

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ  
КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ  
АПРАТУРИ

КУРСОВА РОБОТА

З дисципліни: Автоматизація проектування цифрових пристроїв

На тему: Розробити систему керування для торговельного автомату «Усе по 30»

Студентки 3 курсу групи ДК-91

Котеленець Ю.С.

Напрямок підготовки: Телекомунікації та  
Радіотехніка

Керівник: ст.вик. Антонюк О.І.

Національна оцінка: \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії: \_\_\_\_\_ ст.вик. Антонюк О.І. \_\_\_\_\_  
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2021 рік

## ЗМІСТ

1. Завдання до курсової роботи.....	3
2. Вступ.....	4
3. РОЗДІЛ 1.....	6
3.1. Структурна схема пристрою.....	6
4. РОЗДІЛ 2.....	8
4.1. Розробка пристрою.....	8
4.2. Тестування пристрою.....	12
5. Висновок.....	16
6. Список літератури.....	18
7. Додаток 1.....	19
8. Додаток 2.....	23

## ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ

Розробити систему керування для торгівельного автомату «Усе по 30».

Автомат має можливість продати один з чотирьох напоїв (назви оберіть самостійно) за однаковою ціною – 30 копійок. Автомат може приймати монети трьох номіналів – 5, 10 та 25 копійок. Після завершення внесення суми, автомат підраховує решту, яку необхідно повернути, та видає обраний товар. Внесені кошти та решта повинні відображатися для покупця. Якщо коштів не вистачає – формується сигнал «Замало коштів». Початкова кількість товару – по 5 примірників. Після продажу товару автомат зменшує його кількість. При відсутності товару формується відповідне повідомлення та повертається внесена сума.

- Lab assistants demand a new soda machine for the 6.111 lab. You design the FSM controller.
- All selections are \$0.30.
- The machine makes change. (Dimes and nickels only.)
- Inputs: limit 1 per clock
  - Q - quarter inserted
  - D - dime inserted
  - N - nickel inserted
- Outputs: limit 1 per clock
  - DC - dispense can
  - DD - dispense dime
  - DN - dispense nickel



Рис.1

## ВСТУП

Завдання полягає у тому, що потрібно розробити торговий автомат напоїв, але спершу варто поговорити про те, що таке торговий автомат, актуальність даного виробу, як він працює.

Торговий автомат — пристрій, що в автоматичному режимі (без оператора) здійснює процес обміну товару на гроші чи грошові еквіваленти.

Наявність торгових автоматів дуже важлива, тому що, по-перше, це зручний доступ до продуктів. Автомати зазвичай звикли роздавати цукерки, напої, продукти харчування та інші витратні матеріали, які не потребують для продажу присутність особи. Ці машини задовольняють потреби споживачів у будь-який момент. Враховуючи темпи, якими сьогодні працює світ, це так важливо мати машини, які видають те, що потрібно споживачам. Ці типи машин віддають перевагу багатьом трейдерам через безліч переваг, які вони мають.

Торговий автомат допомагає компаніям збільшити джерела доходу. У світлі того факту, що люди стають все більш зайнятими, зростає попит на швидку їжу. Тому підприємства шукають способи задовольнити цей попит, розміщуючи машини в різних місцях. Торговельні автомати дають споживачам доступ до своїх улюблених продуктів, перебуваючи в дорозі. Це допомагає їм заощадити час. Підприємства мають багато виграти від торгових автоматів. Ці машини допомагають їм заощадити значну суму грошей у вигляді вартості робочої сили, яка часто потрібна, щоб зробити їхню продукцію доступною для споживачів. Підприємства зробили все, щоб встановити машини, які приймають різні типи платежів, включаючи кредитні картки. Це дає можливість більшій кількості споживачів робити покупки.

Інноваційні підприємства шукають способи скористатися перевагами цих машин. В освітніх закладах популярний вендинговий бізнес. Машини дають можливість студентам отримувати напої та закуски зручно та за конкурентоспроможними цінами. Школа може збільшити свій дохід, встановивши машини у стратегічних місцях на території школи. Щоб

забезпечити найкращі результати, важливо забезпечити, щоб студенти отримували здорову поживну їжу за допомогою апаратів.

Отже, можна зробити маленький висновок, що вендинговий бізнес, на даний час, все більше і більше розвивається. Оскільки це має великі переваги, як для споживача, так і для підприємства. Для споживача перш за все, це цілодобова можливість придбати товар, без спілкування з іншими людьми. Для підприємства це економія витрат на зарплатню для персоналу, та цілодобовий дохід.

## РОЗДІЛ 1

## 3.1. Структурна схема пристрою

Структурна схема розробленого пристрою знаходиться на рис.1.1.

