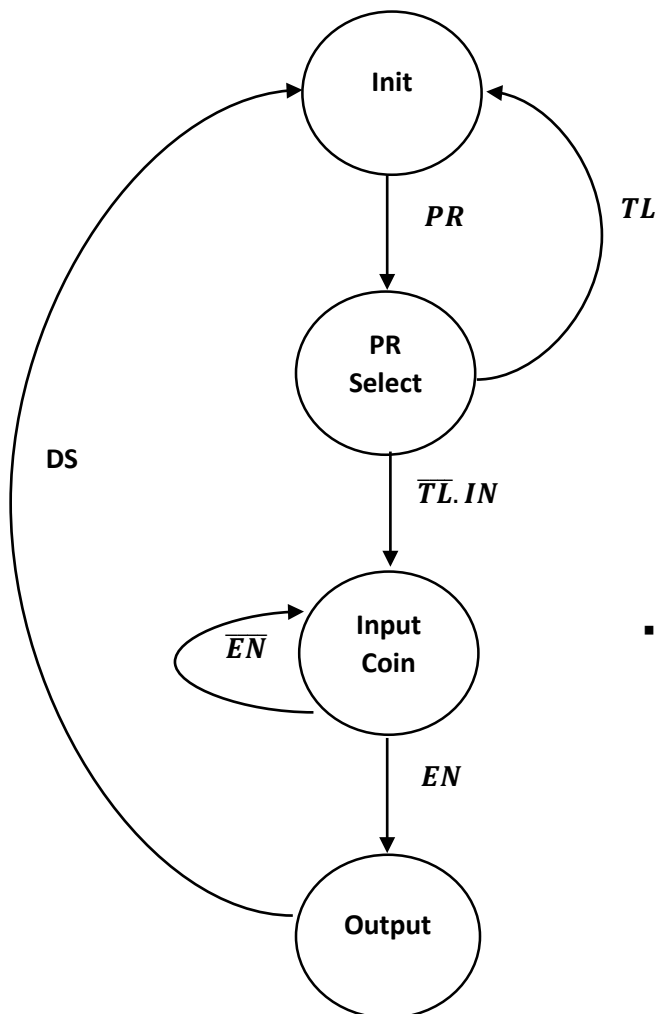


دستگاه **Vending-machine** یک دستگاه خودکار است که اقلامی مانند تنقلات، نوشیدنی ها و ..... را پس از درج پول نقد، کارت اعتباری یا سایر اشکال پرداخت در دستگاه به مصرف کنندگان می دهد. اولین **Vending-machine** مدرن در اوایل دهه 1880 در انگلستان توسعه یافتند و کارت پستال ها را توزیع کردند. ماشین های فروش خودکار در بسیاری از کشورها وجود دارند و در زمان های اخیر، ماشین های فروش تخصصی که محصولات غیر معمول را در مقایسه با اقلام ماشین های فروش سنتی ارائه می دهند، ایجاد شده اند.

برای طراحی مدل منطقی این ماشین ابتدا نمودار ماشین حالت آن را رسم می کنیم و سپس به جدول های مربوطه می پردازیم



▪ State-Machine Diagram for Vending-Machine

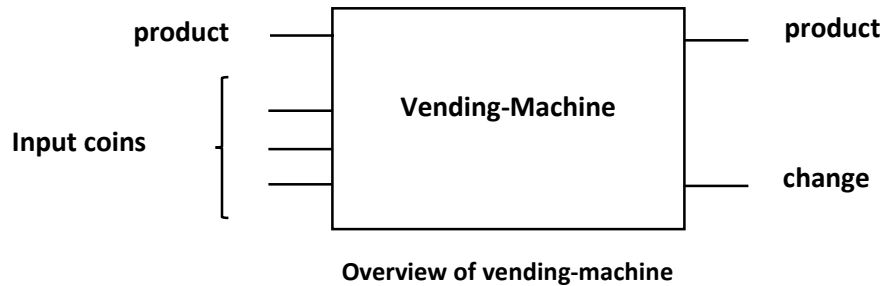
▪ **Input and Output variables for Vending-Machine**

