GROUP NO: - 8

GAURAV KUMAR, MOHD SALMAN, SHEIKH SHAHID

// sign with mantissa

// sign with expo

LXI H,4001H //initialize HL register pair

MVI D,02 //store temporary data in reg D

BEGIN: MVI B,05 // store temporary data in reg B

MVI C,05 // store temporary data in reg C

MOV A,M //copy content to accumulator from memory pointed by memory

pointer

LOOP: RAL //rotate accumulator left with carry

DCR B //decrease content in reg B by 1

JNZ LOOP //jump to LOOP if content in B is not 0

SWAP: JC COND2 //jump to COND2 if carry bit is 1

COND: RRC //rotate accumulator right without carry

DCR C //decrease accumulator content by 1

JNZ COND //jump to COND if content in C is not 0

JZ STORE //jump to STORE if content in C is 0

COND2: RRC //rotate accumulator right without carry

DCR C //decrease accumulator content by 1

JNZ COND2 //jump to COND2 if content in C is not 0

CMA //take complement of accumulator

INR A //increase accumulator content by 1

STORE: MOV M,A //copy to memory from accumulator

INR L //increase L by 1

INR L // increase L by 1

DCR D //decrease D by 1

JNZ BEGIN //jump to BEGIN if D is not 0

LXI H,4000H //again initialize HL pair

MVI B,02H //store 02 in B

MVI C,01H //store 01 in C

BEGIN2: MOV A,M //copy content from memory to accumulator

RAL //rotate accumulator left with carry

JNC SKIP //jump to SKIP if carry is 0

INR C //increase C by 1

RRC //rotate right without carry

MOV M,A //copy data to memory

INR L //increase L by 1

INR L //increase L by 1

DCR B //decrease B by 1

JNZ BEGIN2 //jump to BEGIN2 if B is not 0

JZ SKIP2 //jump to SKIP2 if B is 0

SKIP: RRC //rotate right without carry

MOV M,A //copy data to memory

INR L //increase L by 1

INR L //increase L by 1

DCR B //decrease B by 1

JNZ BEGIN2 //jump to BEGIN2 if B is not 0

JZ SKIP2 //jump to SKIP2 if B is 0

SKIP2: MOV A,C //copy content in C

STA 6000H //store accumulator

LXI H,4000H //point memory at 4000H

MOV B,M //copy content to B from memory

INR L //increase L by 1

MOV C,M //copy content to C from memory

INR L //increase L by 1

MOV D,M //copy content to D from memory

INR L //increase L by 1

MOV E,M //copy content to E from memory

MOV A,C //copy C to accumulator

ADD E //add E to accumulator

STA 7000H //store accumulator to 7000H

MOV E,B //copy B to E

MOV A,D //copy D to accumulator

MVI D,00 //remove data from D

MVI C,08 //copy 08 to C

LXI H,0000 //remove data from HL Pair

LOOP2: RRC //rotate right without carry

JNC SKIP3 //jump to SKIP3 if carry is not 0

DAD D //add DE reg pair to HL reg pair

SKIP3: XCHG //exchange HL pair with DE

DAD H //add HL pair to itself

XCHG //exchange HL pair with DE

DCR C

JNZ LOOP2

DAD H //add HL pair to itself

DAD H //add HL pair to itself

XCHG //exchange HL pair with DE

LXI H,6000H //point memory at 6000H

MOV C,M //copy content from memory to C

STC //set carry bit to 1

LAST: CMC //take complement of carry bit

DCR C //decrease reg C by 1

JNZ LAST //jump to LAST if C is not 0

XCHG //exchange HL pair with DE

MOV A,H //copy H to accumulator

RAR //rotate accumulator right with carry

MOV H,A //copy accumulator to H

MOV A,L //copy L to accumulator

RAR //rotate accumulator right with carry

MOV L,A //copy accumulator to L

SHLD 8000H //store H to 8001H and L to 8000H

//GIVING SIGN TO ADDED EXPO

EXPO SIGN: LXI H,7000 //load 7000H in HL reg pair

MVI D,05 //load 05 in reg D

MVI C,04 //load 04 in C

MOV A,M //copy data from memory to accumulator

STC //set carry

CMC //carry complement

RAL //rotate accumulator left through carry

JC END //jump to END if carry flag is set

RAR //rotate accumulator right with carry

MOV M,A //copy data from accumulator to memory

HLT //termination of program

END: RAR //rotate accumulator right with carry

CMA //take complement of accumulator

INR A //increase accumulator by 1

END2: STC //set carry

CMC //carry complement

RAL //rotate accumulator left through carry

DCR D //decrease reg D by 1

JNZ END2 //jump to END2 if D is not 0

STC //set carry

RAR //rotate accumulator right with carry

END3: STC //set carry

CMC //carry complement

RAR //rotate accumulator right with carry

DCR C //decrease reg C by 1

JNZ END3 //jump to END3 if C is not 0

MOV M,A //copy data from accumulator to memory

HLT //termination of program

# ORG 4000H

# DB 40H,01H,20H,0AH



