**Logic Gates Outputs Program**

#include<iostream>

using namespace std;

class Gates{

public:

int notGate(int input) {

if(input==0)

return 1;

return 0;

}

int andGate(int firstInput,int secondInput) {

if(firstInput==0 || secondInput==0)

return 0;

return 1;

}

int norGate(int firstInput,int secondInput) {

if(firstInput==0 && secondInput==0)

return 1;

return 0;

}

int xorGate(int firstInput,int secondInput) {

if(firstInput==secondInput)

return 0;

return 1;

}

int orGate(int firstInput, int secondInput) {

if(firstInput==1 || secondInput==1)

return 1;

return 0;

}

int nandGate(int firstInput, int secondInput) {

if(firstInput==1 && secondInput==1)

return 0;

return 1;

}

};

**Derived Class File**

#include "baseClassGates.cpp"

class halfAdder : public Gates {

public:

int haSum(int a, int b) {

return xorGate(a, b);

}

int haCarry(int a, int b) {

return andGate(a, b);

}

void haTruthTable() {

cout << "-----TRUTH TABLE-----\n\n" << endl;

cout << "A\t\t" << "B\t\t\t" << "SUM\t\tCARRY" <<endl;

cout << "0\t\t" << "0\t\t\t" << "0\t\t0" <<endl;

cout << "0\t\t" << "1\t\t\t" << "1\t\t0" <<endl;

cout << "1\t\t" << "0\t\t\t" << "1\t\t0" <<endl;

cout << "1\t\t" << "1\t\t\t" << "1\t\t1" <<endl;

cout << "------------------------" << endl;

}

};

class fullAdder : public halfAdder {

public:

int sum\_temp,c1\_temp,c2\_temp;

int faSum(int a, int b, int c) {

return haSum(c,haSum(a,b));

}

int faCarry(int a, int b, int c) {

return orGate(haCarry(a,b),haCarry(haSum(a,b),c));

}

void faTruthTable() {

cout << "-----TRUTH TABLE-----\n\n" << endl;

cout << "A\t\t" << "B\t\t" << "C\t\t\t" << "SUM\t\tCARRY" <<endl;

cout << "0\t\t" << "0\t\t" << "0\t\t\t" << "0\t\t0" <<endl;

cout << "0\t\t" << "0\t\t" << "1\t\t\t" << "1\t\t0" <<endl;

cout << "0\t\t" << "1\t\t" << "0\t\t\t" << "1\t\t0" <<endl;

cout << "0\t\t" << "1\t\t" << "1\t\t\t" << "0\t\t1" <<endl;

cout << "1\t\t" << "0\t\t" << "0\t\t\t" << "1\t\t0" <<endl;

cout << "1\t\t" << "0\t\t" << "1\t\t\t" << "0\t\t1" <<endl;

cout << "1\t\t" << "1\t\t" << "0\t\t\t" << "0\t\t1" <<endl;

cout << "1\t\t" << "1\t\t" << "1\t\t\t" << "1\t\t1" <<endl;

cout << "------------------------------------" << endl;

}

};

class rippleAdder : public fullAdder {

public:

int raCarry(int a, int b, int carryIn) {

return faCarry(a,b,carryIn);

}

int raSum(int a, int b, int carryIn) {

return faSum(a,b,carryIn);

}

};

**Main function file**

#include "derivedClasses.cpp"

int main() {

int choice,loop=1,a,b,c;

char tt\_choice;

cout << "\n Welcome to Our Program\n\n";

while(loop==1) {

cout << "Which operation do you want to execute\n" << endl;

cout << "Half Adder - 1" << endl;

cout <<"Full Adder - 2" <<endl;

cout <<"Ripple Adder - 3" << endl;

cout << "Press the following keys as per your need : ";

cin >> choice;

if(choice==1) {

cout << "\n Half Adder Program";

cout << "\n Enter the value of A : ";

cin >> a;

cout << "\n Enter the value of B : ";

cin >> b;

if(a>=0 && a<=1 && b>=0 && b<=1) {

halfAdder HA;

cout << "\n The Sum is :" << HA.haSum(a,b);

cout << "\n The Carry is :" << HA.haCarry(a,b);

cout << "\n Do You want Truth Table of Half Adder? Then Press 'y' : ";

cin >> tt\_choice;

if(tt\_choice=='y') {

HA.haTruthTable();

}

}

else {

cout<< "\n Value of A & B should be either 1 or 0 \n";

}

cout << "\n Press 1 to continue or press any other key to exit : ";

cin >> choice;

loop = choice;

}

if(choice==2) {

cout << "\n Full Adder Program";

cout << "\n Enter the value of A : ";

cin >> a;

cout << "\n Enter the value of B : ";

cin >> b;

cout << "\n Enter the value of C : ";

cin >> c;

if(a>=0 && a<=1 && b>=0 && b<=1) {

fullAdder FA;

cout << "\n The Sum is :" << FA.faSum(a,b,c);

cout << "\n The Carry is : " << FA.faCarry(a,b,c);

cout << "\n Do You want Truth Table of Half Adder? Then Press 'y' : ";

cin >> tt\_choice;

if(tt\_choice == 'y') {

FA.faTruthTable();

}

}

else {

cout<< "\n Value of A & B should be either 1 or 0 \n";

}

cout << "\n Press 1 to continue or press any other key to exit : ";

cin >> loop;

}

if(choice == 3) {

cout << "/n Ripple Adder Program";

int a0,a1,a2,a3,b0,b1,b2,b3,s0,s2,s3,s1,c1,c2,c3,c0,ci;

cout << "\n Enter the A value : "; cin >> a3 >> a2 >> a1 >> a0;

cout << "\n Enter the B value : "; cin >> b3 >> b2 >> b1 >> b0;

cout << "\n Enter Carry Value "; cin >> ci;

rippleAdder RA;

//for carry values

c0 = RA.raCarry(a0,b0,ci);

c1 = RA.raCarry(a1,b1,c0);

c2 = RA.raCarry(a2,b2,c1);

c3 = RA.raCarry(a3,b3,c2);

//for Sum values

s0 = RA.raSum(a0,b0,ci);

s1 = RA.raSum(a1,b1,c0);

s2 = RA.raSum(a2,b2,c1);

s3 = RA.raSum(a3,b3,c2);

//Output results

cout << "The Ripple Sum is : " << "\t" << s3 << "\t" << s2 << "\t" << s1 << "\t" << s0;

cout << "\n"<< "The Carry Out Value is : " << c3;

cout << "\n Press 1 to continue or press any other key to exit : ";

cin >> loop;

}

}

return 0;

}