APPENDIX 1

Microprogram.

A complete list of all matrix outputs is tabulated in the following pages.

All matrix outputs are inverted and amplified before use and the board and pin numbers from where they are taken to the matrix amplifiers are provided. Matrix outputs that are timed, have the times in which they are used marked in brackets after them.

e. g. VTG (2) means that VTG is strobed by t2.

The tables are arranged in order of function and each function in order of matrix address (where possible).

The manner in which the several matrix addresses are linked together to form processor functions is given on page of this appendix under the title "FLOW DIAGRAM".

NOTE:

The "bars" and suffices (m) have been omitted from the signal names for clarity in printing.

FUNCTION CONTROL

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
C2				
TEST FOR INTERRUPT	AA1	10(A-GJ)	C	
SELECT RELEVANT	AA2	10(A-GJ)	D	
SCR. READ THAT	DTE	19(A-GA)	P	
SCR.	ETJ	15(A-GE)	J	
	WFR	19(A-GA)	M	
	OTJ(3)	13(A-GG)	S	
	OTM(3)	10(A-GJ)	AA	
	READ	11(A-GI)	L	
C0				
READ WORD	AA1	10(A-GJ)	C	
GENERATOR TEST	DTE	19(A-GA)	P	
WHETHER TO ENTER	TWGA	19(A-GA)	W	CONDITIONS
OR OBEY WORD	OTJ(3)	13	S	TEST SIGNAL
GENERATOR	OTM(3)	10(A-GJ)	AA	
	WTM(4)	19(A-GA)	Y	
C1A				
ENTER WORD	AA2	10(A-GJ)	D	WGA
GENERATOR RETURN	MTF	11(A-GI)	AJ	
TO ADDRESS C2	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	
C1B				
OBEY WORD	AA1	10(A-GJ)	C	
GENERATOR	AA3	10(A-GJ)	E	
C3				
INCREMENT SCR	AA3	10(A-GJ)	E	
CONTENT BY ONE	MTF	11(A-GI)	AJ	
AND REWRITE IN	ITF	11(A-GI)	N	
CURRENT SCR	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
	WRITE	11(A-GI)	K	
C4				
DECREMENT BY 1 THE	AA1	10(A-GJ)	C	
EXTRACTED S.C.R.	AA3	10(A-GJ)	E	
LOAD THE STORE	MTF	11(A-GI)	AJ	
ADDRESS REGISTER	DTF1	15(A-GE)	AD	
WITH CORRECTED SCR	DTF2	15(A-GE)	AC	
VALUE AND EXTRACT	OTG(1)*			
THE CONTENTS OF	FTG(2)	11(A-GI)	N	
THAT STORE	OTJ(3)*			
LOCATION	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
C5				
TEST FOR B	AA2	10(A-GJ)	D	
(ADDRESS) MODIFIER	AA3	10(A-GJ)	E	
	TM18	15(A-GE)	T	CONDITIONS TEST SIGNAL

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
C6A				
LOAD I-REGISTER	AA1	10(A-GJ)	C	M18=1
WITH INSTRUCTION.	AA2	10(A-GJ)	D	(B-MODIFIC-
SET UP	AA3	10(A-GJ)	E	ATION
APPROPRIATE	MTF	11(A-GI)	AJ	REQUIRED)
B-REGISTER ADDRESS	DTF1	15(A-GE)	AD	
READ THAT	K1	18(A-GB)	S	
B-REGISTER	ETJ	15(A-GE)	J	
	OTG(1)*			
	OTI(1)*			
	VTG(2)	15(A-GE)	H	
	MTI(2)*			
	OTM(3)	10(A-GJ)	AA	
	OTJ(3)	13(A-GG)	S	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
C6B				
LOAD I-REGISTER	MTF	11(A-GI)	AJ	M18=0
WITH INSTRUCTION.	DTF1	15(A-GE)	AD	(NO B-
PERMIT FUNCTION	SFD	19(A-GA)	V	MODIFICA-
DECODE.	OTG(1)*			TION
	OTI(1)*			REQUIRED)
	VTG(2)	15(A-GE)	H	
	MTI(2)*			
	OTJ(3)	13(A-GG)	S	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
C7				
MODIFY ADDRESS	QTF	11(A-GI)	AH	
PART OF	MTF	11(A-GI)	AJ	
INSTRUCTION BY	SFD	19(A-GA)	V	
ADDING THE	OTG(1)*			
CONTENTS OF M	FTG(2)	11(A-GI)	F	
AND Q	OTJ(3)	13(A-GG)	S	
PERMIT FUNCTION	OTM(3)	10(A-GJ)	AA	
DECODE.	GTM(4)	11(A-GI)	AA	

NOTE: Not all register clear signals are generated by the matrix directly. Signals marked* have several sources (See Figure 35 entitled Waveform Drivers).

SET B-REGISTER – ADD – NEGATE AND ADD – STORE Q REGISTER

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
00				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER EXTRACT	DTF2	15(A-GE)	AC	
THAT OPERAND.	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
01				
SET J-REGISTER TO	AA1	10(A-GJ)	C	
THE ADDRESS OF THE	AA2	10(A-GJ)	D	
CURRENT LEVEL	AA3	10(A-GJ)	E	
B-REGISTER	MTF	11(A-GI)	AJ	
PLACE NEW CONTENT	ETJ	15(A-GE)	J	
OF B-REGISTER IN	K1	18(A-GB)	S	
Q-REGISTER AND	OTG(1)*			
WRITE THAT CONTENT	FTG(2)	11(A-GI)	F	
INTO THE PERTINENT	OTJ(3)	13(A-GG)	S	
B-REGISTER	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	
10				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER EXTRACT	DTF2	15(A-GE)	AC	
THAT OPERAND	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
11				
ADD CONTENTS OF	AA1	10(A-GJ)	C	
A-REGISTER AND	AA2	10(A-GJ)	D	
M-REGISTER PLACE	AA3	10(A-GJ)	E	
RESULT IN	MTF	11(A-GI)	AJ	
ACCUMULATOR	ATF	11(A-GI)	Y	
	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	

FUNCTIONS 0, 1, 2 & 3.

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
20				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER EXTRACT	DTF2	15(A-GE)	AC	
THAT OPERAND	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
21				
TRANSFER THE	AA2	10(A-GJ)	D	
CONTENT OF THE	MTF	11(A-GI)	AJ	
M-REGISTER TO THE	OTG(1)*			
Q-REGISTER	FTG(2)	11(A-GI)	F	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
22				
TRANSFER THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
A-REGISTER TO THE	ATF	11(A-GI)	Y	
M-REGISTER	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
23				
ADD THE TWO'S	AA1	10(A-GJ)	C	
COMPLEMENT OF THE	AA2	10(A-GJ)	D	
M-REGISTER CONTENT	AA3	10(A-GJ)	E	
TO THE Q-REGISTER	MTF	12(A-GH)	P	
CONTENT	1TF	11(A-GI)	N	
PLACE RESULT IN	QTF	11(A-GI)	AH	
ACCUMULATOR	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	
30				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER	DTF2	15(A-GE)	AC	
CLEAR OVERFLOW	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTX(3)	12(A-GH)	X	
	GTJ(4)	11(A-GI)	AF	
31				
RIGHT SHIFT THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
Q-REGISTER ONE	AA3	10(A-GJ)	E	
PLACE. TRANSFER	QTF	11(A-GI)	AH	
THE NEW CONTENT OF	XTF	12(A-GH)	J	
Q-REGISTER TO THE	OTG(1)*			
M-REGISTER AND	RTG(2)	12(A-GH)	R	
WRITE THAT CONTENT	OTM(3)	10(A-GJ)	AA	
INTO THE STORE	GTM(4)	11(A-GI)	AA	
ADDRESS SPECIFIED	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	

*See note at foot of Page 2.

READ WRITE COLLATE JUMP IF A=0 – JUMP

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
40				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER EXTRACT	DTF2	15(A-GE)	AC	
THAT OPERAND	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
41				
TRANSFER THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
M-REGISTER TO THE	AA3	10(A-GJ)	E	
ACCUMULATOR	MTF	11(A-GI)	AJ	
	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	
50				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO THE	DTF1	15(A-GE)	AD	
J-REGISTER CLEAR	DTF2	15(A-GE)	AC	
THE M-REGISTER	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
51				
TRANSFER THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
A-REGISTER TO THE	AA3	10(A-GJ)	E	
M-REGISTER AND	ATF	11(A-GI)	Y	
WRITE THAT CONTENT	OTG(1)*			
IN THE STORE	FTG(2)	11(A-GI)	F	
ADDRESS SPECIFIED	GTM(4)	11(A-GI)	AA	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	
60				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
J-REGISTER. EXTRACT	DTF2	15(A-GE)	AC	
THAT OPERAND	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
61				
PASS CONTENTS OF	AA1	10(A-GJ)	C	
M-REGISTER AND	AA2	10(A-GJ)	D	
A-REGISTER TO THE	AA3	10(A-GJ)	E	
F-UNIT. COLLATE	MTF	11(A-GI)	AJ	
THESE CONTENTS	ATF	11(A-GI)	Y	
INTO ACCUMULATOR	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
70				
TRANSFER THE	AA1	10(A-GJ)	C	
CONTENT OF THE	MTF	11(A-GI)	AJ	
M-REGISTER TO	DTF1	15(A-GE)	AD	
THE Q-REGISTER	DTF2	15(A-GE)	AC	
	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
71				
TRANSFER THE	AA2	10(A-GJ)	D	
CONTENT OF THE	ATF	11(A-GI)	Y	
A-REGISTER TO THE	TM18	15(A-GE)	T	CONDITIONS
M-REGISTER AND	OTG(1)*			TEST SIGNAL
TEST IF THAT	FTG(2)	11(A-GI)	F	
NUMBER IS	OTM(3)	10(A-GJ)	AA	
NEGATIVE	GTM(4)	11(A-GI)	AA	
72A				
DO NOTHING	AA1	10(A-GJ)	C	M18 = 1
	AA2	10(A-GJ)	D	(A NEGATIVE)
	AA3	10(A-GJ)	E	
72B				
CHANGE THE	AA1	10(A-GJ)	C	M18 = 0
CONTENT OF THE	AA2	10(A-GJ)	D	(A POSITIVE)
M-REGISTER TO ITS	MTF	12(A-GH)	P	
TWO'S COMPLEMENT	ITF	11(A-GI)	N	
AND TEST IF THE	TM18	15(A-GE)	T	
CONTENT IS	OTG(1)*			
NEGATIVE	FTG(2)	11(A-GI)	F	
	OTJ(3)	13(A-GG)	S	
	OTM(3)*			
	GTM(4)	11(A-GI)	AA	
73B				
SET J-REGISTER	AA1	10(A-GJ)	C	M18 = 0
WITH ADDRESS OF	AA2	10(A-GJ)	D	(A = 0)
CURRENT LEVEL SCR.	AA3	10(A-GJ)	E	
WRITE THE CONTENT	QTF	11(A-GI)	AH	
OF THE Q-REGISTER	ETJ	15(A-GE)	J	
INTO THAT SCR	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTM(3)*			
	GTM(4)	11(A-GI)	AA	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	
80				
SET J-REGISTER WITH	AA1	10(A-GJ)	C	
ADDRESS OF	ETJ	15(A-GE)	J	
PERTINENT SCR	OTJ	13(A-GG)	S	
81				
WRITE THE CONTENTS	AA1	10(A-GJ)	C	
OF THE M-REGISTER	AA2	10(A-GJ)	D	
INTO THAT SCR.	AA3	10(A-GJ)	E	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	

FUNCTIONS 4, 5, 6, 7 & 8

* See note at foot of Page 2.

