INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONES: 224281/2/3 4/5/6
CABLE: CENTRATOM - TELEX 460392-I

SMR/90 - 34

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

7 September - 2 October 1981

INTRODUCTION TO THE MEK 6802 KIT

R.A. McLAREN

DD Division
CERN
1211 Geneva 23
Switzerland

INTRODUCTION TO MEK6802 KIT

JOB OF MICRO

DO WORK

- INSTRUCTIONS IT UNDERSTANDS
- NECESSARY HARDWARE
- PROGRAM IN MEMORY
- START PROGRAM

IN CASE OF ERRORS

- DEBUG THEM.

JOB REQUIRED

- VAR2 = VAR1 + 5
- MICRO ONLY "UNDERSTANDS" MACHINE CODE
- ? ENGLISH → HLL → MACHINE CODE
- PROBLEM REQUIRES ARITHMETIC
  ∴ USE ACCUMULATOR
- MICRO DOESN'T UNDER. SYMBOLS

VAR1 = LOCATION 10
VAR2 = LOCATION 11

MOVE CONSTANT TO ACC

ADD CONTENTS OF 10

STORE RESULT IN 11

DOES INSTRUCTION SET HAVE MOVE, ADD, STORE?

Instruction Classes

- Data Movement
- Data Modification
- Arithmetic
- Logic
- Jump and Branch
- Data Test
- Condition Code Change
- Index and Stack Ptr
- Interrupt

DATA HANDLING INSTRUCTIONS
(Data Movement)

Ref Manual
Accx ← (M)

FUNCTION	MNEMONIC	OPERATION
LOAD ACMLTR	LDAA	M → A
	LDAB	M → B
PUSH DATA	PSHA	A → M_{SP}, SP − 1 → SP
	PSHB	B → M_{SP}, SP − 1 → SP
PULL DATA	PULA	SP + 1 → SP, M_{SP} → A
	PULB	SP + 1 → SP, M_{SP} → B
STORE ACMLTR	STAA	A → M
	STAB	B → M
TRANSFER ACMLTRS	TAB	A → B
	TBA	B → A

ARITHMETIC INSTRUCTIONS

FUNCTION	MNEMONIC	OPERATION
ADD	ADDA	A + M → A
	ADDB	B + M → B
ADD ACCUMULATORS	ABA	A + B → A
ADD WITH CARRY	ADCA	A + M + C → A
	ADCB	B + M + C → B
COMPLEMENT, 2'S (NEGATE)	NEG	00 − M → M
	NEGA	00 − A → A
	NEGB	00 − B → B
DECIMAL ADJUST, A	DAA	CONVERTS BINARY ADD. OF BCD CHARACTERS INTO BCD FORMAT
SUBTRACT	SUBA	A − M → A
	SUBB	B − M → B
SUBTRACT ACCUMULATORS	SBA	A − B → A
SUBTRACT WITH CARRY	SBCA	A − M − C → A
	SBCB	B − M − C → B

Load Accumulator — LDA

Operation: ACCX ← (M)

Description: Loads the contents of memory into the accumulator. The condition codes are set according to the data.

Condition Codes:
- H: Not affected.
- I: Not affected.
- N: Set if most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- V: Cleared.
- C: Not affected.

Boolean Formulae for Condition Codes:

$N = R_7$

$Z = \overline{R}_7 \cdot \overline{R}_6 \cdot \overline{R}_5 \cdot \overline{R}_4 \cdot \overline{R}_3 \cdot \overline{R}_2 \cdot \overline{R}_1 \cdot \overline{R}_0$

$V = 0$

Addressing Formats:

See Table A-1.

Addressing Modes, Execution Time, and Machine Code (hexadecimal/ octal/ decimal):

(DUAL OPERAND)

Addressing Modes	Execution Time (No. of cycles)	Number of bytes of machine code	Coding of First (or only) byte of machine code		
			HEX.	OCT.	DEC.
A IMM	2	2	86	206	134
A DIR	3	2	96	226	150
A EXT	4	3	B6	266	182
A IND	5	2	A6	246	166
B IMM	2	2	C6	306	198
B DIR	3	2	D6	326	214
B EXT	4	3	F6	366	246
B IND	5	2	E6	346	230

Add Without Carry — ADD

Operation: ACCX ← (ACCX) + (M)

Description: Adds the contents of ACCX and the contents of M and places the result in ACCX.

