Input

8100 : 02

8101 : 03

Output

8102 : 05

/ /

8051: ARITHMETIC OPERATIONS

EXPERIMENT-5.A

ADDITION OF TWO SINGLE BYTE NUMBERS

Aim: To write and execute a program for addition of two single byte numbers using 8051

Address	Instruction	opcode	Explanation
8000	MOV DPTR, #8100	90:81:00	DPTR points to memory location 8100
8003	MOVX A, @DPTR	E0	Content in DPTR is moved to accumulator
8004	MOV R0, A	F8	Content in A is moved to R0
8005	INC DPTR	A3	Increment DPTR
8006	MOVX A, @DPTR	E0	Content in DPTR is moved to accumulator
8007	ADD A, R0	28	Content of A and R0 and result is stored in A
8008	INC DPTR	A3	Increment DPTR
8009	MOVX @DPTR, A	F0	Value in A is moved to location pointed by DPTR
800A	MOV A, #00	74:00	Clear accumulator
800C	ADDC A, #0	34:00	Add accumulator with cassy
800E	INC DPTR	A3	Increment DPTR
800F	MOVX @DPTR, A	F0	Value in A is moved to location pointed by DPTR
8010	SJMP 8010	80:FE	

Result

Program executed and output obtained successfully

Input

8100: 03
8101: 05

Output

8102: 02

/ /

EXPERIMENT 5B

SUBTRACTION OF TWO SINGLE BYTE NUMBERS

Aim: To write and execute a program for subtraction of two single byte numbers using 8051

Address	Instruction	Opcode	Explanation
8000	CLR C	C3	Clear carry
8001	MOV DPTR, #8100	90:81:00	DPTR points to memory location 8100
8004	MOVX A, @DPTR	E0	Move content in DPTR to Accumulator
8005	MOV R0, A	F8	Move content in Accumulator to R0
8006	INC DPTR	A3	Increment DPTR
8007	MOVX A, @DPTR	E0	Move content in DPTR to Accumulator
8008	SUBB A, R0	98	Subtract R0 from A and store in A
8009	INC DPTR	A3	Increment DPTR
800A	MOVX @DPTR, A	F0	Move content in A to memory location pointed by DPTR
800B	SJMP 800B	80:FE	

Result

Program was executed and output obtained successfully.

Input

8100 : 04
8101 : 02

Output

8102 : 08

/ /

EXPERIMENT-5C

MULTIPLICATION OF TWO 8-BIT NUMBERS

Aim: To write and execute a program for multiplication of two 8-bit numbers using 8051

Address	Instruction	Opcode	Explanation
8000	MOV DPTR, #8100	90:81:00	DPTR points to memory location 8100
8003	MOVX A, @DPTR	E3	Move content of DPTR to Accumulator
8004	MOV F0, A	F5: F0	Move content of Accumulator to F0 (B)
8006	INC DPTR	A3	Increment DPTR
8007	MOVX A, @DPTR	E0	Move content of DPTR to Accumulator
8008	MUL AB	A4	Multiply contents of A and B and Result is stored in A
8009	INC DPTR	A3	Increment DPTR
800A	MOVX @DPTR, A	F0	Move content of A to memory location pointed by DPTR
800B	MOV A, F0	E5:F0	Move content of F0 (B) to Accumulator
800E	MOVX @DPTR, A	F0	Move content of A to memory location pointed by DPTR
800F	SJMP 800F	80:FE	

Result

Program was executed and output was obtained successfully.

Input

8100 : 03
8101 : 07

Output

8102 : 02
8103 : 01

/ /

EXPERIMENT-5D

DIVISION OF TWO 8-BIT NUMBERS.

Aim: To write and execute program for division of two 8-bit numbers. using 8051

Address	Instruction	Opcode	Explanation
8000	MOV DPTR, #8100	90:81:00	DPTR points to memory location 8100
8003	MOVX A, @DPTR	E3	Move content of DPTR to Accumulator
8004	MOV F0, A	F5: F0	Move content of Accumulator to F0
8006	INC DPTR	A3	Increment DPTR
8007	MOVX A, @ DPTR	E0	Move contents of DPTR to Accumulator
8008	~~MUL~~ DIV AB	84	Divide content of A by content of B and store quotient in A and remainder in B.
8009	INC DPTR	A3	Increment DPTR
800A	MOVX @DPTR, A	F0	Move content of A to memory location pointed by DPTR
800B	MOV A, F0	E5:F0	Move content of F0 to A
800D	INC DPTR	A3	Increment DPTR
800E	MOVX @DPTR, A	F0	Move content of A to memory location pointed by DPTR
800F	SJMP 800F	80:FE	