JUMP IF A NEGATIVE – COUNT IN STORE – STORE SCR

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
90				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
THE M-REGISTER TO	DTF1	15(A-GE)	AD	
THE Q-REGISTER	DTF2	15(A-GE)	AC	
	OTG(1)*			
	VTG(2)	15(A-GE)	H	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
91				
TRANSFER THE	AA2	10(A-GJ)	D	
CONTENT OF THE	ATF	11(A-GI)	Y	
A-REGISTER TO THE	TM18	15(A-GE)	T	CONDITIONS
M-REGISTER TEST IF	OTG(1)*			TEST SIGNALS
NEGATIVE	FTG(2)	11(A-GI)	F	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
92A				
SET J-REGISTER WITH	AA1	10(A-GJ)	C	M18=1
ADDRESS OF SCR.	AA2	10(A-GJ)	D	(A-NEGATIVE)
WRITE THE CONTENT	AA3	10(A-GJ)	E	
OF THE Q-REGISTER	QTF	11(A-GI)	AH	
INTO THAT SCR.	ETJ	15(A-GE)	J	
	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	
92B				
DO NOTHING	AA1	10(A-GJ)	C	M18=0
	AA2	10(A-GJ)	D	(A-POSITIVE)
	AA3	10(A-GJ)	E	
100				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
THE M-REGISTER TO	DTF1	15(A-GE)		
THE J-REGISTER	DTF2	15(A-GE)		
EXTRACT THAT	OTG(1)*			
OPERAND	VTG(2)	15(A-GE)	H	
	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
101				
INCREMENT THE	AA1	10(A-GJ)	C	
M-REGISTER CONTENT	AA2	10(A-GJ)	D	
BY 1 AND WRITE THE	AA3	10(A-GJ)	E	
NEW CONTENT BACK	MTF	11(A-GI)	AJ	
INTO THE STORE.	ITF	11(A-GI)	N	
	OTG(1)*			
	FTG(2)	11(A-GJ)	F	
	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
	WRITE	11(A-GI)	K	
110				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
THE M-REGISTER TO	DTF1	15(A-GE)	AD	
THE Q-REGISTER	DTF2	15(A-GE)	AC	
SET J-REGISTER	ETJ	15(A-GE)	J	
WITH THE ADDRESS	OTG(1)*			
OF THE SCR AND	VTG(2)	15(A-GE)	H	
READ THE CONTENTS	OTQ(3)*			
OF THAT SCR	OTM(3)	10(A-GJ)	AA	
	GTQ(4)	12(A-GH)	AB	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
111				
TRANSFER ADDRESS	AA2	10(A-GJ)	D	
OF OPERAND FROM	QTF	11(A-GI)	AH	
THE Q-REGISTER TO	OTG(1)*			
THE J-REGISTER	FTG(2)	11(A-GI)	F	
	OTJ(3)*			
	GTJ(4)	11(A-GI)	AF	
112				
COLLATE THE 5	AA1	10(A-GJ)	C	
MOST SIGNIFICANT	AA2	10(A-GJ)	D	
BITS OF THE	MTF	11(A-GI)	AJ	
M-REGISTER CONTENT	DTF2	15(A-GE)	AC	
INTO THE	OTG(1)*			
Q-REGISTER	VTG(2)	15(A-GE)	H	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
113				
COLLATE OUT THE 13	AA1	10(A-GJ)	C	
LEAST SIGNIFICANT	AA2	10(A-GJ)	D	
BITS OF THE	AA3	10(A-GJ)	E	
M-REGISTER CONTENT	MTF	11(A-GI)	AJ	
AND WRITE IN INTO	DTF1	15(A-GE)	AD	
THE ADDRESS	OTG(1)*			
SPECIFIED BY THE	VTG(2)	15(A-GE)	H	
J-REGISTER	OTM(3)	10(A-GJ)	AA	
	GTM(4)	11(A-GI)	AA	
	CLEAR	11(A-GI)	M	
	WRITE	11(A-GI)	K	

FUNCTIONS 9, 10 & 11

*See note at foot of Page 2.

MULTIPLY

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
120				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
M-REGISTER TO THE	OTG(1)*			
J-REGISTER EXTRACT	FTG(2)	11(A-GI)	F	
THAT OPERAND	OTM(3)	10(A-GJ)	AA	
	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
121				
TRANSFER CONTENT	AA2	10(A-GJ)	D	
OF A-REGISTER TO	ATF	11(A-GI)	Y	
Q-REGISTER SET	KTJ	13(A-GG)	U	
PROCESS COUNTER	TXQ1	14(A-GF)	AA	CONDITIONS
TO 17	OTG(1)*			TEST SIGNAL
CLEAR ACCUMULATOR	FTG(2)	11(A-GI)	F	
AND OVERFLOW	OTA(3)*			
COMPARE OVERFLOW	OTJ(3)	13(A-GG)	S	
WITH DIGIT Q1	OTQ(3)*			
	OTX(3)	12(A-GH)	X	
	GTQ(4)	12(A-GH)	AB	
	JTC(5)	11(A-GI)	H	
122A				
ADD THE CONTENTS	AA3	10(A-GJ)	E	X $\overline{Q1}$
OF THE A-REGISTER	ATF	11(A-GI)	Y	&
AND THE M-REGISTER	MTF	11(A-GI)	AJ	C ≠ 0
THEN RIGHT SHIFT	YTX	12(A-GH)	M	
THE RESULT 1 PLACE	OTG(1)*			
INTO THE A-REGISTER	RTG(2)	12(A-GH)	R	
COLLECT UNDERFLOW	OTA(3)*			
DECREMENT COUNTER	OTX(3)	12(A-GH)	X	
CONTENT BY 1	1TC(4)	11(A-GI)	AD	
	GTA(4)	10(A-GJ)	AE	
122B				
SUBTRACT THE	AA3	10(A-GJ)	E	$\overline{X}$ Q1
CONTENT OF THE	ATF	11(A-GI)	Y	&
M-REGISTER FROM	MTF	12(A-GH)	P	C ≠ 0
THAT OF THE	1TF	11(A-GI)	N	
A-REGISTER. SHIFT	YTX	12(A-GH)	M	
THE RESULT RIGHT 1	OTG(1)*			
PLACE INTO THE	RTG(2)	12(A-GH)	R	
A-REGISTER.	OTA(3)*			
COLLECT UNDERFLOW	OTX(3)	12(A-GH)	X	
DECREMENT COUNTER	1TC(4)	11(A-GI)	AD	
CONTENT BY 1	GTA(4)	10(A-GJ)	AE	
122C				
RIGHT SHIFT THE	AA3	10(A-GJ)	E	X ≡ Q1
CONTENT OF THE	ATF	11(A-GI)	Y	&
A-REGISTER 1 PLACE	YTX	12(A-GH)	M	C ≠ 0
COLLECT UNDERFLOW	OTG(1)*			
DECREMENT COUNTER	RTG(2)	12(A-GH)	R	
CONTENT BY 1	OTA(3)*			
	OTX(3)	12(A-GH)	X	
	1TC(4)	11(A-GI)	AD	
	GTA(4)	10(A-GJ)	AE	

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
124				
RIGHT SHIFT THE	AA2	10(A-GJ)	D	
CONTENT OF THE	QTF	11(A-GI)	AH	
Q-REGISTER BY 1	XTF	12(A-GH)	J	
PLACE.	YTX	12(A-GH)	M	
PICK UP CARRY-OVER	TCO	11(A-GI)	W	CONDITIONS
DIGIT FROM THE	OTG(1)*			TEST SIGNAL
A-REGISTER	RTG(2)	12(A-GH)	R	
COLLECT UNDERFLOW	OTX(3)	12(A-GH)	X	
TEST COUNTER FOR	OTQ(3)*			
ZERO CONTENT	GTQ(4)	12(A-GH)	AB	
COMPARE OVERFLOW	TXQ1	14(A-GF)	AA	CONDITIONS
WITH THE DIGIT Q1.				TEST SIGNAL
123A				
ADD THE CONTENTS	AA1	10(A-GJ)	C	X $\overline{Q1}$
OF THE A-REGISTER	AA2	10(A-GJ)	D	&
AND THE M-REGISTER	AA3	10(A-GJ)	E	C = 0
PLACE RESULT IN	ATF	11(A-GI)	Y	
THE A-REGISTER	MTF	11(A-GI)	AJ	
CLEAR OVERFLOW	OTG(1)*			
	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	OTX(3)	12(A-GH)	X	
	GTA(4)	10(A-GJ)	AE	
123B				
SUBTRACT THE	AA1	10(A-GJ)	C	$\overline{X}$ Q1
CONTENT OF THE	AA2	10(A-GJ)	D	&
M-REGISTER FROM	AA3	10(A-GJ)	E	C = 0
THAT OF THE	ATF	11(A-GI)	Y	
A-REGISTER.	MTF	12(A-GH)	P	
PLACE THE RESULT	1TF	11(A-GI)	N	
IN THE A-REGISTER.	OTG(1)*			
CLEAR OVERFLOW	FTG(2)	11(A-GI)	F	
	OTA(3)*			
	OTX(3)	12(A-GH)	X	
	GTA(4)	10(A-GJ)	AE	
123C				
CLEAR OVERFLOW	AA1	10(A-GJ)	C	X ≡ Q1
	AA2	10(A-GJ)	D	&
	AA3	10(A-GJ)	E	C = 0
	OTX(3)	12(A-GH)	X	

FUNCTION 12

*See note at foot of Page 2.

DIVIDE

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
130				
TRANSFER ADDRESS	AA1	10(A-GJ)	C	
OF OPERAND FROM	MTF	11(A-GI)	AJ	
THE M-REGISTER TO	OTG(1)*			
THE J-REGISTER	FTG(2)	11(A-GI)	F	
EXTRACT THAT	OTM(3)	10(A-GJ)	AA	
OPERAND	GTJ(4)	11(A-GI)	AF	
	READ	11(A-GI)	L	
	WRITE	11(A-GI)	K	
131				
RIGHT SHIFT THE	AA2	10(A-GJ)	D	
CONTENT OF THE	QTF	11(A-GI)	AH	
Q-REGISTER 1 PLACE.	OTG(1)*			
CLEAR THE	RTG(2)	12(A-GH)	R	
J-REGISTER	OTQ(3)*			
	OTJ(3)	13(A-GG)	S	
	GTQ(4)	12(A-GH)	AB	
132				
LEFT SHIFT THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
Q-REGISTER BY 1	QTF	11(A-GI)	AH	
PLACE. SET	KTJ	13(A-GG)	U	
COUNTER CONTENT	K2	13(A-GG)	T	
TO 18	OTG(1)*			
	LTG(2)	12(A-GH)	AA	
	OTQ(3)*			
	GTQ(4)	12(A-GH)	AB	
	JTC	11(A-GI)	H	
133				
LEFT SHIFT THE	AA3	10(A-GJ)	E	
CONTENT OF THE	ATF	11(A-GI)	Y	
A-REGISTER 1	YTX	12(A-GH)	M	
PLACE.	OTG(1)*			
COLLECT OVERFLOW	LTG(2)	12(A-GH)	AA	
COMPARE OVERFLOW	OTX(3)*			
WITH SIGN BIT	TXM18	13(A-GG)	V	CONDITIONS TEST SIGNAL
134A				
LEFT SHIFT THE	AA2	10(A-GJ)	D	$X \not\equiv M18$
CONTENT OF THE	AA3	10(A-GJ)	E	&
Q-REGISTER BY 1	QTF	11(A-GI)	AH	$C \neq 0$
PLACE.	YTX	12(A-GH)	M	
COLLECT OVERFLOW	CRS	12(A-GH)	AC	CONDITIONS RETENTION
DECREMENT THE	OTG(1)*			
COUNTER CONTENT	LTG(2)	12(A-GH)	AA	
BY 1. RETAIN	OTX(3)*			
SETTING OF	OTQ(3)*			
CONDITIONS	GTQ(4)	12(A-GH)	AB	
	1TC(4)	11(A-GI)	AD	
136A				
ADD THE CONTENTS	AA3	10(A-GJ)	E	
OF THE A-REGISTER	ATF	11(A-GI)	Y	
AND THE M-REGISTER	MTF	11(A-GI)	AJ	
LEFT SHIFT RESULT	XTF	12(A-GH)	J	
1 PLACE PICKING UP	YTX	12(A-GH)	M	
CARRY-OVER DIGIT	TCO	11(A-GI)	W	
FROM Q. COLLECT	OTG(1)*			
OVERFLOW COMPARE	LTG(2)	12(A-GH)	AA	
OVERFLOW WITH	OTA(3)*			
DIGIT M18. TEST	OTX(3)*			
COUNTER FOR ZERO	GTA(4)	10(A-GJ)	AE	
CONTENT	TXM18	13(A-GG)	V	