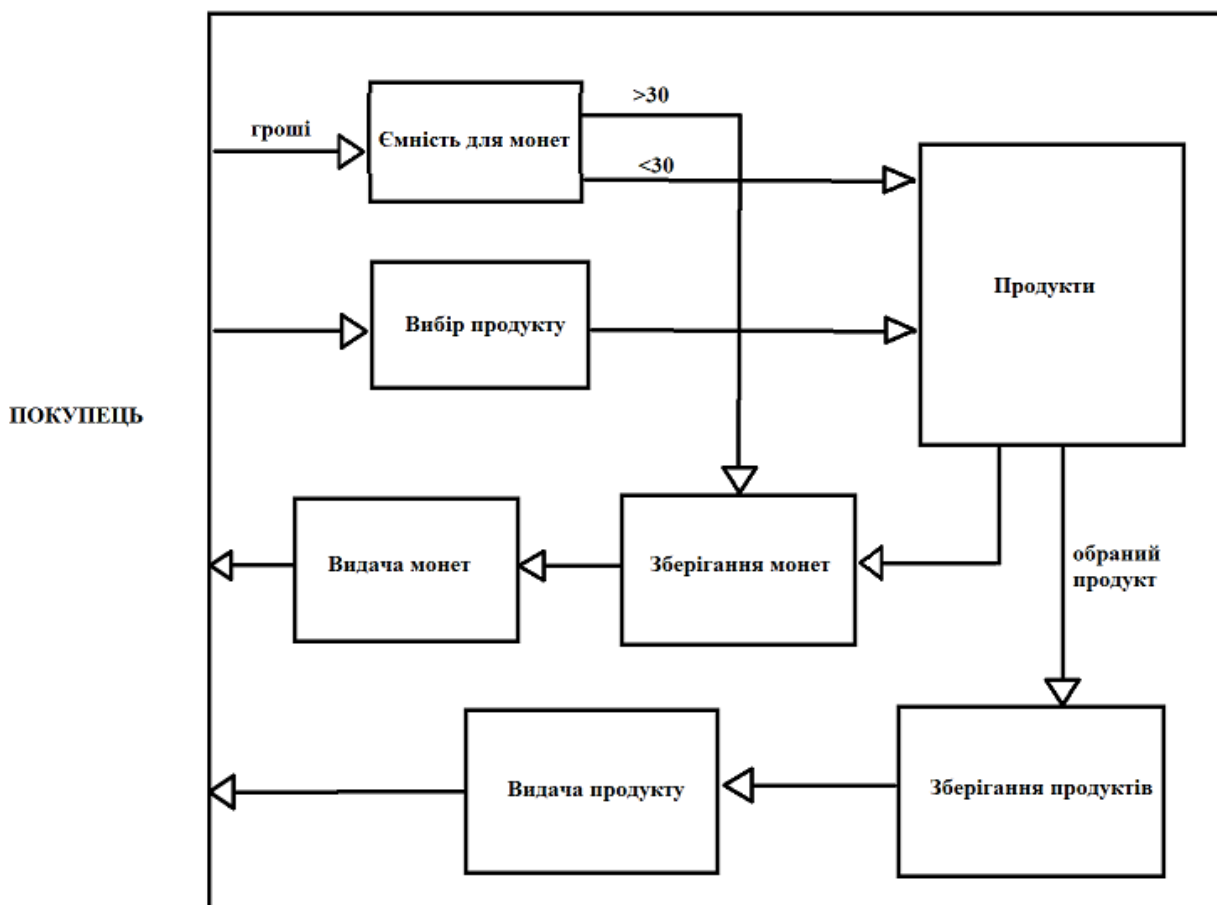


Рис.3.1. Структурна схема торгового автомата

На структурній схемі можна побачити складові торгового автомата та як вони між собою пов'язані.

Отже, для запуску роботи в торговельному автоматі необхідно, щоб покупець вніс кошти, а саме 30 копійок, але при цьому автомат приймає монети лише з номіналом в 5, 10 та 25 копійок. Після внесення коштів, вони потрапляють до місткості для монет, і на цьому моменті може бути два варіанти подальших дій.

Перший варіант, якщо коштів менше ніж 30, а це ціна одного напою, то гроші потраплять до блоку, який відповідає за зберігання монет, а після автомат поверне внесені кошти покупцю, оскільки їх недостатньо для покупки будь-якого з напоїв. Також покупець отримає відповідне повідомлення, про недостатню кількість коштів внесену до автомату.

Другий варіант, якщо коштів достатньо, тобто тридцять або навіть більш як тридцять. Тоді внесені кошти, разом з сигналом про обраний споживачем напій перейдуть до блоку продукти, який внесені кошти перемістить до місткості, яка зберігає їх та якщо це необхідно, то поверне решту. Сигнал з обраним продуктом перейде до місткості, яка відповідає за зберігання продуктів, з даної місткості вже обраний товар піде на видачу клієнту.

Всього торговий автомат має чотири напої, але в обмеженій кількості, кожного напою лише по п'ять екземплярів. Після покупки одного з напоїв, його стає на один менше та коли даний напій закінчиться, то покупець при виборі напою отримає сигнал, що його немає в наявності та отримає внесені кошти назад.

## РОЗДІЛ 2

### 4.1. Розробка пристрою

Для розробки пристрою я буду використовувати середовище ModelSim.