Result

Program was executed and output obtained successfully

Input

0500:02
0501:03

Output

0550:05

8086: ARITHMETIC OPERATIONS

EXPERIMENT - 6.A

ADDITION OF TWO 8-BIT NUMBERS

Aim: To write and execute program for addition of two 8 bit numbers using 8086

Address	Instruction	Opcode	Explanation
0400	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0403	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0406	MOV AL, [SI]	8A:04	Move content in SI location to AL register
0408	INC SI	46	Increment SI register
0409	MOV BL, [SI]	8A:1C	Move content in SI location to BL register
040B	ADD AL, BL	00:D8	Add AL and BL register
040D	MOV [DI], AL	88:05	Move content of AL to location pointed by DI
040F	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500:10
0501: 03
0502: 20
0503: 10

Output

0550:30
0551: 13

/ /

EXPERIMENT - 6.B

ADDITION OF TWO 16-BIT NUMBERS

Aim: To write and execute a program for addition of two 16-bit numbers using 8086

Address	Instruction	Opcode	Explanation
0400	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0403	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0406	MOV AX, [SI]	8B:04	Move the memory content of SI to AX register
0408	MOV BX, [SI+02]	8B:5C:02	Move the memory content of [SI+02] to BX register
040B	ADD AX, BX	01:D8	Add AX and BX register and store the result in AX
040D	MOV [DI], AX	89:05	Move content of AX to memory location pointed by DX
040F	HLT	F4	

Result

Program code executed and output obtained successfully

Input

0500 : B5
0501 : 7A
0502 : 2A
0503 : E5

Output

0550 : DF
0551 : 5F
0552 : 01

/ /

EXPERIMENT - 6C

ADDITION OF TWO 16-BIT NUMBERS CONSIDERING CARRY

Aim: To write and execute a program for addition of two 16-bit numbers considering carry

Address	Instruction	Opcode	Explanation
0400	CLC	F8	Clear carry flag
0401	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0404	MOV SI, 0500	BE:00:05	Move 0500 to SI Register
0407	MOV AX,[SI]	8B:04	Move the memory content of SI to AX register
0409	MOV BX, [SI+02]	8B:BC:02	Move the memory content of [SI+02] to BX register
040C	ADD AX,BX	01:D8	Add Ax and Bx register and store result in Ax register
040E	MOV [DI], AX	89:05	Move content of AX register to memory location in DI register
0410	MOV AX,0000	B8:00:00	Clear Ax register
0413	ADC AX,0000	15:00:00	Add AX with carry and store result in Ax
0416	MOV [DI+02], AX	89:45:02	Move content of AX register to memory location in [DI+02]
0419	HLT	F4	

Result

Program was executed and output was obtained successfully

Input

0500:66
0501:66
0502:11
0503:11

Output

0550:55
0551:55

/ /

EXPERIMENT-6D
SUBTRACTION OF TWO 16-BIT NUMBERS

Aim: To write and execute a program for subtraction of two 16-bit numbers using 8086

Address	Instruction	Opcode	Explanation
0400	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0403	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0406	MOV AX, [SI]	8B:04	Move memory content of SI to AX register
0408	MOV BX, [SI+02]	8B:5C:02	Move memory content of [SI+02] to BX register
040B	SUB AX, BX	29:D8	Subtract Bx from AX and store result in AX register
040D	MOV [DI], AX	89:05	Move content of AX register to memory location in DI
040F	HLT	F4	

Result

Program was executed and output was obtained successfully

Input

0500:18
0501:08
0502:40
0503:10

Output
0550:08
0551:F7

/ /

EXPERIMENT - 6E

SUBTRACTION OF TWO 16 BIT NUMBERS WITH ~~CONSIDERING~~ BORROW

Aim: To write and execute a program for subtraction of two 16-bit numbers with borrow

Address	Instruction	Opcode	Explanation
0400	CLC	F8	Clear carry flag
0401	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0404	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0407	MOV AX, [SI]	8B:04	Move memory content of SI register to AX register
0409	MOV BX, [SI+02]	8B:5C:02	Move memory content of [SI+02] to AX register
040C	SBB AX, BX	19:D8	Subtract Bx from Ax considering the borrow and store results in AX register
040E	MOV [DI], AX	89:05	Move content of AX register to memory location in DI register
0410	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500:1A
0501:2B
0502:4B
0503:12

