NAME: G.PALLAVI

REG NO: 192111601

COURSE NAME: COMPUTER ARCHITECTURE FOR CONTROL SIGNAL GENERATION

COURSE CODE: CSA1290

# EXP 1:

LDA 2050

MOV B,A

LDA 2051

ADD B

STA 2052

HLT

# EXP 2:

LDA 2050

MOV B,A

LDA 2051

SUB B

STA 2052

HLT

# EXP 3:

# LHLD 2500

# XCHG

# LHLD 2502

# MOV A,E

# ADD L

# MOV L,A

# MOV A,D

# ADC H

# MOV H,A

# SHLD 2504

# HLT

# EXP 4:

LHLD 2500

XCHG

LHLD 2502

MOV A,E

SUB L

MOV L,A

MOV A,D

ADC H

MOV H,A

SHLD 2504

HLT

# EXP 5

8 BIT MULTIPLICATION

MVI D,00

MVI A,00

LXI H,4150

MOV B.,M

INX H

MOV C,M

LOOP: ADD B

JNC NEXT

INR D

NEXT: DCR C

JNZ LOOP

STA 4152

MOV A,D

STA 4153

HLT

EXP 6:

8 BIT DIVISION

LXI H,1100

MOV B,M

MVI C,00

INX H

MOV A,M

LOOP: CMP B

JC SKIP

SUB B

INR C

JMP LOOP

SKIP: STA 1102

MOV A,C

STA 1103

HLT

# EXP 18

#include #include

int acum[100]={0} ;

void add(int acum[],int b[],int n);

int q[100],b[100];

int main()

{

int x,y;

printf("Enter the Number :");

scanf("%d%d",&x,&y);

int i=0;

while(x>0||y>0)

{

if(x>0)

{

q[i]=x%2;

x=x/2;

}

Else

{

q[i]=0;

}

if(y>0)

{

b[i]=y%2;

y=y/2;

}

else

{

b[i]=0;

}

i++;

}

int n=i;

int bc[50];

printf("\n");

for(i=0;i<=n;i++)

{

if(bc[i]==0)

{

bc[i]=1; i=n+2;

}

Else

{

bc[i]=0;

}

}

int l;

b[n]=0;

int k=n;

int n1=n+n-1;

int j,mi=n-1;

for(i=n;i!=0;i--)

{

for(j=n;j>0;j--)

{

acum[j]=acum[j-1];

}

acum[0]=q[n-1];

for(j=n-1;j>0;j--)

{

q[j]=q[j-1];

}

add(acum,bc,n+1);

if(acum[n]==1)

{

q[0]=0;

add(acum,b,n+1);

}

else

{

q[0]=1;

}

}

printf("\nQuoient : ");

for( l=n-1;l>=0;l--)

{

printf("%d",q[l]);

}

printf("\nRemainder : ");

for( l=n;l>=0;l--)

{

printf("%d",acum[l]);

}

return 0;

}

void add(int acum[],int bo[],int n)

{

int i=0,temp=0,sum=0;

for(i=0;i<n;i++)

{

sum=0;

um=acum[i]+bo[i]+temp;

If(sum==0)

{

acum[i]=0;

temp=0;

}

else if (sum==2)

{

acum[i]=0;

temp=1;

}

else if(sum==10

{

acum[i]=1;

temp=0;

}

else if(sum==3)

{

acum[i]=1;

temp=1;

}

}

}

# EXP 19

#include <stdio.h>

int main() {

float cachehit, cachemiss;

float cachehitratio;

printf("\n enter the total number of cache hits:");

scanf("%d",&cachehit);

printf("\n enter the number of cache misses:");

scanf("%d",&cachemiss);

cachehitratio=cachehit/(cachehit+cachemiss);

printf("\n Cache Hit Ratio: %f",cachehitratio);

printf("\n Cache Miss Ratio: %f",1-cachehitratio);

return 0;

# }EXP 21

#include<stdio.h>

#include<stdlib.h>

int main(){

int a[10],n,i;

printf("Enter the number to convert: ");

scanf("%d",&n);

for(i=0;n>0;i++)

{

a[i]=n%2;

n=n/2;

}

printf("\nBinary of Given Number is=");

for(i=i-1;i>=0;i--)

{

printf("%d",a[i]);

}

return 0;

}

# EXP 22

#include <stdio.h>

int main()

{

long decimalnum, remainder, quotient,octalnum=0;

int octalNumber[100], i = 1, j;

printf("Enter the decimal number: ");

scanf("%ld", &decimalnum);

quotient = decimalnum;

//Storing remainders until number is equal to zero while (quotient != 0)

{

octalNumber[i++] = quotient % 8;

quotient = quotient / 8;

}

//Converting stored remainder values in corresponding octal number

for (j = i - 1; j > 0; j--)

octalnum = octalnum\*10 + octalNumber[j];

printf("Equivalent octal value of decimal no %d is: %d ", decimalnum,octalnum);

return 0;

}

# EXP 23

#include <stdio.h>

void main()

{

// declaration of variables

int num, binary\_num, decimal\_num = 0, base = 1, rem;

printf (" Enter a binary number with the combination of 0s and 1s \n");

scanf (" %d", &num); // accept the binary number (0s and 1s)

binary\_num = num; // assign the binary number to the binary\_num variable

while ( num > 0)

