Batch-8.

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

DEPARTMENT OF AVIONICS

MICROPROCESSOR & MICROCONTROLLER LAB

8051 MICROCONTROLLER

MAIN FEATURES OF ESA51 Microcontroller

- ESA 51 operates on single +5V power supply either in stand-alone mode using PC keyboard and LCD or with host PC through its RS-232-C / RS 485 interface in serial mode.
- Stand-alone and serial monitor programs support the entry of user programs, editing and debugging facilities like breakpoints (128K), single stepping and full speed execution of user programs.
- Line assembler & disassembler.
- Total on-board memory is 128K bytes of which 96K bytes RAM has battery backup
- provision.
- On-board parallel printer port.
- On-board 8 bit DAC using 0800.
- Optional on-board 12 bit ADC using AD1674.
- 48 I/O lines and four programmable interval timers.
- 13 port lines of 8031 brought out to the connector including INT1, RXD & TXD pins (6 lines are shared for optional ADC).
- Buffered bus signals are available through ribbon cable connector for easy system expansion.
- Driver software for file upload/download to/from host PC.

❖ **CENTRAL PROCESSOR**

8031 MCU @ 11.0592 MHz.

❖ **PROGRAM MEMORY**

ROM : 32K bytes of system firmware using 27C256.

RAM : 32K bytes using 62256.

❖ **DATA MEMORY**

RAM : 64K bytes using 62256 (32K X 2). Upper most 8K bytes are reserved for I/O addressing and I/O expansion.

❖ **MEMORY MAP**

Devices	Address range	Type of memory
27256 at U9	0000-7FFF	Program memory
62256 at U10	8000-FFFF	User program memory
62256 at U11	0000-7FFF	User Data memory
62256 at U12	8000-DFFF	User Data memory

8051 Microcontroller Instruction set

8051 Microcontroller Instruction set

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,R_n	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@R_i	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,R_n	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@R_i	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,R_n	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@R_i	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	R_n	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@R_i	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	R_n	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@R_i	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12

Note: 1. All mnemonics copyrighted © Intel Corp., 1980.

Mnemonic		Description	Byte	
LOGICAL OPERATIONS				
ANL	A,R_n	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@R_i	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,R_n	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@R_i	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,R_n	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@R_i	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
LOGICAL OPERATIONS (continued)				

Mnemonic		Description	Byte	Oscillator Period
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,R_n	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@R_i	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	R_n,A	Move Accumulator to register	1	12
MOV	R_n,direct	Move direct byte to register	2	24
MOV	R_n,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,R_n	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@R_i	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@R_i,A	Move Accumulator to indirect RAM	1	12
MOV	@R_i,direct	Move direct byte to indirect RAM	2	24
MOV	@R_i,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@R_i	Move External RAM (8-bit addr) to Acc	1	24
DATA TRANSFER (continued)				
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24

Mnemonic		Description	Byte	Oscillator Period
MOVX	@R_i,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,R_n	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@R_i	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@R_i	Exchange low-order Digit indirect RAM with Acc	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24

Mnemonic		Description	Byte	Oscillator Period
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	R_n,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@R_i,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	R_n,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

The Instruction Set and Addressing Modes

R_n	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@R_i	8-bit internal data RAM location (0-255) addressed indirectly through register R1or R0.
#data	8-bit constant included in instruction.
#data 16	16-bit constant included in instruction.
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64K byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

EXERCISE

1. Write an ALP in 8051 to
   1.1 Add two 8 bit immediate numbers.
   1.2 Subtract two 8 bit immediate numbers.
   1.3 Multiply two 8 bit immediate numbers.
   1.4 Find the square of an 8 bit number.
2. Write an ALP in 8051 to perform addition of two numbers (8-bit) present in data memory locations 9000 and 9001. Store the sum and carry in 9002 and 9003 data memory.
3. Write an ALP in 8051 to perform subtraction of two numbers (8-bit) present in data memory locations 9100 and 9101.Store result in 9102 data memory.
4. Write an ALP in 8051 to multiply two numbers (8-bit) present in data memory locations 9200 and 9201.Store result in 9202 and 9203 data memory.
5. Write an ALP in 8051 to perform division of two numbers (8-bit) present in data memory locations 9300 and 9301. Store result in 9302 and 9303 data memory.