Output

0550:9E
0551:74
0552:14
0553:03

/ /

EXPERIMENT - 6F

MULTIPLICATION OF TWO 16-BIT NUMBERS

Aim: To write and execute a program for multiplication of two 16-bit numbers

Address	Instruction	Opcode	Explanation
0400	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0403	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0406	MOV AX, [SI]	8B:04	Move memory content of SI to AX register
0408	MOV CX, [SI+02]	8B:4C:02	Move memory content of SI+02 to CX register
040B	MUL CX	F1:E1	Multiply contents of AX and CX register and store lower bit in AX and highest bit in DX
040D	MOV [DI], AX	~~89:45:02~~ 89:05	Move content of AX register to memory location in DI
040F	MOV [DI+02], DX	~~89:55:02~~ 89:15	Move content of DX register to memory location in DI+02
0412	HLT	F4	

Result

Program was executed and output was obtained.

Input

0500: 43
0501: 12
0502: 21

Output

0550: 8D
0551: 16

EXPERIMENT - 6G

DIVISION OF 16-BIT NUMBER BY 8-BIT NUMBER

Aim: To write and execute a program for division of a 16-bit number by an 8-bit number using 8086.

Address	Instruction	Opcode	Explanation
0400	MOV DI, 0550	BF:50:05	Move 0550 to DI register
0403	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0406	MOV AX, [SI]	8B:04	Move memory content of SI into AX register
0408	MOV CL, [SI+02]	8A:4C:02	Move memory content of SI+02 into CL register
040B	DIV CL	F6:F1	Divide AX by CL and store quotient in AL and remainder in AH
040D	MOV [DI], AX	89:05	Move content of AX register to memory location in DI
040F	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500 : 05
0501 : 01
0502 : 02
0503 : 03
0504 : 04
0505 : 05

Output

0601 : 01
0602 : 04
0603 : 09
0604 : 10
0605 : 19

/ /

EXPERIMENT - 7

SQUARES OF 8-BIT NUMBERS IN A BLOCK

Aim : To write and execute a program to find squares of 8-bit numbers stored in a block

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV DI, 0601	BF:01:06	Move 0601 to DI register
0406	MOV CL, [SI]	8A:0C	Move memory content of SI register to CL
0408	INC SI	46	Increment SI
0409	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
040B	MUL AL	F6 : E0	Multiply contents of AL register with AL and store the result in AL register
040D	MOV [DI], AL	88:05	Move content of AL to memory location in DI register
040F	INC DI	47	Increment DI register
0410	DEC CL	FE:09	Decrement CL register (count)
0412	JNZ 0408	75:54	If zero flag is not set then jump to location 0408
0414	HLT	F4	

Result

Program was executed and output was obtained.

Input

0500:03

Output

0601:06

EXPERIMENT - 8

FACTORIAL OF A NUMBER

Aim: To write and execute a program to find factorial of a number

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV DI, 0601	BF:01:06	Move 0601 to DI register
0406	MOV CL, [SI]	8A:0C	Move memory content of SI to CL register
0408	MOV AL, 01	B0:01	Move 01 to AL register
040A	MUL CL	F6:E1	Multiply CL register with AL register and store result in AL register
040C	DEC CL	FE:C9	Decrement CL register
040E	JNZ 040A	75:5A	If zero flag is not set then jump to location 040A
0410	MOV [DI], AL	88:05	Move content of AL register to memory location in DI
0412	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500:05
0501:01
0502:02
0503:03
0504:04
0505:05

Output

0601:09

EXPERIMENT-9

SUM OF ODD NUMBERS FROM A SERIES OF NUMBERS

Aim: Write and execute a program to find sum of odd numbers from a series of numbers

Address	Instruction	Opcode	Explanation
0400	MOV BL, 00	B3:00	Move 00 to BL register
0402	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0405	MOV DI, 0601	BF:01:06	Move 0601 to DI register
0408	MOV CL, [SI]	8A:0C	Move memory content of SI to CL register
040A	INC SI	46	Increment SI register
040B	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
040D	TEST AL, 01	A8:01	Perform bitwise AND operation between AL register and 01 and set the flags accordingly
040F	JZ 0415	74:04	If the zero flag is set then jump to location 0415
0411	ADD AL, BL	00:D8	Add contents of AL and BL registers and store result in AL
0413	MOV BL, AL	88:C3	Move content of AL to BL
0415	DEC CL	FE:C9	Decrement CL register
0417	JNZ 040A	75:F1	If the zero flag is not set then jump to 040A
0419	MOV [DI], AL	88:05	Move content of AL to memory location in DI
041B	HLT	F4	