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
134B				
ADD 1 TO THE	AA2	10(A-GJ)	D	$X \equiv M18$
CONTENT OF THE	AA3	10(A-GJ)	E	&
Q-REGISTER AND	QTF	11(A-GI)	AH	$C \neq 0$
LEFT SHIFT THE	1TF	11(A-GI)	N	
RESULT 1 PLACE.	YTX	12(A-GH)	M	
COLLECT OVERFLOW	CRS	12(A-GH)	AC	CONDITIONS RETENTION
DECREMENT THE	OTG(1)*			
COUNTER CONTENT	LTG(2)	12(A-GH)	AA	
BY 1. RETAIN THE	OTX(3)*			
SETTING OF	OTQ(3)*			
CONDITIONS	GTQ(4)	12(A-GH)	AB	
	1TC(4)	11(A-GI)	AD	
136B				
SUBTRACT THE	AA3	10(A-GJ)	E	
CONTENT OF THE	ATF	11(A-GI)	Y	
M-REGISTER FROM	MTF	12(A-GH)	P	
THAT OF THE	1TF	11(A-GI)	N	
A-REGISTER. LEFT	XTF	12(A-GH)	J	
SHIFT THE RESULT 1	YTX	12(A-GH)	M	
PLACE PICKING UP	TCO	11(A-GI)	W	
CARRY-OVER DIGIT	OTG(1)*			
FROM Q.	LTG(2)	12(A-GH)	AA	
COLLECT OVERFLOW	OTA(3)*			
COMPARE OVERFLOW	OTX(3)*			
WITH SIGN BIT TEST	GTA(4)	10(A-GJ)	AE	
COUNTER FOR ZERO	TXM18	13(A-GG)	V	CONDITIONS TEST SIGNAL
CONTENT				
135				
ADD THE	AA1	10(A-GJ)	C	
"ROUND-OFF"	AA2	10(A-GJ)	D	
CONSTANT TO THE	AA3	10(A-GJ)	E	
Q-REGISTER CONTENT.	QTF	11(A-GI)	AH	
TRANSFER THE	1TF	11(A-GI)	N	
CONTENT OF THE	OTG(1)*			
Q-REGISTER TO THE	FTG(2)	11(A-GJ)	F	
A-REGISTER	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	

FUNCTION 13

*See note at foot of Page 2.

SHIFT (RIGHT)

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
140				
COMPARE THE DIGITS	AA2	10(A-GJ)	D	
M12 AND M13	TM12M13	12(A-GH)	T	)CONDITIONS
DETERMINE THE	TM13	10(A-GJ)	V	)TEST SIGNALS
LOGIC STATE OF M13				
142A				
SET UP THE	AA1	10(A-GJ)	C	M12=M13=1
NUMBER OF PLACES	AA2	10(A-GJ)	D	
OF SHIFT IN THE	MTF	12(A-GH)	P	
COUNTER AND THE	1TF	11(A-GI)	N	
J-REGISTER. RETAIN	CRS	11(A-GI)	AC	CONDITIONS
SETTING OF	OTG(1)*			RETENTION
CONDITIONS	FTG(2)	11(A-GI)	F	
	GTJ(4)	11(A-GI)	AF	
	JTC(5)	11(A-GI)	H	
143A				
TEST COUNTER FOR	AA3	10(A-GJ)	E	
ZERO CONTENT	CRS	11(A-GI)	AC	CONDITIONS
RETAIN SETTING OF	TCO	11(A-GI)	W	RETENTION
CONDITIONS				
144A				
RIGHT SHIFT THE	AA2	10(A-GJ)	D	C=0
CONTENT OF THE	AA3	10(A-GJ)	E	
A-REGISTER 1 PLACE	ATF	11(A-GI)	Y	
COLLECT OVERFLOW	CRS	11(A-GI)	AC	CONDITIONS
DECREMENT COUNTER	YTX	12(A-GH)	M	RETENTION
CONTENT BY 1	OTG(1)*			
RETAIN SETTING OF	RTG(2)	12(A-GH)	R	
CONDITIONS	OTA(3)*			
	OTX(3)	12(A-GH)	X	
	GTA(4)	10(A-GJ)	AE	
	1TC(4)	11(A-GI)	AD	
146A				
RIGHT SHIFT THE	AA3	10(A-GJ)	E	
CONTENT OF THE	QTF	11(A-GI)	AH	
Q-REGISTER 1 PLACE	XTF	12(A-GH)	J	
PICKING UP CARRY	CRS	11(A-GI)	AC	CONDITIONS
OVER DIGIT FROM THE	TCO	11(A-GI)	W	RETENTION
A-REGISTER	OTG(1)*			
RETAIN THE SETTING	RTG(2)	12(A-GH)	R	
OF THE CONDITIONS	OTQ(3)*			
TEST COUNTER FOR	GTQ(4)	12(A-GH)	AB	
ZERO CONTENT				
145				
DO	AA1	10(A-GJ)	C	C=0
NOTHING	AA2	10(A-GJ)	D	
	AA3	10(A-GJ)	E	

(LEFT)

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
142B				
SET UP THE	AA1	10(A-GJ)	C	M12=M13=0
NUMBER OF PLACES	AA2	10(A-GJ)	D	
OF SHIFT IN THE	MTF	11(A-GI)	AJ	
COUNTER AND THE	CRS	11(A-GI)	AC	CONDITIONS
J-REGISTER	OTG(1)*			RETENTION
RETAIN SETTING OF	FTG(2)	11(A-GJ)	F	
CONDITIONS	GTJ(4)	11(A-GI)	AF	
	JTC(5)	11(A-GI)	H	
143B				
TEST COUNTER FOR	AA3	10(A-GJ)	E	
ZERO CONTENT	CRS	11(A-GI)	AC	CONDITIONS
RETAIN SETTING OF	TCO	11(A-GI)	W	RETENTION
CONDITIONS				
144B				
LEFT SHIFT THE	AA2	10(A-GJ)	D	C=0
CONTENT OF THE	AA3	10(A-GJ)	E	
Q-REGISTER 1 PLACE	QTF	11(A-GI)	AH	CONDITIONS
COLLECT OVERFLOW	CRS	11(A-GI)	Y	RETENTION
DECREMENT COUNTER	YTX	12(A-GH)	M	
CONTENT BY 1	OTG(1)*			
RETAIN SETTING OF	LTG(2)	12(A-GH)	AA	
CONDITIONS	OTQ(3)*			
	OTX(3)	12(A-GH)	X	
	GTQ(4)	12(A-GH)	AB	
	1TC(4)	11(A-GI)	AD	
146B				
LEFT SHIFT THE	AA3	10(A-GJ)	E	
CONTENT OF THE	ATF	11(A-GI)	Y	
A-REGISTER 1 PLACE	XTF	12(A-GH)	J	
PICKING UP CARRY	CRS	11(A-GI)	AC	CONDITIONS
OVER DIGIT	TCO	11(A-GI)	W	RETENTION
RETAIN THE SETTING	OTG(1)*			
OF THE CONDITIONS	LTG(2)	12(A-GH)	AA	
TEST COUNTER FOR	OTA(3)*			
ZERO CONTENT	GTA(4)	10(A-GJ)	AE	
145				
DO	AA1	10(A-GJ)	C	
NOTHING	AA2	10(A-GJ)	D	C=0
	AA3	10(A-GJ)	E	

FUNCTION 14

*See note at foot of Page 2.

BLOCK TRANSFER

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
140	AA2	10(A-GJ)	D	
COMPARE THE DIGITS	TM12M13	12(A-GH)	T	)CONDITIONS
M12 AND M13	TM13	10(A-GJ)	V	)TEST SIGNAL
DETERMINE THE				
LOGIC STATE OF M13				
				M12 ≠ M13
14A2				
SELECT PERIPHERAL	AA1	10(A-GJ)	C	
SET COUNTER EQUAL	QTF	11(A-GI)	AM	
TO THE CONTENT OF	OTG(1)*			
THE Q-REGISTER	FTG(2)	11(A-GI)	F	
	MTP(2)	10(A-GI)	Z	
	GTJ(4)	11(A-GI)	AF	
	JTC(5)	11(A-GI)	H	
14A1				
SET THE STARTING	AA1	10(A-GJ)	C	
ADDRESS OF THE	AA2	10(A-GJ)	D	
TRANSFER TO THE	ATF	11(A-GI)	Y	
CONTENT OF THE	OTG(1)*			
A-REGISTER.	FTG(2)	11(A-GI)	F	
SET THE BLOCK	OTJ(3)*			
TRANSFER BISTABLE	GTJ(4)	11(A-GI)	AF	
AND INSPECT THE	SBT(4)	(11A-GI)	R	)CONDITIONS
DIGIT M13	TM13	10(A-GJ)	V	)TEST SIGNAL
14A3C				
CLEAR THE	AA3	10(A-GJ)	E	M13 = 1
M-REGISTER READ	TC0	11(A-GI)	W	
THE SPECIFIED STORE	CRS	11(A-GI)	AC	
LOCATION TEST	OTM(3)	10(A-GJ)	AA	
COUNTER FOR ZERO	READ	11(A-GI)	L	
CONTENT RETAIN	WRITE	11(A-GI)	K	
SETTING OF				
CONDITIONS.				
14A4C				
TRANSFER CONTENT	AA2	10(A-GJ)	D	C≠ 0
OF THE M-REGISTER	AA3	10(A-GJ)	E	
TO THE A-REGISTER	MTF	11(A-GI)	AJ	
RETAIN SETTING OF	CRS	11(A-GI)	AC	
CONDITIONS	OTG(1)*			
	FTG(2)	11(A-GI)	AF	
	OTA(3)*			
	GTA(4)	10(A-GJ)	AE	

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
14A6C				
SELECT PERIPHERAL	AA1	10(A-GJ)	C	
AND AWAIT REPLY	AA2	10(A-GJ)	D	
INCREMENT THE	SZ1	10(A-GJ)	S	
CONTENT OF THE	WFP	10(A-GJ)	J	
J-REGISTER BY 1	JTF	11(A-GI)	AB	
DECREMENT COUNTER	1TF	11(A-GI)	AD	
CONTENT BY 1	CRS	11(A-GI)	AC	
RETAIN SETTING OF	OTG(1)*			
CONDITIONS	FTG(2)	11(A-GI)	F	
	OTJ(3)*			
	GTJ(4)	11(A-GI)	AF	
	1TC(4)	11(A-GI)	N	
14A3D				
TEST COUNTER FOR	AA3	10(A-GJ)	E	M13=0
ZERO CONTENT	TC0	11(A-GI)	W	
RETAIN SETTING OF	CRS	11(A-GI)	AC	
CONDITIONS CLEAR	OTM(3)	10(A-GJ)	AA	
M-REGISTER				
14A4D				
SELECT PERIPHERAL	AA2	10(A-GJ)	D	
AND AWAIT REPLY	AA3	10(A-GJ)	E	
INPUT ONE WORD AND	SZ1	10(A-GJ)	S	
WRITE INTO THE	WFP	10(A-GJ)	J	
STORE ADDRESS	CRS	11(A-GI)	AC	
SPECIFIED	OTG(1)*			
RETAIN THE SETTING	PTG1(2)			
OF CONDITIONS	GTM(4)			
	CLEAR			
	WRITE			
14A6D				
INCREMENT THE	AA1	10(A-GJ)	C	
CONTENT OF THE	AA2	10(A-GJ)	D	
J-REGISTER BY 1	JTF	11(A-GI)	AB	
DECREMENT THE	1TF	11(A-GI)	AD	
COUNTER CONTENT	CRS			
BY 1	OTG(1)*			
RETAIN SETTING	FTG(2)	11(A-GI)	F	
OF CONDITIONS	OTJ(3)*			
	GTJ(4)	11(A-GI)	AF	
	1TC(4)	11(A-GI)	N	
14A5				
RESET THE BLOCK	AA1	10(A-GJ)	C	C=0
TRANSFER	AA2	10(A-GJ)	D	
	AA3	10(A-GJ)	E	
	RBT(4)	11(A-GI)	U	

FUNCTION 14A

*See note at foot of Page 2.