Спочатку варто проаналізувати більш детально вхідні та вихідні сигнали. Вхідними сигналами будуть гроші, які повинен вносити покупець, вибір напою, який теж визначає споживач, а їх всього чотири (вода, лимонад, газована вода, спрайт). Також обов'язково необхідний тактовий сигнал (clk) та сигнал скидання (reset).

```
input [1:0] coin,
input RESET, clk,
input in_soda, in_water, in_sprite, in_lemon,
```

Рис.4.1.1. Вхідні сигнали

Сигнал coin буде 2-розрядний, тому що автомат приймає три різновиди монет: 5, 10, 25 і нам буде достатньо двох розрядів, щоб визначити цю змінну (01 – 5 коп., 10 – 10 коп., 11 – 25 коп.).

На виході повинні бути сигнали напоїв, які видаються покупцю, відповідно вода, лимонад, газована вода, спрайт (out\_soda, out\_water, out\_sprite, out\_lemon). Також вихідним сигналом повинна бути решта, яка повертається покупцеві при внесенні коштів більше та менше ніж 30. Позаяк в умові задачі вказано, що кожного напою по п'ять екземплярів, то необхідно створити вихідний сигнал, який буде відповідати за кількість екземплярів (left\_soda, left\_water, left\_sprite, left\_lemon) і коли вони всі закінчатся, то користувач повинен отримати відповідний сигнал про відсутність товару (сигнал NONE). Також якщо грошей буде недостатньо, то користувач отримає повідомлення, що недостатньо коштів (сигнал NOT\_ENOUGH\_CASH).

```
output reg out_soda, out_water, out_sprite, out_lemon, NOT_ENOUGH_CASH, NONE,
output reg [2:0] left_soda, left_water, left_sprite, left_lemon,
output reg [15:0] left, cash
```

Рис.4.1.2. Вихідні сигнали

Далі необхідно задати параметром напої, які буду дорівнювати 30-ти:

```
parameter soda = 30, water = 30, sprite = 30, lemon = 30;
```

Рис.4.1.3.



Наступним кроком буде створення циклу для сигналу coin:

```
always @(posedge clk)
begin
    case (coin)
        1: cash = cash + 5;
        2: cash = cash + 10;
        3: cash = cash + 25;
        default: cash = cash;
    endcase
end
```

Рис.4.1.4.

У цьому циклі ми задаємо, що якщо змінна coin в положенні 1, то змінна cash, тобто гроші буде дорівнювати  $cash + 5$ . Якщо в положенні 2, то  $cash + 10$ , а для 3-го положення  $cash + 25$  відповідно. За замовчування, якщо жодна з даних умов не виконується, то cash не змінюється. Даний цикл працює по кожному активному передньому фронту тактового сигналу.

Наступним кроком я створила вже основний цикл. Спочатку за допомогою оператора умови (if) задала параметри для вихідних сигналів при подаванні сигналу скидування (reset).

```
if (RESET)
begin
    out_soda = 0;
    out_water = 0;
    out_sprite = 0;
    out_lemon = 0;
    NOT_ENOUGH_CASH = 0;
    NONE = 0;
    left = 0;
    cash = 0;
    left_soda = 5;
    left_water = 5;
    left_sprite = 5;
    left_lemon = 5;
end
```

Рис.4.1.5. При умові, що подається сигнал reset

Як видно з цієї частини коду, якщо сигнал скидування (reset) встановлено в один, то автомат не буде видавати ніякого напою, решти та інших вихідних сигналів. А кількість примірників відновиться до п'яти.

Якщо ж скидування не встановлене, то за допомогою оператора if я перевіряю чи грошей більше або дорівнює 30, тобто чи достатньо коштів для

покупки напою. Якщо недостатньо – покупець отримує сигнал NOT\_ENOUGH\_CASH, тобто недостатньо грошей.

```

        if (cash >= 30)
else
        NOT_ENOUGH_CASH = 1;

```

Рис.4.1.6.

Якщо введена сума більше або дорівнює тридцяти, то в даному операторі починається ще один оператор умови, який перевіряє, який саме напій обрав покупець. Наприклад, якщо покупець уніс достатньо коштів та хоче отримати такий напій, як вода, то умова перевіряє чи справді вхідний сигнал води дорівнює одиниці, якщо ні – на виході сигнал out\_water буде в нулі. Якщо ж все-таки покупець хоче придбати воду і ми цьому впевнились за допомогою попередньої перевірки, то ми повинні впевнитися, що напій вода має достатньо екземплярів, через те, що за умови у нас їх всього п'ять. Якщо кількість примірників води закінчилася, то споживач отримає відповідний сигнал про відсутність обраного напою. Якщо ж є хоч один екземпляр, то автомат видає обраний продукт, відраховує від кількості екземплярів води один, бо кількість примірників зменшилась на один та повертає покупцю решту, якщо це необхідно.

```

if (in_water == 1)
begin
    if (left_water < 1)
        begin
            NONE = 1;
            left = cash;
            cash = 0;
        end
    else begin
        out_water = 1;
        left_water = left_water - 1;
        left = cash - water;
        cash = 0;
        NONE = 0;
        NOT_ENOUGH_CASH = 0;
    end
end
else begin
    out_water = 0;

```

Рис.4.1.7.

