



# VENDED MACHINE

Amirshayan Moghaddas

402243102

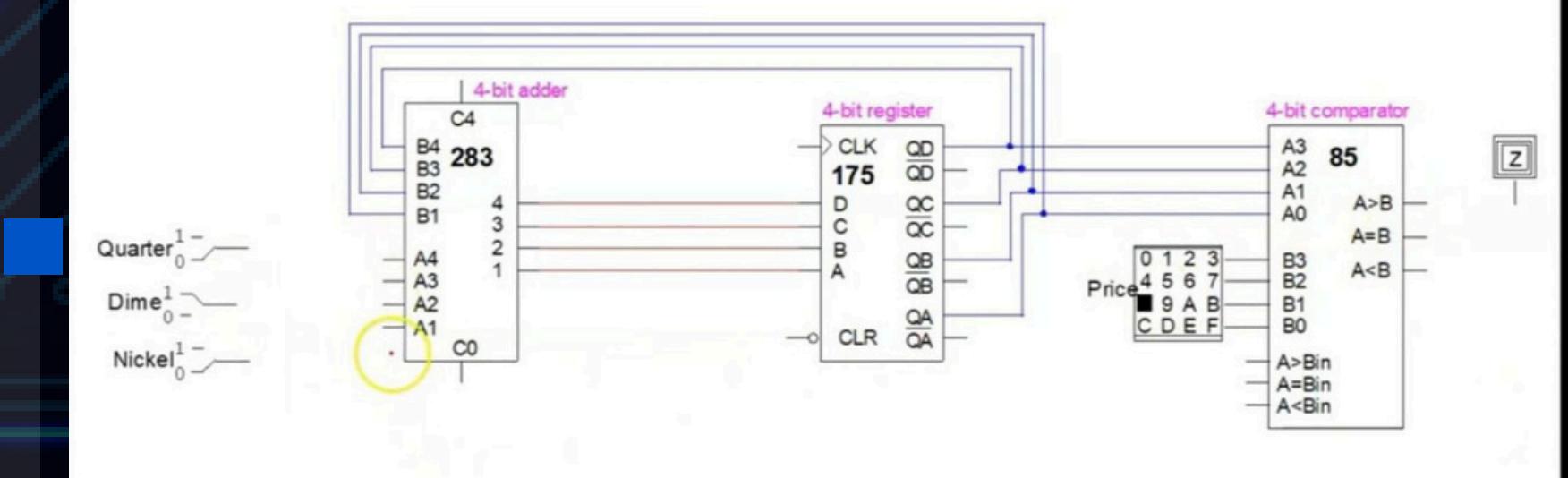
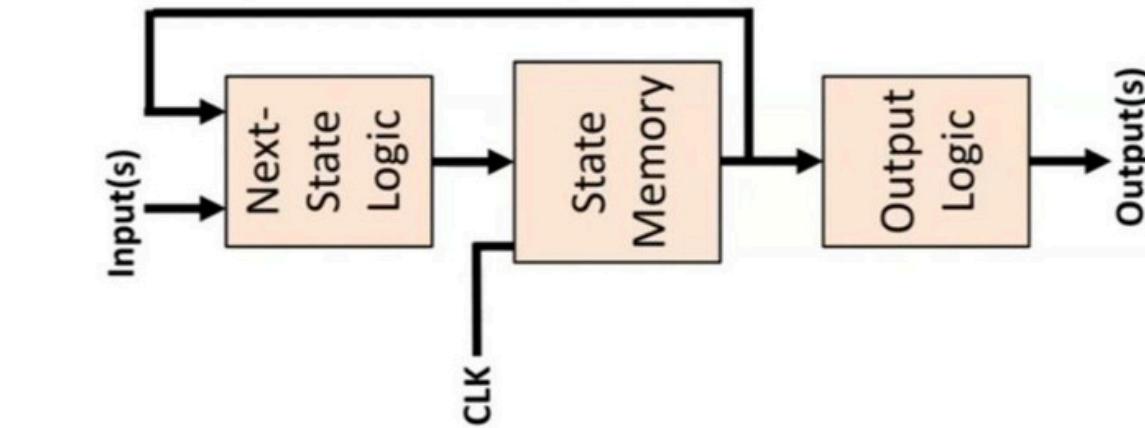
# INTRODUCTION

وندینگ ماشین نوعی ماشین است که به صورت خودکار خرید و فروش محصول را انجام میدهد. حال ما میخواهیم نوعی خاص از وندینگ ماشین را طراحی کنیم. برای اینکار در ۵ مرحله پیش میرویم

# ستفون

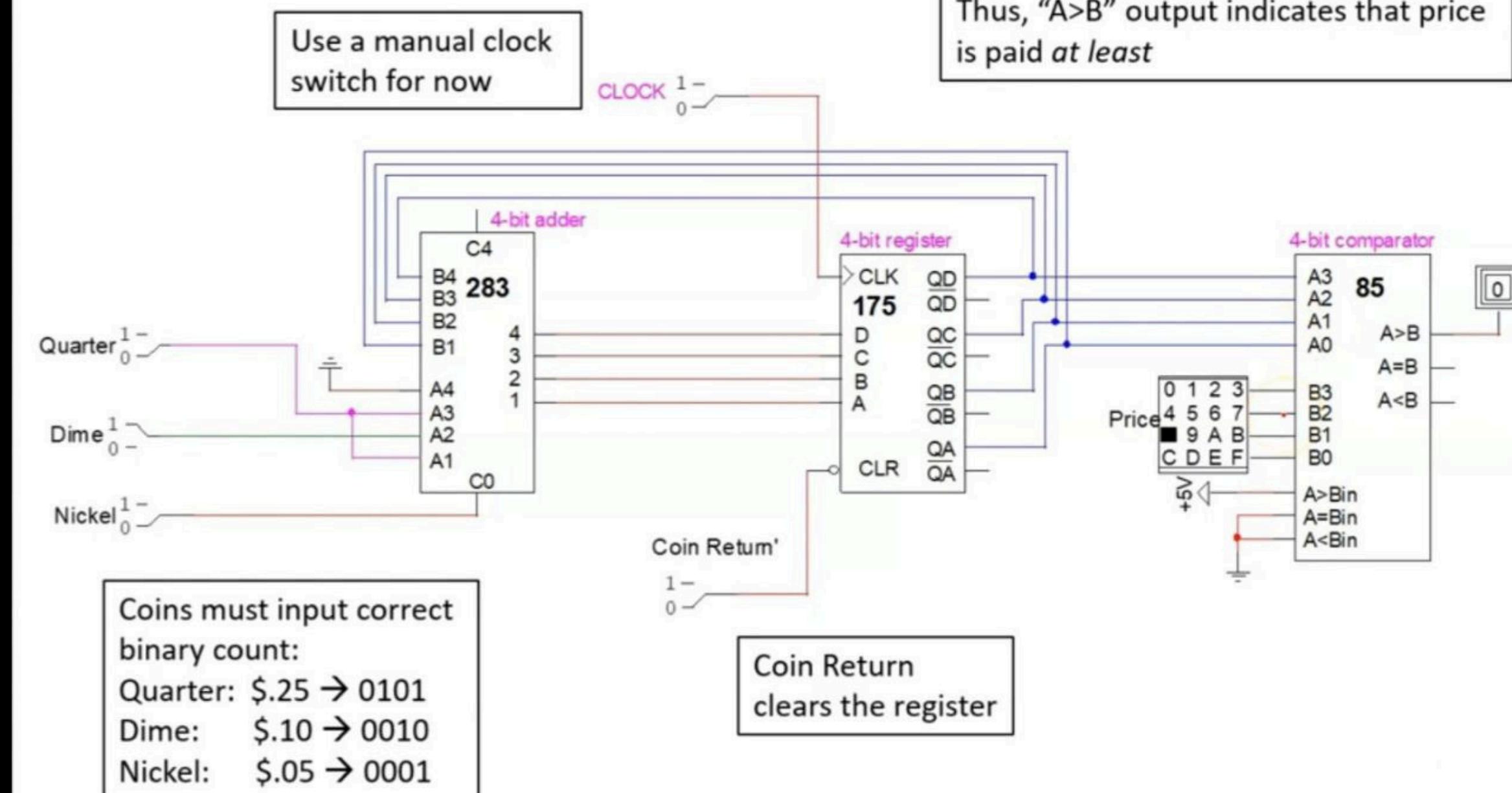
در این مرحله یک محصول به قیمت 40 در نظر میگیریم که میتوانیم به سه روش (dimes=10) (nickels=5) و (quarters=25) خریداری کنیم. این مقادیر را در یک ریجستر چهار بیتی ذخیره میکنیم. برای next state نیاز به یک adder چهار بیتی داریم تا مقادیر گرفته شده را جمع کند. همچنین به یک comprator نیاز داریم تا مقادیر خارج شده از adder را با قیمت محصول مقایسه کند.