INPUT OUTPUT PROGRAM TERMINATE

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS
150 SET UP ADDRESS OF PERIPHERAL AND INSPECT THE DIGIT M13.	AA1 TM13 MTP(2)	10(A-GJ) 10(A-GJ) 10(A-GJ)	C V Z	
151A INSPECT THE DIGIT M12	AA2 TM12	10(A-GJ) 10(A-GJ)	D U	M13=0
152B SEND SELECT TO PERIPHERAL AND AWAIT REPLY THEN INPUT ONE WORD INTO THE ACCUMULATOR	AA1 AA2 AA3 SZ1 WFP OTG(1)* PTG1(2) OTA(3)* GTA(4)	10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ)	C D E S J P AE	M12=0
152A SEND SELECT TO TAPE READER AND AWAIT REPLY THEN INPUT ONE CHARACTER TO THE ACCUMULATOR	AA1 AA2 AA3 SZ1 WFP OTG(1)* PTG2(2) OTA(3)* GTA(4)	10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ)	C D E S J N AE	M12=1
151B INSPECT THE DIGITS M11 AND M12	AA1 AA2 TM11 TM12	10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ)	C D T U	M13=1
153B SEND SELECT TO PERIPHERAL AND AWAIT REPLY	AA1 AA2 AA3 SZ1 WFP	10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ)	C D E S J	M12=0 or M12=1 and M11=0
153A TERMINATE PROGRAM AT CURRENT LEVEL	AA1 AA2 AA3 PTR	10(A-GJ) 10(A-GJ) 10(A-GJ) 10(A-GJ)	C D E AB	M11=M12 =1

MATRIX ADDRESS	MATRIX OUTPUTS	BOARD	PIN	MATRIX CONDITIONS

FUNCTION 15

*See note at foot of Page 2.

Appendix 2: PROGRAM X3 (FUNCTION TESTS)

1. FUNCTION

This program tests that the 900 central processor performs all its functions correctly. Functions 8, 10 and 15 are not tested directly, but it is assumed that they must work if an error character is to be punched out correctly.

2. PROGRAM TYPE

X3 is distributed as a sum-checked binary tape suitable for input by initial instructions. The tape carries a clear store routine at its beginning and end. If, after reading, continuous output occurs the program was not read correctly.

3. PROGRAM ACTION

The first 'clear store' routine is read in by initial instructions and when it has been obeyed the main function test section is read in automatically.

The program is triggered at location 8 and when triggered each of the functions mentioned is tested in the order shown in Tables 1 and 2. This test cycle is repeated 150 times and then a character is read from paper tape. If the character is a blank the test is repeated a further 150 times. Otherwise, the legible punch-out 'FUNCTION TEST OK' is given if the test succeeds and the second clear-store routine is read in and obeyed.

If any of the functions should fail an error character, whose significance can be seen in Table 2, will be punched out.

All errors are punched on paper tape as stated and an error will be one of 57 characters. An error character is output continuously until some action is taken by the operator.

The program may be stopped at any time by pressing the Stop button, but if this is done the visipunched output is lost.

4. OPERATING INSTRUCTIONS

Set the computer mode switch to OPERATE.

Load the program tape in reader (the visipunched leader should not pass under reader).

Set the address keys to 8181 (initial instructions).

Press the Jump button.

After the first clear store routine and the main function test section has been read in set the address keys to 8.

Press the Jump button whereupon the test shall begin.

5. TESTS APPLIED

(1) Function 4 - READ

Load the accumulator from a store location which was set to zero. An error printout will occur if the accumulator is not zero, or Function 7 is failing.

(2) Function 5 - WRITE

Read back into the accumulator the number which was stored by test 1. An error printout will occur if the accumulator is not zero or Function 4 is failing.

(3) Function 7 - JUMP IF ZERO

(a) Load accumulator with -1 and tested for zero content. Error output if computer jumps.

(b) Load accumulator with +2 and test for zero content.

Error output if computer jumps.

(4) Function 9 - JUMP IF NEGATIVE

(a) Load accumulator with zero and test if negative.

Error printout if computer jumps.

(b) Load accumulator with -1 and test if negative.

Error printout if computer does not jump.

(5) Function 1 - ADD

(a) Load the accumulator with zero and add zero.

Test result with Function 7.

Error printout occurs if computer does not jump.

(b) Load the accumulator with -1 and add +1.

Test result with Function 7.

Error printout occurs if computer does not jump.

(c) Load the accumulator with +1 and add -1.

Test result with Function 7.

Error output if computer does not jump.

(d) Load accumulator with 1.01010101010101010 and add 0.10101010101010101

Test result with Function 9.

Error printout if computer fails to jump.

Add +1 to the accumulator of previous test.

Test with Function 7.

Error output if computer fails to jump.

(e) Load accumulator with 0.10101010101010101010 and add 1.01010101010101010

Test with Function 9.

Error output if computer does not jump.

Add +1 to previous accumulator.

Test with Function 7.

Error output if computer fails to jump.

(f) Load accumulator with 1.01010101010101010 and
add 1.01010101010101010 and
add 1.01010101010101010

Test with Function 9.

Error output if computer fails to jump.

Add +1 to accumulator.

Test result with Function 7.

Error printout if computer does not jump.

(g) Load the accumulator with 0.10101010101010101 and
add 0.10101010101010101 and
add 0.10101010101010101

Test result with Function 9.

Error printout if computer fails to jump.

Add +1 to the accumulator.

Test with Function 7.

Error output if computer fails to jump.

(6) Function 2 - NEGATE AND ADD

(a) Load accumulator with zero then negate and add zero.

Test result with Function 7.

Error output if computer fails to jump.

(b) Load the accumulator with -1

Negate and add -1

Test result with Function 7.

Error output if computer fails to jump.

(c) Load accumulator with zero.

Negate and add -1

Test result with Function 9.

Error output if computer fails to jump.

Add +1 to accumulator.

Test with Function 7.

Error output if computer does not jump.

(d) Load accumulator with -1

Negate and add zero.

Test with Function 9.

Error output if computer jump.

Add -1 to accumulator.

Test with Function 7.

Error output if computer does not jump.

(7) Function 6 - COLLATE

(a) Load accumulator with zero.

Collate with -1

Test result with Function 7.

Error output if computer fails to jump.

(b) Load accumulator with -1

Collate with zero.

Test result with Function 7.

Error output if computer fails to jump.

(c) Load accumulator with -1

Collate with -1

Test result with Function 9.

Error if computer fails to jump.

Add +1 to result.

Test result with Function 7.

Error output if computer fails to jump.

(d) Load accumulator with 0000000001111111111(+511)
Collate with 0000111110000011111(+15903)
Negate and add 0.00000000000011111(+31)

Test result with Function 7.

Error output if computer fails to jump.

(8) Function 10 - COUNT IN STORE

Load accumulator with -1

Store in workspace (location 521).

Count in workspace (location 521).

Read location workspace.

Test result with Function 7.

Error printout if computer does not jump.

(9) Function 0 - SET B REGISTER

Set level 1 B-register to 0.00000101010101010

Read the level 1 B-register.

Negate and add 0.00000101010101010

Test result with Function 7.

(10) Function 11 - STORE S.C.R.

Store SCR in location 521.

Read location 521.

Collate out function digits.

Negate and add function digits.

Test with Function 7.

Error if computer fails to jump.

(11) Function 3 - STORE Q-REGISTER

Set Q-register to 1010101010101010 (/5 2730)

Store Q-register (bits 18-2 of Q stored in bits 17-1 of location)

Read store location

Negate and add 0101010101010101 (10 5461)

Test result with Function 7.

Error output if computer fails to jump.

(12) Function 14 - SHIFT

(a) Set Q-register to 101010101010101010 (/5 2730)

Load accumulator with +1

Shift A.Q zero places.

Negate and add +1

Test result with Function 7.

Error output if computer fails to jump.

(b) Set Q-register to 010101010101010101

Load accumulator with 010101010101010101

Shift AQ left one place (A holds 101010101010101010)

Store Q-register (store holds 010101010101010101)

Negate and add 101010101010101010 (/5 2730)

Test result with Function 7.

Error printout if computer fails to jump.

Read back Q-register [c(Q) = 010101010101010101]

Negate and add 010101010101010101

Test result with Function 7.

Error printout if computer fails to jump.

(c) Set Q-register to 010101010101010101

Load accumulator with 010101010101010101

Shift AQ one place right c(A):= 001010101010101010

Store Q-register (store holds 101010101010101010)

Negate and add to accumulator 001010101010101010)

Test result with Function 7.

Error printout if computer fails to jump.

Read back Q-register [c(A):= 010101010101010101]

Negate and add 010101010101010101

Test result with Function 7.

Error printout if computer fails to jump.

(d) Set Q-register to 101010101010101010

Load accumulator with 101010101010101010

Left shift AQ one place c(A) 010101010101010101

Store Q-register (store holds 001010101010101010)

Negate and add to accumulator 010101010101010101

Test result with Function 7.

Error printout if computer fails to jump.

Read back Q-register c(A) 001010101010101010

Negate and add 001010101010101010

Test result with Function 7.

Error printout if computer fails to jump.

(e) Set Q-register to 101010101010101010

Load accumulator 101010101010101010

Right shift AQ one place c(A):= 110101010101010101

Store Q-register (store holds 001010101010101010)

Negate and add 110101010101010101

Test result with Function 7.

Error printout if computer fails to jump.

Read back Q-register 001010101010101010

Negate and add 001010101010101010

Test result with Function 7.

Error printout if computer fails to jump.

(f) Set Q-register to 000000000000000010 (+2)

Load accumulator with zero.

Left shift AQ two places left.

Store Q-register [store holds 000000000000000100 (+4)]

Read back Q-register [c(A):= 0.0000000000000110 (+4)]

Negate and add +4

Test result with Function 7.

Error printout if computer fails to jump.

(g) Set Q-register to +2
Load accumulator with zero
Shift AQ left 4 places.
Store Q-register [store holds +16 (0.00000000000110000)]
Read back Q-register c(A):= +16
Negate and add +16
Test with Function 7.
Error printout if computer fails to jump.

(h) Repeat 'g' but with an 8 place left shift.

(i) Repeat 'g' but with a 16 place left shift.

(j) Repeat 'g' but with a 32 place left shift.

