The Complete Machine Code Tutor

by Malcolm Evans

48K Spectrum

LOADING INSTRUCTIONS

THIS PROGRAM IS ARRANGED ON FOUR SIDES OF TAPE.

TO LOAD THE PROGRAM TYPE LOAD " ", USING SIDE 1. This will load in the screen and program/simulator. When instructed stop the tape. Do NOI press break During the loading of the simulator.

YOU WILL THEN BE ASKED WHETHER YOU WISH TO LOAD LESSONS.

IF NO (N) YOU WILL ENTER STRAIGHT INTO THE EDITOR PART OF THE SIMULATOR. THIS IS NOT RECOMMENDED FOR BEGINNERS. YOU MAY ESCAPE BY PRESSING BREAK.

IF YES (Y) YOU WILL BE REQUESTED TO LOAD THE REQUIRED GROUP OF LESSONS. CHOOSE THE DESIRED SECTION AND START THE TAPE RECORDER. THE GROUP YOU MAVE CHOSEN WILL APPEAR ON THE SCREEN. IF YOU HAVE SELECTED THE WRONG GROUP PRESSING BREAK WILL STOP THE LOAD. AND ENABLE YOU TO LOAD AMOTHER SECTION.

THE PROGRAM AND LESSONS ARE ARRANGED ON THE CASSETTE TAPES AS FOLLOWS:

SIDE	1	•	PROGRAM/SIMULATOR	:	LESSONS	1	-	9
SIDE	2	-	LESSONS 10 - 17	:	LESSOWS	18	-	25
SIDE	3	•	PROGRAM/SIMULATOR	:	LESSONS	18	-	25
SIDE	4	-	LESSONS 26 - 35	:	Lessons	1	-	9

ASSEMBLER INSTRUCTIONS

THE ASSEMBLER ACCEPTS ALL Z80 INSTRUCTIONS AS INCLUDED IN THE LESSON SUMMARIES. IN ADDITION IT MILL ACCEPT THE FOLLOWING INSTRUCTIONS ITSELF:

- DEFB N : DEFINE BYTE. THIS ALLOCATES THE NEXT MEMORY LOCATION FOR STORAGE, (THIS MAY BE LABELLED). AND INITIALLY LOADS IT WITH N. THE SIMULATOR WILL CONTINUALLY DISPLAY THE CONTENTS OF THE LOCATION IN DECIMAL OR HEX.
- DEFW NN : DEFINE WORD, THIS ALLOCATES THE NEXT TWO MENORY LOCATIONS FOR STORAGE, (MAY BE LABELLED), AND INITIALLY LOADS IT WITH THE THO BYTE NUMBER NN. IME SIMULATOR WILL CONTINUALLY DISPLAY THE CONTENTS OF THESE LOCATIONS AS A SINGLE DECIMAL OR MEX MUMBER.
- BIN N : THIS IS THE SAME AS DEFB. EXCEPT THAT THE SIMULATOR DISPLAYS THE MEMORY CONTENTS IN BINARY FORM. N IS A DECIMAL OR HEX NUMBER.

THE ASSEMBLER WILL ALSO ACCEPT LABELS. A LABEL IS DEFINED IN RELATIONSHIP TO NUMBERS AS FOLLOWS:

- DECIMAL NUMBER : ANY STRING CONTAINING ONLY 0-9
- HEXADECIMAL NUMBER : ANY STRING CONTAINING ONLY O-9 or A-F. IT may be DISTINGUISHED FROM DECIMAL BY FOLLOWING IT WITH AN H.
- LABEL : LABELS A MEMORY LOCATION. ANY STRING NOT COVERED BY DECIMAL OR MEXADECIMAL NUMBERS ABOVE, REGISTER NAMES OR CONDITIONAL LETTERS I.E. C. NC. Z, NZ ETC, THE MAXINUM LENGTH IS & CHARACTERS.