<b>Input Symbol</b>	<b>Name</b>	<b>Meaning for Value 1</b>	<b>Meaning for Value 0</b>
PR	Selected product	Product number 1	Product number 0
TP	Stopwatch state	finished	Not finished
IN	Input money	Money inserted	Not inserted
EN	Enable output	Enough money	Not enough money
DS	Dispenser state	dispensed	Not dispensed
<b>Output Symbol</b>	<b>Name</b>	<b>Meaning for Value 1</b>	<b>Meaning for Value 0</b>
PR	Selected product	Product number 1	Product number 0
CH	Money change	Change available	Change not available

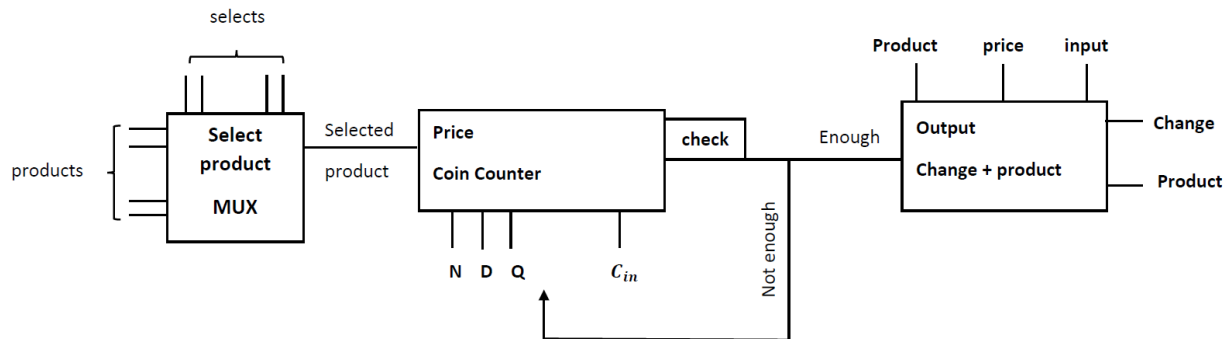
▪ **State table for the Vending-Machine**

<b>State</b>	<b>State code</b>	<b>Transition condition</b>	<b>Nest state</b>	<b>State code</b>	<b>None-Zero output including mealy output</b>
<b>Init</b>	1000	<i>PR</i>	Pr_Select	0100	
<b>Pr_Select</b>	0100	$\overline{TL}.IN$ $TL.\overline{IN}$	Input_coin Init	0010 1000	<i>PR</i>
<b>Input_coin</b>	0010	$EN$ $\overline{EN}$	Output Input_coin	0001 0010	<i>PR, CH</i> <i>PR</i>
<b>output</b>	0001	<i>DS</i>	Init	1000	

برای طراحی مدار منطقی دستگاه **Vending-Machine** ابتدا باید ورودی ها و خروجی های دستگاه را مشخص کنیم. همانطور که در توضیحات پروژه مطرح شده استو این دستگاه ابتدا شکلات انتخاب شده را به عنوان ورودی دریافت میکند و سپس 3 نوع سکه مختلف را به تعداد دلخواه می تواند به عنوان ورودی دریافت کند. در خروجی نیز مقدار پول باقی مانده و شکلات انتخاب شده را نمایش میدهد. شکل زیر درک بهتری از روند کلی دستگاه به ما میدهد:



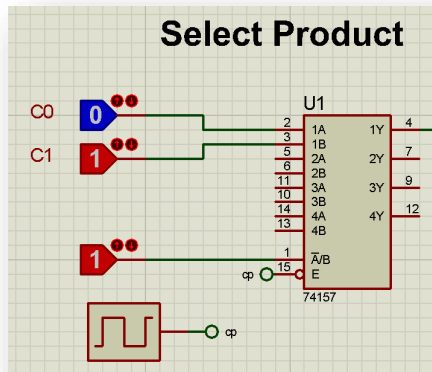
بنابراین دستگاهی که طراحی خواهیم کرد دارای بخش های مختلفی خواهد بود که هر کدام به انجام وظیفه مشخصی خواهند پرداخت. شکل زیر نمای کلی از هر بخش دستگاه را نشان می دهد که به هر کدام جداگانه خواهیم پرداخت.