(k) Set Q-register to +8 (0.00000000000010000)
Load accumulator with zero.
Right shift AQ 2 places.
Store Q-register [store holds (0.00000000000000001)+1]
Read back Q-register c(A):= +1
Negate and add +1
Test result with Function 7.

(l) Repeat 'k' with a 4 place shift.

(m) Repeat 'k' with an 8 place shift.

(n) Repeat 'k' with a 16 place shift.

(o) Repeat 'k' with a 32 place shift.

(p) Set Q-register to +1 0.00000000000000001
Load accumulator with zero.
Shift AQ left 35 places [c(A):= 1.00000000000000000]
Test result with Function 9.
Error output if computer fails to jump.

(q) Load the accumulator with 100000000000000000 (-131072)
Right shift AQ 35 places.
Store accumulator.
Shift left 18 places.
Negate and add +1
Test result with Function 7.
Error printout if computer fails to jump.

(13) Function 12 - MULTIPLY

(a) Load accumulator with /6 3277 (1011001100110011101)
Multiply by /6 8277 (- 78,643)
Store accumulator.
Shift Q into A.
Negate and add /5 6227 (101011100001010011)
Test with Function 7.
Error printout if computer fails to jump.
Read back the accumulator.
Negate and add 5 6225 (001011100001010000)
Test with Function 7.
Error printout if computer fails to jump.

(b) Load the accumulator with /6 3277 (1011001100110011001)
Multiply by +2
Store accumulator and shift Q into A.
Negate and add /9 4917 (110011001100110101)
Test result with Function 7.
Error printout if computer fails to jump.
Read back accumulator.
Negate and add /15 8190 (1.1111111111111110)
Test result with Function 7.
Error printout if computer fails to jump.

(14) Function 13 - DIVIDE

(a) Load accumulator with /6 3277
Multiply by /6 3277
Divide by /6 3277
Store accumulator.
Shift Q into A.
Negate and add /6 3276.
Test result with Function 7.
Error printout if computer fails to jump.
Read back accumulator.
Negate and add /6 3277
Test result with Function 7.

(b) Load the accumulator with /6 3277
Multiply by /6 3277
Divide by /6 3277
Store accumulator.
Shift Q into A.
Negate and add 9 4914.
Test with Function 7.
Error output if computer fails to jump.
Read back accumulator.
Negate and add 4 4915
Test result with Function 7.
Error printout if computer fails to jump.

(c) Load accumulator with /6 3277
Multiply by +2
Divide by +2
Negate and add /6 3277
Test result with Function 7.
Error printout if computer fails to jump.

(d) Load the accumulator with /6 3277

Multiply by +2

Divide by /6 3277

Negate and add +1

Test result with Function 7.

Error printout if computer fails to jump.

(15) B-Line Modification

Set the Level 1 B-register to +2

Read modified [c(A):= 0101010101010101]

Negate and add 0101010101010101

Test result with Function 7.

Error printout if computer fails to jump.

6. ERROR INDICATIONS

Error Character	Function Failing	Cross Refer with Table 1	Locations of test
00000.001	4		12 - 16
00000.010	5		17 - 20
00000.011	7 a		21 - 23
00000.100	7 b		24 - 26
00000.101	9 a		27 - 29
00000.110	9 b		30 - 33
00000.111	1 a		34 - 38
00001.000	1 b		39 - 43
00001.001	1 c		44 - 48
00001.010	1 d		49 - 56
00001.011	1 e		57 - 64
00001.100	1 f		65 - 73
00001.101	1 g		74 - 82
00001.110	2 a		83 - 87
00001.111	2 b		88 - 92
00010.000	2 c		93 - 100
00010.001	2 d		101 - 107
00010.010	6 a		108 - 112
00010.011	6 b		113 - 117
00010.100	6 c		118 - 125
00010.101	6 d		126 - 131
00010.110	10		132 - 140
00010.111	0		141 - 145
00011.000	11		146 - 152
00011.001	3		153 - 159
00011.010	14 a		160 - 166
00011.011	14 b - A.wrong		167 - 174
00011.100	14 b - Q.wrong		175 - 179

Error Character	Function Failing	Cross Refer with Table 1	Locations of test
00011.101	14 c		180 - 191
00011.110	14 c - A.wrong		192 - 199
00011.111	14 d - Q.wrong		200 - 204
00100.000	14 e - A.wrong		205 - 212
00100.001	14 e - Q.wrong		213 - 217
00100.010	14 f - Q.wrong		218 - 226
00100.011	14 g - Q.wrong		227 - 235
00100.100	14 h - Q.wrong		236 - 244
00100.101	14 i - Q.wrong		245 - 253
00100.110	14 j - A.wrong		254 - 260
00100.111	14 k - Q.wrong		261 - 269
00101.000	14 l - Q.wrong		270 - 278
00101.001	14 m- Q.wrong		279 - 287
00101.010	14 n - Q.wrong		288 - 296
00101.011	14 o - Q.wrong		297 - 305
00101.100	14 p - A.wrong		306 - 314
00101.101	14 q - Q.wrong		388 - 396
00101.110	14 q - A.wrong		397 - 400
00101.111	12 a - AR wrong		315 - 322
00110.000	12 a - A.wrong		323 - 327
00110.001	12 b - AR.wrong		328 - 335
00110.010	12 b - A.wrong		336 - 340
00110.011	13 a - Q.wrong		341 - 349
00110.100	13 a - A.wrong		350 - 354
00110.101	13 b - Q.wrong		355 - 363
00110.110	13 b - A.wrong		364 - 368
00110.111	13 c - A.wrong		369 - 375
00111.000	13 d - A.wrong		376 - 381
00111.001	B-modification		382 - 387

7. PROGRAM SHEETS

7.1 Block 1

ADDRESS PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
				Block 1
(8)	0	4	0,2	Set cycle count
	1	5	0,3	= -150
	2	4	1,2	Set character for error punch
	3	5	1,3	
	4	4	2,2	Read zero
	5	5	2,3	Write
	6	7	8,	If READ OK jump to next test
	7	8	408,	READ Error - punch 00000.001
	8	10	1,3	
	9	4	2,3	Read back from store
	10	7	12,	If WRITE OK - jump to next test
	1	8	408,	WRITE Error - punch 00000.010
	2	10	1,3	
	3	4	408,	Jump if zero error punch 00000.011
	5	10	1,3	
	6	4	4,2	Read +2
	7	7	408,	Jump if zero - Error punch 00000.100
	8	10	1,3	
	9	4	2,2	Read zero
	20	9	408,	Jump if -ve Error punch 00000.101

ADDRESS PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	N	NOTES
	1	10	1,3	
	2	4	3,2	Read -1
	23	9	25,	
	4	8	408,	Jump if -ve Error punch 00000.110
	5	10	1,3	
	6	4	2,2	Read zero
	7	1	2,2	Add zero
	8	7	30,	jump if Add OK
	9	8	408,	Add Error punch 00000.111
	30	10	1,3	
	1	4	3,2	Read -1
	2	1	1,2	Add +1
	3	7	35,	If ADD OK jump to next test
	4	8	408,	Add error - punch 00001.000
	5	10	1,3	
	6	4	1,2	Read +1
	7	1	3,2	Add -1
	8	7	40,	If ADD OK jump to next test
	9	8	408,	Add Error punch 00001.001
	40	10	1,3	
	1	4	5,2	Read 101010101010101010
	2	1	6,2	Add 010101010101010101
	3	9	45,	If OK jump to next test
	4	8	408,	Add Error
	5	1	1,2	Add +1
	46	7	48,	If OK JUMP
	7	8	408,	Add Error
	8	10	1,3	

ADDRESS PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	9	4	6,2	Read 0101010101010101
	50	1	5,2	Add 1010101010101010
	1	9	53,	
	2	8	408,	Error
	3	1	1,2	Add +1
	4	7	56,	jump if OK
	5	8	408,	Error
	6	10	1,3	
	7	4	5,2	Read 1010101010101010
	8	1	5,2	Add 1010101010101010
	9	1	5,2	Add 1010101010101010
	60	9	62,	
	1	8	408,	Error
	2	1	4,2	Add +2
	3	7	65,	jump if OK
	4	8	408,	
	5	10	1,3	
	6	4	6,2	Read 0101010101010101
	7	1	6,2	Add 0101010101010101
	8	1	6,2	Add 0101010101010101
	69	9	71,	
	70	8	408,	Error
	1	1	1,2	Add +1
	2	7	74,	
	3	8	408	
	4	10	1,3	
	5	4	2,2	Read zero
	6	2	2,2	Negate and add zero

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	7	7	79,	
	8	8	408,	Error
	9	10	1,3	
	80	4	3,2	Read -1
	1	2	3,2	Negate and add -1
	2	7	84,	
	3	8	408,	Error
	4	10	1,3	
	5	4	2,2	Read zero
	6	2	3,2	Negate and add -1
	7	9	89,	
	8	8	408,	Error
	9	1	1,2	Add +1
	90	7	92,	
	1	8	408,	Error
	2	10	1,3	
	3	4	3,2	Read -1
	4	2	2,2	Negate and add zero
	5	9	408,	Error
	6	1	3,2	Add -1
	7	7	99,	
	8	8	408,	Error
	9	10	1,3	
	100	4	2,2	Read zero
	1	6	3,2	Collate with -1

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	2	7	104,	If OK jump to next test
	3	8	408,	Error
	4	10	1,3	
	5	4	3,2	Read -1
	6	6	2,2	Collate with zero
	7	7	109,	
	8	8	408,	Error
	9	10	1,3	
	110	4	3,2	Read -1
	1	6	3,2	Collate with -1
	2	9	114,	
	3	8	408,	Error
	4	1	1,2	Add +1
	5	7	117,	
	6	8	408,	Error
	7	10	1,3	
	8	4	7,2	Read +511
	119	6	8,2	Collate with +15903
	120	2	9,2	Test result is +31
	1	7	123,	
	2	8	408,	Error
	3	10	1,3	
	4	4	3,2	Read -1
	5	5	2,3	Store in workspace
	6	10	2,3	Count in workspace
	7	4	2,3	Read workspace