Input

0500: 12
0501: 34

Output

0500: 43
0501: 21

/ /

EXPERIMENT-10

REVERSE OF A 16-BIT NUMBER

Aim: To write and execute a program to reverse a 16-bit number using 8086

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
0405	MOV CL, 04	B1:04	Move 04 to CL register
0407	ROR AL, CL	D2:C8	Rotate the bits in AL register to right by value stored in CL times
0409	INC SI	46	Increment SI register
040A	XCHG AL, [SI]	86:04	Exchange content of AL register with content in memory location SI
040C	ROR AL, CL	D2:C8	Rotate the bits in AL register to right side by value stored in CL numbers of times.
040E	DEC SI	4E	Decrement SI register
040F	MOV [SI], AL	88:04	Move contents of AL register to memory location in SI
0411	HLT	F4	

Result

Program was executed and output obtained successfully

Input 1

0500 : 22

output 1

0601 : 01

Input 2

0500 : 54

Output 2

0601 : 00

/ /

EXPERIMENT-11

8-BIT NUMBER IS PALINDROME OR NOT

Aim: To write and execute a program to check whether an 8-bit number is palindrome or not

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV DI, 0601	BE:01:06	Move 0601 to DI register
0406	MOV CL, 04	B1:04	Move 04 to CL register
0408	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
040A	MOV BL, [SI]	8A:1C	Move memory content of SI to BL register
040C	ROR AL, CL	D2:C8	Rotate the bits in AL register to right side by the value in CL number of times
040E	CMP AL, BL	38:D8	Compare content of AL and BL registers and set flags accordingly
0410	JZ 0417	74:05	If zero flag is set then jump to memory location 0417
0412	MOV [DI], 00	C6:05:00	Clear contents of DI register
0415	JMP 041A	EB:03	Jump to location 041A
0417	MOV [DI], 01	C6:05:01	Move 01 to DI register
041A	HLT	F4	Halt

Result

Program was executed and output was obtained successfully

Input

0500 : 02
0501 : 05
0502 : 03

Output

0600 : 01

/ /

EXPERIMENT - 12

DISCRIMINANT OF A QUADRATIC EQUATION

Aim: To write and execute a program to find out the discriminant of a quadratic equation. Express the nature of roots (00: real and equal, 01: real and distinct 02: imaginary)

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV DI, 0600	BF:00:06	Move 0600 to DI register
0406	MOV BL, [SI]	8A:1C	Move memory content of SI to BL register
0408	INC SI	46	Increment SI register
0409	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
040B	MUL AL	F6:E0	Multiply AL register with itself and store result in AL
040D	MOV CL, AL	88:C1	Move content of AL to CL
040F	MOV AL, 04	B0:04	Move 04 to AL register
0411	MUL BL	F6:E3	Multiply content of BL register with AL and store result in AL
0413	INC SI	46	Increment SI register
0414	MOV BL, [SI]	8A:1C	Move memory content of SI register with BL
0416	MUL BL	F6:E3	Multiply content of BL register with AL and store result in AL
0418	CMP CL, AL	38:C1	Compare contents of AL and CL register and modify flags

041A	JNZ 0421	75:05	If zero flag is not set then jump to location 0421
041C	MOV [DI],00	C6:05:00	Move 00 to DI location
041F	JMP 042B	EB:0A	Jump to location 042B
0421	JC 0428	72:05	If carry flag is set then jump to location 0428
0423	MOV [DI],01	C6:05:01	Move 01 to DI memory location
0426	JMP 042B	EB:03	Jump to location 042B
0428	MOV [DI],02	C6:05:02	Move 02 to DI memory location
042B	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500:05
0501:05
0502:04
0503:02
0504:03
0505:01