ERROR MESSAGES - ASSEMBLER

- INSTRUCTION UNKNOWN FIRST PART OF MNEMONIC NOT RECOGNISED
- SPACE MISSING SPACE MISSING AFTER FIRST PART OF MNEMONIC I.E. LDA.5
- MISSING SPACE OR , SPACE OR , MISSING AFTER SECOND PART OF MNEMONIC I.E. LD A 5 OR SUB AM
- ERROR AFTER INST. ERROR AFTER FIRST PART OF MMEMONIC WHICH THE ASSEMBLER WAS UNABLE TO RECOGNISE AS ANY OTHER DEFINED ERROR
- NUMBER MISSING PART OF MNEMONIC MISSING COULD BE A HUMBER
- BRACKET MISSING A BRACKET MISSING
- NUMBER TOO LARGE 16 BIT NUMBER GREATER THAN 65555
- LABEL TOO LONG LABELS CAN ONLY HAVE UPTO 6 CHARACTERS
- LABEL NOT FOUND LABEL APPEARS IN INSTRUCTION BUT NOT DECLARED IN LABEL COLUMN
- No DEFN or DEFN LABEL WITH NO INSTRUCTION
- CAN ONLY ADD IX/IY SBC IX/IY OR ADC IX/IY FOUND
- OFFSET TOO BIG OFFSET IN (1X+D) OR (1Y+D) INSTRUCTION GREATER THAN 255
- OFFSET MISSING NO OFFSET IN (IX+D) OR (IY+D) INSTRUCTION
- ERROR MESSAGE SIMULATOR
- PROGRAM COUNTER HAS JUMPED TO NON VALID ADDRESS
 - Your program has caused the PC to jump to an address which is not the beginning of one of your instruction lines
- YOU ARE TRYING TO RUN CODE IN ALLOCATED STORAGE AREA
 - Your program has caused the PC to jump to an address which has been allocated as storage

YOU ARE ABOUT TO AFFECT MEMORY AREA NOT ALLOCATED TO YOU

-

YOU ARE ABOUT TO LOAD A MEMORY LOCATION NOT ALLOCATED TO YOU LOUR ALLOCATED AREA IS 18584 (4000H) TO (2505) (SAFFH) AND 32000 (7000H) TO 32192 (7000H)

YOU ARE ABOUT TO WRITE TO MEMORY WHICH WILL AFFECT YOUR PROGRAM

- YOU ARE ABOUT TO LOAD A MEMORY LOCATION WITHIN YOUR PROGRAM INSTRUCTIONS.
- THE STACK POINTER IS OUTSIDE ALLOCATED MEMORY AREA
 - THE STACK POINTER HAS MOVED OUTSIDE THE ABEA 32193 (7DCIH) TO 32255 (7DFFH)
- THERE ARE TOO MANY REGISTERS CALLED UP
 - YOUR PROGRAM USES MORE REGISTERS THAN THE SIMULATOR CAN DISPLAY. INE PROGRAM CAN BE RUN BUT OMLY THOSE REGISTERS DISPLATED CAN BE SHOWN

THE LESSONS IN THIS PROGRAM ARE ARRANGED IN FOUR GROUPS. UPON LOADING OF EACH SECTION AN INTRODUCTION APPEARS ON THE SCREEN. THE MENU FOR THE GROUP OF LESSONS CAN THEN BE OBTAINED BY PRESSING SPACE. AT ANY TIME WHILST IN A LESSON OR EXAMPLE PRESSING BREAK (CAPS SHIFT + SPACE) WILL RETURN YOU TO THE MENU.

THE POLLOWING IS A COMPLETE LIST OF ALL KEYS USED AT VARIOUS STAGES WITH A FULL DESCRIPTION OF THEIR FUNCTION.

HENU

- ENTER : WILL ENTER ANY LESSON OR EXAMPLE HIGHLIGHTED ON THE RENU.
- SPACE : PRESSING THIS KEY ALLOWS YOU TO CHOOSE WHICH ITEM TO ENTER
- LESSON AND EXAMPLE TEXT
- SPACE : PRESSING THIS KEY WILL DISPLAY THE MEXT PAGE. AT THE END OF EACH LESSON IT MILL EFFECT A RETURN TO MENU.
- BREAK : WILL RETURN YOU TO THE MENU AT ANY TIME.
- SIMULATOR : ALL KEYS AUTO-RUN
- RUN : THIS CLEARS ALL REGISTERS AND STARTS PROGRAM RUNNING.
- ANY KEY : IF IN RUNNING STATE WILL PERFORM THE HIGHLIGHTED INSTRUCTION.

STOP : THIS STOPS THE RUNNING OF THE PROGRAM (A+SYMBOL SHIFT)

EDIT : PRESSING THIS KEY ENTERS THE EDITOR, (1+Symbol allowing you to modify or re-write the shift) program.

G : THIS SWOPS THE DISPLAY BETWEEN DECIMAL AND HEXADECIMAL NOTATION. THIS KEY IS OPERATIVE ONLY WHEN THE PROGRAM IS NOT RUNNIMG.