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	8	7	3,3	Jump if OK
	9	4	32,2	Set error character for
	130	5	1,3	function 10 = +22
	1	8	408,	Error output
	2	0	10,2	Set B-register
	3	4	1	Read B-register
	4	2	10,2	Test
	5	7	137,	Jump if OK
	6	8	408,	Error
	7	10	1,3	
	8	11	2,3	
	9	4	2,3	
	140	6	11,2	Collate out function bits
	1	2	12,2	Test
	2	7	144,	Jump if OK
	3	8	408,	
	4	10	1,3	
	145	0	5,2	Set Q = 101010101010101010
	6	3	2,3	Store AR
	7	4	2,3	Read back
	8	2	6,2	Test
	9	7	151,	Jump if OK
	150	8	408,	Error
	1	10	1,3	
	2	0	5,2	Q = 101010101010101010
	3	4	1,2	A = 00000000000000001
	4	14	0	Shift zero places
	5	2	1,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	6	7	158,	
	7	8	408,	Error accumulator
	8	10	1,3	
	9	0	6,2	Q = 010101010101010101
	160	4	6,2	A = 010101010101010101
	1	14	1	Shift AQ left one place
	2	3	2,3	Tests accumulator shift
	3	2	5,2	c(5,2) = /5 2730
	4	7	166,	
	5	8	408,	Error - Accumulator
	6	10	1,3	
	7	4	2,3	Tests Q-register shift
	8	2	6,2	
	169	7	171,	
	170	8	408,	Error - Accumulator
	1	10	1,3	
	2	0	6,2	
	3	4	6,2	
	4	14	8191	Shift AQ right 1 place
	5	3	2,3	Store Q
	6	2	13,2	
	7	7	179,	
	8	8	408,	Error - Accumulator
	9	4	2,3	
	180	2	6,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	1	7	183,	
	2	8	408,	Error - Q-register
	3	10	1,3	
	4	0	5,2	
	5	4	5,2	
	6	14	1	Shift AQ left one place
	7	3	2,3	
	8	2	6,2	
	9	7	191,	
	190	8	408,	Error - Accumulator
	1	10	1,3	
	2	4	2,3	
	3	2	13,2	
	194	7	196,	
	5	8	408,	Error - Accumulator
	6	10	1,3	
	7	0	5,2	
	8	4	5,2	
	9	14	8191	Shift right 1 place
	200	3	2,3	
	1	2	14,2	
	2	7	204,	
	3	8	408,	Error - Accumulator
	4	10	1,3	
	5	4	2,3	
	6	2	13,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	7	7	209,	
	8	8	408,	Error - Accumulator
	9	10	1,3	
	210	0	4,2	Q = 0000000000000010
	1	4	2,2	A = 0000000000000000
	2	14	2	Shift left two places
	3	3	2,3	
	4	4	2,3	
	5	2	15,2	
	6	7	218,	
	7	8	408,	Error - Accumulator
	8	10	1,3	
	219	0	4,2	
	220	4	2,2	
	1	14	4	Shift left 4 places
	2	3	2,3	
	3	4	2,3	
	4	2	16,2	
	5	7	227,	
	6	8	408,	Error Q-register
	7	10	1,3	
	8	0	4,2	
	9	4	2,2	
	230	14	8	
	1	3	2,3	
	2	4	2,3	
	3	2	17,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	4	7	236,	
	5	8	408,	Error Q-register
→	6	10	1,3	
	7	0	4,2	
	8	4	2,2	
	9	14	16	
	240	3	2,3	
	1	4	2,3	
	2	2	18,2	
	243	7	245,	
	4	8	408,	Error Q-register
→	5	10	1,3	
	6	0	4,2	
	7	4	2,2	
	8	14	32	
	9	2	19,2	
	250	7	252,	
	1	8	408,	Error Q-register
→	2	10	1,3	
	3	0	20,2	Q = 000000000000001000
	4	4	2,2	A = 0.000000000000000000
	5	14	8190	Q = 000000000000000010
	6	3	2,3	
	7	4	2,3	
	8	2	1,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	9	7	261,	
	260	8	408,	Error Q-register
	1	10	1,3	
	2	0	21,2	Q = 000000000000100000
	3	4	2,2	A = 0
	4	14	8188	Q = 000000000000000010
	5	3	2,3	
	6	4	2,3	
	7	2	1,2	
	268	7	270,	
	9	8	408,	Error Q-register
	270	10	1,3	
	1	0	22,2	Q = 00000000.1000000000
	2	4	2,2	
	3	14	8184	Shift AQ right 8 places
	4	3	2,3	
	5	4	2,3	
	6	2	1,2	
	7	7	279,	
	8	8	408,	Error Q-register
	9	10	1,3	
	280	0	23,2	
	1	4	2,2	
	2	14	8176	Shift AQ 16 places right
	3	3	2,3	Q = 000000000000000010
	4	4	2,3	
	5	2	1,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	6	7	288,	
	7	8	408,	Error Q-register
	8	10	1,3	
	9	0	2,2	Q = 0.00000000000000000
	290	4	19,2	A = 0.01000000000000000
	1	14	8160	Shift AQ 32 places right
	2	3	2,3	
	293	4	2,3	
	4	2	1,2	
	5	7	297,	
	6	8	408,	Error Q-register
	7	10	1,3	
	8	0	1,2	Q = 0.00000000000000001
	9	4	2,2	A = 0.00000000000000000
	300	14	35	Shift left 35 places
	1	9	303,	
	2	8	408,	Error Accumulator
	3	2	23,2	
	4	7	380,	
	5	8	408,	Error Accumulator
391	6	10	1,3	
	7	4	24,2	
	8	12	24,2	
	9	5	2,3	
	310	14	18	
	1	2	25,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	2	7	314,	
	3	8	408,	Error - Q wrong
	4	10	1, 3	
	5	4	2, 3	
	6	2	26, 2	
	317	7	319,	
	8	8	408,	Error Accumulator
	9	10	1, 3	
	320	4	24, 2	
	1	12	4, 2	
	2	5	2, 3	
	3	14	18	
	4	2	27, 2	
	5	7	327,	
	6	8	408,	Error - Q-register
	7	10	1, 3	
	8	4	2, 3	
	9	2	28, 2	
	330	7	332,	
	1	8	408,	Error - Accumulator
	2	10	1, 3	
	3	4	24, 2	
	4	12	24, 2	
	5	13	24, 2	
	6	5	2, 3	
	7	14	18	
	8	2	29, 2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	9	7	341,	
	340	8	408,	Error Q-register
→	1	10	1,3	
	342	4	2,3	
	3	2	24,2	
	4	7	346,	
	5	8	408,	Error Accumulator
→	6	10	1,3	
	7	4	24,2	
	8	12	24,2	
	9	13	30,2	
	350	5	2,3	
	1	14	18	
	2	2	31,2	
	3	7	355,	
	4	8	408,	Error Q-register
→	5	10	1,3	
	6	4	2,3	
	7	2	30,2	
	8	7	360,	
	9	8	408,	Error - Accumulator
	360	10	1,3	
	1	4	24,2	
	2	12	4,2	
	3	13	4,2	
	4	2	24,2	

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	365	7	367,	
	6	8	408,	Error - Accumulator
	7	10	1,3	
	8	4	24,2	
	9	12	4,2	
	370	13	24,2	
	1	2	1,2	
	2	7	374,	
	3	8	408,	Error - Accumulator
	4	10	1,3	
	5	0	4,2	
	6	14	4,2	
	7	2	6,2	
	8	7	393,	
	9	8	408,	Error - Accumulator
	380	10	1,3	
	1	4	23,2	Acc -131072
	2	14	8157	35 places right shift
	3	5	2,3	Store Accumulator
	4	14	18	Shift left 18 places
	5	1	1,2	
	6	7	388,	
	7	8	408,	Error - Accumulator
	8	10	1,3	
	9	4	2,3	
	390	1	1,2	
	1	7	306,	
	2	8	408,	Error - Accumulator

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	3	10	0,3	
	4	4	0,3	Decrement Cycle Count
	5	9	2,	
	6	5	2,2	
	7	15	2048	Input
	8	7	0,	Repeat whole program if blank
	9	0	33,2	
	400	/4	98,2	
	1	15	6144	
	2	14	8184	
	3	15	6144	O/P legible 'FUNCTION
	4	10	1	TEST OK'
	5	4	1	
	6	9	400,	
	7	8	8181	Initial Instructions Input clear Store Routine
	8	4	1,3	
	9	15	6144	Error Character
	410	8	409	Output Continuously

7.2 Block 2 Constants

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
				Block 2, Constants
8	(419)0		-150	Cycle Count
	1		+1	
	2		+0	0.00000000000000000
	3	/15	8191	1.11111111111111111
	4		+2	
	5	/5	2730	1010101010101010
	6	10	5461	0101010101010101
	7		+511	0000000001111111111
	8	1	7711	0000111111000011111
	9		+31	0000000000000011111
	10		+2730	
	1		+8191	
	2	0	139,1	
	3	5	2730	0.0101010101010101
	4	/10	5461	1.0101010101010101
	5		+4	
	6		+16	
	7		+256	
	8	8	0	
	9	4	0	
	20		+8	
	1		+32	
	2		+512	
	3	/0	0	
	4	/6	3277	101100110011001101

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
	25	/5	6227	101011100001010011
	6	5	6225	001011100001010000
	7	/9	4917	110011001100110101
	8	/15	8190	1111111111111111110
	9	/6	3276	101100110011001100
	30	9	4915	010011001100110011
	1	9	4914	010011001100110010
	2		+22	
	3		-64	Count for legible O/P
	4		+2559	
	5		+2313	F
	6		+1	
	7	1	7936	
	8	4	64	
	9	4	128	U
	40	1	8000	
	1		+0	
	2		+767	
	3		+4	N
	4		+4104	
	5	2	32	
	6		+255	
	7	1	7168	
	8	4	322	C
	9	4	385	
	50		+0	C
	1		+257	
	2		+511	T

ADDRESS OF PREVIOUS INSTRUCTION	INSTRUCTION ADDRESS	F or /F	N	NOTES
	3		+1	
	4	4	256	
	5	7	8065	
	6	4	385	T
	7		+0	
	8	2	572	
	9	4	385	
	60	2	641	O
	1		+60	
	2	7	7936	
	3		+1026	
	4		+2048	N
	5	1	16	
	6	7	8000	
	7		+0	
	8		+0	
	9		+0	
	70		+0	
	1		+0	
	2		+257	
	3		+511	T
	4		+1	
	75	7	7936	
	6	4	2441	E
	7	4	393	
	8		+0	
	9	4	4684	
	80	4	4497	S

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	N	NOTES
	1	3	657	
	2		+0	
	3		+257	
	4		+511	T
	5		+1	
	6		+0	
	7		+0	
	8		+0	
	9		+0	
	90	1	7168	
	1	4	322	
	2	4	385	O
	3	1	7234	
	4		+0	
	5		+2303	
	6	1	532	K
	7	4	65	
	8			

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
7.3 Block 3 Miscellaneous				
9	(517)0		+0	Cycle Count
11	1		+0	Error character
13	2		+0	Workspace
	3	10	1,3	
	4	8	132,1	

Appendix 3: PROGRAM X5 (INTERRUPT TEST)

1. FUNCTION

This program tests that the 903 central processor responds correctly to manual interrupts on program levels 1, 2 and 3 and that any program is terminated using the correct SCR and B-register.

2. PROGRAM TYPE

X5 is distributed as a sum-checked binary tape suitable for input by initial instructions. The program uses locations 8-398 inclusive. If, after reading, continuous output occurs the program was not read correctly by the computer.

3. PROGRAM ACTION

Before running the program the interrupt mode switches should be set to the 'MANUAL' (down) position.

The trigger location of the program is 8. When the program is triggered it sets up the SCR's for levels 2, 3 and 4 whilst operating in level 1. Level 1 is then terminated leaving its SCR pointing to the first instruction of the level 1 "interrupt" program. Provided none of the INTERRUPT/LEVEL switches have been pressed the computer enters the level 4 program which is indicated by the visipunched output 4R (R means repeated). If no interrupts are demanded then 4R is punched out at 5 second intervals.

If an INTERRUPT/LEVEL switch is pressed causing a manual interrupt demand then the computer jumps into the new level program. The first action of the interrupting program is to store the accumulator of the interrupted program. A visipunched output "nS" (where n is the level and S means selected) is output showing that the new program has been entered and

if after 5 seconds, no higher level program interrupts then "nT" is visipunched. (T means terminate.) The computer picks up the accumulator of the interrupted program and returns to that program.

NOTE: More than one interrupt may be demanded simultaneously but the highest level program will be selected and as that program is terminated the next lower program which has demanded an interrupt is entered. This action proceeds until the level 4 program is entered.

Between the visipunched output "nS" and the output "nT" there is a 5 second pause to allow the operator to make an interrupt demand if he so wishes.

If, when a program level has been selected, an incorrect SC or B-register was used then X5 gives a visipunched output accordingly, i.e.:-

1AE, 2AE or 3AE (an address error at the level stated owing to the use of the wrong SCR) and/or

1BE, 2BE or 3BE (the wrong B-register was used at the level stated) after which the machine comes to a dynamic stop.

4. OPERATING INSTRUCTIONS

Load X5 in the tape reader, select address 8181 on the word generator and press the JUMP button. When the tape has been read in place the three interrupt mode switches in the MANUAL position, and proceed as outlined in Table 1.

