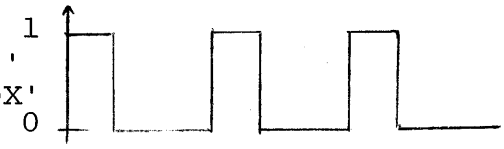


MS2: A7-115
MS3: A7-91 B(7):A11-23
MS4: A7-89 PIN':A29-04

BR', BL'

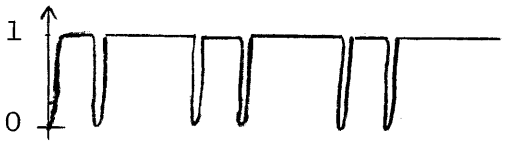


MS2',MS3'
MS4', LQX'
PIN



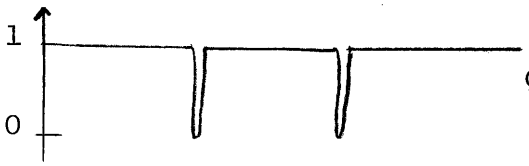
MS2': A7-111
MS3': A7-97 LQX':A23-123
MS4': A7-95 PIN:A28-115

MCLOCK3'
ZCLOCK'

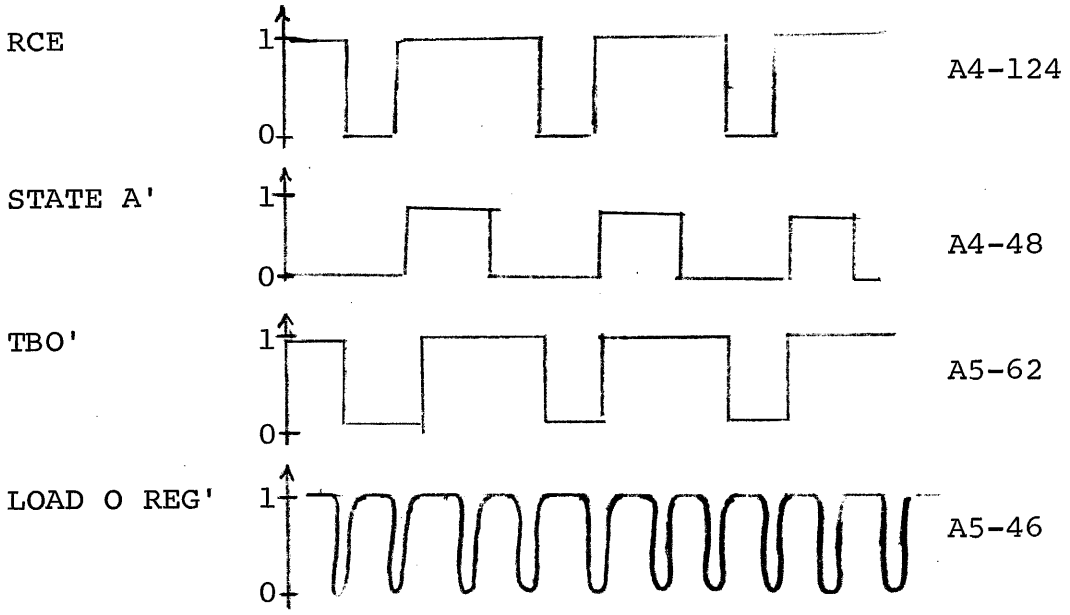


MCLOCK3': A23-09
ZCLOCK': A23-13

QCLOCK'



QCLOCK':A23-06



If OK, step to next test.

TABLE OF RELATIONSHIPS TO BIT NUMBERS

BIT #	DIGIT (OCTAL)	X CARD A9	Y CARD A10	BL' CARD A9	BR' CARD ↙	M CARD A3	Z CARD A28	RØ CARD A3
Ø	4B7	15	58	29	A23-114	1	1	50
1	2B7	16	108	31	" -102	2	3	51
2	1B7	17	96	33	" - 90	3	5	52
3	4B6	18	84	35	" - 78	4	7	53
4	2B6	19	61	37	" - 54	5	9	54
5	1B6	20	48	39	" - 42	6	11	55
6	4B5	21	36	30	" - 30	7	13	56
7	2B5	22	43	32	" - 18	8	15	57
8	1B5	23	120	34	A18-114	9	17	58
9	4B4	24	109	36	" -102	10	19	59
1Ø	2B4	25	97	38	" - 90	11	21	60
11	1B4	26	85	40	" - 78	12	23	61
12	4B3	3	62	41	" - 54	13	25	62
13	2B3	5	49	42	" - 42	14	27	63
14	1B3	7	37	43	" - 30	15	29	64
15	4B2	9	28	44	" - 18	16	31	65
16	2B2	12	121	45	A12-114	17	33	66
17	1B2	14	107	46	" -102	18	35	67
18	4B1	4	95	47	" - 90	19	37	68
19	2B1	6	83	48	" - 78	20	39	69
2Ø	1B1	8	60	49	" - 54	21	41	70
21	4B	10	55	50	" - 42	22	43	71
22	2B	11	56	51	" - 30	23	45	72
23	1B	13	57	52	" - 18	24	47	73

BRANCH': A9 - 69 SC; A4 - 28 BSELECT: A7-38

TEST OF M-,Q-,Z-REGISTER & BOOL BOX

LABEL	ADDRESS	FLOWCHART	REMARKS
		from Page 11	
		5	TEST 1 of M-,Q-,Z-REGISTERS AND M or $\overline{Q}$, Z or $\overline{Q}$ BOOL BOX
T3	6	Z←Q←Y←C← -1B M←X←C← -1B M or $\overline{Q}$, Z or $\overline{Q}$	M-, Z- REGISTERS are loaded with "-1B". BL', BR' show "Ø".
	7	Z←Y←C←Ø M←X←C←Ø M or $\overline{Q}$, Z or $\overline{Q}$ POT	M-, Z- REGISTERS are loaded with "Ø". BL', BR' show "-1B" POT is on.
		SET ? — no	test o.k.? push ADVANCE
		yes	
T4	1Ø	$\overline{Q}$←RØ←Y←C←Ø $\overline{M}$ or Q, $\overline{Z}$, or Q	TEST 2 of M-,Q-,Z-REGISTERS and $\overline{M}$ or $\overline{Q}$, $\overline{Z}$ or Q BOOL BOX Q, RØ are loaded with Ø BL' & BR' pins show "-1B" RØ pins show "Ø"
	11	M←Q←Z←Y←C← -1B RØ←Y←C← -1B M or Q, Z, or Q ALERT	M-, Q-, Z- & RØ-REGISTERS are loaded with -1B BL' & BR' pins show "Ø". RØ pins show "-1B', ALERT is on.
		SET ? — no	
		yes	test o.k.? push ADVANCE
		T5:12	
		to Page 26	

T3: 6 & 7: TEST 1 of M.Q.Z. REGISTER

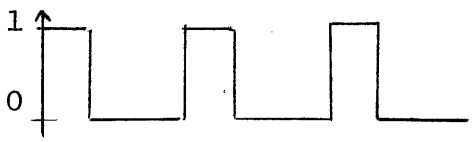
E	6	7	
F	7	10	6

trigger on XS7'
YSØ'

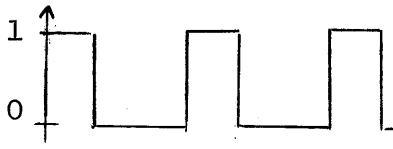
This test loads alternately -1B and ØB into the M- and Z-Registers, while the Q Register is always loaded with -1B. The bool boxes are set to M or Q' (BL = 5B) and Z or Q' (BR = 5B). Since Q = -1B, Q' = ØB therefore M' is displayed on the BL' pins, Z' is displayed on BR' pins. Also the SPECIAL FUNCTION "POT" (MS = 15B) is set. Note that the registers are loaded at the end of the instruction.

For pin locations other than given, see table on page 21.

BL', BR'

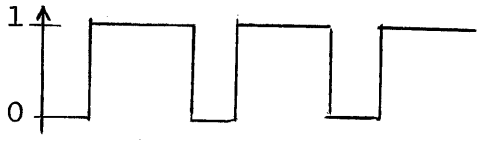


GM'



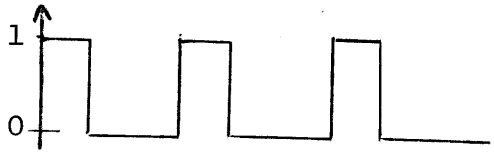
A 14P-1 to -24
(IF CPU check on MQZ Reg. cards)

LQY', B(8)
MS5, POT'



B(8): A11-09
LQY': A23-125
MS5: A7-87
POT': A29-02

MS5', POT



MS5': A7-93
POT: A28-113

If O.K. step to next test.

T4: 1Ø & 11: TEST 2 of M.Q.Z. REGISTER

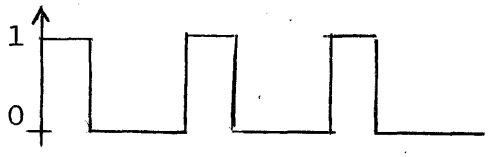
E	10	11	
F	11	12	10

trigger on XS1'
YS1'

This test loads alternately ØB and -1B into the Q- and RØ- Registers, while the M- and Z- Registers are always loaded with -1B.

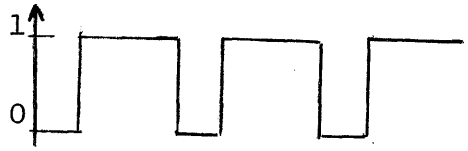
The bool boxes are set to M' or Q (BL = 3B) and Z' or Q (BR = 3B). Since M = -1B and Z = -1B therefore M' = Ø and Z' = Ø and Q' is displayed on the BL' and BR' pins. Also the SPECIAL FUNCTION "ALERT" (MS = 14B) is set.