GENERAL → SPECIFIC



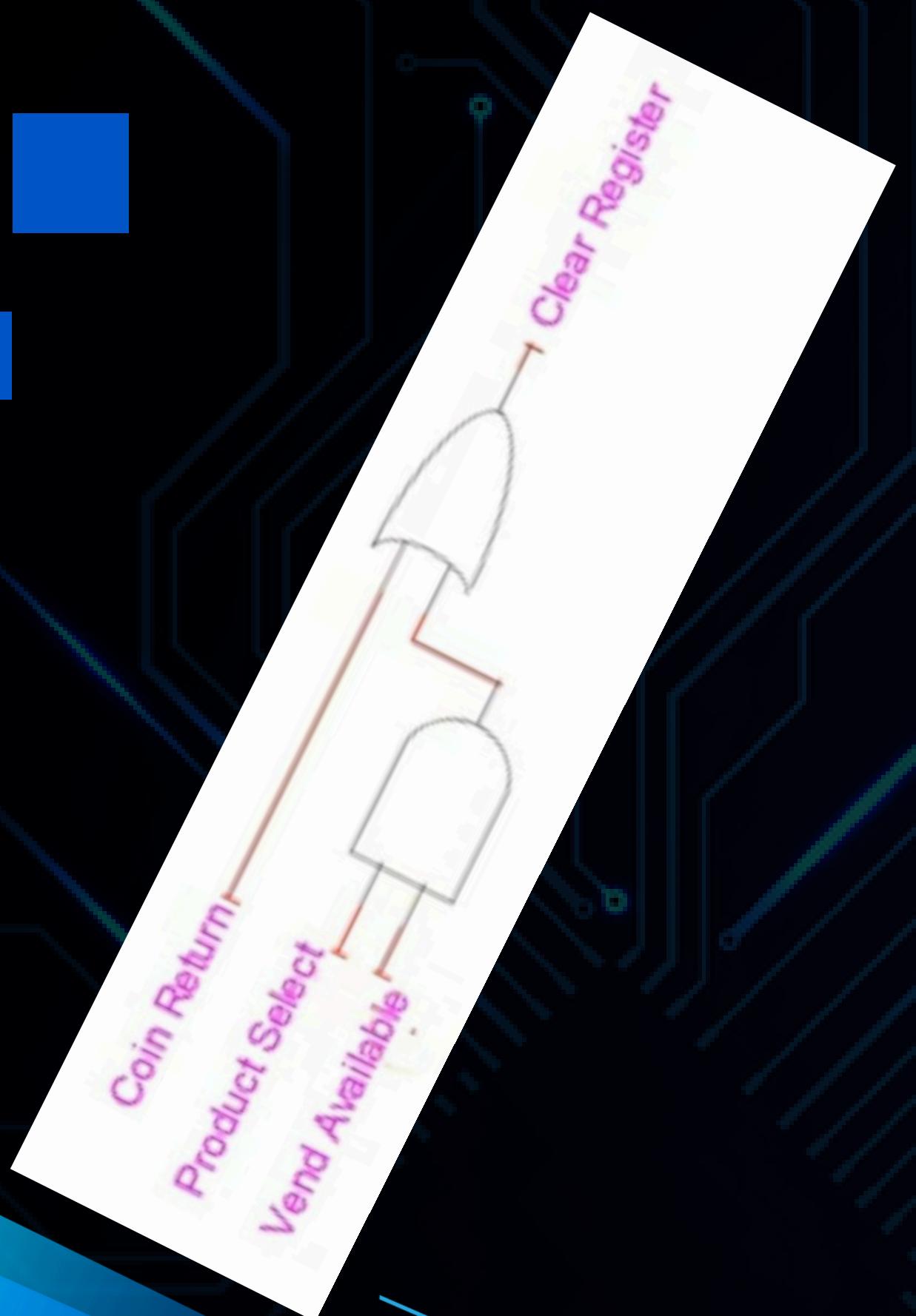
# final circuit of step1

## MODEL A

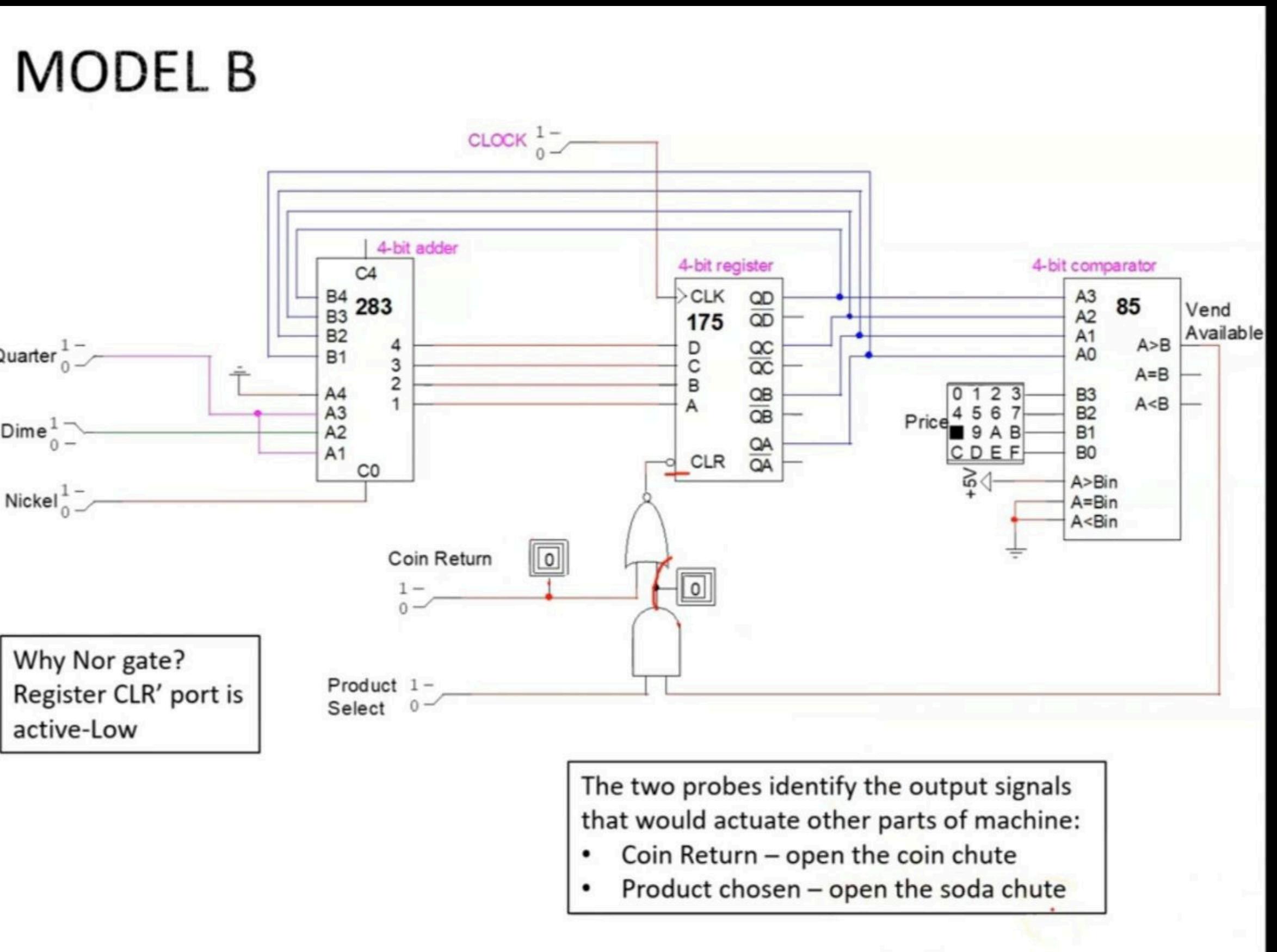


## STEP TWO

در این مرحله قصد داریم قابلیت انتخاب بین چهار محصول را به یوزر بدهیم. به این موضوع باید دقت کنیم که اگر مقدار ورودی برای خرید محصول کافی نبود selector برای انتخاب محصول بی معنی میشود و اگر مقدار ورودی کافی بود selector انتخاب شده و پس از آن مقدار نگهداری شده در register صفر میشود(به برگرداندن پول در مراحل بعد میپردازیم)