Останній цикл буде повторюється для кожного з напоїв. Повний текст написаного коду можна переглянути в розділі Додаток 1.

## 4.2. Тестування пристрою

Щоб перевірити роботу розробленого приладу в середовищі ModelSim необхідно розробити тестовий файл. Повний текст тестового файлу можна переглянути в розділі Додаток 2.

В тестовому файлі я прописала всі можливі ситуації при роботі з торговим автоматом. Тобто всі можливі варіанти вибору напою покупцем. Також ситуацію при якій закінчаться всі екземпляри напоїв та повернення решти покупцеві.

Розробка тестового файлу:

Спочатку необхідно задати вхідні та вихідні параметри з відповідним типом. Тип reg для даних, які ми будемо змінювати та тип wire для інших.

```
reg [1:0] coin;
reg RESET, clk;
reg in_soda, in_water, in_sprite, in_lemon;
wire out_soda, out_water, out_sprite, out_lemon, NOT_ENOUGH_CASH, NONE;
wire [2:0] left_soda, left_water, left_sprite, left_lemon;
wire [15:0] left, cash;
```

Рис.4.2.1.

Наступним кроком потрібно з'єднати провідні порти та сигнали, тобто зазначити прототипи створених сигналів у відповідному модулі та у тестовому.

```
course FSM_controller (.coin(coin),
    .in_soda(in_soda),
    .in_water(in_water),
    .in_sprite(in_sprite),
    .in_lemon(in_lemon),

    .left_soda(left_soda),
    .left_water(left_water),
    .left_sprite(left_sprite),
    .left_lemon(left_lemon),

    .out_soda(out_soda),
    .out_water(out_water),
    .out_sprite(out_sprite),
    .out_lemon(out_lemon),

    .NOT_ENOUGH_CASH(NOT_ENOUGH_CASH),
    .left(left),
    .NONE(NONE),
    .cash(cash),
    .RESET(RESET),
    .clk(clk) );
```

Рис.4.2.2. Прототипи відповідно до основного модуля

Далі створимо одноразовий цикл, щоб задати сигнали скидування (RESET) та тактовий сигнал (clk).

```

initial
begin
    RESET = 1;
    #20 RESET = 0;
    clk = 0;
    forever #10 clk = ~clk;
end

```

Рис.4.2.3.

У цьому циклі ми встановлюємо сигнал скидування на двадцять наносекунд в положення один, а потім в нуль до кінця тестування. Тактовий сигнал буде встановлюватися в один кожні двадцять наносекунд.

Після я створила ще один цикл вже для зміни таких параметрів, як напої та гроші. Спочатку необхідно встановити всі вхідні сигнали напоїв в нуль. Далі змінюємо значення coin від нуля до трьох, цією дією ми будемо додавати гроші в автомат, то якщо спочатку coin в нулі, то у нас нічого не додається до змінної гроші, потім coin в першому положенні, це значить, що до змінної гроші додається п'ять копійок, якщо coin в другому – десять копійок, якщо в третьому – двадцять п'ять копійок.

```

coin = 0;
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;

```

Рис.4.2.4. Зміна положень сигналу coin

Також необхідно змінювати вхідні сигнали напоїв, я робила це по черзі кожен напій, до закінчення всіх примірників, а потім встановлювала наступний напій в одиницю, доки не закінчатся примірники всіх напоїв. Приклад зміни вхідного сигналу:

```

#50 in_soda = 1;
#30 in_soda = 0;