RØ



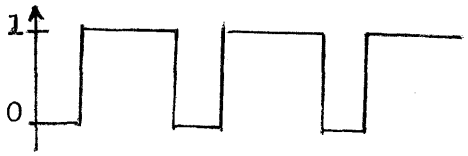
See page 21

BL', BR'



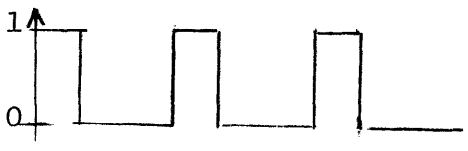
See page 21

LMY, B(6)
MC4
MC3, MC1
ALERT'



LMY: A4-02 MC1: A9-126
B(6): A11-37 MC3: A9-119
ALERT': A29-03 MC4: A9-120

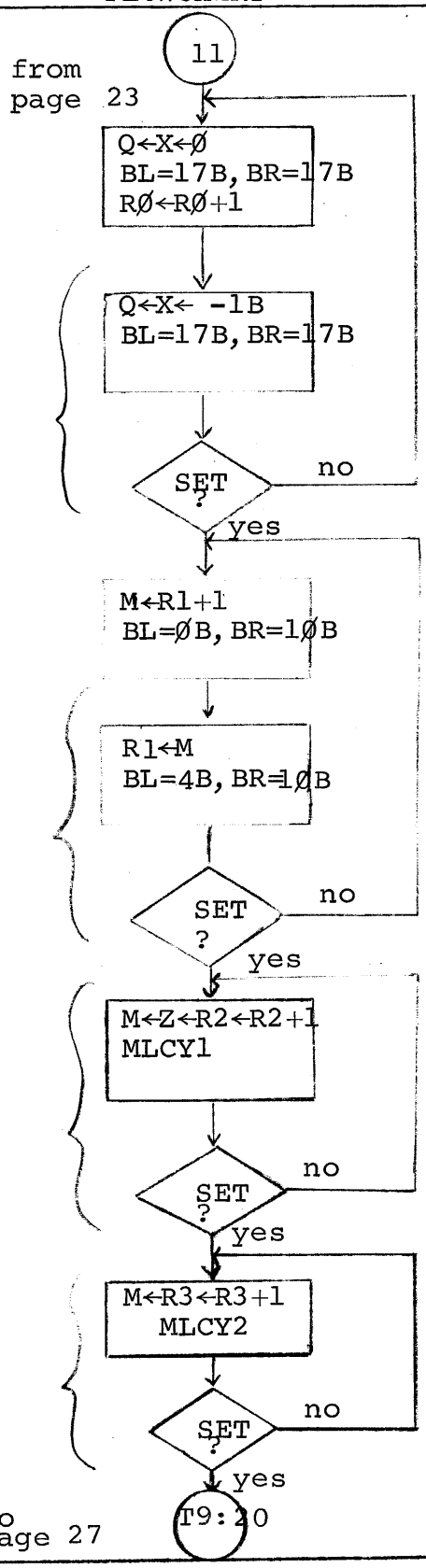
LMY',LMYA'
LZY', MC4'
MC3', MC1'
ALERT



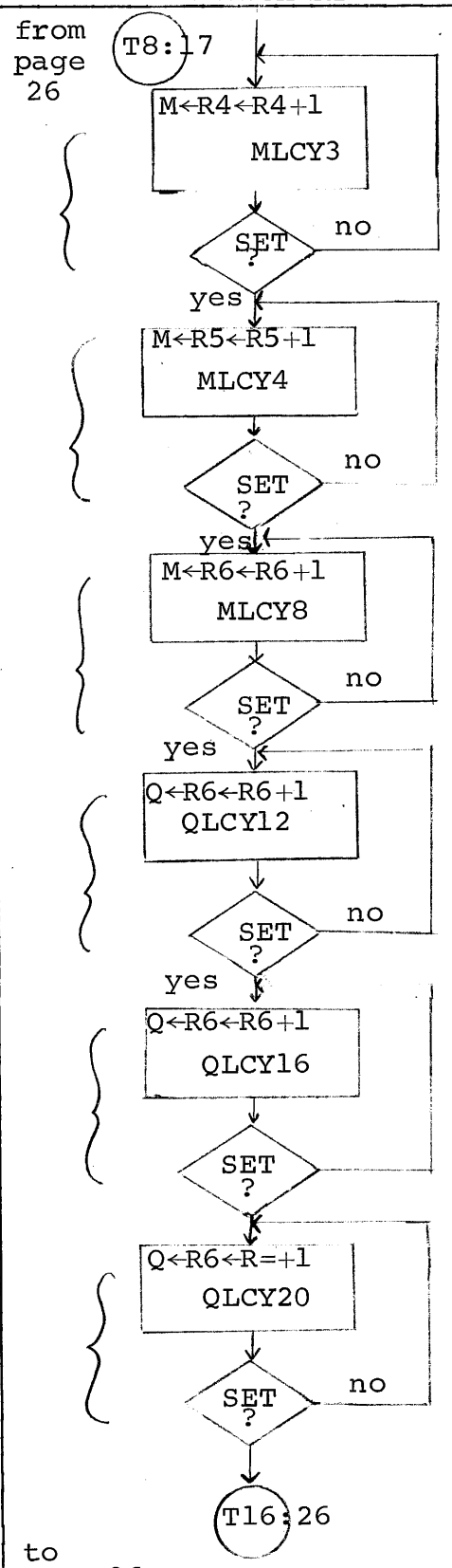
LMY' : A5-21 MC1':A9-122
LMYA': A23-88 MC3':A9-115
LZY' : A23-70 MC4':A9-118
ALERT: A28-107

If O.K. step to next test

TEST OF HOLDING REGISTER & LEFT CYCLING

LABEL	ADDRESS	FLOWCHART	REMARKS
			M = -1B, Z = -1B from last test Test of Bool Box & Increment of RØ Q is loaded with Ø from "C" & "X". RØ is incremented on "Y". Bool Boxes are set to M' or Q', Z' or Q' BL' & BR' pins show Q. Q is loaded with -1B from "C" & "X". Bool Boxes are set to M' or Q', Z' or Q'. BL' & BR' pins show Q. Z= -1B, Q= -1B from last test Test of Bool Box & Increment of R1 Right Bool Box is set to "Ø". R1 is incremented & loaded into "M". All BR' pins show "1'. R1 is loaded from M through the ADDER Test of R2 & Cycle by 1 R2 is incremented and loaded into R2, Z, & M. BL is left cycled by 1 onto the X-Bus. Test of R3 & Cycle by 2 R3 is incremented and loaded into R3 & M and left cycled by 2.
T5	12		
	13		
T6	14		
	15		
T7	16		
T8	17		

TEST OF HOLDING REGISTER & LEFT CYCLING

LABEL	ADDRESS	FLOWCHART	REMARKS
			

T9 2Ø — Test of R4 & cycle by3
R4 is incremented and loaded into R4 & M. M is left cycled by 3 onto the X-Bus.

T1Ø 21 — Test of R5 & cycle by 4
Test as before.

T11 22 — Test of R6 & Cycle by 8
Test as before.

T12 23 — Test of R6 & Cycle by 12
Test as before but Q-Register is used now.

T13 24 — Test of R6 & Cycle by 16
Test as before.

T14 25 — Test of R6 & Cycle by 20
Test as before.

T5: 12 & 13: Test of Bool Box & RØ

E	12	13	
F	13	14	12

trigger on XS3, YS1

Q is alternately loaded with "Ø" and "-1B". M and Z are "-1B" from previous test. The Bool Boxes are set to M' or Q', Z' or Q'. M = -1B therefore M' = Ø, Z = -1B therefore Z' = Ø and Q is displayed on the BL; and BR; pins.

RØ is incremented by 1.

BL', BR' — see page 21

TYW, LRØ — TYW: A4-109; LRØ: A5-26

TYW', THY', IHR', LRØ' — TYW': A20-003; THY': A25-113; IHR': A25-112; LRØ': A4-110

RØ CLOCK' — ROØ CLOCK': A25-50

Check of increment RØ: (Times not in machine cycles)

trigger on RØ(Ø)* "+" (see page 24)

RØ (Ø)

RØ(1)

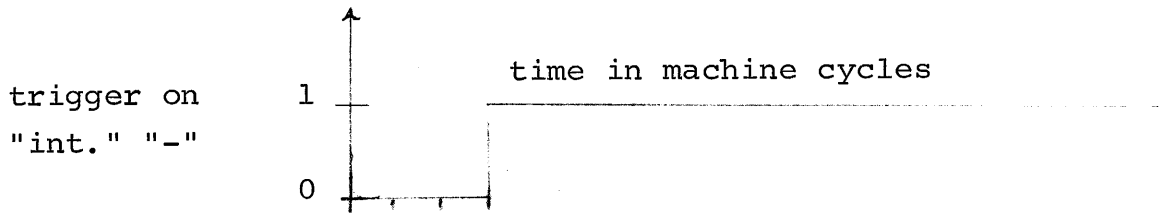
GRØ(Ø)'

Going from RØ(Ø) toward RØ(23), the frequency must double from bit to bit! The GRØ' signals must be the inversed of the RØ signals.

	A25		A20		A14
GRØ(Ø)'	122	GRØ(8)'	122	GRØ(16)'	122
GRØ(1)'	98	GRØ(9)'	98	GRØ(17)'	98
GRØ(2)'	88	GRØ(1Ø)'	88	GRØ(18)'	88
GRØ(3)'	86	GRØ(11)'	86	GRØ(19)'	86
GRØ(4)'	62	GRØ(12)'	62	GRØ(20)'	62
GRØ(5)'	37	GRØ(13)'	37	GRØ(21)'	37
GRØ(6)'	02	GRØ(14)'	02	GRØ(22)'	02
GRØ(7)	27	GRØ(15)'	27	GRØ(23)'	27

*NOTE: For better observation move probe and trigger from RØ(Ø) to RØ(1) to·····until the scope picture improves.

Check carries (0.4V/2.4V) on A25 pins 1,4,6,8,58,111



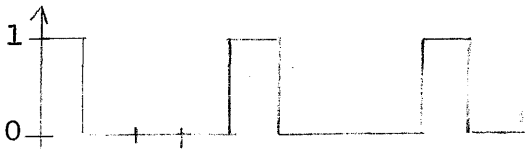
T6: 14 & 15: Test of Bool Box & R1

E	14	15	trigger on XS 5, YS 1
F	15	16	14

Z and Q are - 1B from previous test. The right bool box is set to "Ø" and therefore all BR' pins show always "1".

R1 is incremented and stored into M from where it is loaded into R1.

