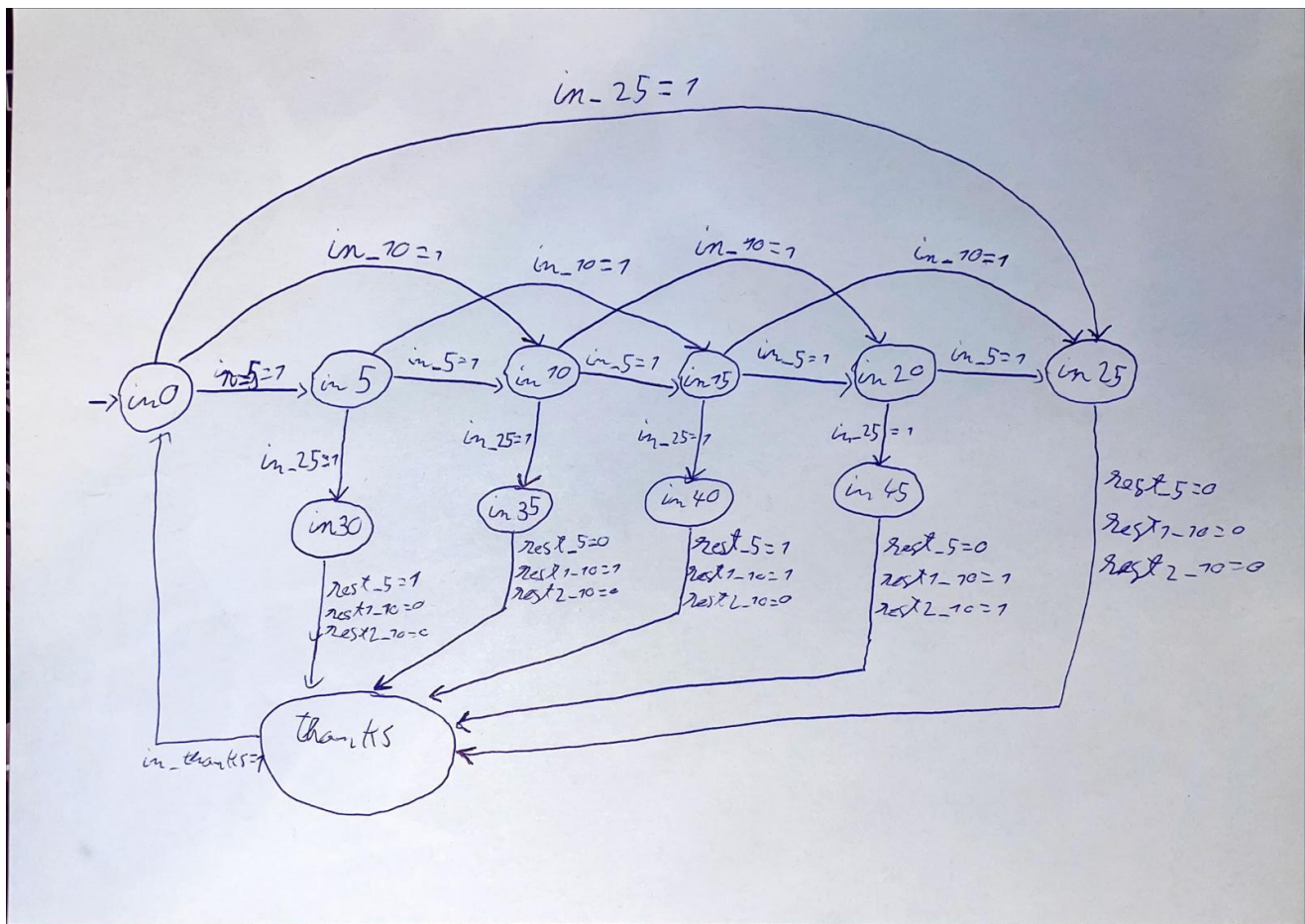


Tema 2 – Mereu Ioan Flaviu 333CC

1. Diagrama de stari:



2. Cod CPP + rezultate

```

1 #include <iostream>
2 #include "Vending_machine.h"
3
4
5 using namespace std;
6
7 void money_in(bool in_5, bool in_10, bool in_25) {
8
9     bool out_drink, in_thanks, rest_5, rest1_10, rest2_10;
10
11     vending_machine(in_5, in_10, in_25, in_thanks, out_drink, rest_5, rest1_10, rest2_10);
12
13     cout << "intra 5 = " << in_5 << ", intra 10 = " << in_10 << ", intra 25 = " << in_25 << ", in_thanks = " << in_thanks << endl;
14     cout << "out_drink = " << out_drink << ", rest bancnota 5 = " << rest_5 << ", rest bancnota 10 = " << rest1_10 << ", rest banc
15     cout << endl;
16 }
17
18 void press_thanks() {
19
20     bool out_drink, in_thanks = 1, rest_5, rest1_10, rest2_10;
21
22     vending_machine(0, 0, 0, in_thanks, out_drink, rest_5, rest1_10, rest2_10);
23
24     cout << "in_5 = 0, in_10 = 0, in_25 = 0, in_thanks = " << in_thanks << endl << "S-a apasat THANK YOU!!!" << endl;
25     cout << "out_drink = " << out_drink << ", rest bancnota 5 = " << rest_5 << ", rest bancnota 10 = " << rest1_10 << ", rest banc
26     cout << endl;
27 }
28
29 void vending_machine(bool in_5, bool in_10, bool in_25, bool in_thanks, bool& out_drink, bool& rest_5, bool& rest1_10, bool& rest2_
30
31 #pragma HLS INTERFACE ap_none port=in_5
32 #pragma HLS INTERFACE ap_none port=in_10
33 #pragma HLS INTERFACE ap_none port=in_25
34 #pragma HLS INTERFACE ap_none port=in_thanks
35 #pragma HLS INTERFACE ap_none port=out_drink
36 #pragma HLS INTERFACE ap_none port=rest_5
37 #pragma HLS INTERFACE ap_none port=rest1_10

```

```