Table 1 - SEQUENCE OF DEMANDS X5

Operators Action	Correct output response
Trigger program location 8	"4R"
Press level 3 within 5 seconds	"3S"
Press level 2 within 5 seconds	"2S"
Press level 1 within 5 seconds	"1S"
NIL	"1T"
NIL	"2T"
NIL	"3T"
NIL	"4R"
NIL	"4R"
The operator may now stop the program or continue with level selections at will.	

5. PROGRAM SHEETS

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
Absolute Addresses	8	4	155	Set L2 S. C. R. := +52
		5	2	
		4	156	Set L3 S. C. R. := +86
		5	4	
		4	157	Set L4 S. C. R. := +118
		5	6	
		14	0	do nothing
		14	0	
		4	173	Retrieve lower level acc.
	17	15	7168	Terminate L1
	18	5	173	Store lower L acc.
		4	177	
		11	188	Enter routine to output "1S"
		8	189	
340		14	0	
		11	145	check L1 S. C. R.
		4	145	
		2	0	
		6	170	
		1	176	
	28	7	31	
		4	160	L1 S. C. R. error
		8	146	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		0	145	check L1 B-register
		4	1	
		2	145	
		7	37	
		4	160	L1 B-register error
		8	152	
		14	47	
	38	10	171	wait 5 seconds
		4	171	
		9	37	
		4	172	reset wait count
		5	171	
		4	178	
		11	188	Enter routine to
		8	189	Output "1T"
		14	0	
		14	0	
	48	14	0	
		8	16	
		4	174	Retrieve lower level acc.
		15	7168	Terminate L2
		5	174	
		4	179	
		11	341	Enter routine to output "2S"
		8	342	
		14	0	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		11	145	Check L2 S.C.R.
	58	4	145	
		2	2	
		6	170	
		1	176	
		7	65	
		4	161	L2 S.C.R. Error
		8	146	
		0	145	Check L2 B-register
		4	3	
		2	145	
	68	7	71	
		4	161	L2 B-register Error
		8	152	
		14	47	wait 5 seconds
		10	171	
		4	171	
		9	71	
		4	172	reset wait count
		5	171	
		4	180	
	78	11	341	Enter routine to output "2T"
		8	342	
		14	0	
		14	0	
		14	0	
		8	50	
	84	4	175	Retrieve lower L acc.

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		15	7168	Terminate L3
		5	175	Store lower L acc.
		4	181	
	88	11	360	Enter routine to output "3S"
		8	361	
		14	0	
		11	145	Check L3 S.C.R.
		4	145	
		2	4	
		6	170	
		1	176	
		7	99	
		4	162	L3 S.C.R. Error
	98	8	146	
		0	145	Check L3 B-register
		4	5	
		2	145	
		7	105	
		4	162	L3 B-register Error
		8	152	
		14	47	Wait 5 seconds
		10	171	
		4	171	
	108	9	105	
		4	172	Reset wait count
		5	171	
		4	182	
		11	360	Enter routine to output "3T"
		8	361	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		14	0	
		14	0	
		14	0	
		8	84	
	118	4	182	
		11	379	Enter routine to output "4R"
		8	380	
		14	0	
		14	0	
		14	0	
		11	145	Check L4 S.C.R.
		4	145	
		2	6	
		6	170	
	128	1	176	
		7	132	
		4	163	L4 S.C.R. Error
		8	146	
		0	145	Check L4 B-register
		4	7	
		2	145	
		7	138	
		4	163	L4 B-register Error
		8	152	
	138	14	47	Wait 5 seconds
		10	171	
		4	171	
		9	138	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	N	NOTES
		4	172	Reset wait count
		5	171	
		8	118	
		+	0	W.S.
		11	188	Output level number for
		8	189	Error type A.
	148	4	184	
		11	188	Enter routine to punch
		8	189	error output "AE" or "BE"
		8	151	Dynamic Stop on Error
		11	188	Output level number for
		8	189	Error type B.
		8	186	
		+	52	
		+	86	
		+	118	
	158	+	18	
		+	144	
		+	303	
		+	313	Error 'N'
		+	323	Addresses
		+	333	
	164	+	0	
		+	0	
		+	114	
		+	65	
	168	+	85	
		+	66	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		+	8191	
		-	16384	
		-	16384	
	173	+	0	
		+	0	
		+	0	
		-	2	
	177	+	213	
	178	+	223	
		+	233	
		+	243	
		+	253	"read" addresses
		+	263	
		+	273	
		+	283	
		+	293	
		4	185	Part of error O/P
		8	149	
	188	+	0	Level 1 S. C. R. link
		1	202	Set read inst.
		5	194	
		4	201	
		5	164	
		0	164	
		+	0	For read inst.
	192	15	6144	Output
		10	164	"IS", "1T"
		4	164	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	N	NOTES
	198	9	192	
		0	188	
		8	334	
		-	10	
		/4	0	
		+	0	"1"
		+	65	
		+	255	
		+	1	
		+	0	
	208	+	0	"S"
		+	113	
		+	137	
		+	137	
		+	70	
		+	0	"1"
		+	65	
		+	255	
		+	1	
		+	0	
	218	+	128	"T"
		+	128	
		+	255	
		+	128	
		+	128	
		+	64	"T"
		+	135	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		+	137	
		+	113	"2"
		+	0	
	228	+	0	
		+	113	
		+	137	"S"
		+	137	
		+	70	
		+	64	
		+	135	
		+	137	"2"
		+	113	
		+	0	
	238	+	128	
		+	128	
		+	255	"T"
		+	128	
		+	128	
		+	128	
		+	145	
		+	177	"3"
		+	202	
		+	132	
	248	+	0	
		+	113	
		+	137	"S"
		+	137	
		+	70	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	N	NOTES
		+	128	
		+	145	
		+	177	"3"
		+	202	
		+	132	
	258	+	128	
		+	128	
		+	255	"T"
		+	128	
		+	128	
		+	24	
		+	40	
		+	72	"4"
		+	255	
		+	0	
	268	+	255	
		+	136	
		+	148	"R"
		+	98	
		+	1	
		+	127	
		+	136	
		+	136	"A"
		+	136	
		+	127	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
	278	+	0	
		+	255	
		+	145	"E"
		+	129	
		+	129	
		+	255	
		+	137	
		+	137	"B"
		+	119	
		+	0	
	288	+	0	
		+	255	
		+	145	"E"
		+	129	
		+	129	
		+	0	
		+	0	
		+	0	
		+	0	
		+	0	
	298	+	0	
		+	65	
		+	255	"T"
		+	1	
		+	0	
		+	0	
		+	0	
		+	0	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		+	0	
		+	0	
	308	+	64	
		+	135	
		+	137	"2"
		+	113	
		+	0	
		+	0	
		+	0	
		+	0	
		+	0	
		+	0	
	318	+	178	
		+	145	
		+	177	"3"
		+	202	
		+	132	
		+	0	
		+	0	
		+	0	
		+	0	
		+	0	
	328	+	16	
		+	48	
		+	80	"4"
		+	255	
		+	0	
	333	+	0	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	N	NOTES
200		4	333	
		15	6144	
		15	6144	Output 5 spaces shared
		15	6144	by "1S", "1T" and
	338	15	6144	Error Prints
		15	6144	
		/8	1	
		+	0	Link
		1	202	
		5	347	
		4	201	
		5	359	
		0	359	
		+	0	
	348	15	6144	Punch legible characters
		10	359	"2S", "2T"
		4	359	
		9	345	
		15	6144	
		15	6144	
		15	6144	Output 5 blanks
		15	6144	
		15	6144	
		0	341	Exit
	358	/8	1	
		+	0	
		+	0	For link
		1	202	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		5	366	
		4	201	
		5	378	
		0	378	
		+	0	Read set here
		15	6144	Output legible characters
	368	10	378	"3S", "3T"
		4	378	
		9	364	
		15	6144	Output 5 blanks
		15	6144	
		15	6144	
		15	6144	
		15	6144	
		0	360	
		/8	1	
	378	+	0	
		8	120	Link
		1	202	
		5	385	
		4	201	
		5	398	
		0	398	
		+	0	
		15	6144	Punch legible character
		10	398	"4R"
	388	4	398	
		9	383	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F	INSTRUCTION N	NOTES
		15	6144	
		15	6144	
		15	6144	6 blanks
		15	6144	
		15	6144	
		15	6144	
		0	379	
	397	/8	1	

Appendix 4: PROGRAM X8

1. PURPOSE

This program tests that the processor will 'trace' a level 4 program on program level 1 when the interrupt mode switch for level 1 is in the TRACE position.

2. PROGRAM TYPE

X8 is distributed as a sum-checked binary tape suitable for input by initial instructions. It occupies addresses

8- 75 inclusive
768- 781 inclusive
1024-3125 inclusive

If continuous punch-out occurs after the program has been read in, then an error has occurred during reading.

3. PROGRAM ACTION

The program is entered at location 8 on level 1. The section of program in locations 8-43 inclusive writes a series of blocks of instructions into locations 1024-3123 inclusive which are to act as the level 4 program. The SCR for level 4 is then set to 1024 so that on termination of level 1 the first of these instructions will be obeyed. While the interrupt mode switch is in the ON-LINE position the processor exits level 1 at the end of the program in that level and proceeds to obey the instructions in the level 4 program.

At the end of the level 4 program a chararacter is input by the reader to show that the level 4 program is finished after which the computer jumps back to the start of the level 4 program. This cycle is repeated approximately 20 times a second.

If the interrupt mode switch for level 1 is put in the TRACE position the reading speed should slow to approximately 1 character per second, if the level 4 program is being traced.

The TRACE position of the interrupt mode switch causes a permanent interrupt demand. Therefore, after each instruction is obeyed in level 4 program, level 1 becomes effective (except after the 'function 0' instructions). When the level 1 program first interrupts the processor jumps to location 47 which is the first instruction of the group which stores the current level 4 SCR. Level 1 is then terminated so that the next level 4 instruction may be obeyed. The second level 1 interrupt causes entry at location 51 where a routine is entered for checking that the level 4 SCR is being incremented by 1 with each traced instruction. When an error is found, a check is made to see whether location 3125 has been reached, and that no interrupt has occurred after a "0" instruction.

If it was the end of the level 4 program or the interrupt had not occurred after a "0" instruction a jump in level 1 is made to location 47 so as to set location 780 to the current level 4 SCR value. Otherwise, an error print-out is given and the test repeated.

4. OPERATING INSTRUCTIONS

Load the tape in the reader set up address 8181 and press the JUMP button.

When the program section has been read in the blank tape at the end of it will be read at approximately 20 c.p.s. If now, the interrupt mode switch is put in the TRACE position the blank tape will be read at approximately 1 c.p.s. If there is no punch out during this time then the test is successful.

5. ERROR INDICATIONS

If an error occurs a character will be punched which is the binary value of the last function obeyed of the traced (level 4) program.