```

Рис.4.2.5. Зміна вхідного сигналу in\_soda

## Результат моделювання:

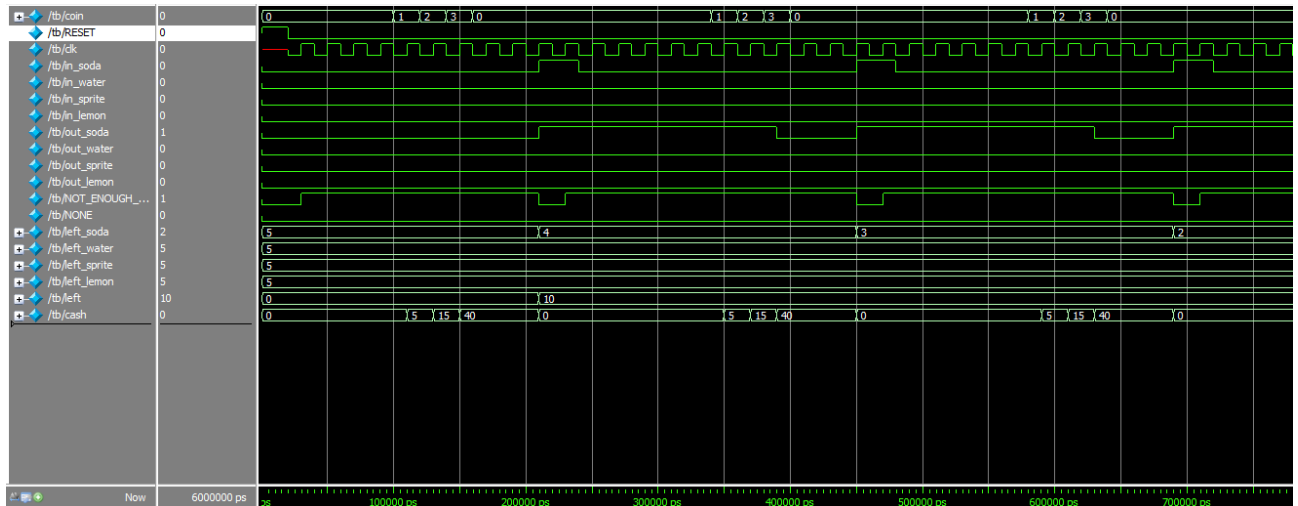


Рис.4.2.1. Результати моделювання при виборі напою soda

З моделювання можна побачити, що пристрій правильно працює, з кожним наступним сигналом `in_soda` кількість екземплярів зменшується на один та при потраплянні в автомат суми більшої ніж тридцять, покупець отримує решту.

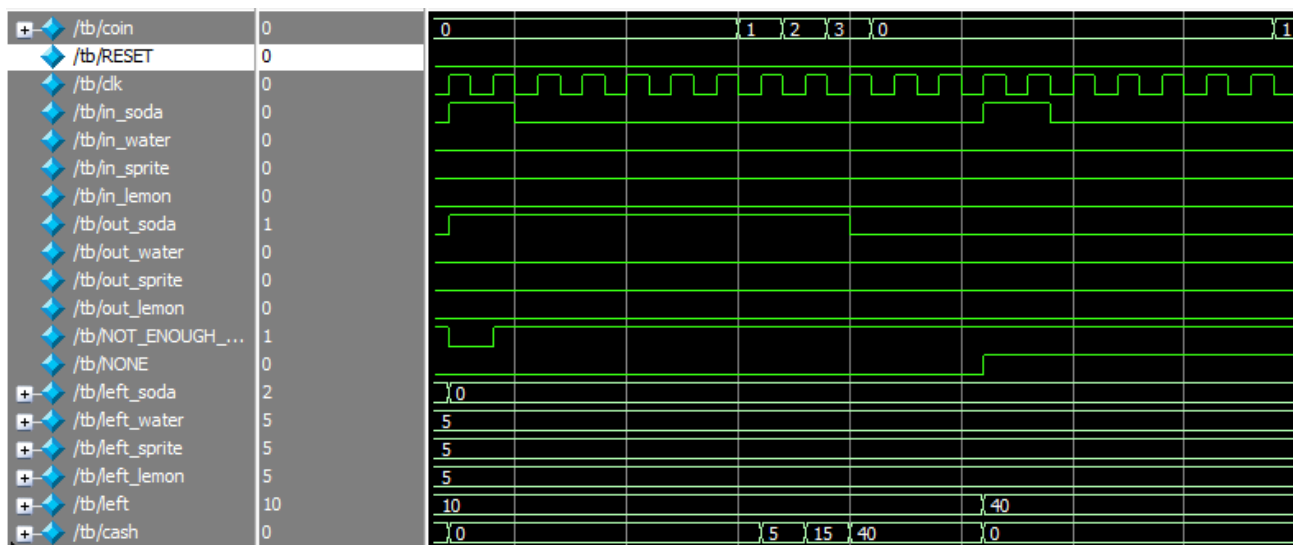


Рис.4.2.2. Екземплярів недостатньо

На даному фото можна побачити ситуацію, коли закінчується відповідний напій в автоматі. Сигнал, який відповідає за повідомлення про відсутність напою встановлюється в 1 та клієнт отримує всі внесені кошти.