37 #pragma HLS INTERFACE ap_none port=rest1_10
38 #pragma HLS INTERFACE ap_none port=rest2_10
39 #pragma HLS INTERFACE ap_ctrl_none port=return
40
41 static State current_state;
42 State next_state = current_state;
43
44 bool out_drink_ok;
45 bool out_5, out1_10, out2_10;
46
47
48 switch (current_state) {
49
50 case in0:
51     out_drink_ok = 0;
52     out_5 = 0; out1_10 = 0; out2_10 = 0;
53
54     if (in_5)
55         next_state = in5;
56     else if (in_10)
57         next_state = in10;
58     else if (in_25)
59         next_state = in25;
60     else
61         next_state = in0;
62     break;
63
64 case in5:
65     out_drink_ok = 0;
66     out_5 = 0; out1_10 = 0; out2_10 = 0;
67
68     if (in_5)
69         next_state = in10;
70     else if (in_10)
71         next_state = in15;
72     else if (in_25)
73         next_state = in30;
74

```

Synthesis Summary(solution1) Vending_machine.cpp x vending_machine_csim.log

```
73     next_state = in30;
74     else
75     next_state = in5;
76     break;
77
78 case in10:
79     out_drink_ok = 0;
80     out_5 = 0; out1_10 = 0; out2_10 = 0;
81
82     if (in_5)
83     next_state = in15;
84     else if (in_10)
85     next_state = in20;
86     else if (in_25)
87     next_state = in35;
88     else
89     next_state = in10;
90     break;
91
92 case in15:
93     out_drink_ok = 0;
94     out_5 = 0; out1_10 = 0; out2_10 = 0;
95
96     if (in_5)
97     next_state = in20;
98     else if (in_10)
99     next_state = in25;
100    else if (in_25)
101    next_state = in40;
102    else
103    next_state = in15;
104    break;
105
106 case in20:
107     out_drink_ok = 0;
108     out_5 = 0; out1_10 = 0; out2_10 = 0;
109
```

