**B31DG-EMBEDDED SOFTWARE**

**ASSIGNMENT- 1.**

**C++ PROGRAM**

The program has 4 possible system modes:

1. If the mode is 1, and **Switch2** is 0, the normal waveform is generated. If **Switch2** is 1, 3 pulses are removed from the pulse block (i.e. **c - 3** pulses in the block) until **Switch2** is set back to 0.
2. If the mode is 2, and **Switch2** is 0, the normal waveform is generated. If **Switch2** is 1, the inverted form of the complete **SignalA** waveform is generated (from the largest pulse to the shortest) until **Switch2** is set back to 0.
3. If the mode is 3, and **Switch2** is 0, the normal waveform is generated. If **Switch2** is 1, an extra 3 pulses are inserted into the pulse block (i.e. **c + 3** pulses in the block) until **Switch2** is set back to 0.
4. If the mode is 4, and **Switch2** is 0, the normal waveform is generated. If **Switch2** is 1, **d** and **b** times are halved until **Switch2** is set back to 0.

**PROGRAM**

#include <iostream>

#include <vector>

const int PULSE\_CYCLE\_LENGTH = 50;// initialize the pulse cycle length 50us

struct WaveformParameters {

int a, b, c, d;

};

class Signal {

// use public and private functions for subroutines

public:

signal(int mode, int switch1, int switch2, WaveformParameters params): mode\_(mode), switch1\_(switch1), switch2\_(switch2), params\_(params) {}

std::vector<int> GenerateSignalA() {

std::vector<int> signal; // If the mode is 1, and **Switch2** is 0, the normal waveform is generated.

if (switch1\_ == 0) {

int num\_pulses = params\_.c;

if (switch2\_ == 1) {// If **Switch2** is 1 use switch case operation for various modes

switch (mode\_) {

case 1://mode-1-3 pulses are removed from the pulse block

num\_pulses -=3;

break;

case 2://mode2= Invert signal

num\_pulses -=num\_pulses;

break;

case 3://mode3= extra pulses are inserted into the pulse block

num\_pulses += 3;

break;

case 4:// mode 4=Half d and b time

num\_pulses =num\_pulses/2;

break;

}}

for (int i = 0; i < num\_pulses; ++i) {

signal.push\_back(params\_.a + i \* PULSE\_CYCLE\_LENGTH);

signal.push\_back(params\_.b);

}

signal.push\_back(params\_.d);

}

return signal;

}

private://function call the inputs and parameters from the subroutine

int mode\_, switch1\_, switch2\_;

waveformParameters params\_;

};

int main() {

int mode, switch1, switch2;

waveformParameters params;

std::cin >> mode >> switch1 >> switch2 >> params.a >> params.b >> params.c >>

params.d;

signal signal(mode, switch1, switch2, params);

std::vector<int> signalA = signal.GenerateSignalA();

std::vector<int> signalB = signal.GenerateSignalB();

// Print signalA and signalB

return 0;

}

#include <iostream>

using namespace std;//initialize the input parameters, modes, and switches.

void generateWaveform(int switch1, int switch2, int a, int b, int c, int d, int mode) {

//use the switch statement to get values for switches and modes for generating waveform.

if (switch1 == 0) {//if switch1=0 the stream of pulses is enabled.

cout << "Stream of pulses enabled." << endl;

}

else {//if switch1=0 the stream of pulses is disabled.

cout << "Stream of pulses disabled." << endl;

}//if switch2=0 the same waveform is generated.

if (switch2 == 0) {

cout << "Running in normal mode." << endl;

}

else {//if switch2=0 the new waveform is generated.

cout << "Running in new waveform cycle mode." << endl;

//use switch case statement for generating pulses of Signal A and B waveform

switch (mode) {

case 1:

if (switch2 == 1) {

c = c - 3;

cout << "3 pulses removed from the pulse block." << endl;

}break;

case 2:

if (switch2 == 1) {

cout << "Generating inverted form of complete Signal A waveform." << endl;

}break;

case 3:

if (switch2 == 1) {

c = c + 3;

cout << "3 pulses inserted into the pulse block." << endl;

}break;

case 4:

if (switch2 == 1) {

b = b / 2;d = d / 2;

cout << "Time for b and d halved." << endl;

}break;

default:

cout << "Invalid mode. Running in normal mode." << endl;

break;