BREAK : WILL RETURN YOU TO THE MENU.

ONCE COMPLETED IF YOU WISH TO REPEAT THE EXERCISE THEN PRESS RUN.

EDITOR	:	ALL KEYS AUTO-RUN
CURSOR KEYS CAPS'SHIFT)	:	PRESSING THESE WILL ALLOW YOU TO MOVE THE CURSOR IN THE DIRECTION OF THE ARROWS,
E (6 + Symbol Shift)	:	THIS CLEARS THE EDITOR SCREEN AND MEMORY AND RETURNS THE CURSOR TO THE START OF THE SCREEN
SPACE	:	TABS TO BEGINNING OF INSTRUCTION WHEN IN LABEL COLUMN. ELSEWHERE A SPACE WILL BE CREATED.
ENTER	:	THIS HOVES THE CURSOR TO THE START OF THE NEXT LINE.
DELETE (0 + CAPS SHIFT)	:	THIS DELETES THE CHARACTER TO THE Left of the cursor and shifts the Cursor one space to the left.
STOP	:	ÎNITIATES ASSEMBLY OF THE PROGRAM ON SCREEN. ÎF ASSEMBLED CORRECTLY THE SIMULATOR MODE IS ENTERED. ÎF AN ERROR IS FOUND THEN AN ERROR STATEMENT IS DISPLAVED AGAINST THE APPROPRIATE LINE OF THE PROGRAM AND THE EDITOR WAITS FOR CORRECTION.
BREAK	:	WILL RETURN YOU TO THE MENU.

THE EDITOR ALWAYS DISPLAYS LETTERS IN CAPITALS. However CAPS SHIFT IS NOT REQUIRED.

INTRODUCTION TO LESSONS

THE COMPLETE MACHINE CODE TUTOR CONTAINS 35 LESSONS COVERING ALL THE INSTRUCTIONS ON THE Z8D PROCESSOR, WHICH IS THE PROCESSOR IN YOUR SPECTRUM COMPUTER.

ALL THE LESSONS ARE DEALT WITH IN GREAT DETAIL ON SCREEN, AND IN MANY CASES ARE FOLLOWED BY EXAMPLE PROGRAMS, WHICH YOU CAN USE AS EXERCISES BY MODIFYING THEM YOURSELF, THERE IS NO DANGER THAT YOU MIGHT CRASH THE SYSTEN.

THERE NOW FOLLOWS A LIST OF ALL THE LESSON HEADINGS. TOGETHER WITH AN INDICATION OF THOSE LESSONS THAT ARE FOLLOWED BY EXAMPLES. UNDER EACH HEADING THERE IS A SUMMARY OF INSTRUCTIONS WHICH MILL BECOME CLEAR TO YOU AS YOU PROGRESS THROUGH THE IUTOR. IMESE SUMMARIES ARE INTENDED AS A PERRAMENT RECORD OF INSTRUCTIONS, TO WHICH YOU CAN EASILY REFER FOR REVISION PURPOSES. WITHOUT HAVING TO REFER BACK TO THE SCREEN TEXT.

LESSON 1 -	REGISTERS AND MENORY
	INITIALLY WE ONLY CONSIDER A.B.C.D. E.H. AND L REGISTERS.
LESSON 2 -	SIMPLE LOAD INSTRUCTIONS
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
LD ala'	WHERE B AND R' ARE ANY OF THE FOLLOWING: A.B.C.D.E.H AND L.
LD R.N	WHERE N IS A NUMBER 0 - 255
LD A.(NN)	WHERE NH IS A MEMORY LOCATION 0 - 65535
LD (NN).A	EXAMPLES FOLLOW THIS LESSON
LESSON 3 -	· REGISTER PAIRS
	A NUMBER IN A REGISTER PAIR IS 256 X The High Byte + The Low Byte.
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
LD DD.NN	WHERE DD IS ANY REGISTER PAIR
	BC, DE, AND HL. HN IS A NUMBER O - 65535
LD DD.(NN) LD (NN),DD	WHERE NU IS ADDRESS OF A MEMORY LOCATION U - 55535.
EX DE.HL	EXCHANGES REGISTER CONTENTS.
	Examples follow this lesson

