Name: Mahmoud Raza kheyrati fard Student number: 9832110

Digital circuit design Final project: vending machine

Professor: Dr. Farshad KhunJush

Second semester 1398- 1399

Name: Mahmoud Raza kheyrati fard Student number: 9832110

Vending machine documentation

Features:

User:

- Buy products
- Charge his own wallet
- Can see how much money in his wallet right now

Machine owner:

- Charge products in the machine
- Change price of products
- Can see how much money in the machine
- Receive money from machine
- See the machine logs (all events)

Name: Mahmoud Raza kheyrati fard Student number: 9832110

General features:

Write Modular application

Seven-segment implementation

Display failure or successful message for all events

Error handling

Choose readable names

Test bench for each Verilog module

Clear Comments in the code

Program databases

DATABASE NAME	ACTION
LOGS.TXT	Store all machine logs
CUSTOMERMONEY.TXT	Store customer money
MACHINESAVEDMONEY.TXT	Store machine money
STUFF.TXT	Store products in the machine

Name: Mahmoud Raza kheyrati fard Student number: 9832110

The application made of 13 modules that builds separately.
Each module does one specific action not more!

Names of the modules:

1. **Main:** all components are connected here
2. **buyProductModule:** buy method done by this module
3. **chargeMachineModule:** charge products count
4. **recieveMoneyModule:** owner can recive money from machine
5. **changePriceModule :** the owner can change products price

6. **cashHandler:** all things related to increase and decrease money of the machine and customer done here
7. **storeHandler:** all things related to buy and sell products handled here
8. **saveMachineLogs:** save all events that happened to the machine eq. buy , charge , change price , … .
9. **sevenSegmentDisplay:** binary to seven-segment display with DP LED.
10. **setMoeny:** set customer money and machine money
11. **getMoney :** give saved customer money and machine money

12. **writeStuff:** write products to the database.

13. **readStuff:** read products from database.

The design of the application describes in the second page.

Name: Mahmoud Raza kheyrati fard Student number: 9832110

Main design:

Input/ output

```
input mainClock;
input [2:0] mode;
input [2:0] productCode;
input [3:0] productCount, recieveAmount, newPrice, chargeCustomerAmount;
output wire A, B, C, D, E, F, G, DP;
output reg [3:0]customerMoney , machineMoney;
```

A, B, C, D, E, F, G, DP; are segments of display

modules:

1. sevenSegmentDisplay
2. buyProductModule
3. chargeMachineModule
4. recieveMoneyModule
5. changePriceModule
6. cashHandler

Name: Mahmoud Raza kheyrati fard Student number: 9832110

productivity table:

mode	action	Related module
0	**Buy product**	buyProductModule
1	**Charge machine**	chargeMachineModule
2	**Receive money**	recieveMoneyModule
3	**Show machine logs**	-
4	**Change product price**	changePriceModule
5	**Show customer money**	getMoney
6	**Show machine money**	getMoney
7	**Charge customer money**	cashHandler

All stuff done by the separate modules.

Name: Mahmoud Raza kheyrati fard Student number: 9832110

buyProductModule Design:

buy method done here

input / output

```
input clock;
input [2:0] itemCode;
input [3:0] itemCount;
output reg DP;
```

item Code: is product code that in the stuff.txt

item count: count of product that user want to buy

DP: represent the error (active low)

Modules:

1. **getMoney**
2. **readStuff**
3. **cashHandler**
4. **cashHandler**
5. **storeHandler**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

changeMachineModule Design:

charge products int the machine done here

input/ output

```
input clock;
input [2:0] productCode;
input [3:0] productCount;

output reg DP;
```

modules:

1. **storeHandler: to increase count of product**
2. **saveMachineLogs: to save log**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

recieveMoneyModule Design:

the owner can receive money from machine

input/ output

```
input clock;
input [3:0] amount;
```

modules:

1. **getMoney : to get machine money**
2. **cashHandler : to decrease machine money amount**
3. **saveMachineLogs : to save the log**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

changePriceModule Design:

the owner can change price of the product with the help of this module

input/ output

```
input clock;
input [2:0] productCode;
input [3:0] newPrice;
```

modules:

1. **storeHandler : to change the price**
2. **saveMachineLogs : to save the log**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

cashHandler Design:

all stuff that relate to the customer money and machine money done here

input/ output

```
input clock, mode, func;
input [3:0] amount;
output reg res;
```

modules:

1. **setMoney**
2. **getMoney**

productivity table:

mode	func	action
0	**0**	**Purchase (decrease customer money)**
0	**1**	**Charge (increase customer money)**
1	**0**	**Increase machine money**
1	**1**	**Receive money from machine**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

storeHandler Design:

all stuff that relate to the products and stuff.txt done here

input/ output

```
input clock;
input [1:0] mode;
input [2:0] productCode;
input [3:0] itemCount, newPrice;
```

productivity table:

mode	action
00	**Charge product by the owner of machine**
01	**Buy by the customer**
10	**Update product price**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

saveMachineLogs Design:

save every event log in the logs.txt

input/ output

```
input clock, param1;
input [1:0] operator ;
input [3:0]  param2, param3, param4;
```

productivity table:

operator	action	Param1	Param2	Param3	Param4
00	**buy**	**status**	**productCode**	**Item count**	**price**
01	**Charge machine**	**status**	**productCode**	**Added count**	**-**
10	**Receive money**	**status**	**amount**	**-**	**-**
11	**Change price**	**status**	**ProductCode**	**New price**	**-**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

writeStuff Design:

save every product in stuff.txt

input/ output

```
input clock;
input [10:0] p0, p1, p2, p3, p4
```

p0,p1,p2,p3,p4 are the products

Name: Mahmoud Raza kheyrati fard Student number: 9832110

readStuff Design:

read every product in stuff.txt

input/ output

```
input clock;
output reg [10:0] p0, p1, p2, p3, p4;
```

p0,p1,p2,p3,p4 are the products

Name: Mahmoud Raza kheyrati fard Student number: 9832110

setMoney Design:

write the customer and machine money to the related database

input/ output

```
input clock, mode;
input [3:0] value;
```

productivity table:

mode	action
0	**Set Machine money**
1	**Set customer money**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

getMoney Design:

read the customer and machine money to the related database

input/ output

```
input clock, mode;
output reg [3:0] value;
```

productivity table:

mode	action
0	**Set Machine money**
1	**Set customer money**

Name: Mahmoud Raza kheyrati fard Student number: 9832110

sevenSegmentDisplay Design:

represent binary number into the seven-Segment Display

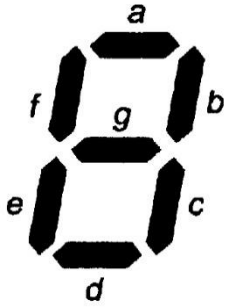
input/ output

```
input clock, isError;
input [3:0] binaryNumber;
output reg A, B, C, D, E, F, G,DP;
```

A-G are segments of display

DP follow isError input

Productivity table:



Name: Mahmoud Raza kheyrati fard Student number: 9832110

Segments (✓ = ON)							Display
a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0
	✓	✓					1
✓	✓		✓	✓		✓	2
✓	✓	✓	✓			✓	3
	✓	✓			✓	✓	4
✓		✓	✓		✓	✓	5
✓		✓	✓	✓	✓	✓	6
✓	✓	✓					7

Segments (✓ = ON)							Display
a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓	✓	8
✓	✓	✓			✓	✓	9
✓	✓	✓		✓	✓	✓	A
		✓	✓	✓	✓	✓	b
✓			✓	✓	✓		C
	✓	✓	✓	✓		✓	d
✓			✓	✓	✓	✓	E
✓				✓	✓	✓	F

Note: the Verilog code based on the table above for example for F in hex we had this Verilog code :

```
4'b1111: begin   //f
  A = 1;
  B = 0;
  C = 0;
  D = 0;
  E = 1;
  F = 1;
  G = 1;
end
```

NOTE: all components are edge triggered and on the positive edge of the clock, and all components synchronized with the clock.