}}}

int main() {//initialize the switch inputs and parameters

int switch1, switch2, a, b, c, d, mode;//call the input switch values to print the output modes

cout << "Enter the value of switch1 (0/1): ";

cin >> switch1;

cout << "Enter the value of switch2 (0/1): ";

cin >> switch2;

cout << "Enter the value of a: ";

cin >> a;

cout << "Enter the value of b: ";

cin >> b;

cout << "Enter the value of c: ";

cin >> c;

cout << "Enter the value of d: ";

cin >> d;

cout << "Enter the value of mode (1/2/3/4): ";

cin >> mode;

generateWaveform (switch1, switch2, a, b, c, d, mode);

return 0;

}

#include <iostream>

#include <string>

using namespace std;

//initialize the alphabet in ascending order from a to m

int number\_of\_alphabet(char alphabet) {

if (alphabet >= 'a' && alphabet <= 'm') {

return alphabet - 'a' + 1;//initialize the alphabet in descending order from n to z

} else if (alphabet >= 'n' && alphabet <= 'z') {

return 13 - (alphabet - 'n');

} else if (alphabet >= 'A' && alphabet <= 'M') {

return alphabet - 'A' + 1;

} else {

return 13 - (alphabet - 'N');

}}

int rem(int num, int den) {

return num % den;

}

int main() {//get the first and last name inputs from the console

string first\_name, last\_name;

cout << "Enter your first name: ";

cin >> first\_name;

cout << "Enter your last name: ";

cin >> last\_name;//calculate the parameter values from the first 5 values of the last name

int length = last\_name.length();

if (length < 5) {

last\_name += last\_name.substr(0, 5-length);

}

int a = number\_of\_alphabet(last\_name[0]) \* 100;

int b = number\_of\_alphabet(last\_name[1]) \* 100;

int c = number\_of\_alphabet(last\_name[2]) + 4;

int d = number\_of\_alphabet(last\_name[3]) \* 500;

int mode = rem((last\_name[4] <= 'Z' ? last\_name[4] - 'A' + 1 : last\_name[4] - 'a' + 1), 4) + 1;

//Print the parameters and modes for waveform

cout << "The parameter set for the name " << " is " << a << " μS, " << b << " μS, " << c << ", " << d << " μS, and mode " << mode << "." << endl;

return 0;

}

**OUTPUT -RESULT**

Shape

Description automatically generatedText

Description automatically generated

**EXPLANATION**

This program generates two waveform signals, **SignalA** and **SignalB**.

The waveform consists of repeated cycles, and **SignalB** should be 50 microseconds.

There are two switch inputs to generate the signals, **Switch1** and **Switch2**, and four parameters, **a**, **b**, **c**, and **d**.

If **Switch1** is 0, it enables the stream of pulses; if **Switch1** is 1, it disables the stream of pulses. Similarly, if **Switch2** is 0, the program runs in normal mode, and if **Switch2** is 1, it runs with a new waveform cycle.

The new waveform cycle parameters are defined as follows: **a** is the width of the first pulse, the 2nd pulse is **a + 50 microseconds**, the 3rd pulse is **a + 100 microseconds**, and so on. **b** is the width of the space between pulses (all spaces are the same). **c** is the number of pulses in a block, and **d** is the space between pulse blocks.

This is a C++ program that takes the user's first and last name as input, calculates a set of parameters for the name and outputs it to the console.

The program uses the function "number\_of\_alphabet" to calculate the position of an alphabet in the English alphabet (from 1 to 26) and maps it to a number.

The program then concatenates the first 5 characters of the last name and calculates the values of the parameters a, b, c, d and mode by using the first 5 characters of the last name as input. The value of a and b are calculated as the position of the first and second characters of the last name multiplied by 100. The value of c is calculated as the position of the third character of the last name plus 4. The value of d is calculated as the position of the fourth character of the last name multiplied by 500. The value of mode is calculated as the remainder of the position of the fifth character of the last name divided by 4 plus 1.

Finally, the program outputs the values of the parameters to the console.

**FIRST NAME: GAUTAMI**

**LAST NAME: ALAGARSAMY**

**Parameters calculation : A=1,L=12,A=1,G=7 and A=1.**

1. Parameter 'a' calculated as first letter\*100 = 1\*100 = 100=>**a = 100**
2. Parameter 'b' calculated as second letter\*100 = 12\*100 = 1200 => **b = 1200**
3. Parameter 'c' calculated as third letter+4 = 1+4=5=> **c = 5**
4. Parameter 'd' calculated as fourth letter\*500 = 7\*500 = 3500=> **d = 3500**

Mode is calculated as rem(fifth letter/4) + 1 = rem(1/4) + 1 = 2. Hence, we use **mode 2.**