

# Código

## main.cpp

```
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <cstdio>
#include "registros.h"

using namespace std;

int main(void){
    srand(time(NULL));
    archivoRegistros r = archivoRegistros();

    //1.- Reset
    r.operacionAsincrona(true);
    r.get();
    getchar();

    //2.- Banco[1] = 89
    r.operacionSincrona(89,1,0,0,true,false,false,false);
    r.get();
    getchar();

    //3.- Banco[2] = 72
    r.operacionSincrona(72,2,0,0,true,false,false,false);
    r.get();
    getchar();

    //4.- Banco[3] = 123
    r.operacionSincrona(123,3,0,0,true,false,false,false);
    r.get();
    getchar();

    //5.- Banco[4] = 53
    r.operacionSincrona(53,4,0,0,true,false,false,false);
    r.get();
    getchar();

    //6.- Leer Banco[1] y Banco[2]
    r.operacionAsincrona(false,1,2);
    r.get();

    //7.- Leer Banco[3] y Banco[4]
    r.operacionAsincrona(false,3,4);
```

```

    getchar();

    //8.- Banco[2] = Banco[1] << 3
    r.operacionSincrona(0,2,1,3,true,true,true,false);
    r.get();
    getchar();

    //9.- Banco[4] = Banco[3] >> 5
    r.operacionSincrona(0,4,3,5,true,true,false,false);
    r.get();
    getchar();

    //10.- Leer Banco[1] y Banco[2]
    r.operacionAsincrona(false,1,2);
    getchar();

    //11.- Leer Banco[3] y Banco[4]
    r.operacionAsincrona(false,3,4);
    getchar();

    //12.- get
    r.get();
    getchar();

    //1.- Reset
    r.operacionAsincrona(true);
    r.get();
    getchar();
    return 0;
}

```

## registros.h

```

using namespace std;

class archivoRegistros {
public:
    unsigned short writeData;
    unsigned char writeReg;
    unsigned char readReg1;
    unsigned char readReg2;
    unsigned char shamt;
    bool WR;
    bool SHE;
    bool DIR;
    bool CLR;

```

```

        unsigned short readData1;
        unsigned short readData2;
        int Banco[16];

        ~archivoRegistros();
        archivoRegistros();
        void set();
        void get();
        void operacionSincrona(unsigned short writeData, unsigned char writeReg,
        unsigned char readReg1, unsigned char shamt, bool WR, bool SHE, bool DIR, bool
CLR);
        void operacionAsincrona(bool CLR);
        void operacionAsincrona(bool CLR,
                unsigned char readReg1, unsigned char readReg2);
private:
        void clr();
};

```

## registros.cpp

```

#include "registros.h"
#include <ctime>
#include <cstdlib>
#include <iostream>
#include <cstdio>

using namespace std;

archivoRegistros::archivoRegistros(){
        writeData = 0;
        writeReg = 0;
        readReg1 = 0;
        readReg2 = 0;
        shamt = 0;
        WR = false;
        SHE = false;
        DIR = false;
        CLR = false;
        readData1 = 0;
        readData2 = 0;
}

archivoRegistros::~archivoRegistros(){
}

```

```

void archivoRegistros::set(){
    for(char i = 0; i < 16; i++) {
        Banco[i] = rand();
    }
}

void archivoRegistros::get(){
    printf("Banco\n");
    for(char i = 0; i < 16; i++) {
        printf("-----\n");
        printf("%2d :%16d\n",i, Banco[i]);
    }
    printf("\n");
}

void archivoRegistros::clr(){
    for(char i = 0; i < 16; i++) {
        Banco[i] = 0;
    }
}

void archivoRegistros::operacionSincrona(unsigned short writeData, unsigned char
writeReg,
unsigned char readReg1, unsigned char shamt, bool WR, bool SHE, bool DIR, bool
CLR){
    if(CLR){
        clr();
    }else if(WR){
        if(SHE){
            if(DIR){
                Banco[writeReg] = Banco[readReg1] << shamt;
            }else{
                Banco[writeReg] = Banco[readReg1] >> shamt;
            }
        }else{
            Banco[writeReg] = writeData;
        }
    }
}

void archivoRegistros::operacionAsincrona(bool CLR){
    clr();
}

void archivoRegistros::operacionAsincrona(bool CLR,
unsigned char readReg1, unsigned char readReg2){

```

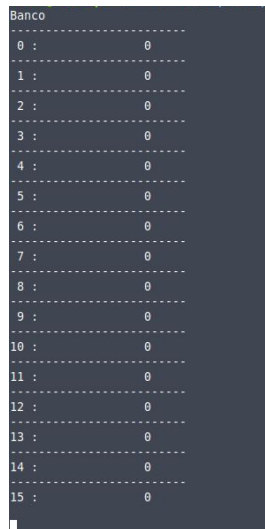
```

        if(CLR){
            clr();
        }else{
            readData1 = Banco[readReg1];
            readData2 = Banco[readReg2];
            printf("readData1: %d\n",readData1);
            printf("readData2: %d\n",readData2);
        }
    }
}