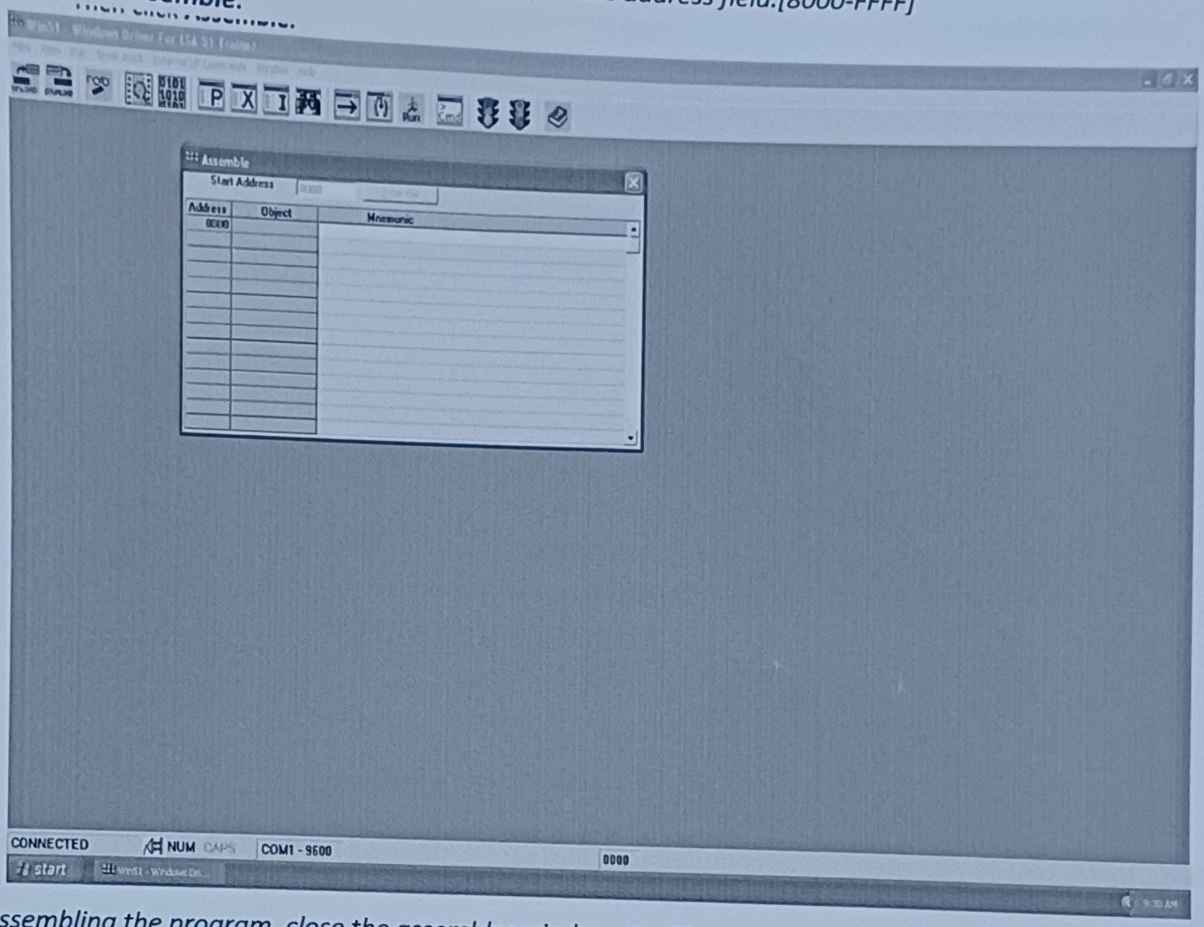
Procedure to use the ESA51 Microcontroller trainer Kit and Win51 software:

1. Turn on the computer and 8051 microcontroller kit.
2. Double click Win 851 on the desktop.
3. For assembling the program,

   View —> Line Assembler *or* from the tool bar 'Assembler' icon *or* Control+L.

   Type a location from which the program starts in the address field.[8000-FFFF]

   Then click ***Assemble.***



After assembling the program, close the assembler window. It will automatically save the program. 4. For viewing and substituting contents in external and internal data memory locations, View —> Memory space *or* click 'view memory' from the Tool bar.

That is, click & from the Tool bar. Type the memory address and fill with the respective data.

X I

5. For running the program

   Click '**Set PC**' and type the **starting address** of the program in the address field .

   Click '**Run**'.

6. To view the contents in Registers

   View —>Registers.

7. For disassembling the program,

   View —> Disassembly *or* from the tool bar '**Disassembly**' icon *or* **Control+C.**

   Type the location of the program. Select the line which has to be modified, then click ***Assemble*** *and type.*

Note:

1. The "B" register is only used by two 8051 instructions: **MUL AB** and **DIV AB**. In other cases to use this register, use its address, that is '**0F0**' along with the mnemonics and operands.

2. Use **LJMP 3** or **LJMP 0** or **LABEL: SJMP LABEL** to halt the program execution.
3. Refer the tutorials available on the desktop folder named '8051' for more details about the 8051 microcontroller and instruction set.