Vending-machine part by part illustration

## 1. Select product

این بخش دستگاه کالای انتخابی توسط کاربر را تشخیص می دهد و برای استفاده در دیگر بخش های مدار ذخیره میکند. پیاده سازی این بخش توسط یک مالتی پلکسر انجام میشود که کالاهای مختلف به پین های ورودی مالتی پلکسر متصل می شوند و انتخاب توسط پین های سلکت مالتی پلکسر انجام میشود. در این پروژه چون دو کالا برای خرید وجود دارد، از یک مالتی پلکسر 2 به 1 استفاده کردیم که تنها پین سلکت آن به یک دکمه انتخاب متصل است.



**74157 IC**  
Quad 2-to-1 line MUX with Enable

## 2. Coin-counter

### 3. check

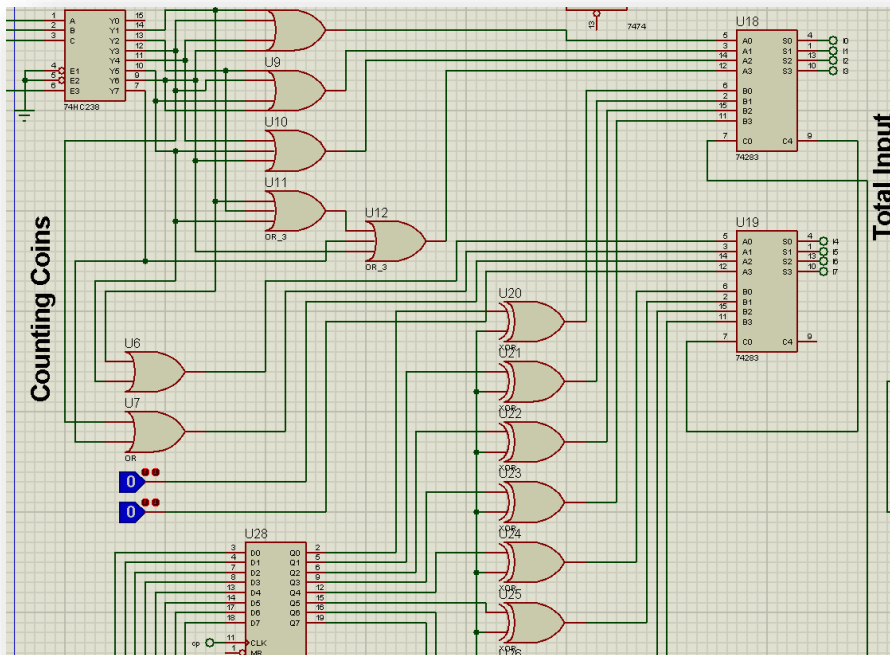
این بخش از سیستم سکه های مختلف را از کاربر دریافت می کنند و مقدار معادل را در ۸ بیت ذخیره میکند. با توجه به توضیحات پروژه امکان دریافت سه نوع سکه مختلف از کاربر وجود دارد:

- N : Nickel – 5 cents
- D : Dime – 10 cents
- Q : Quarter – 25 cents

ورودی های دریافت شده با استفاده از مدار منطقی طراحی شده به مقدار معادل 8 بیتی تبدیل می شوند و سپس پس با مقدار قبلی وارد شده جمع شده و در یک متغیر ۸ بیتی ذخیره میشوند. عمل ذخیره سازی با استفاده از یک **D-type Flip-Flop** 8 بیتی انجام میشود.

برای طراحی مدار منطقی مورد نیاز ابتدا جدول درستی با 3 ورودی و 8 خروجی را رسم میکنیم و سپس رابطه هر کدام از خروجی ها را بر حسب ورودی ها به دست میآوریم در انتها مدار را با استفاده از دیکودر 3 به 8 پیاده سازی میکنیم.