```

## Pruebas

### 1.- Reset

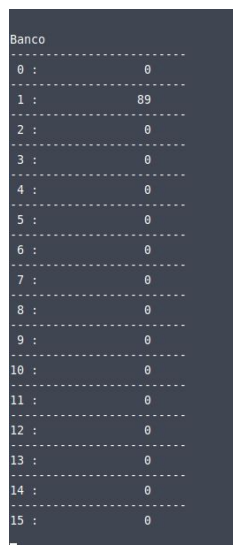


```

Banco
-----
0 :      0
-----
1 :      0
-----
2 :      0
-----
3 :      0
-----
4 :      0
-----
5 :      0
-----
6 :      0
-----
7 :      0
-----
8 :      0
-----
9 :      0
-----
10 :     0
-----
11 :     0
-----
12 :     0
-----
13 :     0
-----
14 :     0
-----
15 :     0
-----

```

### 2.- Banco[1] = 89



```

Banco
-----
0 :      0
-----
1 :     89
-----
2 :      0
-----
3 :      0
-----
4 :      0
-----
5 :      0
-----
6 :      0
-----
7 :      0
-----
8 :      0
-----
9 :      0
-----
10 :     0
-----
11 :     0
-----
12 :     0
-----
13 :     0
-----
14 :     0
-----
15 :     0
-----

```

### 3.- Banco[2] = 72

Banco		
0 :		0
1 :		89
2 :		72
3 :		0
4 :		0
5 :		0
6 :		0
7 :		0
8 :		0
9 :		0
10 :		0
11 :		0
12 :		0
13 :		0
14 :		0
15 :		0

4.- Banco[3] = 123

Banco		
0 :		0
1 :		89
2 :		72
3 :		123
4 :		0
5 :		0
6 :		0
7 :		0
8 :		0
9 :		0
10 :		0
11 :		0
12 :		0
13 :		0
14 :		0
15 :		0

5.- Banco[4] = 53

Banco		
0 :		0
1 :		89
2 :		72
3 :		123
4 :		53
5 :		0
6 :		0
7 :		0
8 :		0
9 :		0
10 :		0
11 :		0
12 :		0
13 :		0
14 :		0
15 :		0

6.- Leer Banco[1] y Banco[2]

```

Banco
-----
0 :          0
-----
1 :          89
-----
2 :          72
-----
3 :         123
-----
4 :          53
-----
5 :           0
-----
6 :           0
-----
7 :           0
-----
8 :           0
-----
9 :           0
-----
10 :          0
-----
11 :          0
-----
12 :          0
-----
13 :          0
-----
14 :          0
-----
15 :          0
-----
readData1: 89
readData2: 72

```

7.- Leer Banco[3] y Banco[4]

```

0 :          0
-----
1 :          89
-----
2 :          72
-----
3 :         123
-----
4 :          53
-----
5 :           0
-----
6 :           0
-----
7 :           0
-----
8 :           0
-----
9 :           0
-----
10 :          0
-----
11 :          0
-----
12 :          0
-----
13 :          0
-----
14 :          0
-----
15 :          0
-----
readData1: 89
readData2: 72
readData1: 123
readData2: 53

```

8.- Banco[2] = Banco[1] << 3

```

Banco
-----
0 :          0
-----
1 :          89
-----
2 :         712
-----
3 :         123
-----
4 :          53
-----
5 :           0
-----
6 :           0
-----
7 :           0
-----
8 :           0
-----
9 :           0
-----
10 :          0
-----
11 :          0
-----
12 :          0
-----
13 :          0
-----
14 :          0
-----
15 :          0
-----

```

9.- Banco[4] = Banco[3] >> 5

```
Banco
-----
0 :      0
-----
1 :      89
-----
2 :     712
-----
3 :     123
-----
4 :       3
-----
5 :       0
-----
6 :       0
-----
7 :       0
-----
8 :       0
-----
9 :       0
-----
10 :      0
-----
11 :      0
-----
12 :      0
-----
13 :      0
-----
14 :      0
-----
15 :      0
-----
```

10.- Leer Banco[1] y Banco[2]

```
Banco
-----
0 :      0
-----
1 :      89
-----
2 :     712
-----
3 :     123
-----
4 :       3
-----
5 :       0
-----
6 :       0
-----
7 :       0
-----
8 :       0
-----
9 :       0
-----
10 :      0
-----
11 :      0
-----
12 :      0
-----
13 :      0
-----
14 :      0
-----
15 :      0
-----

readData1: 89
readData2: 712
-----
```

11.- Leer Banco[3] y Banco[4]

```
0 :      0
-----
1 :      89
-----
2 :     712
-----
3 :     123
-----
4 :       3
-----
5 :       0
-----
6 :       0
-----
7 :       0
-----
8 :       0
-----
9 :       0
-----
10 :      0
-----
11 :      0
-----
12 :      0
-----
13 :      0
-----
14 :      0
-----
15 :      0
-----

readData1: 89
readData2: 712
-----

readData1: 123
readData2: 3
-----
```



12.- Get()

Banco		
0 :		0
1 :		89
2 :		712
3 :		123
4 :		3
5 :		0
6 :		0
7 :		0
8 :		0
9 :		0
10 :		0
11 :		0
12 :		0
13 :		0
14 :		0
15 :		0

13.- Reset

Banco		
0 :		0
1 :		0
2 :		0
3 :		0
4 :		0
5 :		0
6 :		0
7 :		0
8 :		0
9 :		0
10 :		0
11 :		0
12 :		0
13 :		0
14 :		0
15 :		0