RRN2,BL1'
LRN2',TXW'
TAX'

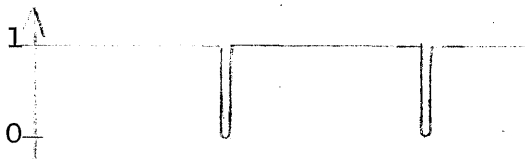


BL1' : A 4 - 89
LRN2': A 5 - 6
RRN2 : A25 - 73
TXW' : A25 - 28
TAX' : A24 - 07

BL1,TXW
TAX,LRN2
RRN2',
LMYA'

BL1 : A23 - 10
LRN2 : A 5 - 12
RRN2': A25 - 68
TXW : A 4 -113
TAX : A 5 - 73
LMYA': A 4 - 04

R1 CLOCK'

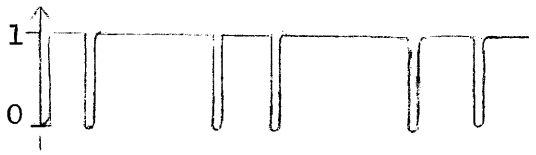


R1 CLOCK': A25 - 64

STOPB'

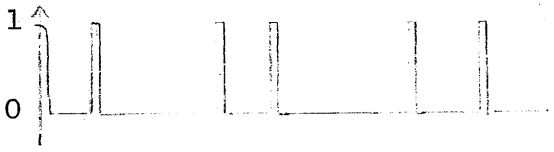
STOPB': A4 - 24

I2 CLOCKA'

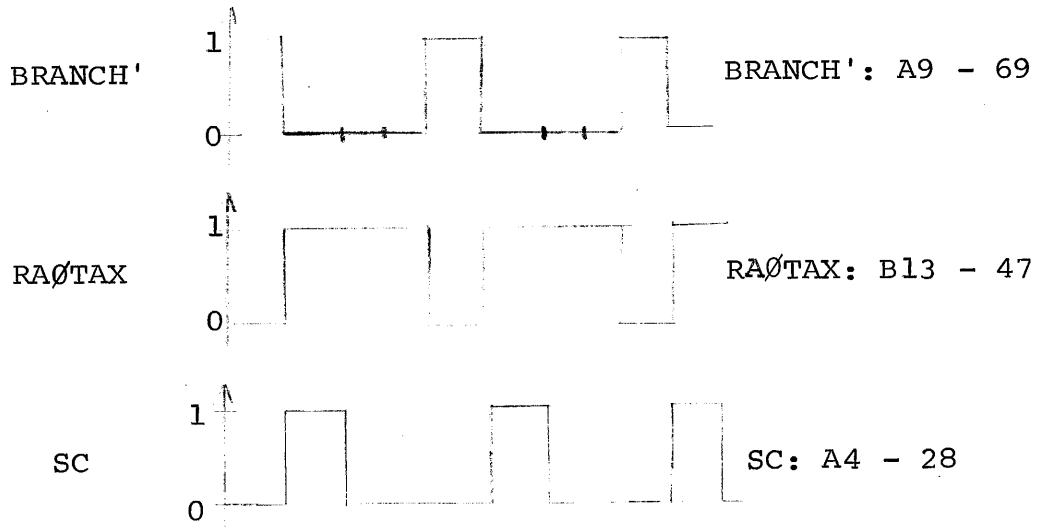


I3 CLOCKA1' = A5 - 30
I3 CLOCKA2' = A5 - 119
I2 CLOCKA1' : A5 - 24
I2 CLOCKA2' : A5 - 110
I2 CLOCKA3' : A5 - 112

I2 CL

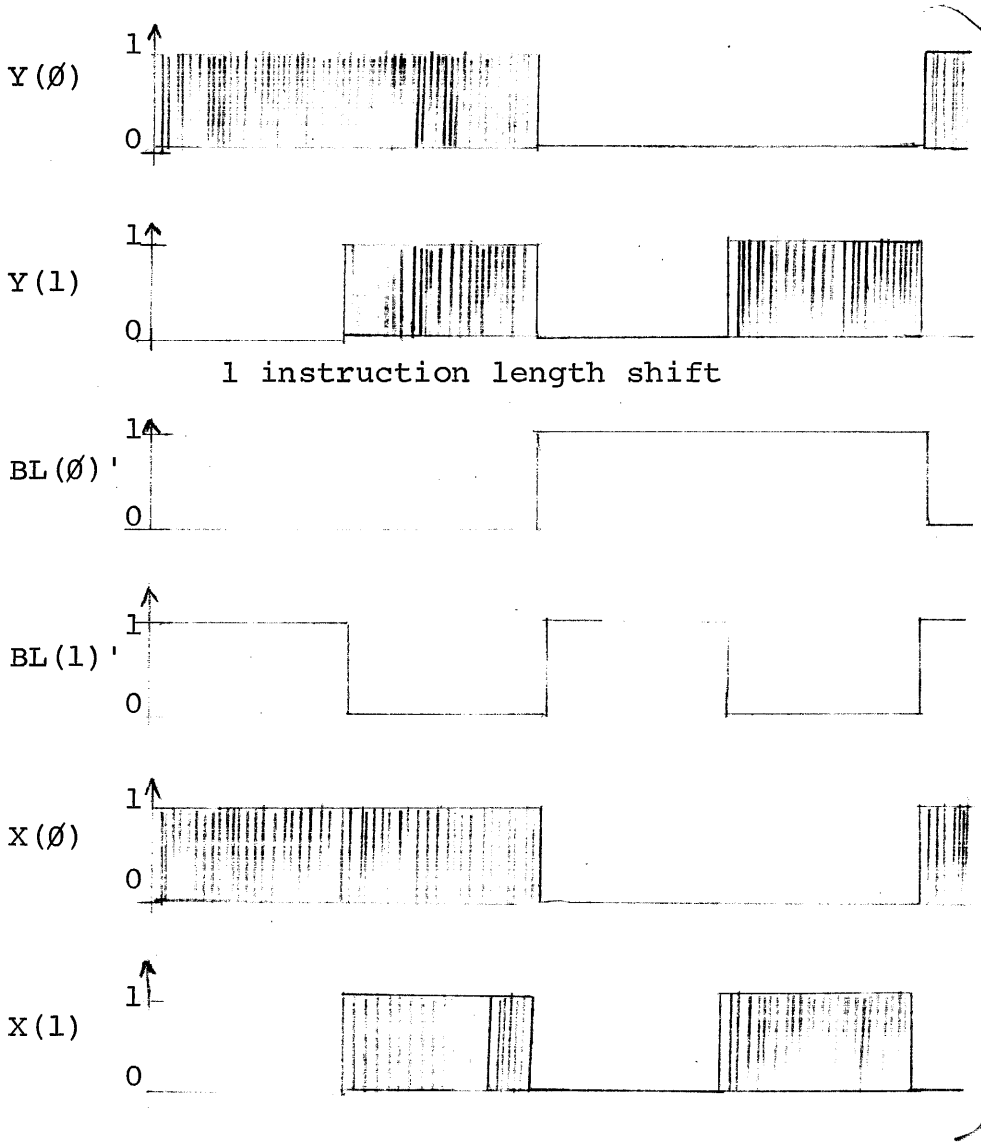


I2 CL : A10 - 77

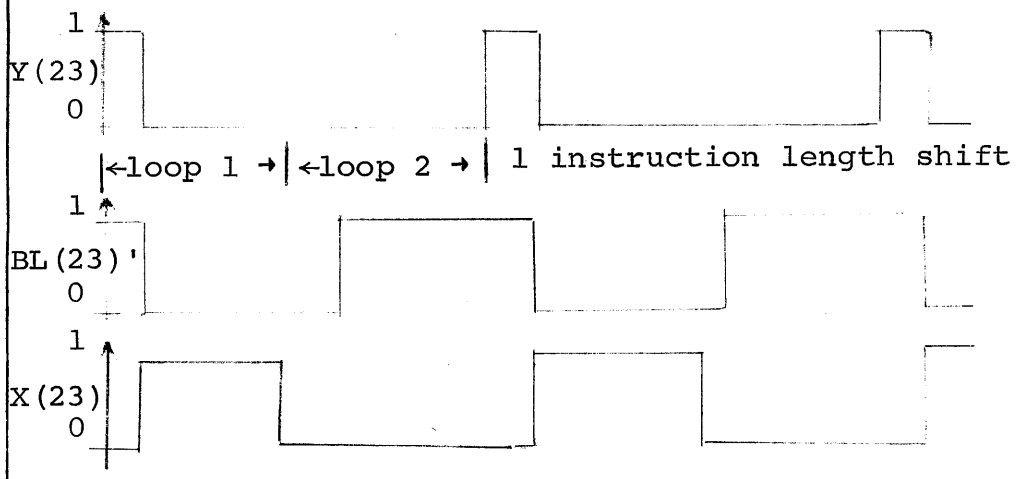

BRANCH'
1
0
BRANCH': A9 - 69
RAØTAX
1
0
RAØTAX: B13 - 47
SC
1
0
SC: A4 - 28

CHECK OF INCREMENT AND LOADING OF R1

trigger on Y(Ø) "+" (see page 24)



trigger on Y(23) "+"



Tests T 7:16: to T 14:25: Tests of Holding Register and Cycler.

This test checks Holding Register and Cycler, but also Bool Box and M.Q.Z. Registers. The different Holding Registers are incremented and loaded either into the Q- or M Register. The contents of M or Q are fed through the Bool Box and cycled onto the X-Bus.

