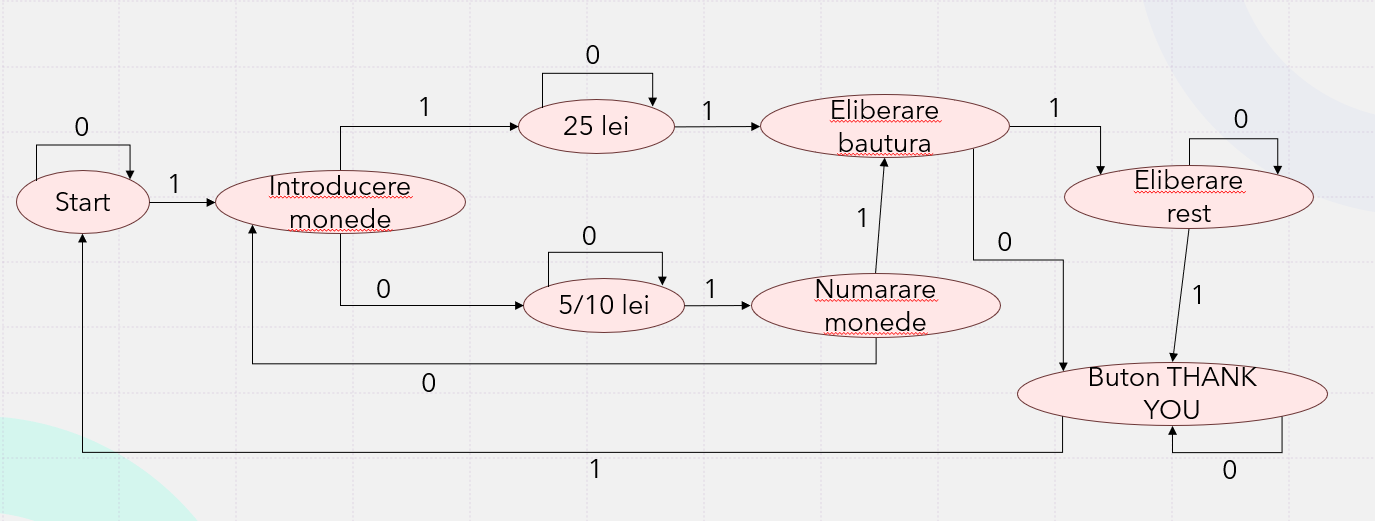
**Tema 2**

Movileanu Raluca 331CB

A diagram of a company

Description automatically generated1.Diagrama de stări (format MEALY) pentru mașina care eliberează băutura răcoritoare.

2.Codul C++ folosit pentru a descrie implementarea hardware dar și pentru simulare

**vending\_machine.cpp:**

**#include** "vending\_machine.h"

**#include** <iostream>

**#define** **PRICE** 25

**void** **vending\_machine**(**ap\_uint**<5> moneda, *State* &stare\_curenta, **bool** &bautura\_eliberata, **bool** &buton\_TY\_apasat, **int** &rest, **int** &suma) {

// Verificam daca suntem in starea BUTON\_THANK\_YOU si butonul a fost apasat

**if** (stare\_curenta == *BUTON\_THANK\_YOU* && buton\_TY\_apasat) {

std::cout << "Multumim!\n";

// Eliberam restul daca este necesar

**if** (rest > 0) {

std::cout << "Rest eliberat: " << rest << " lei.\n";

}

// Resetam starea automatului pentru urmatorul client

suma = 0;

rest = 0;

stare\_curenta = *START*;

bautura\_eliberata = **false**;

buton\_TY\_apasat = **false**;

**return**;

}

// Procesam moneda introdusa

**if** (moneda > 0) {

stare\_curenta = *INTRODUCERE\_MONEDA*;

suma += moneda;

// Setam starea corespunzatoare in functie de valoarea monedei

**if** (moneda == 25) {

stare\_curenta = *MONEDA\_25\_LEI*;

} **else** **if** (moneda == 5 || moneda == 10) {

stare\_curenta = *MONEDA\_5\_10\_LEI*;

}

}

// Verificam daca suma acumulata este suficienta pentru a elibera bautura

**if** (suma >= PRICE && !bautura\_eliberata) {

bautura\_eliberata = **true**;

rest = suma - PRICE;

stare\_curenta = (rest > 0) ? *ELIBERARE\_REST* : *ELIBERARE\_BAUTURA*;

std::cout << "Bautura racoritoare a fost eliberata.\n";

// Anuntam clientul daca trebuie sa astepte restul

**if** (rest > 0) {

std::cout << "Asteptati restul: " << rest << " lei.\n";

} **else** {

std::cout << "Multumim! Asteptam urmatorul client.\n";

}

}

// Dupa eliberarea bauturii, setam starea pentru butonul THANK YOU

**if** (stare\_curenta == *ELIBERARE\_BAUTURA*) {

stare\_curenta = *BUTON\_THANK\_YOU*;

}

}

vending\_machine.h:

**#ifndef** VENDING\_MACHINE\_H\_

**#define** **VENDING\_MACHINE\_H\_**

**#include** <ap\_int.h>

**enum** *State* {

*START*,

*INTRODUCERE\_MONEDA*,

*MONEDA\_25\_LEI*,

*MONEDA\_5\_10\_LEI*,

*NUMARARE\_MONEDA*,

*ELIBERARE\_BAUTURA*,

*ELIBERARE\_REST*,

*BUTON\_THANK\_YOU*

};