# final circuit of step2



## STEP THREE

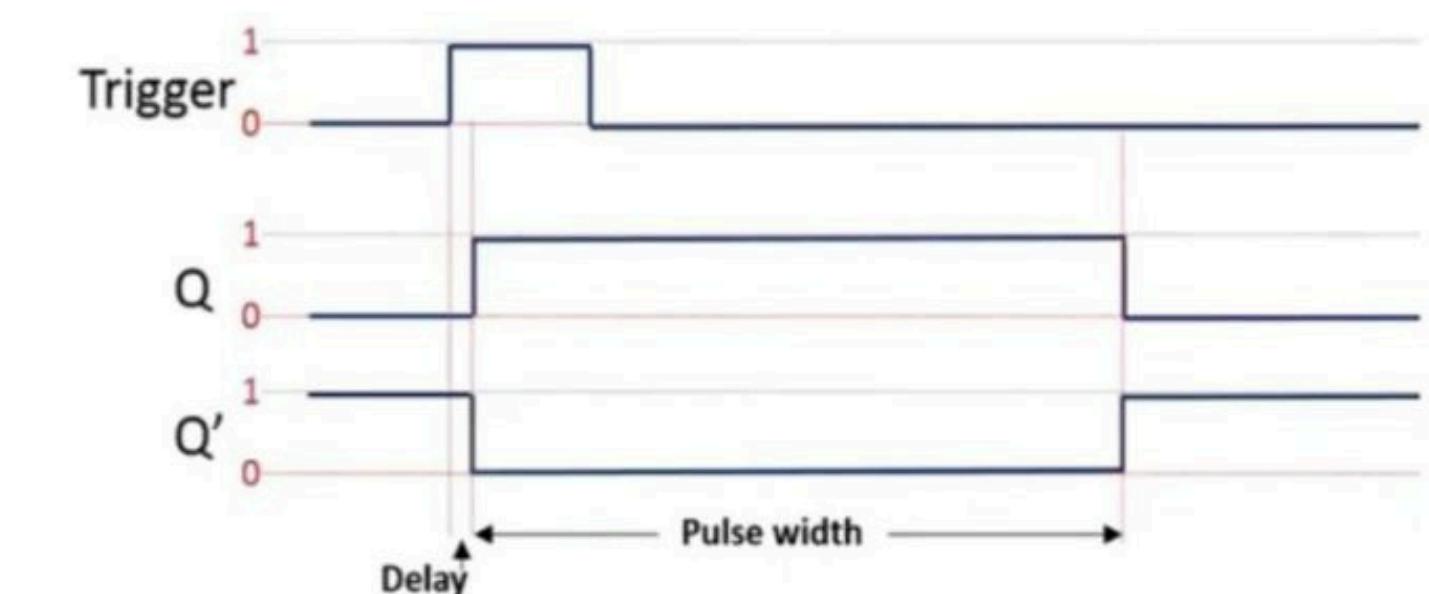
در مرحله قبل برای ذخیره شدن اطلاعات در رجیستر نیاز بود که یوزر به صورت دستی هر بار clock را active high کند تا اطلاعات ذخیره شده و به مرحله بعد برود. حال ما در این مرحله قصد داریم که اطلاعات به صورت اتوماتیک و one shot در رجیستر ذخیره گردد. برای حل این روش دو راه وجود دارد:

Asynchronous: one-shot device to delay positive edge

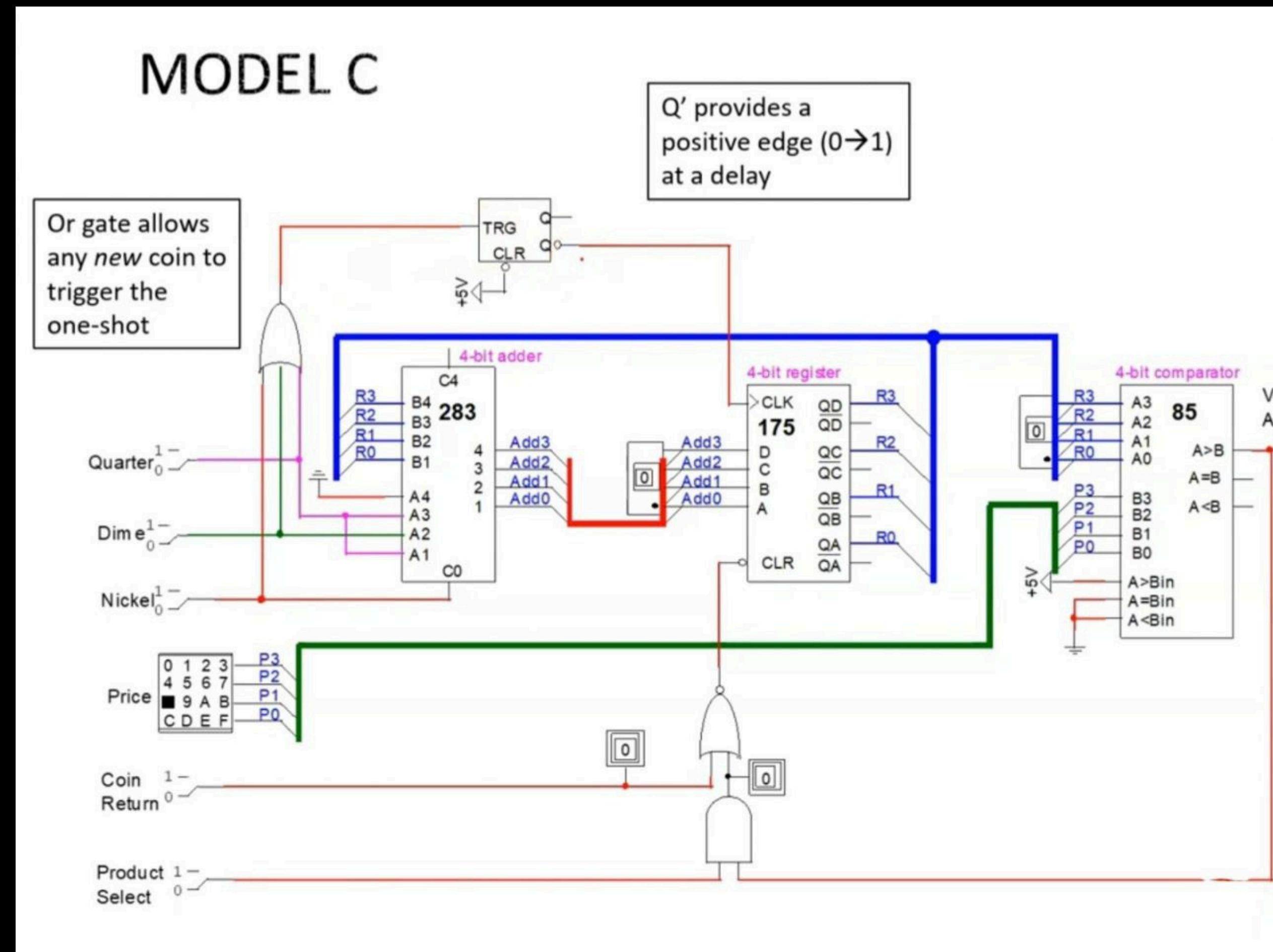
Synchronous: FSM to determine when clock is enabled