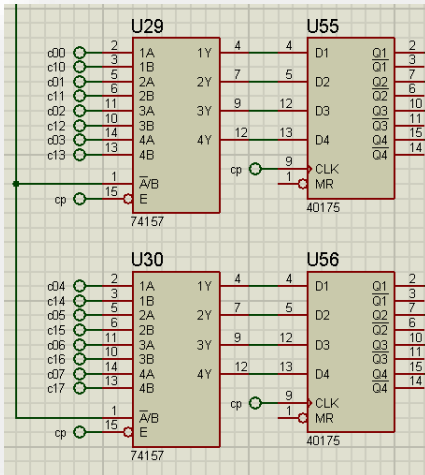
N	D	Q	S7	S6	S5	S4	S3	S2	S1	S0	m
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	1	0	1	0	2
0	1	1	0	0	1	0	0	0	1	1	3
1	0	0	0	0	0	0	0	1	0	1	4
1	0	1	0	0	0	1	1	1	1	0	5
1	1	0	0	0	0	0	1	1	1	1	6
1	1	1	0	0	1	0	1	0	0	0	7

**74HC238 IC****3-to-8 line Decoder with Enable****74238 IC****two 4-bit full-Adder cascaded****74273 IC****8-bit D-type Flip-Flop with clear and Reset**

کاربر تا زمانی که مقدار پول وارد شده از قیمت کالای انتخاب شده کمتر باشد میتواند سکه وارد کند. تشخیص کافی بودن ورودی اعمال شده توسط بخش **check** سیستم انجام می شود. ابتدا قیمت کالای انتخاب شده توسط یک مالتی پلکسر انتخاب شده و سپس با استفاده از یک **Comparator** با پول وارد شده مقایسه می شود اگر پول وارد شده بیشتر از قیمت کالای انتخاب شده باشد، بخش **output** مدار فعال میشود تا خروجی ها نمایش داده شوند.

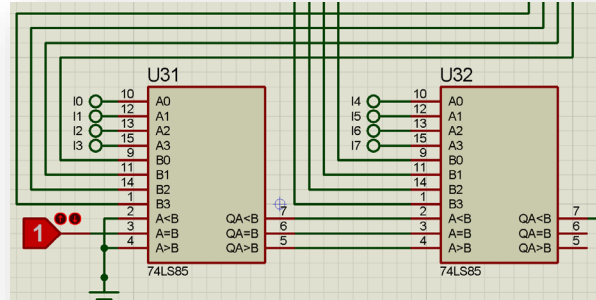
برای انتخاب قیمت کالای انتخاب شده از دو تراشه **74157** استفاده می کنیم تا 8 بیت قیمت را سلکت کنیم. همچنین از دو 4 D-type Flip-Flop بیتی استفاده میکنیم تا خروجی های مالتی پلکسر را ثابت نگه داریم.

برای اینکه بتوانیم قیمت کالا را با پول وارد شده مقایسه کنیم باید از یک **comparator** 8 بیتی استفاده کنیم اما چون پروتئوس این تراشه را به صورت آماده ندارد، دو **comparator** چهار بیتی را با هم **cascade** میکنیم تا یک **comparator** 8 بیتی بسازیم.