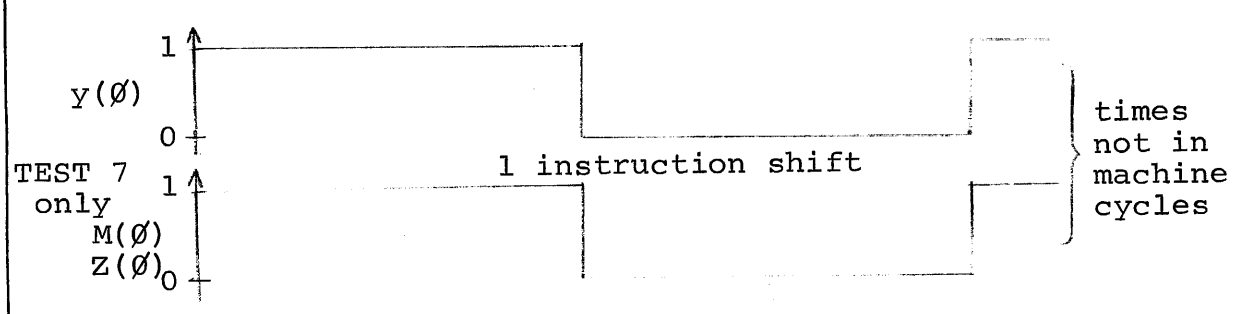
E	A	
F	B	A

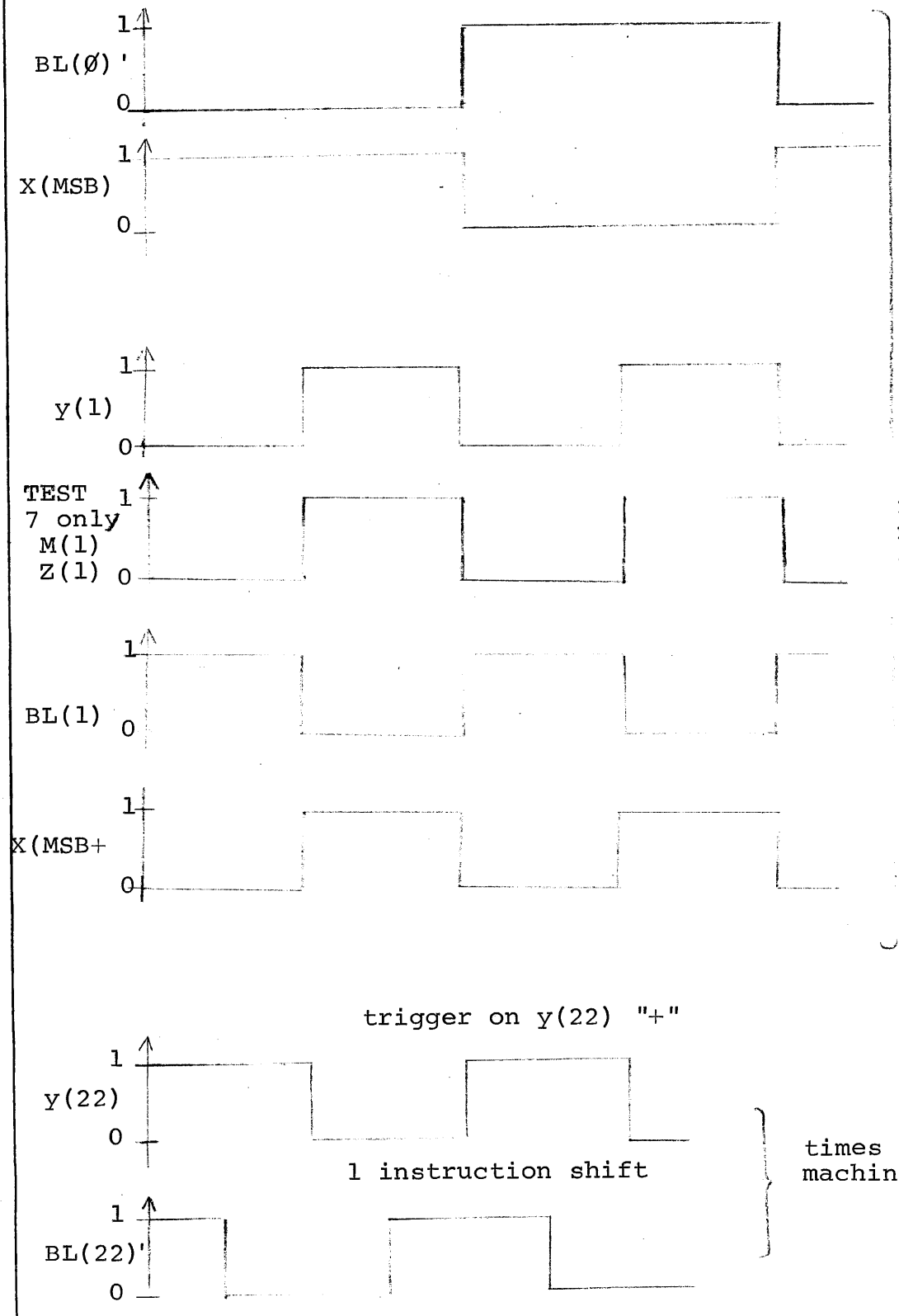
trigger on y(Ø) "+"

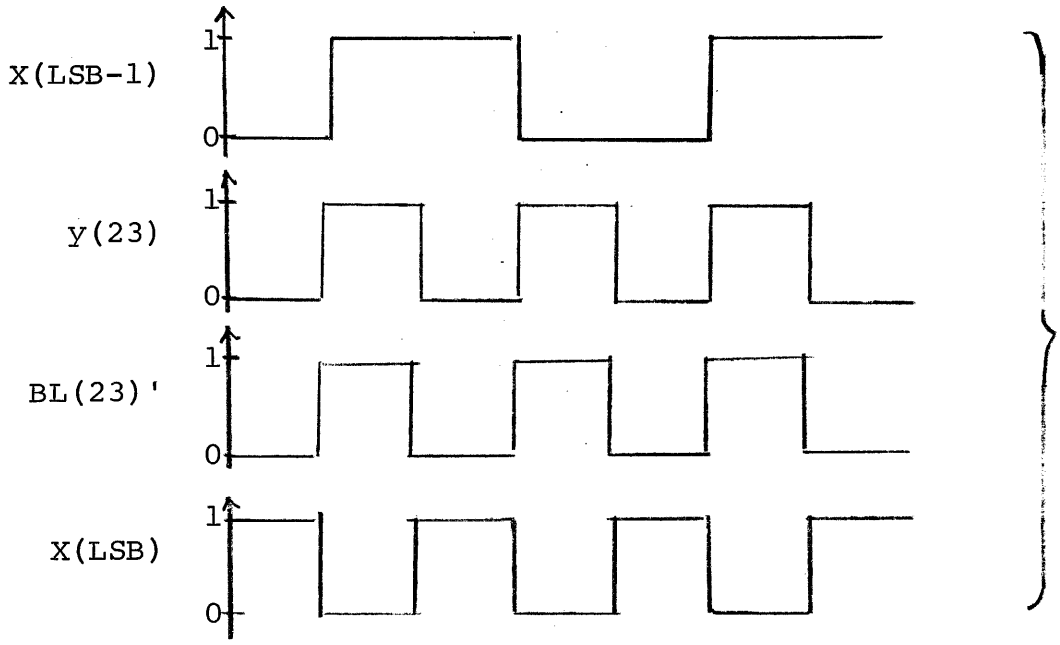
TEST	A	B	ys	xs	(MSB) MOST SIGNI- FICANT BIT	(LSB) LEAST SIGNI- FICANT BIT	RE- GISTERS USED
7	16	17	1	7	X (23)	X (22)	M,Z, R2
8	17	20	2	0	X (22)	X (21)	M, R3
9	20	21	2	1	X (21)	X (20)	M, R4
10	21	22	2	2	X (20)	X (19)	M, R5
11	22	23	2	3	X (16)	X (15)	M, R6
12	23	24	2	4	X (12)	X (11)	Q, R6
13	24	25	2	5	X (8)	X (7)	Q, R6
14	25	26	2	6	X (4)	X (3)	Q, R6

Counting toward double frequency

X (MSB), X (MSB+1), ... X(23), X(Ø), ... X (LSB-1), X (LSB)

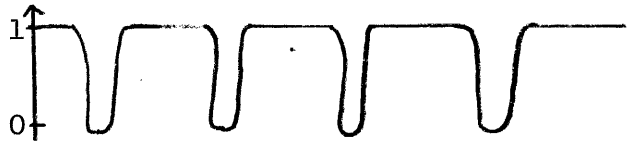



1
BL(Ø)'
0
1
X(MSB)
0
1
y(1)
0
TEST
7 only
M(1)
Z(1)
1
0
times not in machine cycles
1
BL(1)
0
1
X(MSB+
0
trigger on y(22) "+"
1
y(22)
0
1 instruction shift
1
BL(22)'
0
times in machine cycles



Going from y(Ø), BL(Ø)', X(MSB) toward y(23), BL(23)', X(LSB), the frequency must double from bit to bit.

R2CLOCK to R6CLOCK according to register used



Check the following signals for levels.

LCY01B' }
LCY20B' }

TEST	7	8	9	10	11	12	13	14
LCYXB'	1	2	3	4	8	12	16	20
A19	117	123	125	124	12	16	11	13

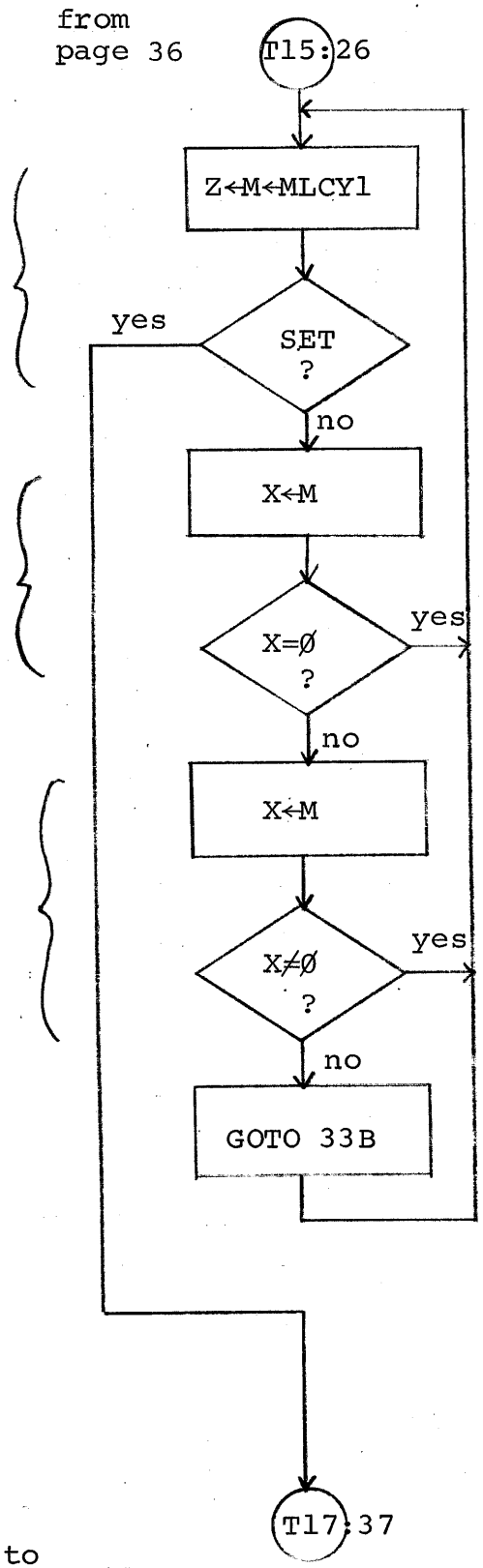
	Test	7	9
RRNØ :	A20-69	0	1
RRNØ':	A20-75	1	0
RRN1 :	A20-72	1	0
RRN1':	A20-70	0	1
LRNØ :	A5-2	0	1
LRNØ':	A5-8	1	0
LRN1 :	A5-4	1	0
LRN1':	A5-1Ø	0	1

BRANCH CONDITION TEST

LABEL	ADDRESS	FLOWCHART	REMARKS
		from page 27	
		T14:25	
		M←1	Load M-Register with "1B"
T15	26	SET ? — yes	
		no	
	27	X←C←Ø	Load X-Bus with "Ø" & test not branch on X*≠Ø
		X≠Ø ? — yes	
		no	
	30	X←C←Ø	Load X-Bus with "Ø" & test not branch on X*<Ø
		X<Ø ? — yes	
		X←C←Ø	Load X-Bus with "Ø" & test branch on X*=Ø
	31	X=Ø ? — yes	
		no	
	32	GOTO 26B	Not executed if test succeeds
		T16:33	
		to Page 37	

*for CPU: Z=Ø from previous test. Branches on Z instead on X.