LESSON 4	-	INDIRECT ADDRESSING THE FOLLOWING INSTRUCTIONS ARE COVERED:
LD R,(HL)		WHERE R IS ANY SINGLE REGISTER A.B.C.D.E.H. OR L.
LD (HL),R		
LD A.(BC)		
LD A.(DE)		
LD (BC).A		
LD (DE),A		EXAMPLES FOLLOW THIS LESSON
LESSON 5	-	ADDITIONS AND THE CARRY FLAG
		ADDITIONS WITH ACCUMULATOR AND HL REGISTER PAIR ARE DISCUSSED AS WELL AS ADD WITH CARRY.
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
ADD A.M ADD A.M ADD A.(HL)		WHERE N IS A NUMBER 0 - 255 WHERE R IS ANY SINGLE REGISTER
ABB HE:BE		
ADC A.H ADC A.R ADC A.(HL)		ADD WITH CARRY
ABE HE:BE		2 EXAMPLES FOLLOW THIS LESSON
LESSON 6	•	SUBTRACTION AND THE CARRY FLAG
		SUBTRACTION WITH AND WITHOUT CARRY ON THE ACCUMULATOR AND HL REGISTER PAIR ARE DISCUSSED.
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
SUB N SUB P SUB (HL)	}	SUBTRACT FROM A. N. R. OR (HL)
SBC A.N.	}	SUBTRACT FROM A WITH CARRY
發 [] []	Ş	SUBTRACT FROM HL WITH CARRY
ξĘ.	•	SET CARRY FLAG COMPLIMENT CARRY FLAG
		2 EXAMPLES FOLLOW THIS LESSON

LESSON 7	•	INCREMENT AND DECREMENT INSTRUCTIONS THE FOLLOWING INSTRUCTIONS ARE COVERED:
NC PHL)		
EC (HL) EC DD		EXAMPLES FOLLOW THIS LESSON
LESSON 8	-	THE ZERO FLAG
		NO NEW INSTRUCTIONS ARE COVERED IN THIS LESSON, WHICH IS INCLUDED TO SHOW YOU THE EFFECT ON THE ZERO FLAG OF ALL THE INSTRUCTIONS COMSIDERED IN PREVIOUS LESSONS.
		À TABLE OF THE EFFECTS OF ALL INSTRUCTIONS ON ALL FLAGS IS GIVEN IN APPENDIX (A)
		Examples Follow THIS LESSON
LESSON 9	-	COMPARE
		IF N IS THE NUMBER WITH WHICH A IS COMPARED, THEN THE FOLLOWING RESULTS:-
		CARRY ZERO
		A GREATER THAN N U U
		THE FOLLOWING INSTRUCTIONS ARE COVERED.
(P N (HL)		COMPARES A WITH N (0-255) Compares A with register R Compares A with register R Compares A with nemory location (HL)
		EXAMPLES FOLLOW THIS LESSON
LESSON 10	-	CONDITIONAL & UNCONDITIONAL JUNPS
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
≸71⊾)		
-B MC. MM		JUMP IF CARRY FLAG NOT SET
P MZ.m		JUNP IF CARRY PLAG SET JUNP IF ZERO FLAG NOT SET
JP Z.m		JUNP IF ZERO FLAG SET
		EXAMPLES FOLLOW THIS LESSON

LESSON 11 -	RELATIVE JUMPS
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
JRE	WHERE E IS THE DISPLACEMENT IN THE RANGE 127 TO -128
JR NC,E JR C,E JR NZ,E JR Z,E	
DJNZ E	DECREMENT AND JUMP ON NON ZERO
	2 EXAMPLES FOLLOW THIS LESSON
LESSON 12 -	The Stack
	THE STACK AND THE STACK POINTER ARE INTRODUCED.
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
PUSH DD	WHERE DD IS AF, BC, DE, OR HL. FROM NOW ON DD CAN BE CONSIDERED AS BC, DE, HL, OR SP.
POP DD	
LD SP. NN LD SP. (NN) LD SP. (NN), SP LD SP. HL	
ADC HL .SP SBC HL .SP	
INC SP DEC SP	
EX (SP).HL	F
	EXAMPLES FOLLOW THIS LESSON
LESSON 13 -	CALLS TO SUBROUTINES
CA11	RET UNCONDITIONAL
CALL NC. NN	RET NC NO CARRY
CALL MZ. NN	RET NZ NOT ZERO
CALL Z.NN	REI Z ZERO SET
LESSON 14 -	2 Examples follow this lesson Binary Notation
	THIS LESSON IS ABOUT A MHOLE NEW NUMBER
	BASE - BINARY. THIS IS A SYSTEM OF USING ONLY TWO DIFFERENT NUMBERS, ONE
	AND ZERO, IN EACH DIGIT COLUMN. A
	BINARY DIGIT. THIS IS BECAUSE IT CAN
	ONLY HOLD A VALUE OF ONE OR ZERO. AFTER
	FIND THIS REFERENCE CHART USEFUL:

VALUES	OF	Віт	7 то 3	11 0:							
BIT Nu	MBER	1:	7	6	5	9	3	2	1	0	
VALUES	:	:	128	64	32	16	8	4	2	1	
			Examp	LES P	OLLON	THIS	LES	SON			
LESSON	15	-	HEXAD	ECIMA	L NOT	ATION					
			HEXAD GREAT FOLLO INVAL	ECIMA LENG WING UABLE	L NOT	ATION THIS WILL RENCE	IS LES PRO	DISC SON, IVE A	USSE BUT	D AT	
			Decim	AL	Bina	RY	Hex	ADEC	IMAL		
			0								
			EXAMPL	LES F	OLLOW	THIS	LES	sox			
LESSON	16	•	BINARY THE IN ACCUMU EXAMPL	V COD NSTRU ALATO LES F	ED DE CTION R) IS OLLOW	CIMAL DAA INTR THIS	Not (Dec Duc LES	ATIO IMAL ED. SON	N Adj	UST	
LESSON	17	-	POSITI THE IN (COMPL AS THE TABLE ON THE IN APP EXAMPL	IVE & STRUE EMEN OVEI OF TE OVEI ENDI ES F	NEGA CTION T) ANI RFLOW RFLOW RFLOW X (A) DLLOW	TIVE I S INTI D NEG AND : FECTS AND : THIS	NUMB RODU (NE SIGN OF SIGN LES	ER N GATE FLA ALL FLA SON	OTAT ARE), A GS. INST GS 1	ION CPL S WELL A RUCTIO S GIVE	NS
LESSON	18	-	PARITY THE PA INTROD A TABL INSTRU IN APP	RITY NUCED E OF ICTIOI ENDID	FLAG THE I VS ON C (Å)	AND S	ITS I IS OI PARI	USES F ALI Ty Fi	ARE L	IS 61V	EN

LESSON 19	-	THE FLAG REGISTER AND AF REGISTER Pair
LESSON 20	-	SIGN & P/V FLAGS IN INSTRUCTIONS
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
JP PO. NN		CALL PO.NN REI PO PARITY ODD = 0
IP P ww		CALL P MM RET P Sign AVE
JP N. NN		CALL M.NN RET M SIGN -VE
		IF THE CONDITION IS NOT MET THE PROGRAM WILL NOT JUMP, CALL A SUBROUTINE OR RETURN
		EXAMPLES FOLLOW THIS LESSON
LESSON 21	-	BIT MANIPULATION
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
SET N: (HL)		WHERE N IS THE BIT NUMBER 0-7
RES M.P.		
Bit NJ(HL)		r
		EXAMPLES FOLLOW THIS LESSON
LESSON 22	-	LOGICAL INSTRUCTIONS THE FOLLOWING INSTRUCTIONS ARE COVERED:
AND N AND R AND (HL)		
OR N		
OR (HL)		
808 N		
XOR (HL)		
		EXAMPLES FOLLOW THIS LESSON
LESSON 23	-	SHIFT INSTRUCTIONS
		SHIFT INSTRUCTIONS ARE PICTORIALLY ILLUSTRATED IN APPENDIX (B).
		THE FOLLOWING INSTRUCTIONS ARE COVERED:
SRA (HL)		DIVIDES THE AND THE NUMBERS BY 2
SRL R SRL (HL)		DIVIDES VE NUMBERS 0 - 255 BY 2
SLA R SLA (HL)		MULTIPLIES THE AND THE NUMBERS BY 2
		EXAMPLES FOLLOW THIS LESSON