**74157 IC**  
Quad 2-to-1 line MUX

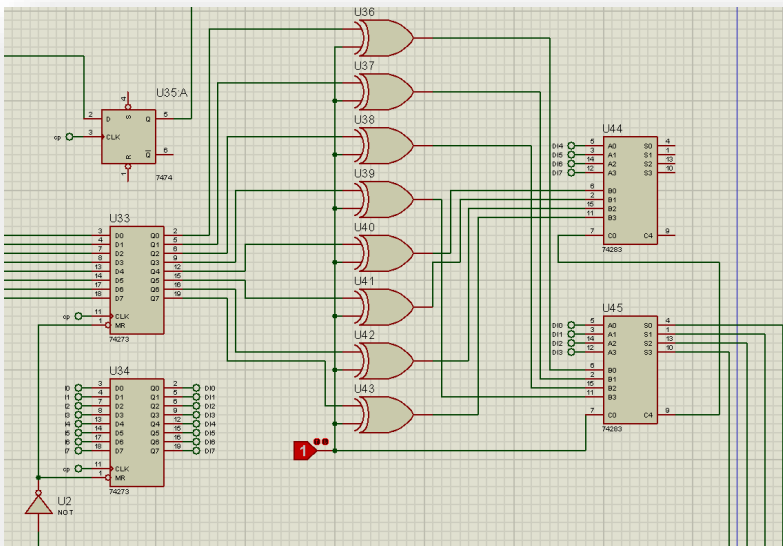
**40175 IC**  
4-bit D-type Flip-Flop with clear and Reset



**74LS85 IC**  
two 4-bit comparators cascaded

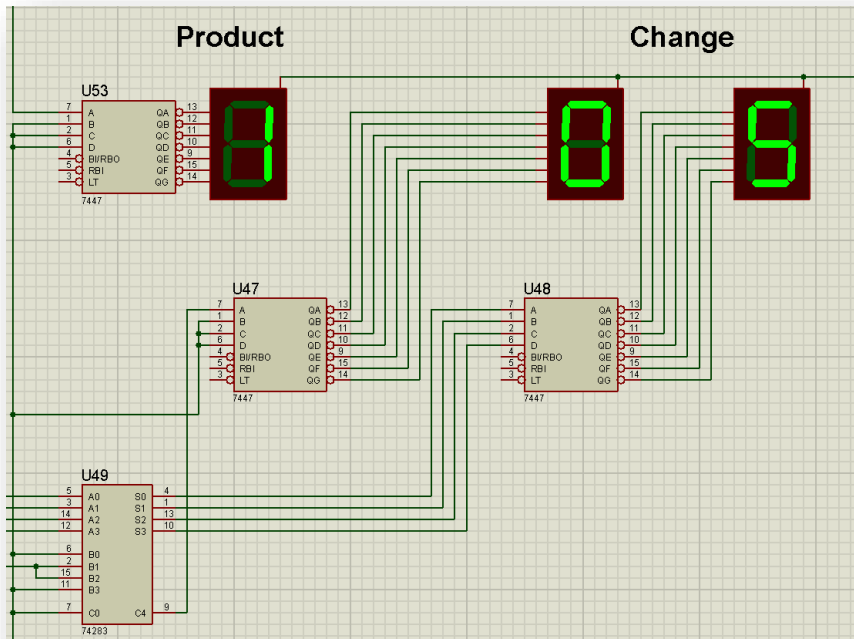
## 4. Output

بعد از اینکه مقدار پول ورودی از قیمت کالای انتخاب شده بیشتر شد و یا بهتر است بگوییم کمتر نباشد بخش خروجی مدار فعال میشود. زمانی که بخش خروجی مدار فعال می شود، مقدار پول وارد شده و قیمت کالا به یک **Binary subtractor** انتقال می یابد تا قیمت کالا از پول وارد شده کم شود و باقی مانده محاسبه شود.



در نهایت بعد از اینکه مقدار پول باقی مانده را محاسبه کردیم، این مقدار را برای نمایش به بخش سون سگمنت متصل میکنیم. مقدار پول باقی مانده را روی 2 سون سگمنت نمایش می دهیم. برای این کار برای هر سون سگمنت یک مبدل Binary به BCD استفاده می کنیم.

برای نمایش کالای انتخاب شده نیز از یک سون سگمنت جداگانه استفاده می کنیم.