Synthesis Summary(solution1) Vending_machine.cpp x vending_machine_csim.log

```
109
110     if (in_5)
111     next_state = in25;
112     else if (in_10)
113     next_state = in30;
114     else if (in_25)
115     next_state = in45;
116     else
117     next_state = in20;
118     break;
119
120 case in25:
121     out_drink_ok = 1;
122     out_5 = 0; out1_10 = 0; out2_10 = 0;
123
124     next_state = thanks;
125     break;
126
127 case in30:
128
129     out_drink_ok = 1;
130     out_5 = 1; out1_10 = 0; out2_10 = 0;
131
132     next_state = thanks;
133     break;
134
135 case in35:
136
137     out_drink_ok = 1;
138     out_5 = 0; out1_10 = 1; out2_10 = 0;
139
140     next_state = thanks;
141     break;
142
143 case in40:
144
145     out_drink_ok = 1;
```

Synthesis Summary(solution1) Vending_machine.cpp x vending_machine_csimpl.log

```
144
145     out_drink_ok = 1;
146     out_5 = 1; out1_10 = 1; out2_10 = 0;
147
148     next_state = thanks;
149     break;
150
151 case in45:
152
153     out_drink_ok = 1;
154     out_5 = 0; out1_10 = 1; out2_10 = 1;
155
156     next_state = thanks;
157     break;
158
159 case thanks:
160
161     out_drink_ok = 0;
162     out_5 = 0; out1_10 = 0; out2_10 = 0;
163
164     if (in_thanks == 1)
165         next_state = in0;
166     else
167         next_state = thanks;
168     break;
169
170 default:
171     return;
172 }
173
174 current_state = next_state;
175
176 rest_5 = out_5;
177 rest1_10 = out1_10;
178 rest2_10 = out2_10;
179 out_drink = out_drink_ok;
180
```

Synthesis Summary(solution1) Vending_machine.cpp vending_machine_csimpl.log Vending_machine_tb.cpp x

```
1 #include <iostream>
2 #include "Vending_machine_tb.h"
3
4 using namespace std;
5
6 int main() {
7
8     int status = 0;
9
10    cout << "Test 1: " << endl;
11    money_in(0, 1, 0); // 10
12    money_in(0, 1, 0); // 10
13    money_in(0, 1, 0); // 10
14    money_in(0, 0, 0); // Dispense drink and change.
15    press_thanks(); // Press thank you button.
16
17    cout << endl << "Test 2: " << endl;
18    money_in(0, 1, 0); // 10
19    money_in(0, 0, 1); // 25
20    money_in(0, 0, 0); // Dispense drink and change.
21    press_thanks(); // Press thank you button.
22
23    return status;
24 }
```

Console x Errors Warnings Guidance Properties Man Pages Git Repositories Modules/Loops