LESSON 24 -	ROTATE INSTRUCTIONS
	ROTATE INSTRUCTIONS ARE PICTORIALLY ILLUSTRATED IN APPENDIX (8)
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
RLC 2	ROTATE & LEFT. CARRY DUPLICATES
RLCA	ROTATE À LEFT. CARRY DUPLICATES
8-9	ROTATE R AND CARRY LEFT
RLA	ROTATE A AND CARRY LEFT
RRC P	ROTATE R RIGHT, CARRY DUPLICATES
RRCA	ROTATE À RIGHT, CARRY DUPLICATES
RR 9	ROTATE R AND CARRY RIGHT
RRA (HL) RRA	ROTATE A AND CARRY RIGHT
	EXAMPLES FOLLOW THIS LESSON
LESSON 25 -	DECIMAL ROTATE
	DECIMAL ROTATE INSTRUCTIONS ARE PICTORIALLY ILLUSTRATED IN APPENDIX (B)
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
RLD	ROTATE LEFT DECIMAL (x10)
RRD	ROTATE RIGHT DECIMAL(/10)
	EXAMPLES FOLLOW THIS LESSON
LESSON 26 -	INDEX REGISTERS
	The IX or IY register can replace the HL register in all instructions except ADC HL.DD SBC HL.DD and EX DE.HL
	THE FOLLOWING INSTRUCTIONS ARE COVERED:
LR 8, (1X+D)	LD SP.IX
FR SIXERS:	
ADD A. (IX+D)	INC (1X+p) AND (1X+p)
ADC A.(IX+D)	
SBC A. (IX+D)	CP (IX+p)
ADD IX.DD	INC IX DEC 1X
SLA (IX+D)	SRA (IX+D) SBL (IX+D)
	RL (IX+D) RRC (IX+D)
SET N (IX+D)	RES N.(IX+D) BIT N.(IX+D)
JP (IX)	
	EXAMPLES FOLLOW THIS LESSON

LESSON 27	-	THE ALTERNATIVE SET OF REGISTERS THE FOLLOWING INSTRUCTIONS ARE COVERED:
EX AF.AF' EXX		EXCHANGES THE CONTENTS OF AF AND AF' EXCHANGES BC, DE AND HL, WITH BC', DE' AND HL' RESPECTIVELY.
		EXAMPLES FOLLOW THIS LESSON
LESSON 28	•	INPUT AND OUTPUT INSTRUCTIONS THE FOLLOWING INSTRUCTIONS ARE COVERED:
IN A.(N) IN B.(C) IN F.(N) QUI (N).A		WHERE N IS THE NUMBER OF THE INPUT PORT (0-255)
OUT (C),R		EXAMPLES FOLLOW THIS LESSON
LESSON 29	-	BLOCK INSTRUCTIONS
LESSON 30	-	BLOCK TRANSFER INSTRUCTIONS THE FOLLOWING INSTRUCTIONS ARE COVERED:
EBIR EBBR		POINTER INCREMENTED POINTER INCREMENTED AND REPEATED UNTIL NUMBER FOUND OR BC=0 POINTER DECREMENTED POINTER DECREMENTED AND REPEATED UNTIL NUMBER FOUND OR BC=0 FAMBLES EQUION THIS I SESON
LESSON 31	-	BLOCK SEARCH THE FOLLOWING INSTRUCTIONS ARE COVERED-
EBIR EBBR		POINTER INCREMENTED POINTER INCREMENTED NUMBER FOUND OR BLOO POINTER DECREMENTED POINTER DECREMENTED AND REPEATED UNTIL NUMBER FOUND OR BLOO
		EXAMPLES FOLLOW THIS LESSON
LESSON 32	-	BLOCK INPUT/OUTPUT INSTRUCTIONS THE BLOCK INPUT INSTRUCTIONS COVERED ARE:
IN R IND OTDR		INCREMENTING INCREMENTING AND REPEATING DECREMENTING AND REPEATING DECREMENTING AND REPEATING

OUT I OT IR OUT D OT DR		THE BLOCK OUTPUT INSTRUCTIONS COVERED ARE: INCREMENTING INCREMENTING AND REPEATING DECREMENTING DECREMENTING
		EXAMPLES FOLLOW THIS LESSON
LESSON	33 -	PROCESSOR CONTROL INSTRUCTIONS THE FOLLOWING INSTRUCTIONS ARE COVERED:
NOP HALT RST W		инеке н - ООН. ОВН. 10Н. 18Н. 20Н. 28Н.
EB &:¤ EB { :å		
		EXAMPLES OF THE USE OF THE REFRESH REGISTER FOLLOW THIS LESSON.
LESSON 3	54 -	INTERRUPTS THE FOLLOWING INSTRUCTIONS ARE COVERED:
EI		ENABLE INTERRUPTS
DI		DISABLE INTERRUPTS
IR2	}	INTERRUPT MODES
RET I RETN		Return from Interrupt Return from Non-Maskable Interrupt
LESSON 3	5 -	FINALE