**74283 IC**

**4-bit Full-Adder**

**7447 IC**

**BCD-to-Seven-Segment Decoders/Drivers**

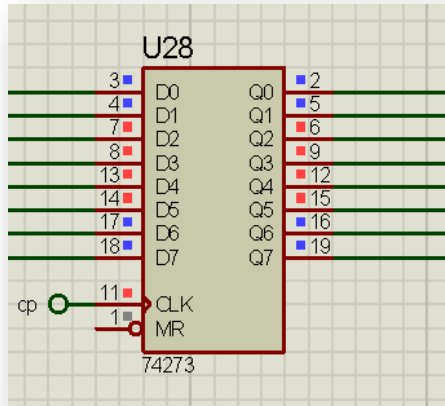
**Common-Anode 7Segment, Green**

بررسی رفتار مدار در سناریو های مختلف:

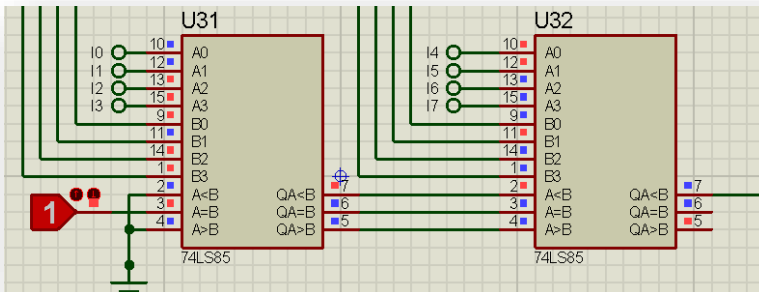
۱. سناریو اول: انتخاب شکلات نوع 1 به قیمت 45 سنت و پول ورودی 60 سنت (25 + 25 + 10)

نتایج مورد انتظار:

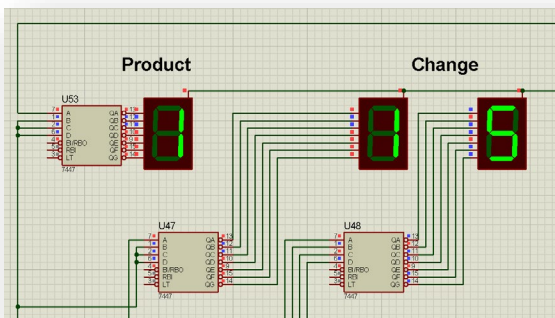
- ذخیره ورودی :  $60 = (00111100)_2$
- تشخیص کافی بودن ورودی
- نمایش باقی مانده 15 سنت و کالای انتخاب شده 1



ذخیره ورودی به درستی انجام شده است.



مقایسه ورودی با قیمت کالا به درستی انجام شده است.



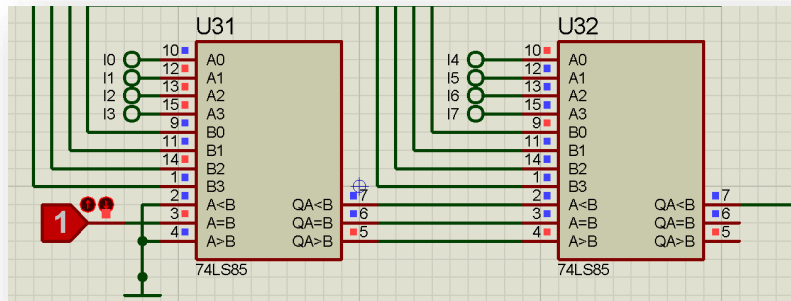
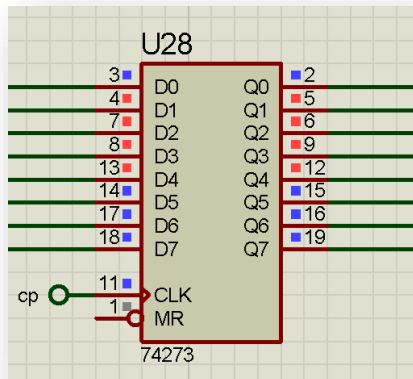
خروجی به درستی نمایش داده شده است.