Output

0501:01
0502:02
0503:03
0504:04
0505:05

EXPERIMENT - 13

SORT AN ARRAY IN ASCENDING ORDER

Aim: Write and execute a program to sort an array of numbers in ascending order

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV CL, [SI]	8A:0C	Move memory content of SI register to CL
0405	DEC CL	FE:09	Decrement CL register
0407	MOV SI, 0500	BE:00:05	Move 0500 to SI register
040A	MOV CH, [SI]	8A:2C	Move memory content of SI to CH register
040C	DEC CH	FE:CD	Decrement CH register
040E	INC SI	46	Increment SI register
040F	MOV AL, [SI]	8A:04	Move memory content of SI to AL register
0411	INC SI	~~3A:04~~ 46	Increment SI register
0412	CMP AL, [SI]	3A:04	Compare AL and content of SI and set flags accordingly
0414	JC 041C	72:06	If cassy flag is set then jump to location 041C
0416	XCHG AL, [SI]	86:04	Exchange content of AL and [SI]
0418	DEC SI	4E	Decrement SI register
0419	XCHG AL, [SI]	86:04	Exchange content of AL and [SI]
041B	INC SI	46	Increment SI register
041C	DEC CH	FE:CD	Decrement CH register
041E	JNZ 040F	75:EF	If zero flag is not set then jump

			to location 040F
0420	DEC CL	FE:C9	Decrement CL register
0422	JNZ 0407	75:E3	If zero flag is not set then jump to location 0407
0424	HLT	F4	

Result

Program was executed and output obtained successfully

Input

0500: FF
0501: 00

Output

0700: 55
0701: 02

/ /

EXPERIMENT-14

CONVERT HEX TO BCD CONVERSION

Aim: To write and execute a program to convert a 16-bit number from hexadecimal to BCD

Address	Instruction	Opcode	Explanation
0400	MOV SI, 0500	BE:00:05	Move 0500 to SI register
0403	MOV DI, 0700	BF:00:07	Move 0700 to DI register
0406	MOV CL, 0A	B1:0A	Move 0A to CL register
0408	MOV AX, [SI]	8B:04	Move memory content BI to AX register
040A	DIV CL	F6:F1	Divide AX by CL and store quotient in AL and remainder in AH
040C	MOV [DI], AH	88:25	Move content of AH register to memory location pointed by DI
040E	INC DI	47	Increment content of DI register
040F	CMP AL, CL	38:C8	Compare AL and CL register and set flag register accordingly
0411	JC 0419	72:06	If carry flag is set then jump to location 0419
0413	MOV [SI], AL	88:04	Move content of AL with SI register
0415	MOV AH, 00	B4 :00	Move 00 to AH register
0417	JMP 0408	EB:EF	Jump to location 0408
0419	MOV [DI], AL	88:05	Move content of AL register with memory location in DI
041B	HLT	F4	

Input

	Forward	Reverse
0500	0A	09
0501	06	05
0502	05	06
0503	09	0A

EXPERIMENT-15

STEPPER MOTOR

Aim: To write and execute a program to implement stepper moter in 8086 kit

Address	Instruction	Opcode	Explanation
0400	MOV AL,80	B0:80	Move 80 to AL
0402	OUT 66,AL	E6:66	Out AL to port 66
0404	MOV CL,04	B1:04	Move 04 to CL register
0406	MOV BX,500	BB:00:05	Move 500 to BX register
0409	MOV AL,[BX]	8A:07	Move memory content of Bx to AL register
040B	OUT 60H,AL	E6:60	Out AL to port 60
040D	CALL 0415	E8:05:00	Call subroutine at 0415 frddag
0410	INC BX	43	Increment BX register
0411	LOOPNZ 0409	E0:F6	If zero flag is not set then loop to location 0409
0413	JMP 040A	EB:EF	Jump to location 040A
0415	PUSH CX	51	Push CX register to stack
0416	MOV CX,0FFFF	B9:FF:FF	Move OFFFF to CX register
0419	LOOPNZ 0419	E0:FE	If zero flag is not set then loop to location 0419
041B	POP CX	59	POP stack top to CX register
041C	RET	C3	Return from Subroutine

Result

Program was executed and output was obtained successfully

EXPERIMENT - 16

8255 INTERFACE PROGRAM

Aim: To write and execute a program to interface 8255 into 8286

Address	Instruction	Opcode	Explanation
0400	MOV AL,80	B0:80	Move 80 to AL register
0402	OUT 26, AL	E6:26	Out AL to port 26
0404	MOV AL,FF	B0:FF	Move FF to AL register
0406	NOT AL	F6:D0	Logical Not performed to AL
0408	OUT 20, AL	E6:20	Out AL to port 20
040A	OUT 22, AL	E6:22	Out AL to port 22
040C	OUT 24, AL	E6:24	Out AL to port 24
040E	MOV CX, FFFF	B9:FF:FF	Move FFFF to Cx register
0411	LOOPNZ 0411	E0:FE	If zero flag is not set then loop to location 0411
0413	JMP 0406	EB:F1	Jump to location 0406

Result

Program was executed and output obtained successfully