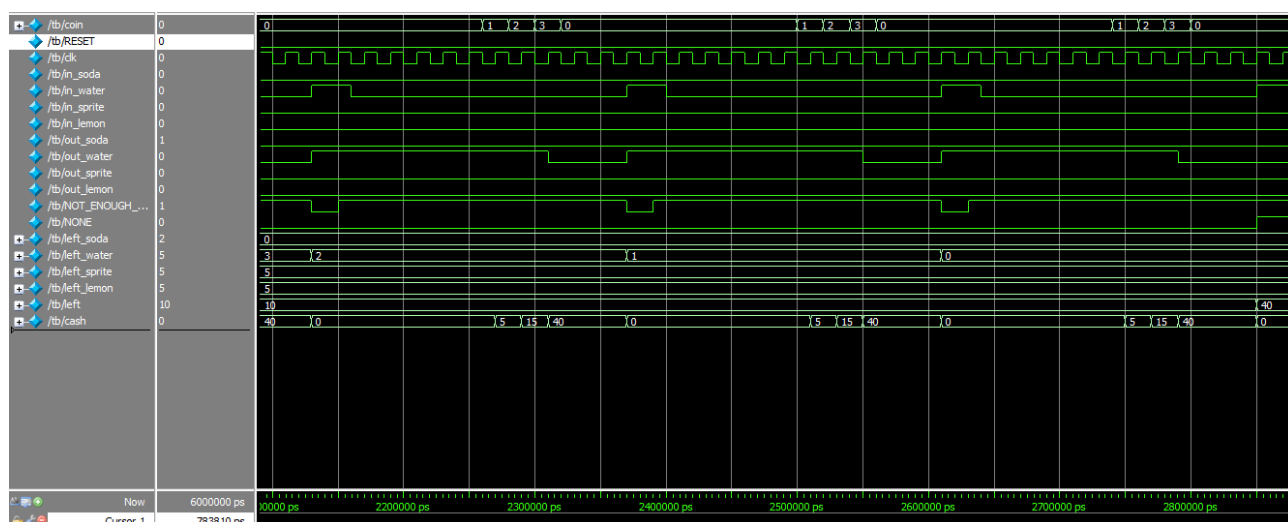


Рис.4.2.3. Вибір покупцем води

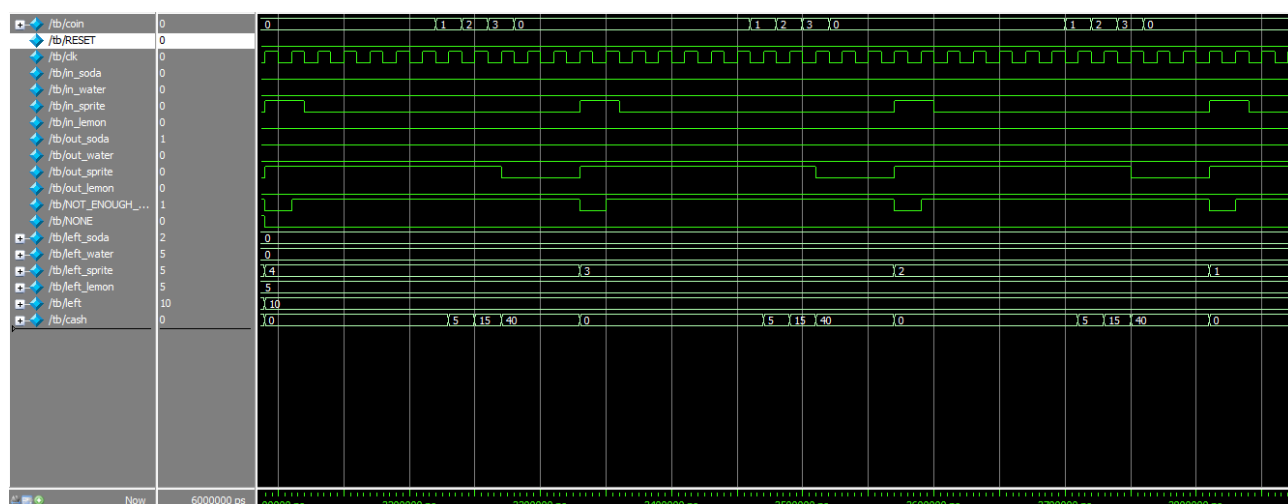


Рис.4.2.4. Вибір покупцем спрайт

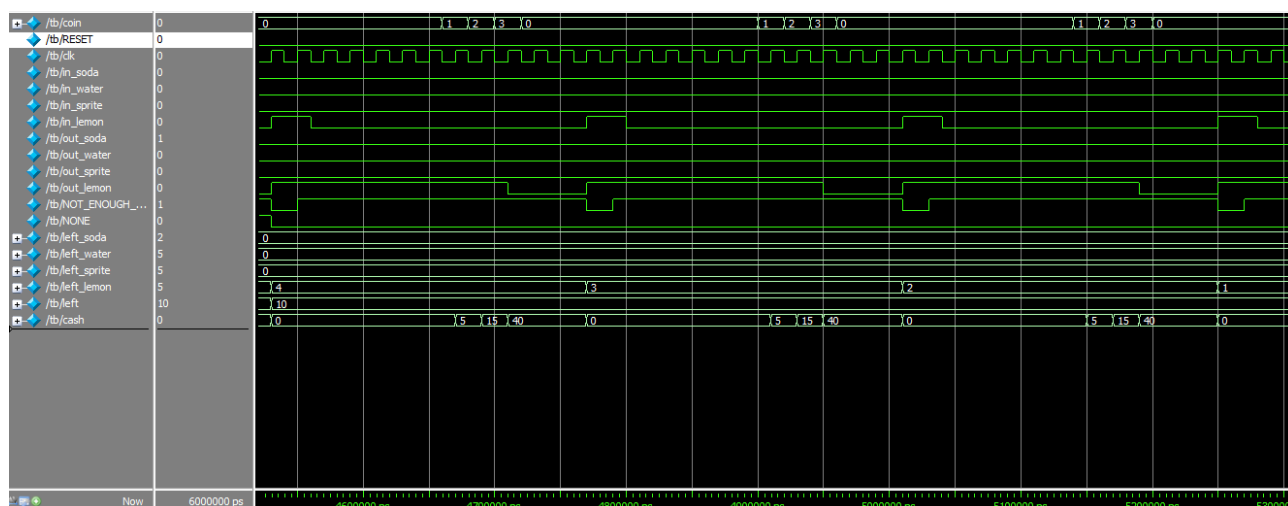


Рис.4.2.5. Вибір покупцем лимонаду

## ВИСНОВОК

Протягом курсової роботи було розроблено торговельний автомат для продажу чотирох видів напоїв з обмеженою кількістю екземплярів. Також автомат повертає решту покупцеві при внесенні суми більше вартості напою, при відсутності товару. При відсутності товару також покупець отримує відповідний сигнал. Якщо внесених коштів недостатньо для покупки одного з напоїв клієнт отримує відповідний сигнал, який сповіщає його, що внесених грошей недостатньо та отримує свої кошти назад. За результатами тестування приладу, можна переконатися, що все працює коректно.

Файли розробленого приладу та тестового файлу до нього можна знайти за посилання: <https://github.com/juliakotelenets/Vending-Machine.git>.

У висновку я б хотіла привести декілька переваг та недоліків вендингового автомату.

Торгові автомати мають мобільність, їх можна переміщати в нові місця, і вони будуть продовжувати надавати послуги як звичайно, немає касира, вони дають клієнтам вільний вибір для покупки продуктів у будь-який час доби, ви можете купувати потрібний вам товар 24 години на добу, протягом всього року.

Різноманітність товарів, із якими може працювати торговельний автомат. За допомогою автомата можна продавати різноманітні асортименти продуктів: фрукти, напої, напої, сигарети та інші продукти.

Більшість торгових автоматів зручні і дозволяють клієнтам економити час, вони знижують накладні витрати, не наймаючи персонал, збільшують прибуток, приймають кредитні картки та пропонують здоровішу їжу.

Торгові автомати працюють 24 години на добу 365 днів на рік, вони не вимагають ні матеріалів, ні особливої уваги, вони не займають багато часу, вам знадобиться небагато часу, щоб підтримувати їх у робочому стані та забезпечувати продуктами.

Торгові автомати вимагають інвестицій тільки в торгове обладнання, у них немає витрат на рекламу та маркетинг.



Торговими автоматами дуже легко управляти, так як ніхто не повинен бути присутнім при продажі товарів, вам потрібно тільки встановити автомат і ви можете просто спостерігати за його роботою здалеку, немає жодних проблем з оплатою, так як для покупки товарів потрібна миттєва оплата готівкою/карткою.