**void** **vending\_machine**(**ap\_uint**<5> moneda, *State* &stare\_curenta, **bool** &bautura\_eliberata, **bool** &buton\_TY\_apasat, **int** &rest, **int** &suma);

**#endif**

vending\_machine\_tb.cpp:

**#include** "vending\_machine\_tb.h"

**#include** "vending\_machine.h"

**#include** <iostream>

**const** **char**\* **state\_to\_string**(*State* state) {

**switch** (state) {

**case** *START*: **return** "START";

**case** *INTRODUCERE\_MONEDA*: **return** "INTRODUCERE\_MONEDA";

**case** *MONEDA\_25\_LEI*: **return** "MONEDA\_25\_LEI";

**case** *MONEDA\_5\_10\_LEI*: **return** "MONEDA\_5\_10\_LEI";

**case** *NUMARARE\_MONEDA*: **return** "NUMARARE\_MONEDA";

**case** *ELIBERARE\_BAUTURA*: **return** "ELIBERARE\_BAUTURA";

**case** *ELIBERARE\_REST*: **return** "ELIBERARE\_REST";

**case** *BUTON\_THANK\_YOU*: **return** "BUTON\_THANK\_YOU";

**default**: **return** "STARE NECUNOSCUTA";

}

}

**void** **vending\_machine\_test**() {

*State* stare\_curenta = *START*;

**bool** bautura\_eliberata = **false**;

**bool** buton\_TY\_apasat = **false**;

**ap\_uint**<5> moneda;

**int** rest = 0;

**int** suma = 0;

std::cout<<"Test1: cand introducem 25 lei\n";

moneda = 25;

**vending\_machine**(moneda, stare\_curenta, bautura\_eliberata, buton\_TY\_apasat,rest,suma);

std::cout<<"\nTest2: cand introducem o bancnota de 5 lei si doua de 10 lei\n";

bautura\_eliberata = **false**;

stare\_curenta = *START*;

rest = 0;

suma = 0;

moneda = 5;

std::cout<<"Introducem 5 lei\n";

**vending\_machine**(moneda, stare\_curenta, bautura\_eliberata, buton\_TY\_apasat,rest,suma);

std::cout << "Stare dupa introducerea monedei de 5 lei: " << **state\_to\_string**(stare\_curenta) << std::**endl**;

std::cout << "Bautura eliberata: " << bautura\_eliberata << std::**endl**;

std::cout<<"Introducem inca 10 lei\n";

moneda = 10;

**vending\_machine**(moneda, stare\_curenta, bautura\_eliberata, buton\_TY\_apasat,rest,suma);

std::cout << "Stare dupa introducerea monedei de 10 lei: " << **state\_to\_string**(stare\_curenta) << std::**endl**;

std::cout << "Bautura eliberata: " << bautura\_eliberata << std::**endl**;

std::cout<<"Mai introducem inca 10 lei\n";

**vending\_machine**(moneda, stare\_curenta, bautura\_eliberata, buton\_TY\_apasat,rest,suma);

bautura\_eliberata = **false**;

std::cout<<"\nTest3: daca nu s-ar fi eliberat mai devreme bautura si mai introduceam 10 lei\n";

**vending\_machine**(moneda, stare\_curenta, bautura\_eliberata, buton\_TY\_apasat,rest,suma);

std::cout << "Stare dupa introducerea monedei de 10 lei: " << **state\_to\_string**(stare\_curenta) << std::**endl**;

}

**int** **main**() {

**vending\_machine\_test**();

**return** 0;

}

vending\_machine\_tb.h:

**#ifndef** VENDING\_MACHINE\_TB\_H\_

**#define** **VENDING\_MACHINE\_TB\_H\_**

**#include** "vending\_machine.h"

**void** **vending\_machine\_test**();

**#endif**

3.Rezultatul Simularii.

INFO: [SIM 2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSIM start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

INFO: [SIM 4] CSIM will launch GCC as the compiler.

Compiling ../../../vending\_machine\_tb.cpp in debug mode

Generating csim.exe

Test1: cand introducem 25 lei

Bautura racoritoare a fost eliberata.

Multumim! Asteptam urmatorul client.

Test2: cand introducem o bancnota de 5 lei si doua de 10 lei

Introducem 5 lei

Stare dupa introducerea monedei de 5 lei: MONEDA\_5\_10\_LEI

Bautura eliberata: 0

Introducem inca 10 lei

Stare dupa introducerea monedei de 10 lei: MONEDA\_5\_10\_LEI

Bautura eliberata: 0

Mai introducem inca 10 lei

Bautura racoritoare a fost eliberata.

Multumim! Asteptam urmatorul client.

Test3: daca nu s-ar fi eliberat mai devreme bautura si mai introduceam 10 lei

Bautura racoritoare a fost eliberata.

Asteptati restul: 10 lei.

Stare dupa introducerea monedei de 10 lei: ELIBERARE\_REST

INFO: [SIM 1] CSim done with 0 errors.

INFO: [SIM 3] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSIM finish \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A black rectangular object with a black background

Description automatically generatedA screenshot of a computer

Description automatically generated 4. Raportul VITIS HLS care demonstrează că avem un circuit secvențial

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated 5. Raportul VITIS HLS care arată resursele hardware folosite în cazul utilizării unui FPGA Artix 7