Condition Codes:
- H: Set if there was a carry from bit 3; cleared otherwise.
- I: Not affected.
- N: Set if most significant bit of the result is set; cleared otherwise.
- Z: Set if all bits of the result are cleared; cleared otherwise.
- V: Set if there was two's complement overflow as a result of the operation; cleared otherwise.
- C: Set if there was a carry from the most significant bit of the result; cleared otherwise.

Boolean Formulae for Condition Codes:

$H = X_3 \cdot M_3 + M_3 \cdot \overline{R}_3 + \overline{R}_3 \cdot X_3$

$N = R_7$

$Z = \overline{R}_7 \cdot \overline{R}_6 \cdot \overline{R}_5 \cdot \overline{R}_4 \cdot \overline{R}_3 \cdot \overline{R}_2 \cdot \overline{R}_1 \cdot \overline{R}_0$

$V = X_7 \cdot M_7 \cdot \overline{R}_7 + \overline{X}_7 \cdot \overline{M}_7 \cdot R_7$

$C = X_7 \cdot M_7 + M_7 \cdot \overline{R}_7 + \overline{R}_7 \cdot X_7$

Addressing Formats:

See Table A-1

Addressing Modes, Execution Time, and Machine Code (hexadecimal/ octal/ decimal):

(DUAL OPERAND)

Addressing Modes	Execution Time (No. of cycles)	Number of bytes of machine code	Coding of First (or only) byte of machine code		
			HEX.	OCT.	DEC.
A IMM	2	2	8B	213	139
A DIR	3	2	9B	233	155
A EXT	4	3	BB	273	187
A IND	5	2	AB	253	171
B IMM	2	2	CB	313	203
B DIR	3	2	DB	333	219
B EXT	4	3	FB	373	251
B IND	5	2	EB	353	235

Store Accumulator **STA**

Operation: M ← (ACCX)

Description: Stores the contents of ACCX in memory. The contents of ACCX remains unchanged.

Condition Codes:
- H: Not affected.
- I: Not affected.
- N: Set if the most significant bit of the contents of ACCX is set; cleared otherwise.
- Z: Set if all bits of the contents of ACCX are cleared; cleared otherwise.
- V: Cleared.
- C: Not affected.

Boolean Formulae for Condition Codes:

$N = X_7$

$Z = \overline{X}_7 \cdot \overline{X}_6 \cdot \overline{X}_5 \cdot \overline{X}_4 \cdot \overline{X}_3 \cdot \overline{X}_2 \cdot \overline{X}_1 \cdot \overline{X}_0$

$V = 0$

Addressing Formats:

See Table A-2.

Addressing Modes, Execution Time, and Machine Code (hexadecimal/ octal/ decimal):

Addressing Modes	Execution Time (No. of cycles)	Number of bytes of machine code	Coding of First (or only) byte of machine code		
			HEX.	OCT.	DEC.
A DIR	4	2	97	227	151
A EXT	5	3	B7	267	183
A IND	6	2	A7	247	167
B DIR	4	2	D7	327	215
B EXT	5	3	F7	367	247
B IND	6	2	E7	347	231

LDAA

ADDA M

STAA

PROBLEM (11) = (10) + 5

?