Торгові автомати економлять час, не потрібно витратити дорогоцінний час на приготування гарячого напою, так як торговий автомат може видати напій приблизно за 20 секунд, заощаджуючи масу часу.

Торгові автомати мають багато недоліків як для власника, так і для клієнтів, вони не дають можливості торгуватися, тому застосовуються фіксовані ціни, що може бути недружелюбним як для клієнта, так і для власника.

У цьому виді бізнесу поширені випадки шахрайства з боку клієнтів, які вигадують способи злому системи автомата для видачі продуктів.

Інвестор зазнає великих втрат, стикаючись з випадками руйнування автомата або неправильного програмування, наприклад безперервної видачі продукту через технічні помилки.

Торгові автомати виставляються в громадських місцях і можуть зазнати вандалізму з боку некерованих груп або заздрісних конкурентів.

## СПИСОК ЛІТЕРАТУРИ

1. Discuss on the importance of Vending Machines [Електронний ресурс] – Режим доступу: <https://cutt.ly/yYkFc7O> (дата звернення 01.12.2021р).
2. Automatic vending machines Advantages and Disadvantages [Електронний ресурс] – Режим доступу: <https://cutt.ly/hYkFlMX> (дата звернення 05.12.2021р).
3. Advantages and Disadvantages of Vending Machines [Електронний ресурс] – Режим доступу: <https://cutt.ly/fYkFfXA> (дата звернення 05.12.2021р).
4. Торговельний автомат [Електронний ресурс] – Режим доступу: <https://cutt.ly/wYkFyg5> (дата звернення 04.12.2021р).

## ДОДАТОК 1

```

module course (input [1:0] coin,
               input RESET, clk,
               input in_soda, in_water, in_sprite, in_lemon,
               output reg out_soda, out_water, out_sprite, out_lemon, NOT_ENOUGH_CASH,
               NONE,
               output reg [2:0] left_soda, left_water, left_sprite, left_lemon,
               output reg [15:0] left, cash
               );

```

```

Parameter soda = 30, water = 30, sprite = 30, lemon = 30;

```

```

always @(posedge clk)

```

```

begin

```

```

    case(coin)

```

```

        1: cash = cash + 5;

```

```

        2: cash = cash + 10;

```

```

        3: cash = cash + 25;

```

```

        Default: cash = cash;

```

```

    endcase

```

```

end

```

```

always @(posedge clk or posedge RESET)