{

rem = num % 10; /\* divide the binary number by 10 and store the remainder in rem

variable. \*/

decimal\_num = decimal\_num + rem \* base;

num = num / 10; // divide the number with quotient

base = base \* 2;

}

printf ( " The binary number is %d \t", binary\_num); // print the binary number

printf (" \n The decimal number is %d \t", decimal\_num); // print the decimal

}

# EXP 24

#include <stdio.h>

int main()

{

float cr;

int p,p1,i;

float cpu[5];

float cpi,ct,max;

int n=1000;

for(i=0;i<=4;i++)

{

cpu[5]=0;

}

printf("\n Enter the number of processors:");

scanf("%d",&p);

p1=p;

for(i=0;i<p;i++)

{

printf("\n Enter the Cycles per Instrcution of processor:");

scanf("%f",&cpi);

printf("\n Enter the clockrate in GHz:");

scanf("%f",&cr);

ct=1000\*cpi/cr;

printf("The CPU time is: %f",ct);

cpu[i]=ct;

}

max=cpu[0];

//printf("%f", max);

for(i=0;i<p1;i++)

{

if(cpu[i]<=max)

max=cpu[i];

}

printf("\n The processor has lowest Execution time is: %f ", max);

return 0;

}

# EXP 17

#include <stdio.h>

#include <math.h>

int a = 0,b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0};

int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};

int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0};

void binary(){

a1 = fabs(a);

b1 = fabs(b);

int r, r2, i, temp;

for (i = 0; i < 5; i++){

r = a1 % 2;

a1 = a1 / 2;

r2 = b1 % 2;

b1 = b1 / 2;

anum[i] = r;

anumcp[i] = r;

bnum[i] = r2;

if(r2 == 0){

bcomp[i] = 1;

}

if(r == 0){

acomp[i] =1;

}

}

//part for two's complementing

c = 0;

for ( i = 0; i < 5; i++){

res[i] = com[i]+ bcomp[i] + c;

if(res[i] >= 2){

c = 1;

}

else

c = 0;

res[i] = res[i] % 2;

}

for (i = 4; i >= 0; i--){

bcomp[i] = res[i];

}

//in case of negative inputs

if (a < 0){

c = 0;

for (i = 4; i >= 0; i--){

res[i] = 0;

}

for ( i = 0; i < 5; i++){

res[i] = com[i] + acomp[i] + c;

if (res[i] >= 2){

c = 1;

}

else

c = 0;

res[i] = res[i]%2;

}

for (i = 4; i >= 0; i--){

anum[i] = res[i];

anumcp[i] = res[i];

}

}

if(b < 0){

for (i = 0; i < 5; i++){

temp = bnum[i];

bnum[i] = bcomp[i];

bcomp[i] = temp;

}

}

}

void add(int num[]){

int i;

c = 0;

for ( i = 0; i < 5; i++){

res[i] = pro[i] + num[i] + c;

if (res[i] >= 2){

c = 1;

}

else{

c = 0;

}

res[i] = res[i]%2;

}

for (i = 4; i >= 0; i--){

pro[i] = res[i];

printf("%d",pro[i]);

}

printf(":");

for (i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}

void arshift(){//for arithmetic shift right

int temp = pro[4], temp2 = pro[0], i;

for (i = 1; i < 5 ; i++){//shift the MSB of product

pro[i-1] = pro[i];

}

pro[4] = temp;

for (i = 1; i < 5 ; i++){//shift the LSB of product

anumcp[i-1] = anumcp[i];

}

anumcp[4] = temp2;

printf("\nAR-SHIFT: ");//display together

for (i = 4; i >= 0; i--){

printf("%d",pro[i]);

}

printf(":");

for(i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}

void main(){

int i, q = 0;

printf("\t\tBOOTH'S MULTIPLICATION ALGORITHM");

printf("\nEnter two numbers to multiply: ");

printf("\nBoth must be less than 16");

//simulating for two numbers each below 16

do{

printf("\nEnter A: ");

scanf("%d",&a);

printf("Enter B: ");

scanf("%d", &b);

}while(a >=16 || b >=16);

printf("\nExpected product = %d", a \* b);

binary();

printf("\n\nBinary Equivalents are: ");

printf("\nA = ");

for (i = 4; i >= 0; i--){

printf("%d", anum[i]);

}

printf("\nB = ");

for (i = 4; i >= 0; i--){

printf("%d", bnum[i]);

}

printf("\nB'+ 1 = ");

for (i = 4; i >= 0; i--){

printf("%d", bcomp[i]);

}

printf("\n\n");

for (i = 0;i < 5; i++){

if (anum[i] == q){//just shift for 00 or 11

printf("\n-->");

arshift();

q = anum[i];

}

else if(anum[i] == 1 && q == 0){//subtract and shift for 10

printf("\n-->");

printf("\nSUB B: ");

add(bcomp);//add two's complement to implement subtraction

arshift();

q = anum[i];

}

else{//add ans shift for 01

printf("\n-->");

printf("\nADD B: ");

add(bnum);

arshift();

q = anum[i];

}

}

printf("\nProduct is = ");

for (i = 4; i >= 0; i--){

printf("%d", pro[i]);

}

for (i = 4; i >= 0; i--){

printf("%d", anumcp[i]);

}

}

# EXP 25

LDA 2500

MOV B,A

LDA 2501

STA 2500

MOV A,B

STA 2501

HLT

# EXP 20

LDA 3000

CMA

STA 3001

ADI 01

STA 3002

HLT