6. PROGRAM SHEETS

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
	0			Level 1 SCR
	1			Level 1 B-register
	2			not used
	3			
	4			
	5			
	6			Level 4 SCR
	7			Level 4 B-register
	8	0	768	Write Function 0 in each location from 1024-1523
	9	4	769	
	10	/5	1524	
	11	10	1	
	12	4	1	
	13	9	9	
	14	0	768	Write Function 1 in each location from 1524-2023
	15	4	770	
	16	/5	2024	
	17	10	1	
	18	4	1	
	19	9	15	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
	20	0	768	Write Function 2 in each location from 2024-2523
	21	4	771	
	22	/5	2524	
	23	10	1	
	24	4	1	
	25	9	21	
	26	0	781	Write Function 12 in each location from 2524 to 2573
	27	4	772	
	28	/5	2574	
	29	10	1	
	30	4	1	
	31	9	27	
	32	0	781	Write Function 13 in each location from 2574-2623
	33	4	773	
	34	/5	2624	
	35	10	1	
	36	4	1	
	37	9	33	
	38	0	768	Write Function 14 in each location from 2624-3123
	39	4	774	
	40	/5	3124	
	41	10	1	
	42	4	1	
	43	9	39	
	44	4	775	Set start address for level 4 program
	45	5	6	
	46	15	7168	Terminate and start level 4 program

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
1st TRACE ENTRY	47	4	6	Level 4 S.C.R. (less F bits) to 780
	48	6	777	
	49	5	780	
	50	15	6168	
SUCCESSIVE TRACE ENTRY (S)	51	10	780	
	52	4	6	Check that level 4 S.C.R. and 780 remain in step
	53	6	777	
	54	2	780	
	55	7	50	
	56	4	780	If not check for end of Level 4 program (reset 780 to first address of level 4 program
	57	2	776	
	58	7	47	
	59	4	780	
	60	2	778	Check if in block of 0 instructions (no interrupt possible following fn 0).
	61	9	64	
	62	7	64	
	63	8	68	
	64	4	780	
	65	2	779	
	66	9	68	
	67	8	47	
	68	4	6	Obtain F-bits of last instruction of level 4 program that was obeyed.
	69	1	75	
	70	5	1	
	71	/4	0	
	72	14	8179	

ADDRESS OF PREVIOUS INSTRUCTION	ADDRESS	INSTRUCTION F or /F	INSTRUCTION N	NOTES
	73	15	6144	Error O/P
	74	8	44	Repeat test
	75	-1		
	768		-500	Constants
	769	0	0	
	770	1	0	
	771	21	0	
	772	12	0	
	773	13	0	
	774	14	0	
	775		+1024	
	776		+3126	
	777		+65535	
	778		+1024	
	779		+1524	
	780		-1	(Count)
	781		-50	
	3124	15	2048	Input a character from paper tape
	3125	8	1024	Repeat level 4 program.

Appendix 5: PROGRAM X.6 (INSTRUCTION TIMING)

1. FUNCTION

To enable an operator, using a stop watch, to time any of the instructions of the 900 instruction code. It is also possible to include variations where the timing of an instruction depends on the accumulator contents or the particular address within the instruction word.

2. STORE USED

Locations	8 - 69	inclusive	(Set up program)
Locations	999 - 3499	"	(W/S + 2500 instructions)
Locations	3500 - 3514	"	(print count Program)

3. TAPES

Alternative tapes using input modes 1 and 3 respectively are available; the latter is titled 903 x 6.

The tapes are punched complete with directories for input by 903 T2.

The tapes include standard parameter tapes, punched complete with directories for input by 903 T2, and standard address tapes.

4. ENTRY POINT AND METHOD OF USE

Read in the library tape by 903 T2 (trigger 7749 = 1111001000101) then select the required parameter tape and again read in by 903 T2. If instructions 14 or 15 have been selected then the Address tape must be in the Tape Reader when triggering the program, trigger is to location 8.

The program will then write 2500 identical instructions in locations 1000 to 3499. These instructions will be obeyed consecutively and repeated, every 20th cycle the program will output "1101100" on paper tape.

The repetition rate of this output can now be judged and 20 such outputs timed with a stop watch. The measured time in seconds will then be the time in microseconds for one of the selected instructions. Note that input/output instructions can only be timed when a test set is connected to the computer.

The maximum error will be in the order of .08 microseconds, due to the control instructions in the timing loop.

A further parameter tape may now be read in by 903 T2 (which is not overwritten) and the program re-triggered as before. When testing functions 14 and 15 a succession of address tapes may be used without re-reading the parameter tape.

5. STANDARD PARAMETER AND ADDRESS TAPES

These are punched in the following order on the library tapes:-

Parameter tape for function				0	
"	"	"	"	1	
"	"	"	"	2	
"	"	"	"	3	
"	"	"	"	4	
"	"	"	"	5	
"	"	"	"	6	
"	"	"	"	7,	accumulator
"	"	"	"	7,	"
"	"	"	"	7,	"
"	"	"	"	8	
"	"	"	"	9	accumulator
"	"	"	"	9	"
"	"	"	"	10	
"	"	"	"	11	
"	"	"	"	12	
"	"	"	"	13	
"	"	"	"	14	

Address tape for address 2 (left shift - 2 places)
" " " " 8190 (right shift - 2 places)
" " " " 2048 (block input - 2 words)
" " " " 4096 (block output - 2 words)

Parameter tape for function 15
Address tape, address 0 (input)
" " " 4096 (output)

Parameter tape, function 15 }
Address tape, address 7168 } (program terminate)

Parameter tape, function 4 modified

6. PREPARATION OF TAPES

The parameter tapes are provided but they can be punched as follows:-

```
&
+52
*
a
b
c
)
```

If a = +0, the program will expect to read an address tape. This is only used for instructions 14 or 15.

If a = +1, the program will write the address of the next location in all those locations containing the jump instruction being timed. This is only used for instructions 7, 8 and 9.

If a = -1, the program will write address 999 into all those locations containing the instructions being timed. This is used for all the remaining instructions.

Where b will denote the accumulator contents during the test cycle. This enables the timing of instruction 7 and 9 to be varied according to the state of the Accumulator.

Where c will denote the selected instruction the address bits always being zero. The modifier digit may be included if required.

The Address tape for use with instructions 14 and 15 must be punched as follows:-

" " run out" " x x x x" " run out" "

where x x x x denotes the digits of the address.

NOTES

There is no error detection during input of the address tape so no other characters may be punched (i.e. no signs, no spaces, nor any erases).

When testing the block transfer instructions, the number of words transferred per instruction is normally 2. If desired this can be altered by means of an addition to the parameter tape which is punched as follows:

```
&
+52
+3514
*
+0
+3516
14  0
*
+d
)
```

where d is the number of words to be transferred. Note that b is set to +3516 so that words are transferred into locations 3516 onwards.

When testing the program terminate instruction (15 7168) the level 4 S. C. R. location (6) must be loaded with +1001 so that the change from level 1 to level 4 does not affect the sequence. This can be done by an addition to the parameter tape as follows:-

```
&
+6
+52
*
+1001
*
+ 0
etc.
```

It is not possible to test a modified program terminate instruction.

APPENDIX 6: 903 INITIAL INSTRUCTIONS TEST XINIT

1. Purpose

To test that the Initial Instructions in store locations 8180 - 8191 are protected until a level terminate instruction is obeyed, and that after such an instruction the Initial Instructions may be overwritten.

Configuration

This program is for use only on machines with more than 8K of store.

2. Operating Instructions

2.1 The program tape is input under Initial Instructions, (8181 set up on word generator and "JUMP" depressed).

2.2 Phase 1

Enter the program at entry point 21. The program will attempt to overwrite the Initial Instructions, which should be protected. If the Initial Instructions are protected, the message

I.I. PROTECTED

will be output on the control teleprinter (see Section 4).

If the program succeeds in overwriting the Initial Instructions, it will output an error message (see 3.1).

2.3 Phase 2

The program will obey a level terminate instruction and will then attempt to overwrite the Initial Instruction locations, which should not now be protected. If the attempt succeeds, the program will output the message

I.I. OVERWRITTEN

on the control teleprinter (see Section 4).

If the attempt fails, the program will output an error message (see Section 3.2).

2.4 Phase 3

The program will overwrite the Initial Instructions with a block of machine code and will test that the machine code block is obeyed correctly. If the program is obeyed correctly the program will output the message

PROGRAM OBEYED OVER I.I.

on the control teleprinter (see Section 4). If the program is not obeyed correctly an error will be output (see Section 3.3).

2.5 Phase 4

If phase 3 is completed successfully the message

ENTER PHASE 4

will be output on the control teleprinter. The operator will then re-enter the program at location 22 by depressing RESET and JUMP. A level terminate will then be obeyed and locations 8180-8191 will be examined. These should contain the instructions stored during phase 3. If the instructions are then obeyed correctly the message

ENTER PHASE 5

will be output on the control teleprinter (see Section 4). If an error occurs a message will be output (see Section 3.4).

2.6 Phase 5

If phase 4 is completed successfully the operator will press RESET and JUMP. The program will then check locations 8180-8191 which should now contain Initial Instructions.

If the instructions are correct the message

END XINIT

will be output on the control teleprinter (see Section 4). The program will then enter a dynamic stop.

If an error occurs a message will be output (see Section 3.5).

3. Method

3.1 Phase 1

The program attempts to write a pattern of alternate ones and zeros in each of the Initial Instruction locations. The pattern 1010101010101010 will be written into the first location, and its complement in the next location. This will be repeated throughout the area of Initial Instructions. The Initial Instructions will then be read out and checked. If any Instruction is overwritten the program will output the message

ERR1 LOCN <LOCATION IN DECIMAL>
EXPECTED <WORD IN BINARY>
READ <WORD IN BINARY>

If the Initial Instructions have not been overwritten, the program will output the message

I.I. PROTECTED

which indicates the phase has been successful.

3.2 Phase 2

The program will obey a level terminate instruction, and then attempt to overwrite locations 8180 8191, as in Phase 1.

If these locations are overwritten successfully the message

I.I. OVERWRITTEN

will be output on the control teleprinter (see Section 4).

If the locations are not overwritten correctly the message

ERR2 LOCN <LOCATION IN DECIMAL>
EXPECTED <WORD IN BINARY>
READ <WORD IN BINARY>

will be output on the control teleprinter (see Section 4).

3.3 Phase 3

The program writes into each location the Instruction 10 TESTCOUNT and into location 8192 an instruction to jump back into the main program. The contents of the locations 8180 8191 are then checked, to ascertain that they contain the instruction 10 TESTCOUNT. The program then clears TESTCOUNT location and jumps to 8180. When all instructions in locations 8180 8191 have been obeyed, control is returned to the main program from location 8192, and TESTCOUNT is examined.

If TESTCOUNT holds the value 12 then the message

PROGRAM OBEYED OVER I.I.

will be output on the control teleprinter (see Section 4). The program then stops.

If TESTCOUNT does not hold the value 12 the program will output the message

ERR 3 TESTCOUNT = <n>

where n is the decimal value held in TESTCOUNT.

If locations 8180-8191 do not contain the correct instructions the message

ERR 3 LOCN <LOCATION NUMBER IN DECIMAL>
EXPECTED <10 TESTCOUNT, IN BINARY>
READ <WORD IN BINARY>

3.4 Phase 4

The message

ENTER PHASE 4

will be output on the control teleprinter (see Section 4). The program will then enter a dynamic stop. The operator will then set up the entry point 22 on the word generator and depress Jump. The program will obey a level terminate and then check that locations 8180 8191 contain the instructions written in during Phase 3.

If the Instructions are not correct the message

ERR 4 LOCN <LOCATION IN DECIMAL>
EXPECTED <WORD IN BINARY>
READ <WORD IN BINARY>

will be output on the control teleprinter (see Section 4).

3.5 Phase 5

If phase 4 is completed successfully the message

ENTER PHASE 5

will be output on the control teleprinter. The operator will then re-enter at 23. Locations 8180 8191 will be examined, and these should contain Initial Instructions.

If these instructions are not correct the message

ERR 5 LOCN <LOCATION IN DECIMAL>
EXPECTED <WORD IN BINARY>
READ <WORD IN BINARY>

will be displayed on the control teleprinter (see Section 4).

If the instructions are correct the message

END XINIT

will be output, and the program will enter a dynamic stop.

4. Further Information

If a control teleprinter is not fitted all messages will be directed to the paper tape punch.