```

```

    if(RESET)

```

```

        begin

```

```

            out_soda = 0;

```

```

            out_water = 0;

```

```

            out_sprite = 0;

```

```

            out_lemon = 0;

```

```

            NOT_ENOUGH_CASH = 0;

```

```

            NONE = 0;

```

```

            left = 0;

```

```

            cash = 0;

```

```

            left_soda = 5;

```

```

            left_water = 5;

```

```

            left_sprite = 5;

```

```

            left_lemon = 5;

```

```

        end

```

```

    else begin

```

```

        if(cash >= 30)

```

```

            begin

```

```

//soda
if(in_soda == 1)
    begin
        if(left_soda < 1)
            begin
                NONE = 1;
                left = cash;
                cash = 0;
            end
        else begin
            out_soda = 1;
            left_soda = left_soda - 1;
            left = cash - soda;
            cash = 0;
            NONE = 0;
            NOT_ENOUGH_CASH = 0;
        end
    end
end
else begin
    out_soda = 0;
end
//water
if(in_water == 1)
    begin
        if(left_water < 1)
            begin
                NONE = 1;
                left = cash;
                cash = 0;
            end
        else begin
            out_water = 1;
            left_water = left_water - 1;
            left = cash - water;
            cash = 0;
            NONE = 0;
            NOT_ENOUGH_CASH = 0;
        end
    end
end
else begin
    out_water = 0;
end
end

```

```

//sprite
if(in_sprite == 1)
    begin
        if(left_sprite < 1)
            begin
                NONE = 1;
                left = cash;
                cash = 0;
            end
        else begin
            out_sprite = 1;
            left_sprite = left_sprite - 1;
            left = cash - sprite;
            cash = 0;
            NONE = 0;
            NOT_ENOUGH_CASH = 0;
        end
    end
end
else begin
    out_sprite = 0;
end
//lemon
if(in_lemon == 1)
    begin
        if(left_lemon < 1)
            begin
                NONE = 1;
                left = cash;
                cash = 0;
            end
        else begin
            out_lemon = 1;
            left_lemon = left_lemon - 1;
            left = cash - lemon;
            cash = 0;
            NONE = 0;
            NOT_ENOUGH_CASH = 0;
        end
    end
end
else begin
    out_lemon = 0;
end
end

```

```
        end
    else
        NOT_ENOUGH_CASH = 1;
    end
end
endmodule
```

## ДОДАТОК 2

```

`timescale 1ns / 10ps

module tb;
    reg [1:0] coin;
    reg RESET, clk;
    reg in_soda, in_water, in_sprite, in_lemon;
    wire out_soda, out_water, out_sprite, out_lemon, NOT_ENOUGH_CASH, NONE;
    wire [2:0] left_soda, left_water, left_sprite, left_lemon;
    wire [15:0] left, cash;

    course FSM_controller (.coin(coin),
                           .in_soda(in_soda),
                           .in_water(in_water),
                           .in_sprite(in_sprite),
                           .in_lemon(in_lemon),

                           .left_soda(left_soda),
                           .left_water(left_water),
                           .left_sprite(left_sprite),
                           .left_lemon(left_lemon),

                           .out_soda(out_soda),
                           .out_water(out_water),
                           .out_sprite(out_sprite),
                           .out_lemon(out_lemon),

                           .NOT_ENOUGH_CASH(NOT_ENOUGH_CASH),
                           .left(left),
                           .NONE(NONE),
                           .cash(cash),
                           .RESET(RESET),
                           .clk(clk)
                          );

    initial begin
        RESET = 1;
        #20 RESET = 0;
        clk = 0;
        forever #10 clk = ~clk;
    end
end

```

```

initial begin
    in_soda = 0;
    in_water = 0;
    in_sprite = 0;
    in_lemon = 0;

    coin = 0;
    #100 coin = 1;
    #20 coin = 2;
    #20 coin = 3;
    #20 coin = 0;
    #50 in_soda = 1;
    #30 in_soda = 0;

    #100 coin = 1;
    #20 coin = 2;
    #20 coin = 3;
    #20 coin = 0;
    #50 in_soda = 1;
    #30 in_soda = 0;

    #100 coin = 1;
    #20 coin = 2;
    #20 coin = 3;
    #20 coin = 0;
    #50 in_soda = 1;
    #30 in_soda = 0;

    #100 coin = 1;
    #20 coin = 2;
    #20 coin = 3;
    #20 coin = 0;
    #50 in_soda = 1;
    #30 in_soda = 0;

    #100 coin = 1;
    #20 coin = 2;
    #20 coin = 3;
    #20 coin = 0;
    #50 in_soda = 1;
    #30 in_soda = 0;

```



```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_soda = 1;
#30 in_soda = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_water = 1;
#30 in_water = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_sprite = 1;
#30 in_sprite = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
```

```
#100 coin = 1;
#20 coin = 2;
#20 coin = 3;
#20 coin = 0;
#50 in_lemon = 1;
#30 in_lemon = 0;
end

initial #6000 $finish;
endmodule
```