BRANCH CONDITION TEST



LABEL	ADDRESS	REMARKS
T16	33	Cycling of "1" in Z-& M-Register. M=1B from last test.
	34	Load M onto the X-Bus & test not branch on X*=Ø
	35	Load M onto the X-Bus & test branch on X*≠Ø
	36	Not executed if test succeeds

*Note: for CPU: branch on Z instead of branch on X

LABEL	ADDRESS	FLOWCHART	REMARKS
		from page 37 → T16:23	
T17	37	Z←RØ←4B7	Load negative sign bit into Z- & RØ Registers
		SET ? — yes	
		no → X←Z	Load Z onto the X-Bus & test not branch on X*≥Ø
	40	X≥Ø ? — yes	
		no → Y←RØ	Load RØ onto the Y-Bus & test not branch on RØ≥Ø
	41	RØ≥Ø ? — yes	
		no → X←Z	Load Z onto the X-Bus & test branch on X*<Ø
	42	X<Ø — yes	
		no → GOTO 37B	Not executed if test succeeds
	43		
		to page 39 ← T18:44	*Note: for CPU: branch on Z instead of branch on X

TEST OF BRANCH CONDITION & INCREMENT OF Z

LABEL	ADDRESS	FLOWCHART	REMARKS
		from page 38	
		T17:37	
		Z←RØ←$\bar{Z}$	Invert Z and load it into Z- & RØ-Registers (Z is either -1B or 37777777B)
T18	44	SET ? (yes)	
		no	
	45	Y←RØ	Load RØ onto the Y-Bus & test not branch on RØ<Ø (branch on RØ<Ø, in 2nd loop)
		RØ<Ø ? (yes)	
		no	
	46	X←Z	Load Z onto the X-Bus & test not branch on X*<Ø
		X<Ø ? (yes)	
		no	
	47	GOTO 44B	
		Z←Z+1	increment Z-Register by 1
T19	50	SET ? (no)	
		yes	
		T20Z:51	
		to page 43	

*Note: for CPU: branch on Z instead of branch on X

T 15: to T 18:, 26: to 47: BRANCH CONDITION TEST

This test tests different branch conditions for branch and not - branch. The program loop through which the program runs determines the successful completion of the test.

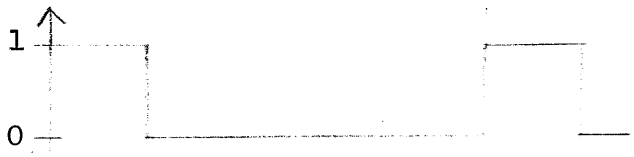
T 15: trigger on xs 7, ys 2

E	26	27	30	31	
F	27	30	31	32	2 6

loop for successful test. 33 should not be fetched!

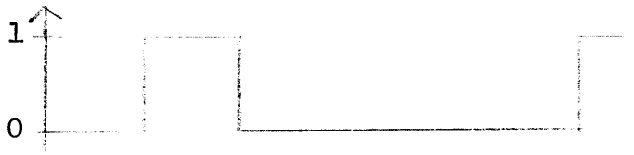
TCX' : A24 - 1
VCY' : A 5 - 38
MC2 : A 9 - 124
LOC : A13 - 6
X(23): A 4 - 11
BLØ : A 4 - 96
LMX : A 4 - 5
B(9): A11 - 1
LOC' : A13 - 21
BLØ' : A23 - 1
BRØ' : A23 - 2
LMX' : A 5 - 17
VCY : A 5 - 40
MC2' : A 9 - 125
FLUSH: A24 - 112

LOC,X(23)
BLØ,LMX
MC2,VCY'
B(9),TCX'



LOC',LMX'
BLØ',BRØ'
VCY ,MC2'
FLUSH

MC 5



A9 - 116

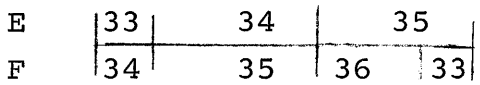
MC 5'

A9 - 117

LMXA'

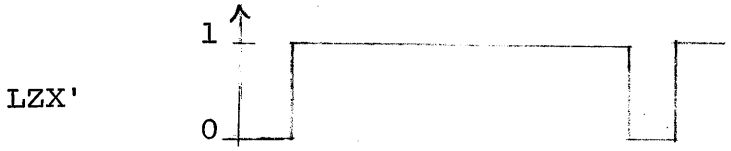
A23-126

T16: trigger on XS4, YS3

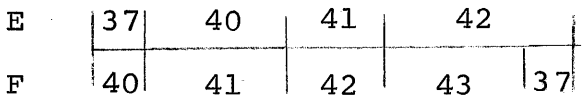


loop for successful test

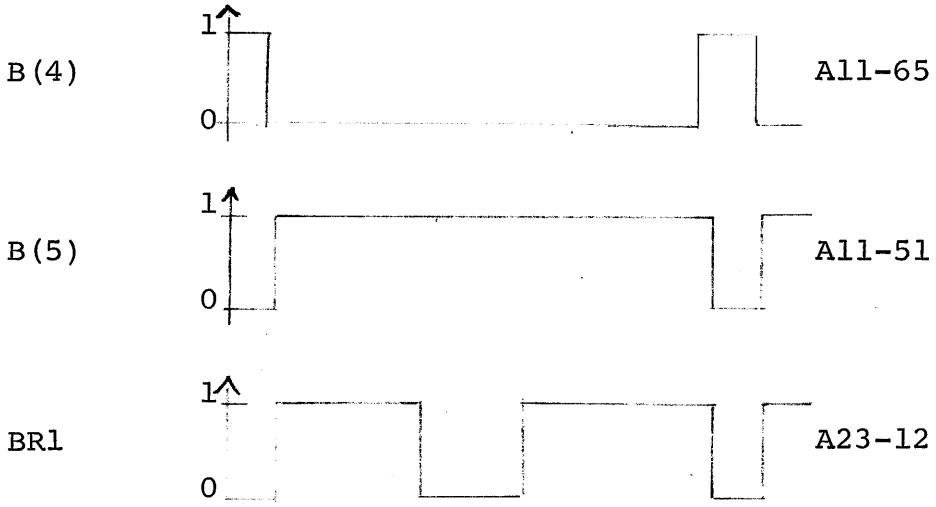
37 should not be fetched!



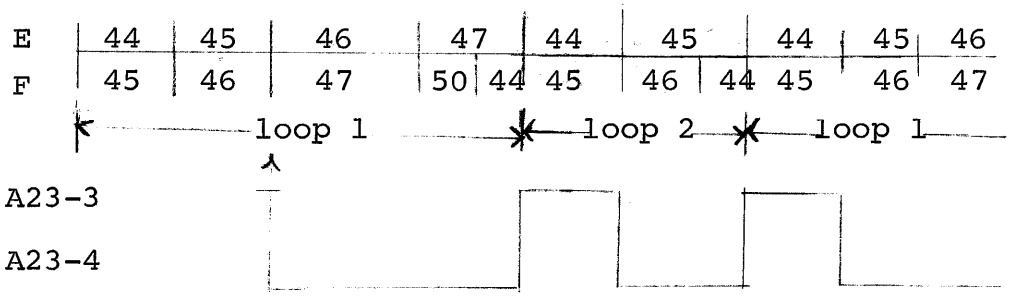
T17: trigger on XS0, YS4



loop for successful test
44 should not be fetched!



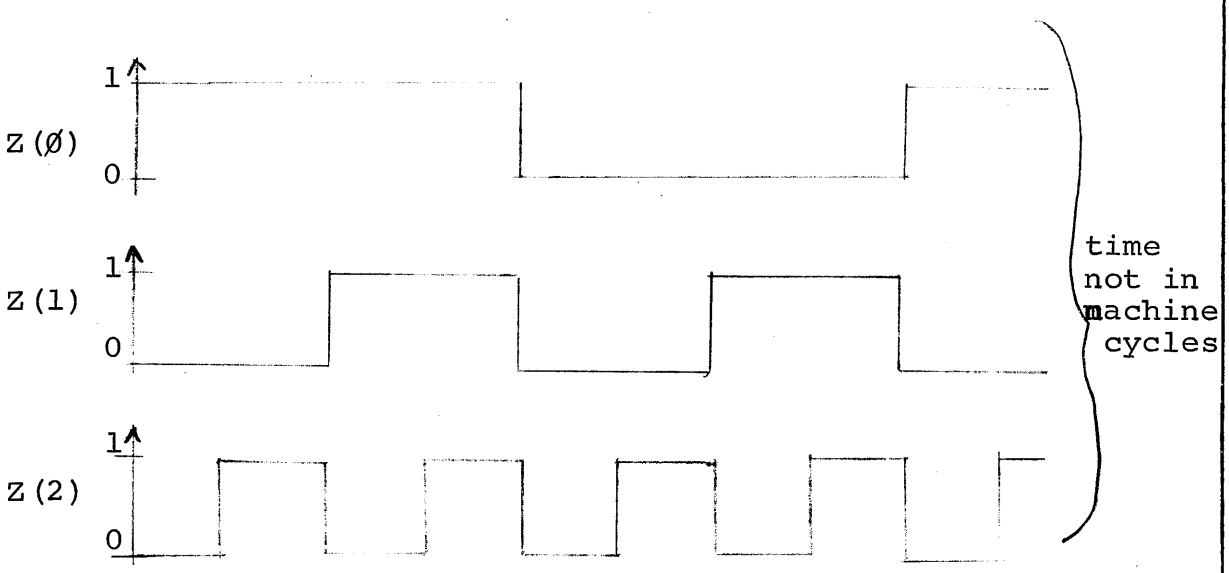
T18: trigger on XS7, YS4



T19: 50: Increment of Z

E	50	
F	51	50

trigger on Z(Ø) "+"



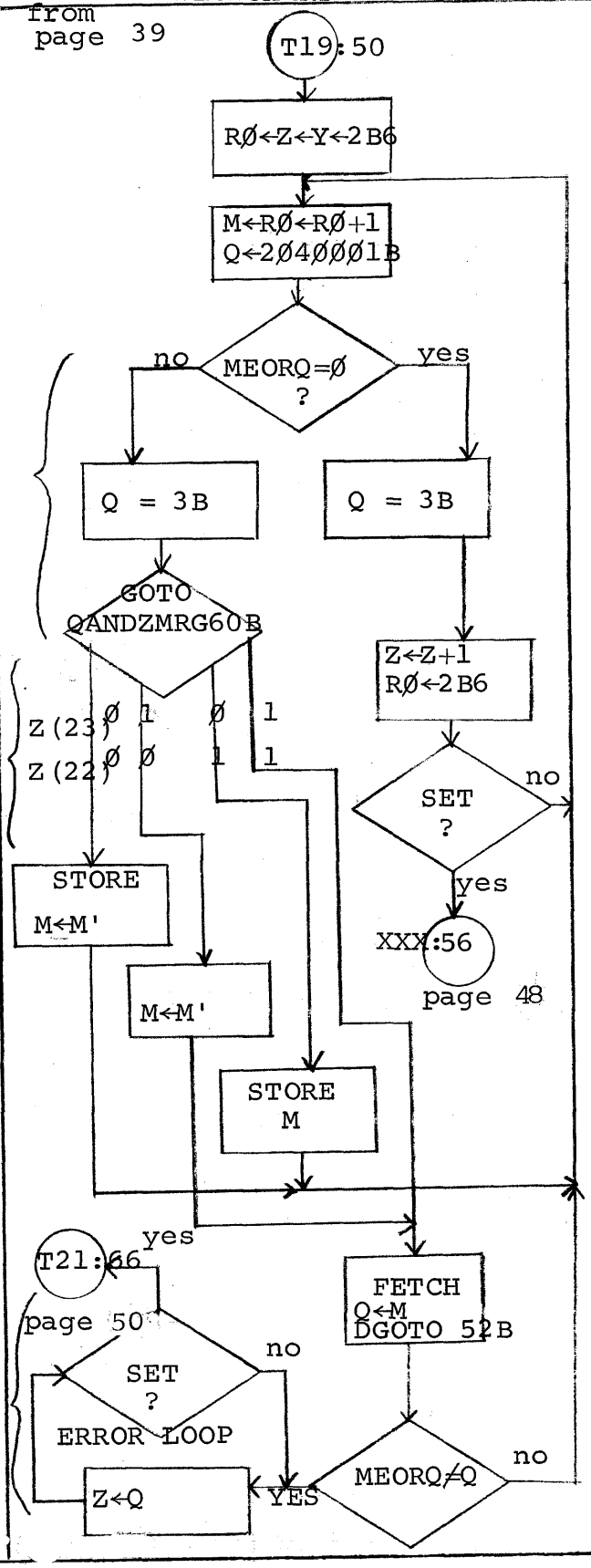
going from Z(Ø) toward Z(23) the frequency must double from bit to bit.