Vitis HLS Console

```
intra 5 = 0, intra 10 = 0, intra 25 = 0, in_thanks = 0
out_drink = 1, rest bancnota 5 = 1, rest bancnota1 10 = 0, rest bancnota2 10 = 0

in_5 = 0, in_10 = 0, in_25 = 0, in_thanks = 1
S-a apasat THANK YOU!!!
out_drink = 0, rest bancnota 5 = 0, rest bancnota1 10 = 0, rest bancnota2 10 = 0

Test 2:
intra 5 = 0, intra 10 = 1, intra 25 = 0, in_thanks = 0
out_drink = 0, rest bancnota 5 = 0, rest bancnota1 10 = 0, rest bancnota2 10 = 0

intra 5 = 0, intra 10 = 0, intra 25 = 1, in_thanks = 0
out_drink = 0, rest bancnota 5 = 0, rest bancnota1 10 = 0, rest bancnota2 10 = 0

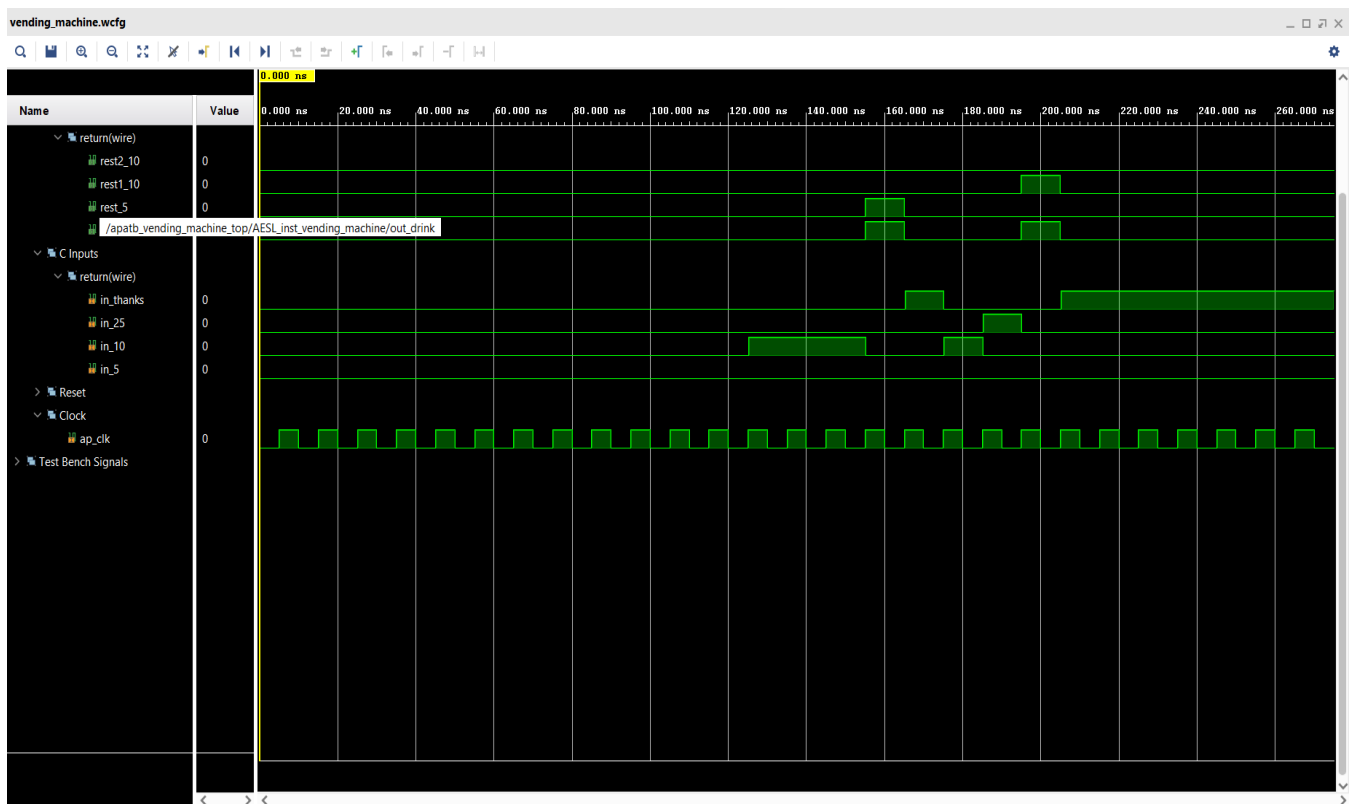
intra 5 = 0, intra 10 = 0, intra 25 = 0, in_thanks = 0
```