• ADDRESSING MODES

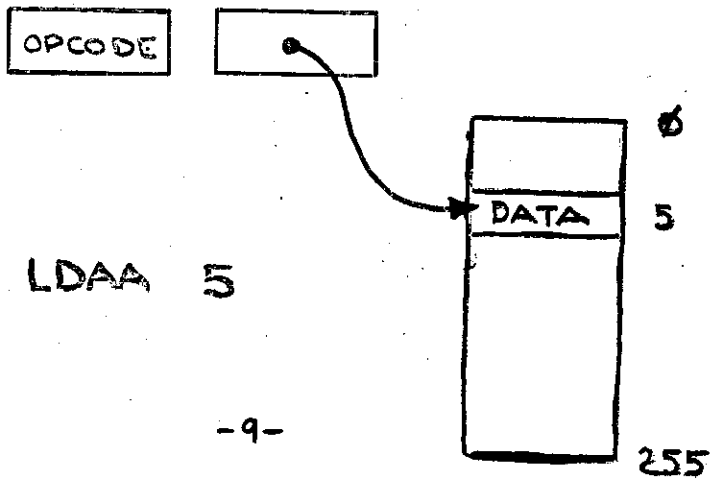
ADDRESSING MODES

SIMPLE :

- ACC / INHERENT / IMPLIED
  NO OPERAND REQUIRED
- IMMEDIATE

LDAA #5

- DIRECT



LDAA 5

PROGRAM IS NOW

86 05
9B 10
97 11

- THIS IS HEXIDECIMAL

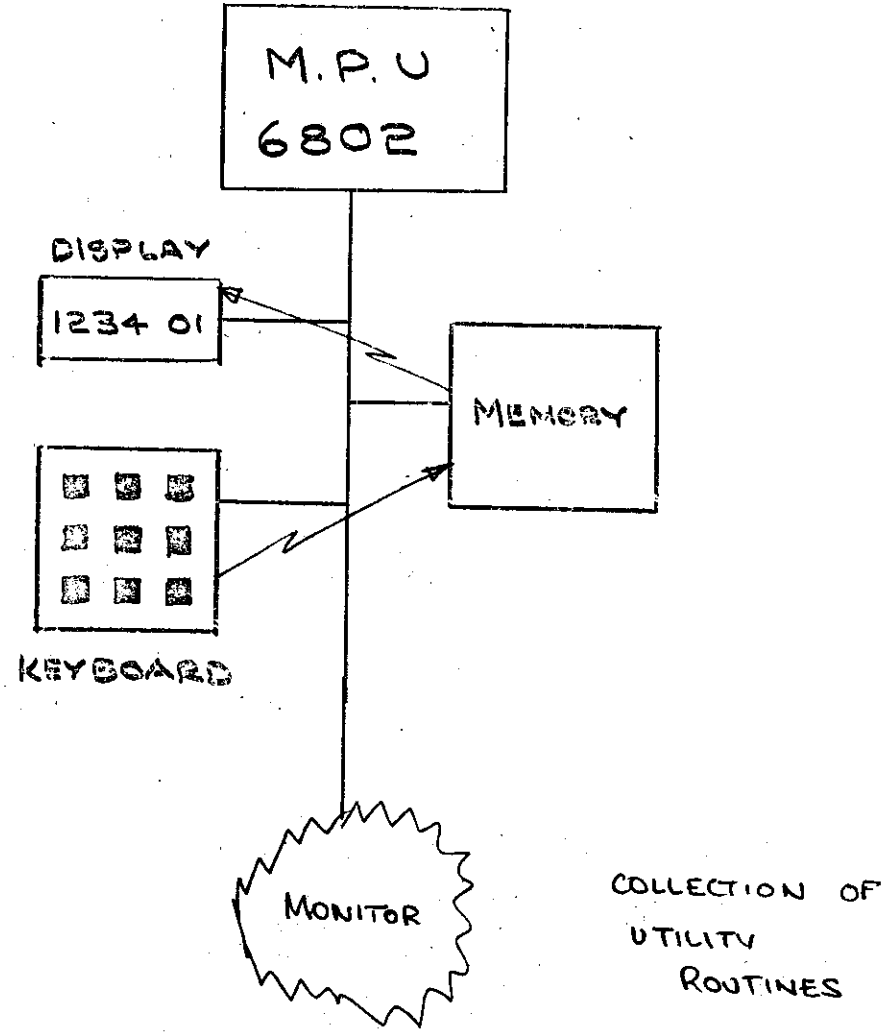
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

$86 = 1000 0110

- NOW WE HAVE MACHINE CODE !