که ما از روش اول برای حل این مسئله استفاده میکنیم. در این روش به روش clear به همراه trigger استفاده میکنیم. فیلی فلاپ بعد از یک دیلی مشخص اطلاعات را از register به comperator ارسال میکند.

- The One-shot can provide a positive edge (through  $Q'$ ), at a long enough delay to ensure Adder has processed
- Should be triggered if any coin is deposited



# final circuit of step3



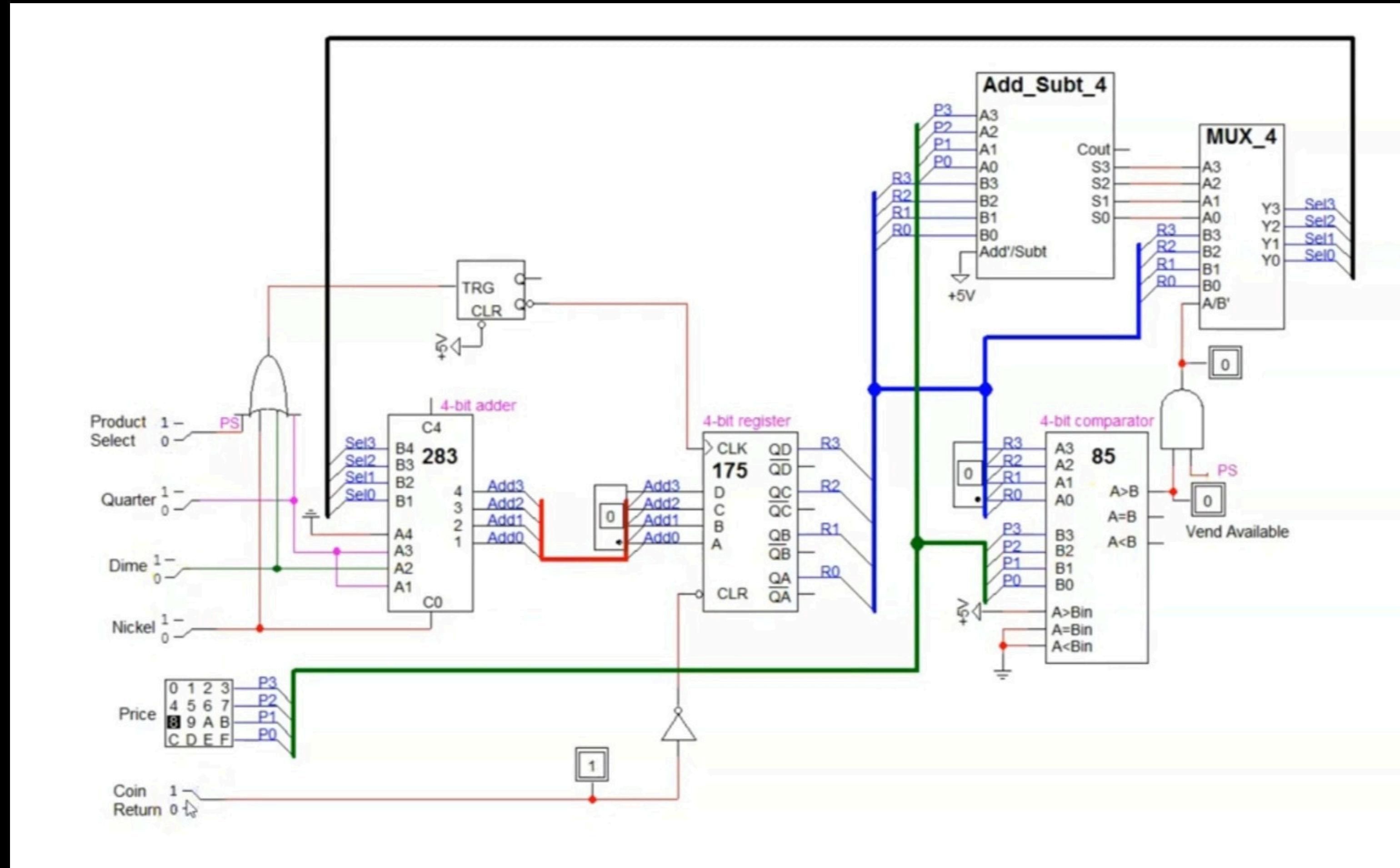
# STEP FOUR

در این مرحله میخواهیم مقدار پول اضافی را ذخیره کنیم و برگردانیم. برای اینکار به دو operation نیاز داریم. اولین subtractor operation یک میزان اختلاف پول ورودی با قیمت محصول را محاسبه میکند:

$$\text{Register} - \text{Product Price} = \text{Change}$$

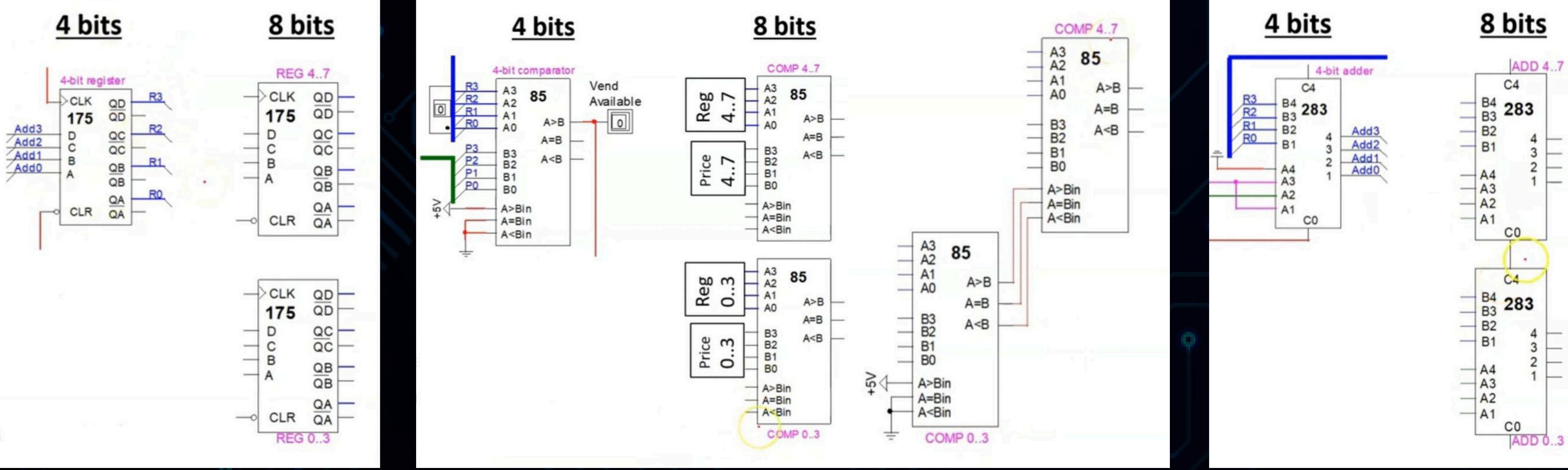
دومین operation یک Data Selector است. در مدار یک رجیستر وجود دارد ولی دو امکان برای ذخیره داده در این رجیستر هست. اولی مقدار جمع شده و دومی مقدار change که در قسمت بالا توضیح داده شد. برای حل این مشکل از یک MUX چهار بیتی استفاده میکنیم.