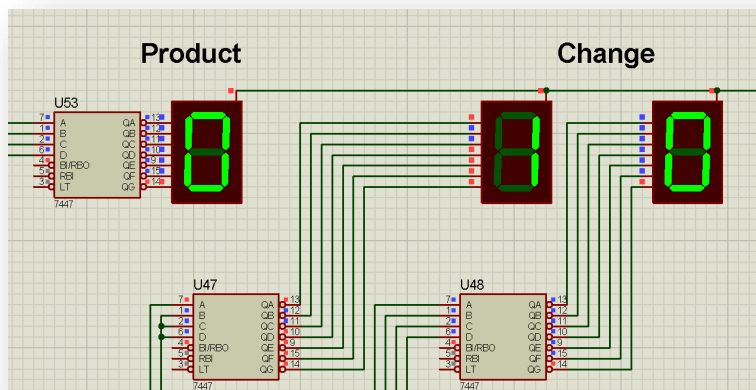
II. سناریو دوم: انتخاب شکلات نوع 0 به قیمت 20 سنت و پول ورودی 30 سنت (5 + 5 + 10 + 10)

نتایج مورد انتظار:

- ذخیره ورودی :  $30 = (00011110)_2$
- تشخیص کافی بودن ورودی
- نمایش باقی مانده 10 سنت و کالای انتخاب شده 0



مقایسه ورودی با قیمت کالا به درستی انجام شده است.

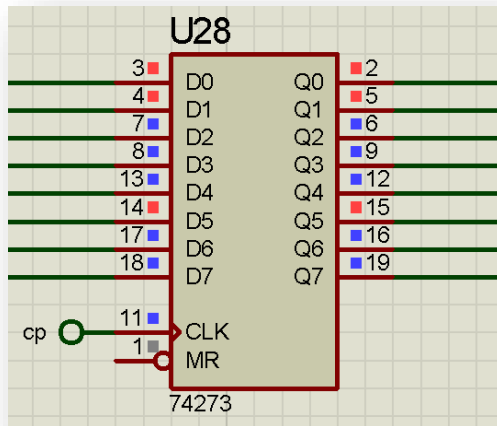


خروجی به درستی نمایش داده شده است.

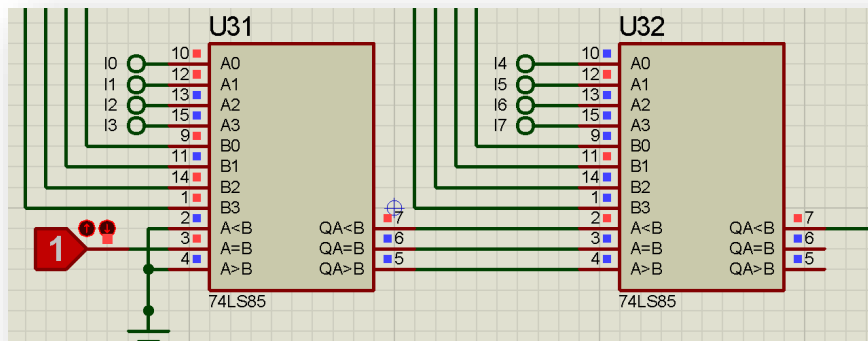
III. سناریو سوم: انتخاب شکلات نوع 1 به قیمت 45 سنت و پول ورودی 35 سنت (10 + 25)

نتایج مورد انتظار:

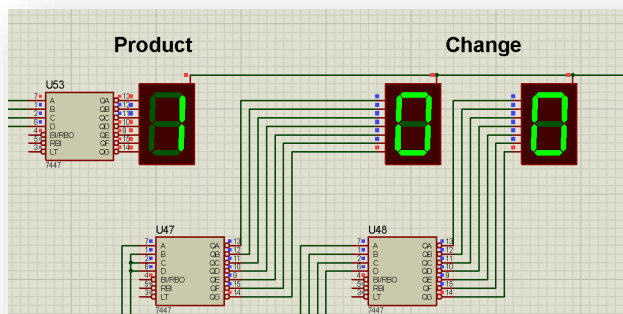
- ذخیره ورودی :  $35 = (00100011)_2$
- تشخیص کافی نبودن ورودی
- نمایش 0 در باقی مانده



ذخیره ورودی به درستی انجام شده است.



مقایسه ورودی با قیمت کالا به درستی انجام شده است.



خروجی به درستی نمایش داده شده است.