DO WORK

- INSTRUCTIONS IT UNDERSTANDS
- NECESSARY HARDWARE
- PROGRAM IN MEMORY
- START
- DEBUG



FUNCTIONS ?

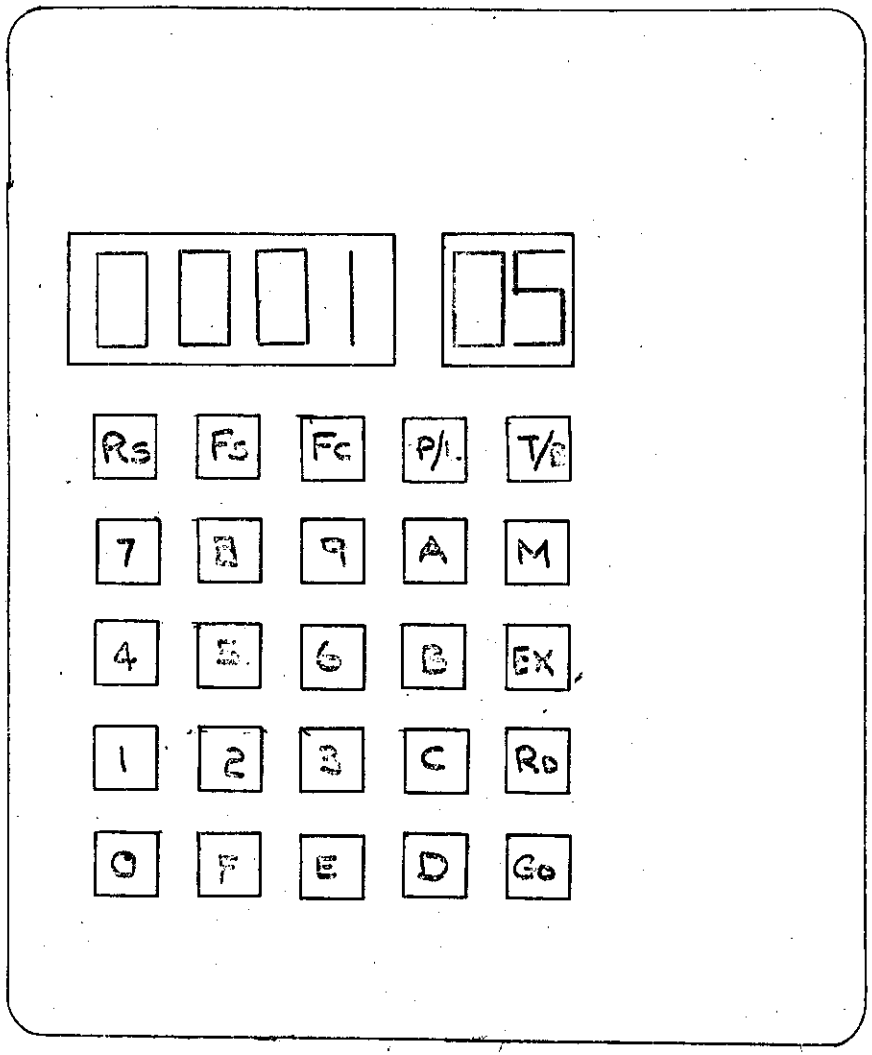
- LOAD MEMORY
- DISPLAY MEMORY
- START PROGRAM
- CHANGE / EXAMINE REGISTERS
- EMERGENCY STOP
  RESET / ESCAPE
- CONTROLLED EXECUTION
  BREAK-POINTS / TRACE

SEE REF MANUAL (CHAPTER 3)
* COURSE NOTES.

EXERCISES

1 MORE INSTRUCTION

- STOP EXECUTION
  SWI → $3F


0001
05
Rs
Fs
Fc
P/C
T/B
7
8
9
A
M
4
5
6
B
Ex
1
2
3
C
Rb
0
F
E
D
Go


= B
= 6