# final circuit of step4



# STEP FIVE

در این مرحله میخواهیم یک ورودی دیگر نیز اضافه کنیم تا چهار ورودی داشته باشیم. اگر بخواهیم هم بر پایه سنت هم بر پایه دلار(۱۰۰ سنت) ورودی بگیریم بهتر است تا اعداد را به جای چهار بیت در هشت بیت نمایش بدم. در این صورت همهی دیوایس های استفاده شده که شامل: (register, adder, comparator) هشت بیتی میشوند:



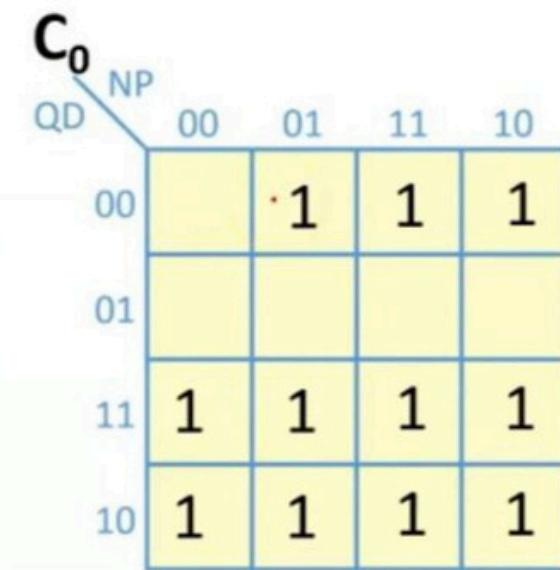
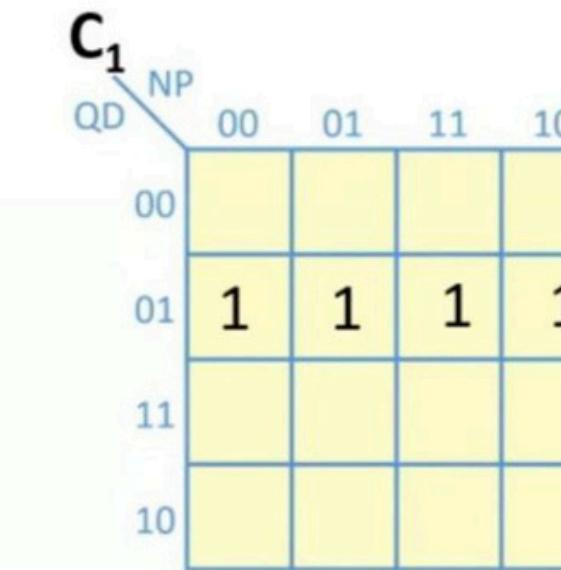
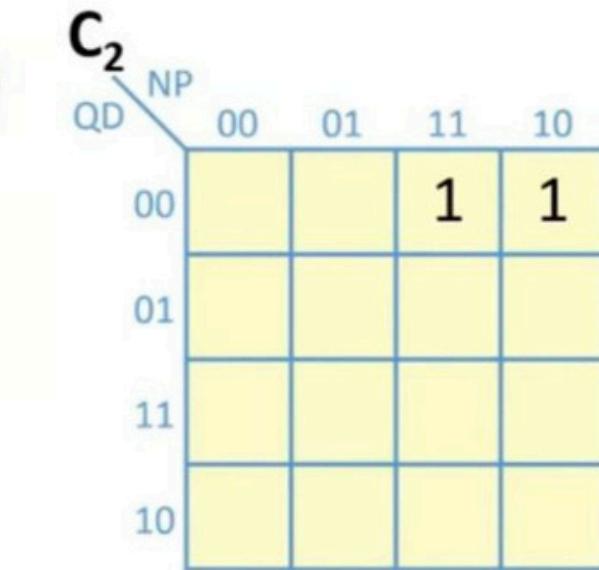
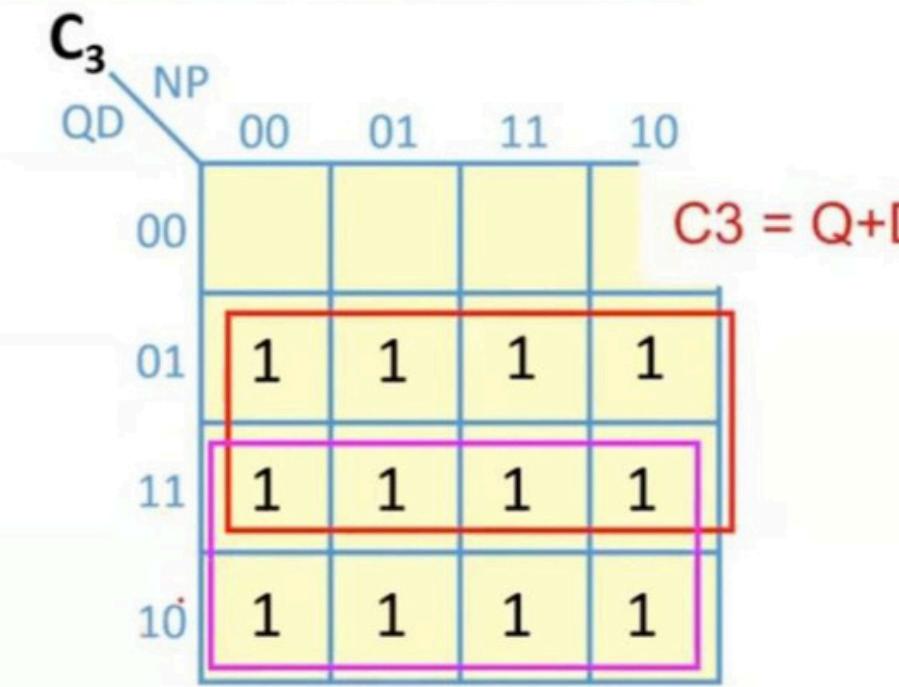
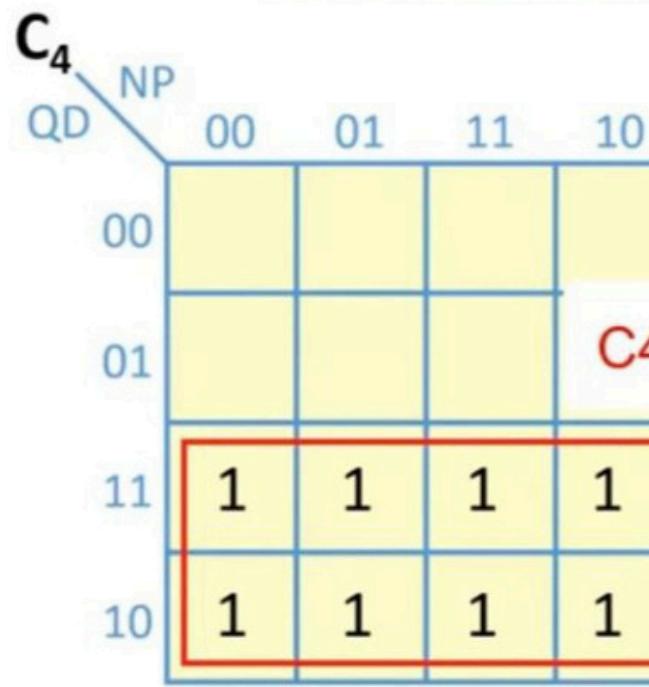
برای گرفتن چهار ورودی به یک encoder نیاز داریم که به این شکل بدست می‌آید:

## ENCODER FUNCTION TABLE (WITH PRIORITY)

Q	D	N	P	\$.	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
0	0	0	0	00	0	0	0	0	0
0	0	0	1	01	0	0	0	0	1
0	0	1	X	05	0	0	1	0	1
0	1	X	X	10	0	1	0	1	0
1	X	X	X	25	1	1	0	0	1

## PRIORITY ENCODER CONT.

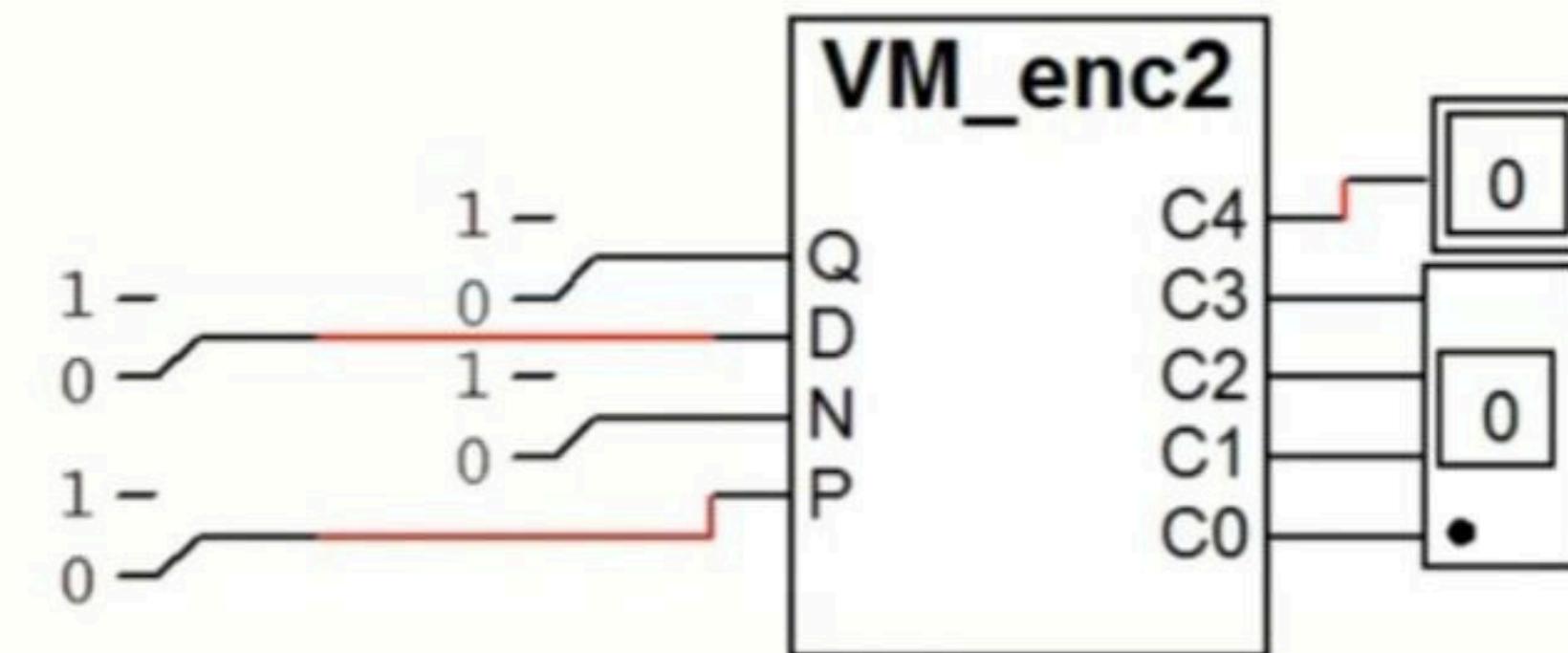
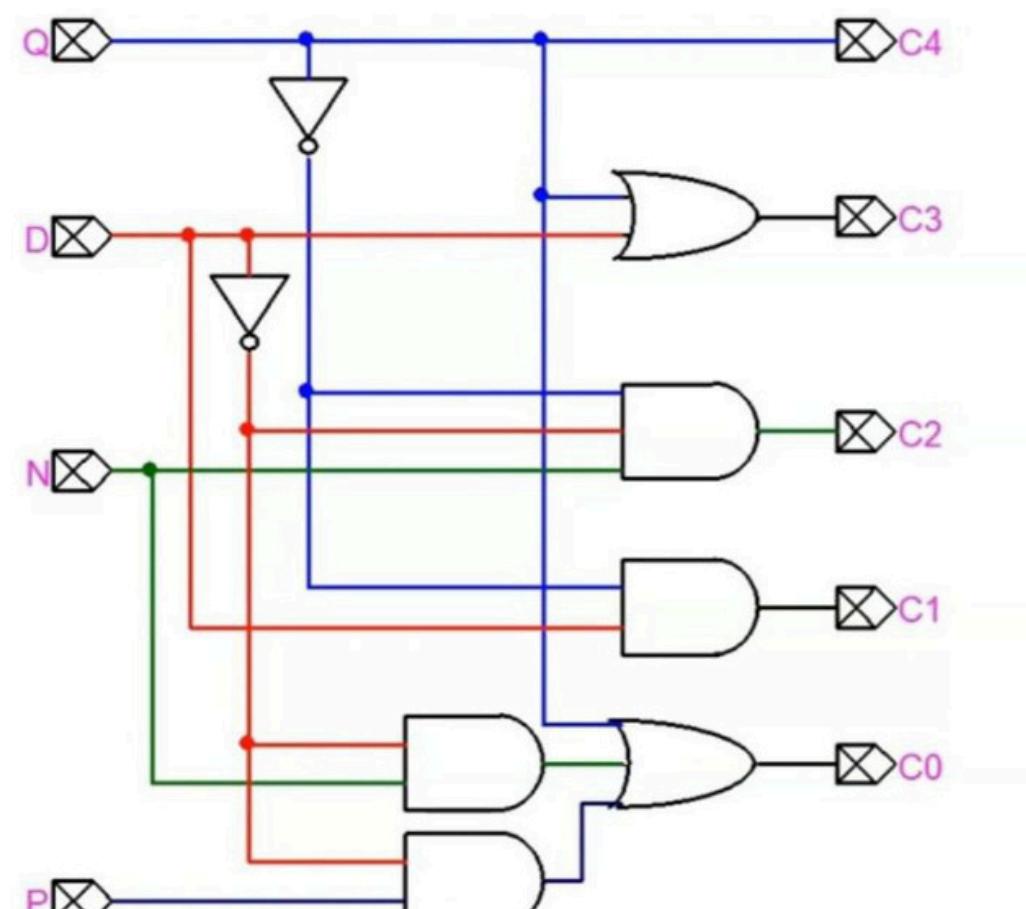
Q	D	N	P	\$.	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
0	0	0	0	00	0	0	0	0	0
0	0	0	1	01	0	0	0	0	1
0	0	1	X	05	0	0	1	0	1
0	1	X	X	10	0	1	0	1	0
1	X	X	X	25	1	1	0	0	1