13 V TIREJAN	SINGLE	INSTRUCTIONS ON REGISTER INSTRUCT	5101.	
Institut 1 cm	CABAY	ZERO	۲ ۲	5161
ALL LD	•	•	•	•
EXCEPT LDA. 1 & LDA.R	•		2	-
ALB, AC, SM, SK, C	-	-	7	-
IN. KC	•	-	~	-
M			•	-
K 6	-		>	-
SHIFT & BOTATE SPECIFICALLY ON A	-	•	•	•
ML 9, ND0	٠	•	۰.	-
ALL OTHER SHIFT & HOTATES ON N	-	-	•	
AND. OR. YOR	0	-	œ	-
917	•	•	~	~
ST.RS	•	•	•	•
tot		•	•	•
XF	-	•	•	•
NOTATION . NOT AFFECTED	>	P/V INDICATES ON	ERFLON	AFECTED ACCORDING
P PA INDICATES PARI	17 0	R.A6 RESEI		TO INSTRUCTION
1 FUG SET	~	R.M. STATE DILLOND	5	

Prendix. A	EFFECT OF MEGISTE	INSTRUCTIONS ON	FLAKS S		
INSTRUCTION	CURY	2230	æ	۲	SIGN
TT TD INSTRUCTIONS	•	•		•	•
LL EXCRANGE INSTRUCTIONS	•	•		•	•
8	•	•		•	•
K. 39C	-	-		~	•
M. DEC	•	•		•	•
NSN. POP	•	•		•	•
DIATION • NOT AFFECTED		P/V INDICATES O	MBRON	I AFFECTE	ED ACCOR
P P/V INDICATES) H.A.G. RESET P. F.A.G. STATE UNDIA	Ę	IO INS	

APPENDIX	•		EFFECT	OF INSTR	UCTION	S ON PAIRS		
			NISC	ELLANEOUS	INSTRI	JCT TONS		
	INSTRUCT	ION	CARR	7	ZER	~	٩/٧	SIGN
ALL JP, JR	CALLS.R	ET	٠		٠		•	•
2 DJNZ								
LD1,LD0			٠		٠		i(2)	•
LDIR LDDR			•		٠		0	+
CP1, CP1R, (CPD, CPDR		٠		9	\$	i(2)	
IN ASKNOC	OUT (N)	~ v	•		٠		•	٠
OUT (C).	~	^						
IN R. (C)			٠		-		٩	
INI. IND.	DUTI. OUT	e	٠		2	2	۰.	¢.
INIR, INDR.	OTIR.OT	ž	٠		-		4	<u>ر.</u>
NOP. HALT	, DI. EI.	, In	•		٠		•	٠
NOTES:	(I) PA	V DISPLAYS STA	TE OF 1	NTERRUPT	ENABLE	FLA6		
-	(2) PV	V = 0 1F BC =		- V/4 10	-			
-	(3) 7.	- 1 IF A - (H		012=0				
-	- Z (b)	- 1 IF B - O	÷ 1	07 2 - 1				



GLOSSARY

ASSEMBLY LANGUAGE - A LANGUAGE USING MNEMONICS TO REPRESENT MACHINE CODE OPERATIONS. A LOW-LEVEL LANGUAGE. AN ASSEMBLY LANGUAGE PROGRAM CAN NOT ITSELF BE RUN UNTIL IT IS ASSEMBLED.

BINARY - Two. IN BINARY ARITHMETIC THE DIGITS O AND I ARE USED TO REPRESENT NUMBERS.

BINARY CODED DECIMAL (BCD) - A SYSTEM WHERE A NYBBLE REPRESENTS ONE DECIMAL NUMBER. INEREORE A BYTE CAN REPRESENT TWO DECIMAL NUMBERS.

BIT - ONE SINGLE BINARY DIGIT, EITHER A ONE OR A ZERO.

BUG - AN ERROR OR UNDESIRABLE ASPECT IN A PROGRAM, which prevents a program from working correctly or not at all.

BYTE - A group of binary bits, usually 8, considered as one unit.

CHARACTER - AN ELEMENT OF A SET OF SYMBOLS, SUCH AS A LETTER OR NUMBER, OR SPECIAL SYMBOL.

CHIP - COMMON NAME FOR INTEGRATED CIRCUIT. DERIVED FROM THE SMALL PIECE OF SILICON ON WHICH THE INTEGRATED CIRCUIT IS CHEMICALLY FORMED.

COMPUTER - A MACHINE THAT ACCEPTS DATA, ACTS UPON IT, and supplies the results of the processing as a result of certain instructions. A collective noum describing the processor and I/O devices.