Note: To improve scope picture, change probe and trigger from Z(Ø) toward Z(23) until picture is stable!

Z(XX)' on A28 should be checked, however, may not be terminated now. Also, the carrys from the adder will be checked now. Check for termination (no ringing) and levels. Note the carrys from the most significant bit are not used except in the CPU and hence won't be terminated.

	A24,A19,A13
C(0,8,16)A	114
C(0,8,16)B	066
C(0,8,16)C'	070
C(0,8,16)D'	118
C(0,8,16)E	068
C(0,8,16)F'	072

MEMORY TEST AND INCREMENT OF Z

LABEL	ADDRESS	FLOWCHART	REMARKS
			RØ: ADDRESS REGISTER M: DATA REGISTER Z: COMPUTE INSTRUCTION LOOP INITIALIZATION 1st MEMORY ADDRESS -1=2B6 Z(22) & Z(23) is set to "Ø" PREPARATION for ADDRESS COMPARE MEMORY ADDRESS is incremented DATA is incremented Last MEMORY ADDRESS+1= 2Ø4ØØØ1B ADDRESS COMPARE All cells addressed? SET MASK for Z(22) & Z(23) COMPUTED GOTO GOTO 60 to 63 according to the values of Z(22) & Z(23) START NEXT LOOP increment Z re-initialize RØ Test o.k.? Push ADVANCE. Store RØ' into RØ-Address Invert M to RØ' Store RØ into RØ-Address Load Q with compare data FETCH from RØ-Address DGO to 52B COMPARE DATA. ERROR?
T20A	51		
	52		
	53		
	54		
	55		
T20B	60		
	61		
	62		
	63		
	65		
	64		

T20 A/B: 51 to 55 and 60 to 65: MEMORY TEST

This is the first actual program that is more than a simple instruction loop and is intended to test all memory locations.

51: Initialization: RØ = Memory Address Register, Z computes GOTO.

2B6 is loaded into Z and RØ. This sets Z(22) & Z(23) to "Ø" and sets RØ(4) to "1". Z(22) & Z(23) determine the loop through which the program goes and RØ(4) addresses the private memory.

52: Increment of RØ and loading of address reference:

2040001B is the highest private memory address +1. Since RØ is incremented at the beginning of the loop, the program must branch and reinitialize for new loop when the address register contains 2040001B.

Increment RØ (Memory Address) to address next location and load M with new contents of RØ.

53: Address compare and setting of mask:

Compare RØ with 2040001B and branch to reinitialization (55B) when both are the same.

Set mask in Q to mask up all bits but Z(22) & Z(23).

54: Computed GOTO:

Merge 60B from C-field with Z(22) & Z(23) and branch to this computed location (the O-Register is loaded with the computed address from the X-Bus).

55: Reinitialization for new loop:

Increment Z and reset RØ to "2B6".

60: Store RØ':

Invert M = RØ and store the new contents into the memory location addressed by RØ, and go to increment of RØ (52B).

61: Invert M = RØ and go to fetch from memory (63B).

62: Store RØ:

Store M = RØ into the memory location addressed by RØ.

63: Fetch from memory, DGO:

Load M = RØ or M = RØ', depending on loop, into Q. Fetch the content of the memory location addressed by RØ and load it into M. DGO to 52B (the 0-Register is loaded with 52B from the B-field).

64: Data compare:

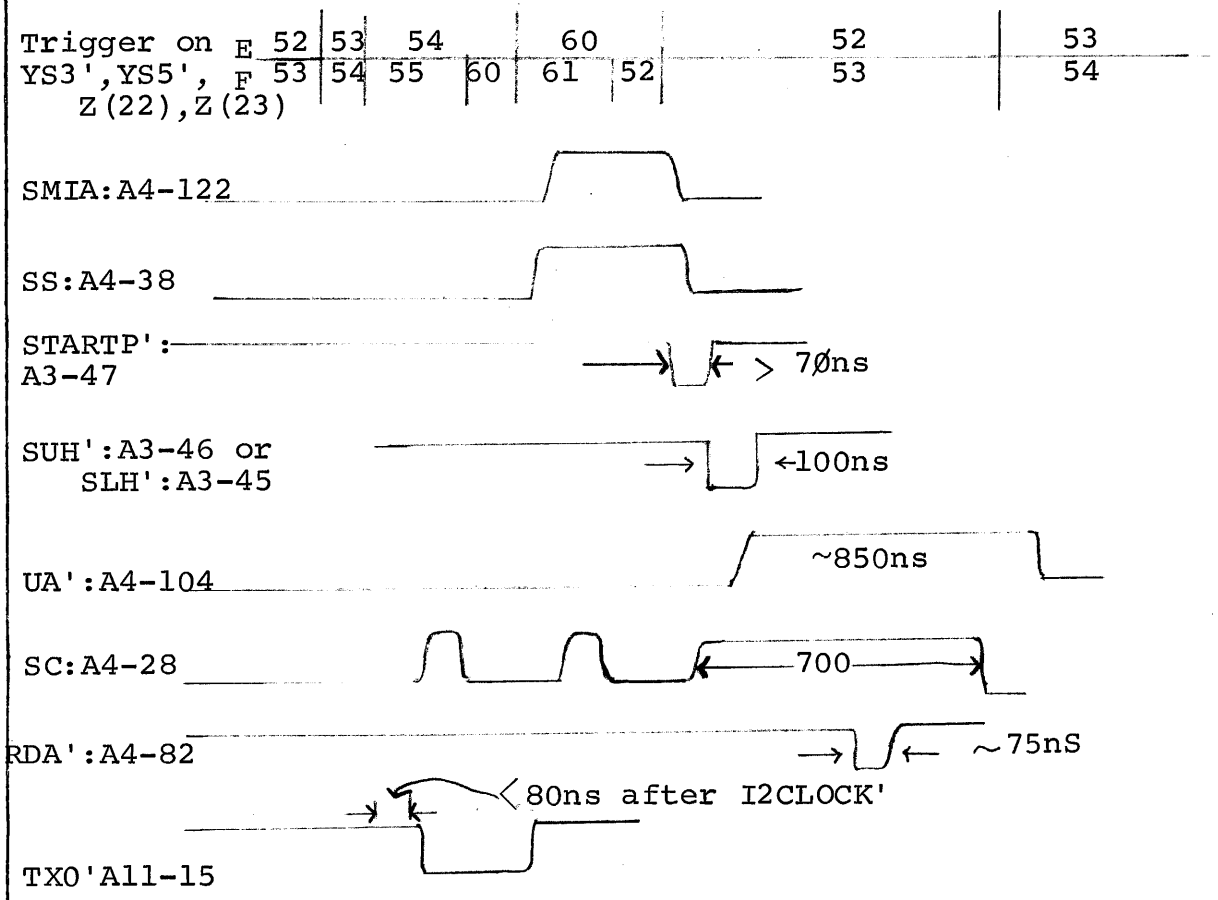
MEORQ compares the reference data in Q with the data in M fetched from memory.
If the data is not the same, the program branches to error loop (65B) (the 0-Register is reloaded with 65B and the DGO from instruction 63B is over-written.

65: Error loop:

The instruction branches to itself until ADVANCE is pressed. Q is loaded into Z to make the data available at the output pins.

Test: Check whether Z(23) is incrementing at >50ms/ cycles to see whether the test is running at all. Check the program loops.

STORE RØ' (Z(22), Z(23) = ØØ) RUN at 100ns



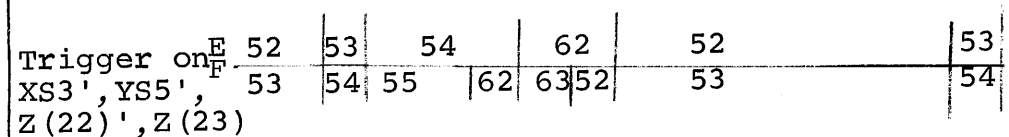
SMIA':A7-121, SFA':A7-17, SSA':A7-39, SHA':A7-61 } >4.0V