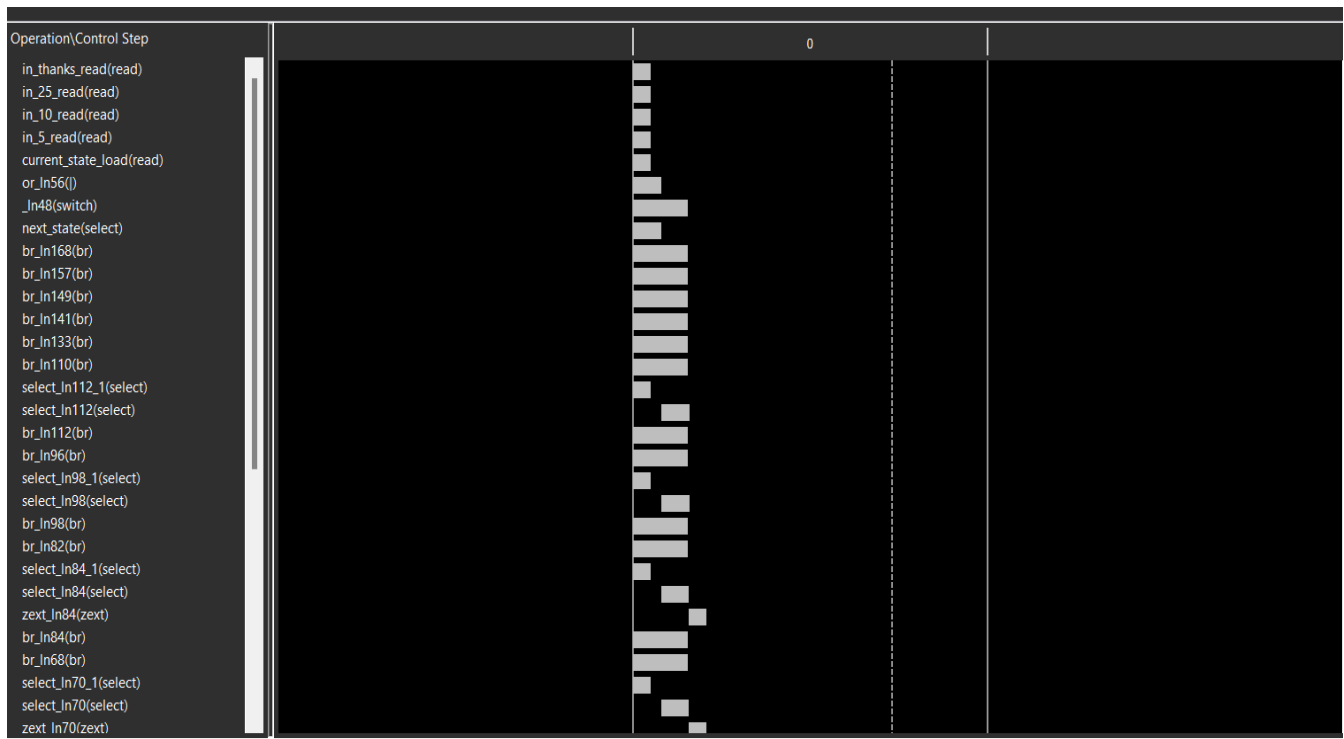
In primul test se introduc bancnote de 10,10,10 si da rest 5.

In al doilea test se introduc 10,25 si da rest 10.

3. Raport care demonstreaza ca circuitul este secvential:

Formele de unda :





4. Raportul care arată resursele hardware folosite în cazul utilizării unui FPGA Artix 7

Synthesis Summary Report of 'vending_machine'

General Information

Date: Mon Feb 12 01:07:33 2024

Version: 2021.1 (Build 3247384 on Thu Jun 10 19:36:33 MDT 2021)

Project: Tema_2_APM

Solution: solution1 (Vivado IP Flow Target)

Product family: artix7

Target device: xc7a100t-csg324-2L

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	3.182 ns	2.70 ns

Performance & Resource Estimates

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM	
vending_machine				-	0	0.0		-	1	-	no	0	0	5	147	0

HW Interfaces

REGISTER

Interface	Mode	Bitwidth
in_10	ap_none	1
in_25	ap_none	1
in_5	ap_none	1
in_thanks	ap_none	1
out_drink	ap_none	1
rest1_10	ap_none	1
rest2_10	ap_none	1
rest_5	ap_none	1

ConsoleErrorsWarningsGuidancePropertiesMan PagesGit RepositoriesModules/Loops

Vitis HLS Console