CRASH - TERM USED TO DESCRIBE THE COMPUTER 'LOCKING UP' AND NOT ACCEPTING ANY INPUT FROM THE KEYBOARD. THE OWLY SOLUTION IS TO TURN THE COMPUTER OFF AND THEN ON AGAIN.

CURSOR - A FLASHING THIN LINE USED TO INDICATE WHERE DATA IS EXPECTED TO BE ENTERED ON A VDU.

DATA - A PIECE OF INFORMATION WHICH THE COMPUTER CAN PROCESS.

EDITING - THE PROCESS OF CHANGING DATA BEFORE IT IS COMMITTED TO THE PROCESSOR.

EXECUTE - TO CARRY OUT THE INSTRUCTIONS IN A PROGRAM. A MICROPROCESSOR EXECUTES A PROGRAM BY READING AND ACTING ON THE INSTRUCTIONS. GRAPHICS - TERM DESCRIBING THE DISPLAY OF DATA IN PICTORIAL FORM. PICTURES ON SCREEN ARE DISPLAYED USING PIXELS.

HARDWARE - PARTS OF THE COMPUTER THAT PHYSICALLY EXIST. THE COMPUTER AND A PRINTER FOR EXAMPLE.

HEXADECIMAL - A NUMBER BASE USING 16 DIFFERENT DIGITS FOR EACH NUMBER COLUNN. THE DIGITS 0-9 AND A-F ARE COMMONLY USED.

INSTRUCTION - A CERTAIN ACTION TO BE TAKEN BY THE PROCESSOR. A MACHINE CODE PROGRAM IS MADE UP OF INSTRUCTIONS.

MACHINE CODE - BINARY REPRESENTATION OF THE INSTRUCTIONS OF THE NICROPROCESSOR. MACHINE CODE CAN BE ACTED UPON BY THE MICROPROCESSOR MITHOUT ANY FURTHER TRANSLATION.

MICROPROCESSOR - AN INTEGRATED CIRCUIT THAT CONTAINS ALL THE COMPONENTS TO PERFORM THE BASIC DATA PROCESSING OPERATIONS, ALL IN OME PACKAGE, A MICROPROCESSOR MUST BE CONNECTED TO MEMORY AND 1/0 DEVICES BEFORE IT CAN BE USED.

MMEMONIC - A group of 3/4 characters representing a machine code instruction. Each mnemonic is translated by an assembler into a machine code instruction.

NYBBLE - A group of four bits. There are two nybbles per byte.

OBJECT PROGRAM - A PROGRAM IN MACHINE CODE. THE SOURCE PROGRAM, WHICH CANNOT BE EXECUTED BY THE PROCESSOR, IS ASSEMBLED BY THE ASSEMBLER WHICH GENERATES AN OBJECT PROGRAM. THIS OBJECT PROGRAM RESIDES IN MEMORY, AND CAN BE EXECUTED BY THE PROCESSOR.

OPERATING SYSTEM - A MACHINE CODE PROGRAM, PART OF THE systems software, which enables the processor to perform the data processing and control functions.

PAGE - WHEN USED IN CONJUNCTION WITH MEMORY, MEANS 256 BYTES OF MEMORY.

PROGRAM - A COLLECTION OF INSTRUCTIONS TO MAKE THE MICROPROCESSOR PERFORM A CERTAIN TASK.

RAM - RANDON ACCESS MEMORY. THIS KIND OF MEMORY HAY BE WRITTER TO OR READ FROM. THIS KIND OF MEMORY USED TO STORE THE PROGRAM THAT IS BEING DEVELOPED. IF YOU TURN THE COMPUTER OFF, ALL DATA CONTAINED IN RAM WILL BE LOST.

ROM - READ UNLY MEMORY. THIS KIND OF MEMORY IS SET UP AT THE FACTORY WHERE THE COMPUTER IS MADE. IT USUALLY HOUSES THE OPERATING SYSTEM AND OTHER PROGRAMS NECESSARY EACH TIME THE COMPUTER IS TURNED OM. JURNING THE COMPUTER OFF AND THEN BACK ON AGAIN HAS NO EFFECT ON FOM.

SOFTWARE - A NON-PHYSICAL PART OF A COMPUTER SUCH AS A PROGRAM.

SOURCE PROGRAM - THE PROGRAM THAT CONSISTS OF MNEMONICS THAT CAN BE UNDERSTOOD BY HUMANS. THIS PROGRAM CANNOT BE EXECUTED UNTIL IT IS ASSEMBLED.

NOTES

NOTES

Copyright 1984 New Generation Software Ltd. Beth, Avon.