S:A4-79, GH':A4-54 } >2.4V

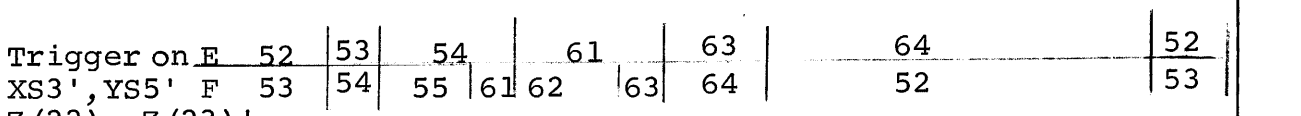
GS':A4-80, H:A4-52, SH:A4-4Ø } <0.4V

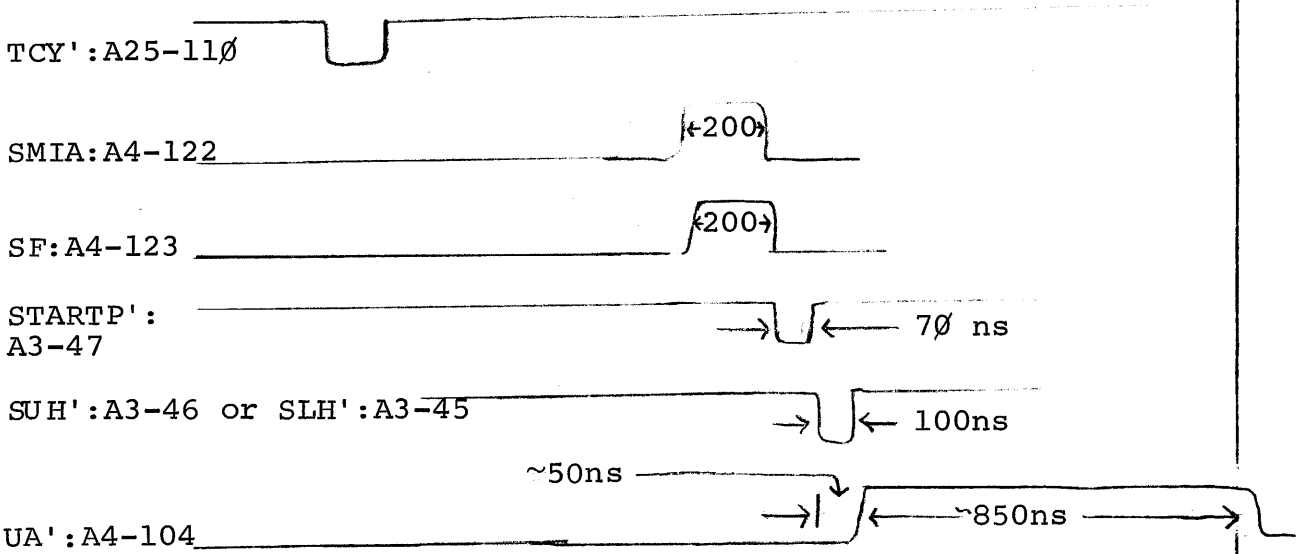
The number of machine cycles spent executing instruction number 52 is dependent upon UA' and RDA, and hence may vary.

STORE RØ (Z(22) = 1, Z(23) = Ø)



FETCH RØ' (Z(22) = Ø, Z(23) = 1)



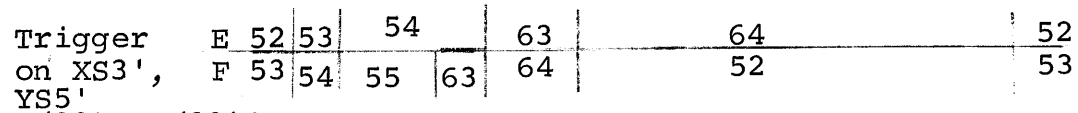


M2(XX) bus: Check for levels and patterns.
(This is data back from core.)

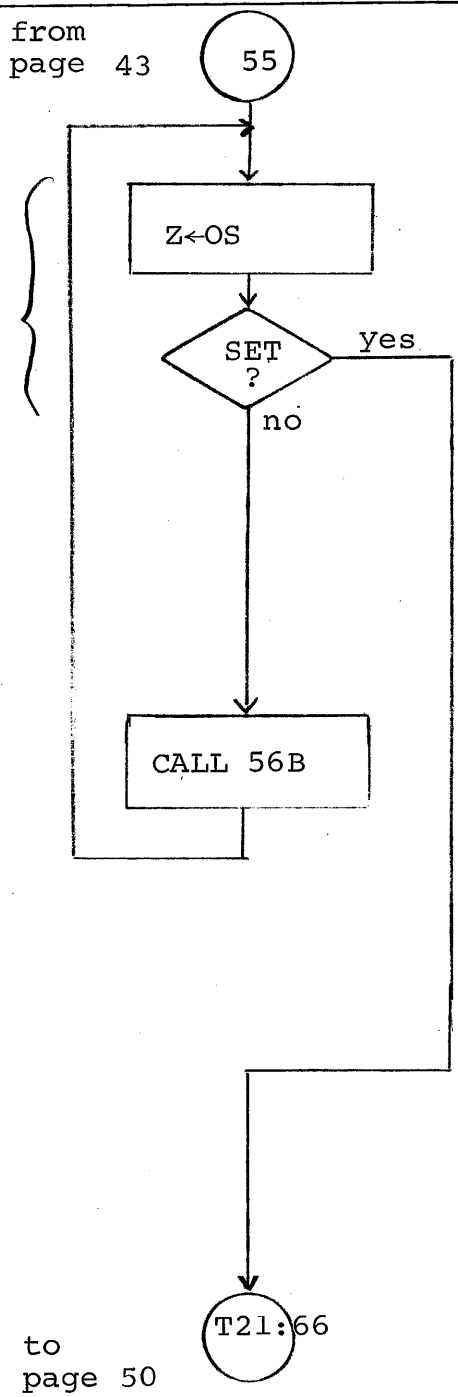
F:A4-56 >2.4V

GF':A4-58 <0.4V

FETCH RØ (Z(22) = 1, Z(23) = 1)



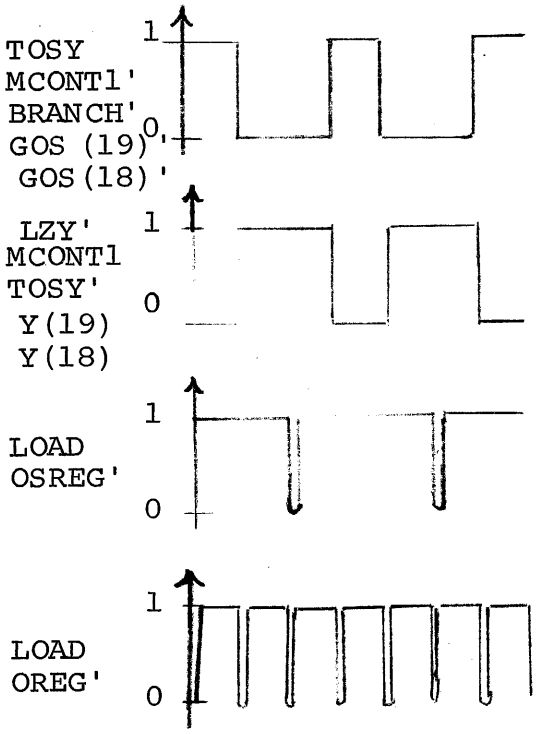
Test of CALL & Return

LABEL	ADDRESS	FLOWCHART	REMARKS
			
XXX	56		load OS into Z-Register to make the contents of OS available at the pins. Contents of OS = 60B
	57		The contents of the 0-Register are loaded into the OS-Register. The program branches to 56B in the same manner as an unconditional branch.

XXX: 56, 57: Test of CALL & Return

This test tests the OS-Register and the signals involved in calling subroutine.

E	56	57
F	57	60 56



TOSY: A07-27
MCONT1'
GOS(19)':A014-85
GOS(18)':A014-97

LZY':A5-23
TOSY': A11-22
MCONT1: A05-55
Y(19):A1Ø-83
Y(18):A1Ø-95

A11-13

A11-07

Y(ØØ) - Y(17) } '0'
Y(2Ø) - Y(23) }

GOS(ØØ)' - GOS(17)' } >2.4V
GOS(2Ø)' - GOS(23)' }

SCRATCHPAD TEST

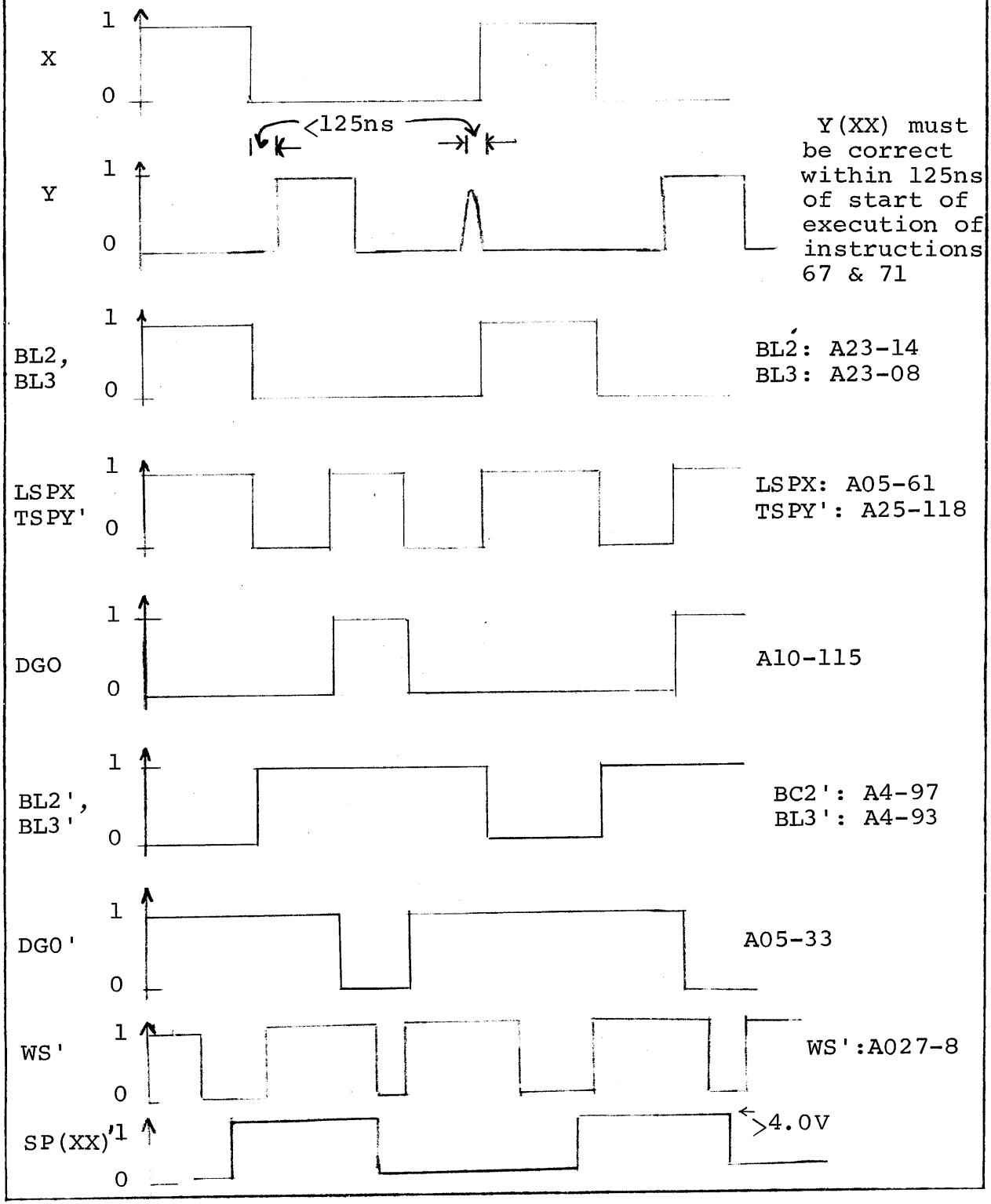
LABEL	ADDRESS	FLOWCHART	REMARKS
		from page 48 XXX:56 / 65	
		SKØ← -1B	Load Scratchpad Ø with "-1B"
T21	66	yes SET ? no	
	67	Z ← SKØ	Load Scratchpad Ø into Z-Register
	70	SKØ←ØB DGOTO	Load Scratchpad Ø with "Ø" DGO to T21:66:
	71	Z ← SKØ M ← 3	Load Scratchpad Ø into Z-Register Load 3B into M (not used in test)
		T22:72	
		to page 52	

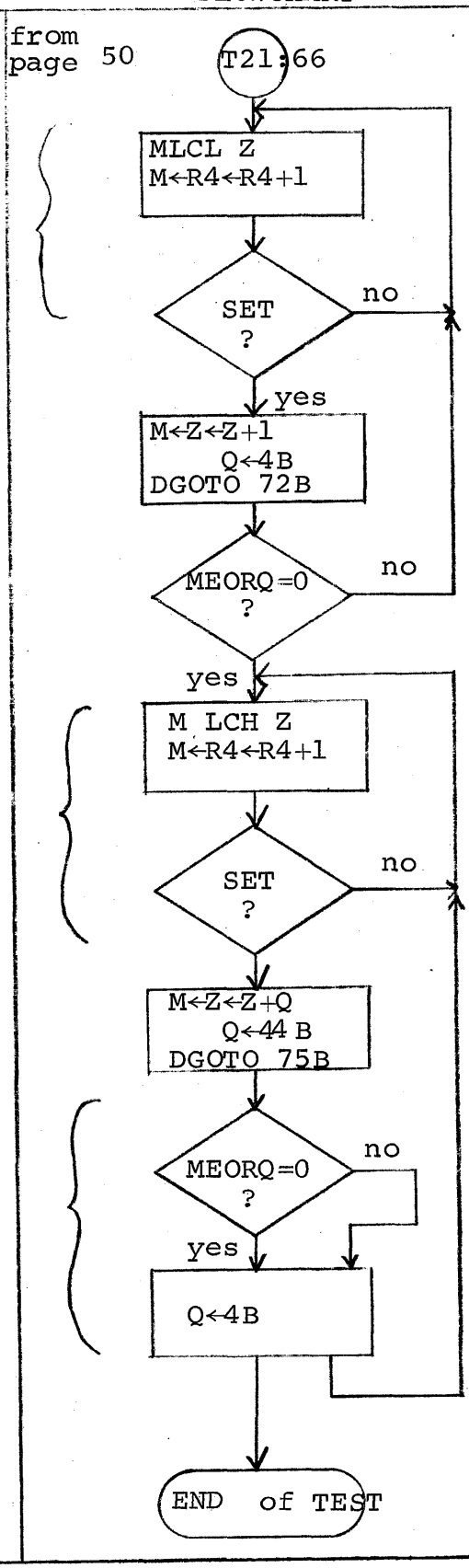
Scratchpad Test

This test loads alternately "-1B" and "Ø" into the scratchpad Ø.

E	66	67	70	71
F	67	70	71	66

trigger on XS7, YS6



LABEL	ADDRESS	FLOWCHART	REMARKS
			*ADVANCE must be pushed to step to the next left cycle! Z = Ø from last test.
T22	72		Test of LCLZ R4 is incremented and loaded into M. M is left cycled according to Z(22) and Z(23) _ _ _ _ _ increment of Z by 1. Q is loaded with 4B to be compared against Z. DGO to 72B _ _ _ _ _ Branch to 75B when Z = 4B.
	73		
	74		
	75		Test of LCHZ R4 is incremented and loaded into M. M is left cycled according to Z(19), Z(20), and Z(21)
	77		increment of Z by 4. Q is loaded with 44B to be compared against Z. DGO to 75B. _ _ _ _ Branch to ØB when Z = 44B.
T1	0		Reload Q with 4B to be added to Z. End of Test. Program branches back to beginning.

T22: 72 to 77: Test of Left Cycle by Z.

This tests test Left Cycle by the high order bits of Z [Z(19), Z(20) & Z(21)] and by the low order bits of Z[Z(22) & Z(23)]. R4 is incremented and loaded into M, from where it is left cycled. To change to the next left cycle "ADVANCE' has to be pushed. Initially left cycle by "Ø" tested. Check increment and cycling on X and TAX A' & B'.

ADDRESS	Z	LEFT CYCLE	X(MSB)	X(LSB)	
72	Ø	Ø (TAXA')	X(Ø)	X(23)	CCFZA'
72	1	1 (LCY1B')	X(23)	X(22)	
72	2	2 (LCY2B')	X(22)	X(21)	
72	3	3 (LCY3B')	X(21)	X(20)	
75	4	4 (LCY4B')	X(2Ø)	X(19)	CCFZB'
75	1Ø	8 (LCY8B')	X(16)	X(15)	
75	14	12 (LCY12B')	X(12)	X(11)	
75	2Ø	16 (LCY16B')	X(8)	X(7)	
75	24	20 (LCY20B')	X(4)	X(3)	
75	3Ø	Ø (TAXB')	X(Ø)	X(23)	
75	34	4 (LCY4B')	X(2Ø)	X(19)	
75	4Ø	Ø (TAXB')	X(Ø)	X(23)	

Since the LCY gates have already been checked, it is only necessary to check that the bit mentioned in the X(LSB) column be checked to see that it increments once every 2 machine cycles.

Check that all the LCYXXB', TAXA', and TAXB', and CCFZA', and CCFZB', are all <.4V, when mentioned in the above chart and otherwise are >2.4V. Also check B(0), B(1), B(2), and B(3) and B(7) for <.4V as they have not been checked.

B(Ø):A11-119

B(1):A11-107	LCY01B':A24-117	LCY08B':A24-12	CCFZA':A5-11
B(2):A11-93	LCY02B':A24-123	LCY12B':A24-16	CCFZB':A5-97
B(3):A11-79	LCY03B':A24-125	LCY16B':A24-11	TAXA':A24-8
B(7):A11-23	LCY04B':A24-124	LCY2ØB':A24-13	TAXB':A24-1Ø

Concluding Remarks

Since this test is only a basic test, not every part of the processor is thoroughly tested. Tests like "MICRO-PROCESSOR ACCEPTANCE PROCEDURE" working paper MPAP/W-18.1, "MICROPROCESSOR ROM ACCEPTANCE" Memo from CHUCK THACKER, January 12, 1970, to check the final ROM boards, "SCRATCH PAD TEST" N.ESKR 11/21/69, diagnostic programs and dynamic debugging routines are intended to check individual parts of the system in more detail.

APPENDIX:

see also:

page 6: xs-, ys- and cs- pins

page 14: I-Register pins

page 21/22: Bus pins

TABLE #I : BOOL BOX CODE

BLØ-BL3	Left Bool Box Output
ØØ	M·Q
Ø1	M=Q
Ø2	Q
Ø3	$\overline{M}$+Q
Ø4	M
Ø5	M+$\overline{Q}$
Ø6	M+Q
Ø7	1
1Ø	Ø
11	$\overline{M}$·$\overline{Q}$
12	$\overline{M}$·Q
13	$\overline{M}$
14	M·$\overline{Q}$
15	$\overline{Q}$
16	M(EOR)Q
17	$\overline{M}$+$\overline{Q}$

BRØ-BR3	Right Bool Box Output
ØØ	Z·Q
Ø1	Z=Q
Ø2	Q
Ø3	$\overline{Z}$+Q
Ø4	Z
Ø5	Z+$\overline{Q}$
Ø6	Z+Q
Ø7	1
1Ø	Ø
11	$\overline{Z}$·$\overline{Q}$
12	$\overline{Z}$·Q
13	$\overline{Z}$
14	Z·$\overline{Q}$
15	$\overline{Q}$
16	Z(EOR)Q
17	$\overline{Z}$+$\overline{Q}$

Branch Conditions

Ø	Never branch
1	Always branch
2	X= Ø
3	X≠ Ø
4	X<Ø
5	X≥Ø
6	X>Ø
7	Y≥ Ø
1Ø	Y<Ø
11	RØ<Ø
12	RØ≥Ø
13	X≤Ø
14	X'∧777777B = Ø (X(6) - X(23) = 777777B)
15	X'∧777777B ≠ Ø (X(6) - X(23) ≠ 777777B)
16	Z≥Ø
17	Z<Ø
2Ø	Always branch
21	Y∧7≠Ø (Y(23)∨Y(22)∨Y(21)= 1)
22	BL=Ø
23	BL≠Ø
24	Y(23) = Ø
25	Y(23) ≠ Ø

Branch Conditions - 2

26	Attention latch 1 = Ø (also resets the latch)
27	(RSLAT 1 = Ø) ∨ (RSLAT 2 = Ø)
3Ø	Protect ≠ X
31	RSLAT 2 = Ø
32	Special flag A = Ø
33	Special flag A ≠ Ø
34	Attention latch 2 = Ø
35	Attention latch 3 = Ø
36	Attention latch 1 ≠ Ø

} Also resets the latch

37	Undecoded
4Ø	undefined
41	undefined
42	Local memory parity error = 1 (resets the latch)
43	undefined
44	Central memory parity error = 1 (resets the latch)
45	breakpoint ≠ Ø

Special Conditions

Ø	No activity
1	LCY 1
2	LCY 2
3	LCY 3
4	LCY 4
5	LCY 8
6	LCY 12
7	LCY 16
1Ø	LCY 2Ø
11	LCL Z (CCFZA)
12	LCH Z (CCFZB)
13	SKZ - Reference scratchpad with address in Z (SPFZ)
14	ALERT
15	POT
16	PIN
17	Request Strobe #1
2Ø	Unprotect
21	Unusable
22	Load memory request priority field (LPF)
23	Reset Request Strobe latch #1 (Occurs at end of instruction)
24	Reset Central Memory Request (Local & Central Memory)
25	Request Protect

Special Conditions - 2

26	Reset T.U. (or other device attached to I/O connector)	
27	undecoded	
3Ø	Set special flag A	Occurs at end of instruction
31	Reset special flag A	Occurs at end of instruction
32	Reset Request Strobe Latch #2	Occurs at end of instruction
33	Request Strobe #2	
34	undefined	
35	undecoded	
36	undecoded	
37	undecoded	
4Ø	Release	Memory Reference
41	Prestore	Memory Reference
42	Store	Memory Reference
43	Store & Hold	Memory Reference
44	Fetch	Memory Reference
45	Fetch & Hold	Memory Reference
47	Prefetch	Memory Reference
6Ø	Set Bank B	
61	Set Bank A	
62	Clear all CPU Maps	
64	Fetch	ODDWORD FETCH
65	Fetch & Hold	ODDWORD FETCH