## PRIORITY ENCODER CIRCUIT

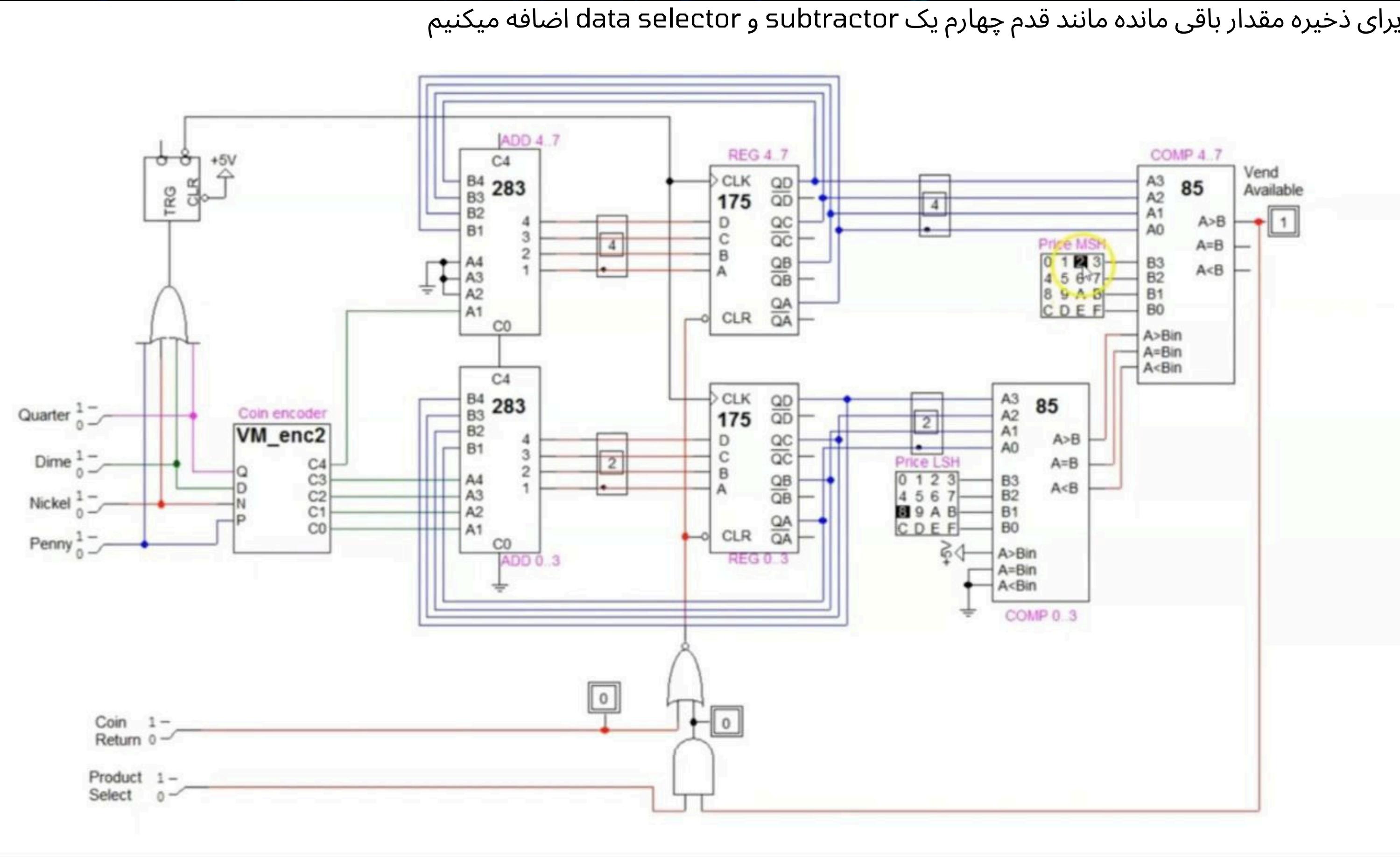
$$\begin{aligned}C_4 &= Q \\C_3 &= Q + D \\C_2 &= Q'D'N \\C_1 &= Q'D \\C_0 &= Q + D'N + D'P\end{aligned}$$

Q	D	N	P	\$.	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>
0	0	0	0	00	0	0	0	0	0
0	0	0	1	01	0	0	0	0	1
0	0	1	X	05	0	0	1	0	1
0	1	X	X	10	0	1	0	1	0
1	X	X	X	25	1	1	0	0	1

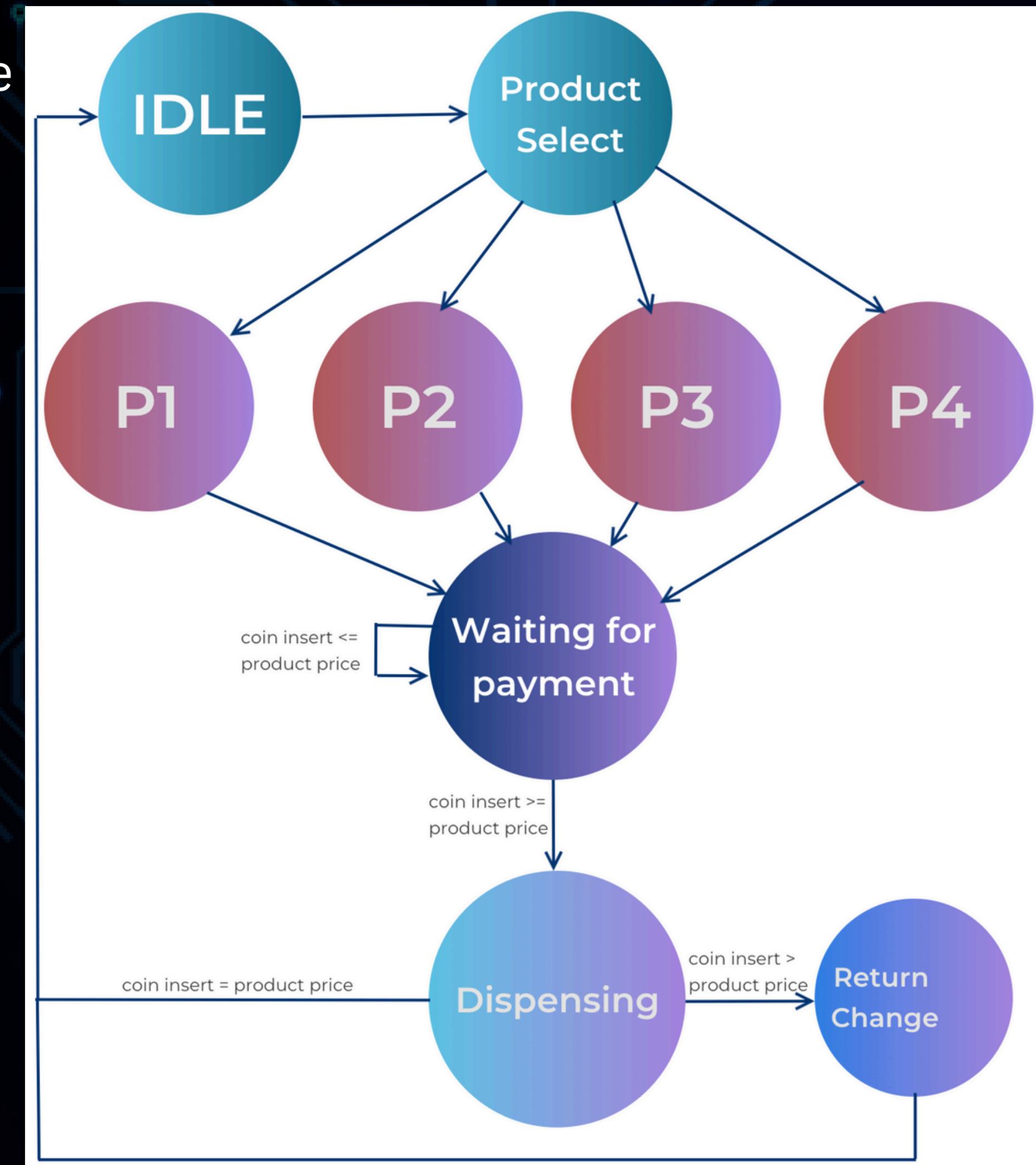


# final circuit of step5

یرای ذخیره مقدار باقی مانند قدم چهارم یک data selector و subtractor اضافه میکنیم



Here is a simple state diagram for our Vending machine:



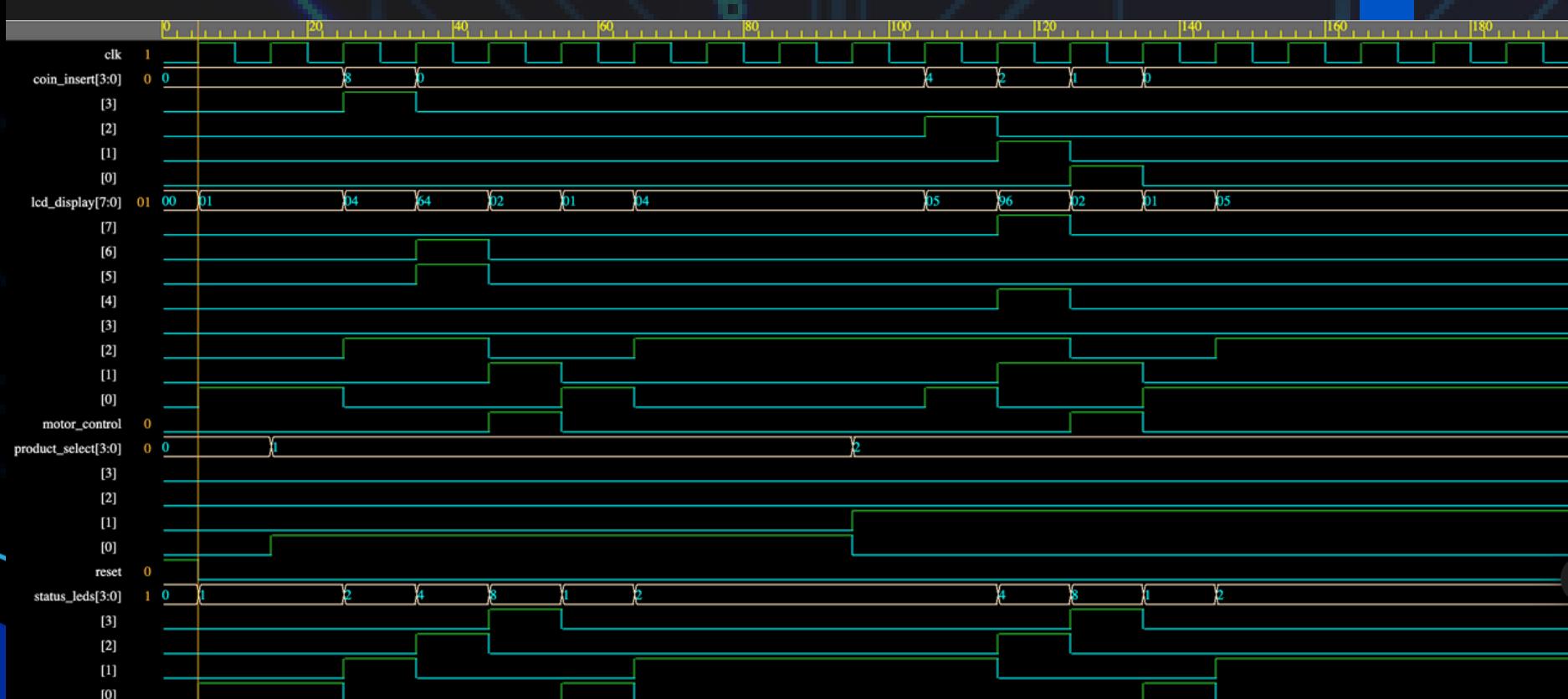
```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```

مرحله يك وندينگ ماشين ريست است  
پس در state IDLE هستيم  
و Lsd هم اين state را نشان ميدهد

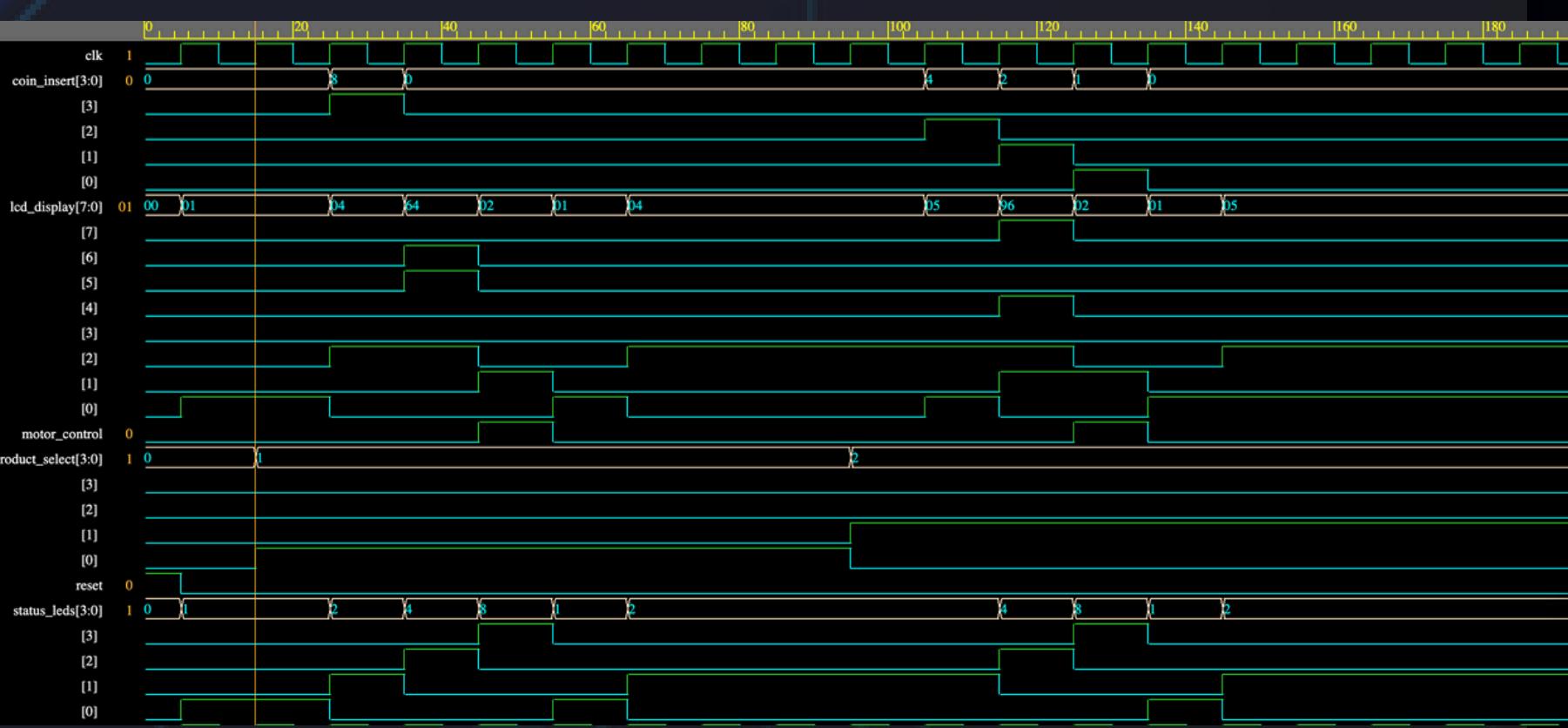


```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```



مرحله دو محصول اول را انتخاب  
ميكنيم و ماشين منتظر ميماند تا  
سکه وارد شود

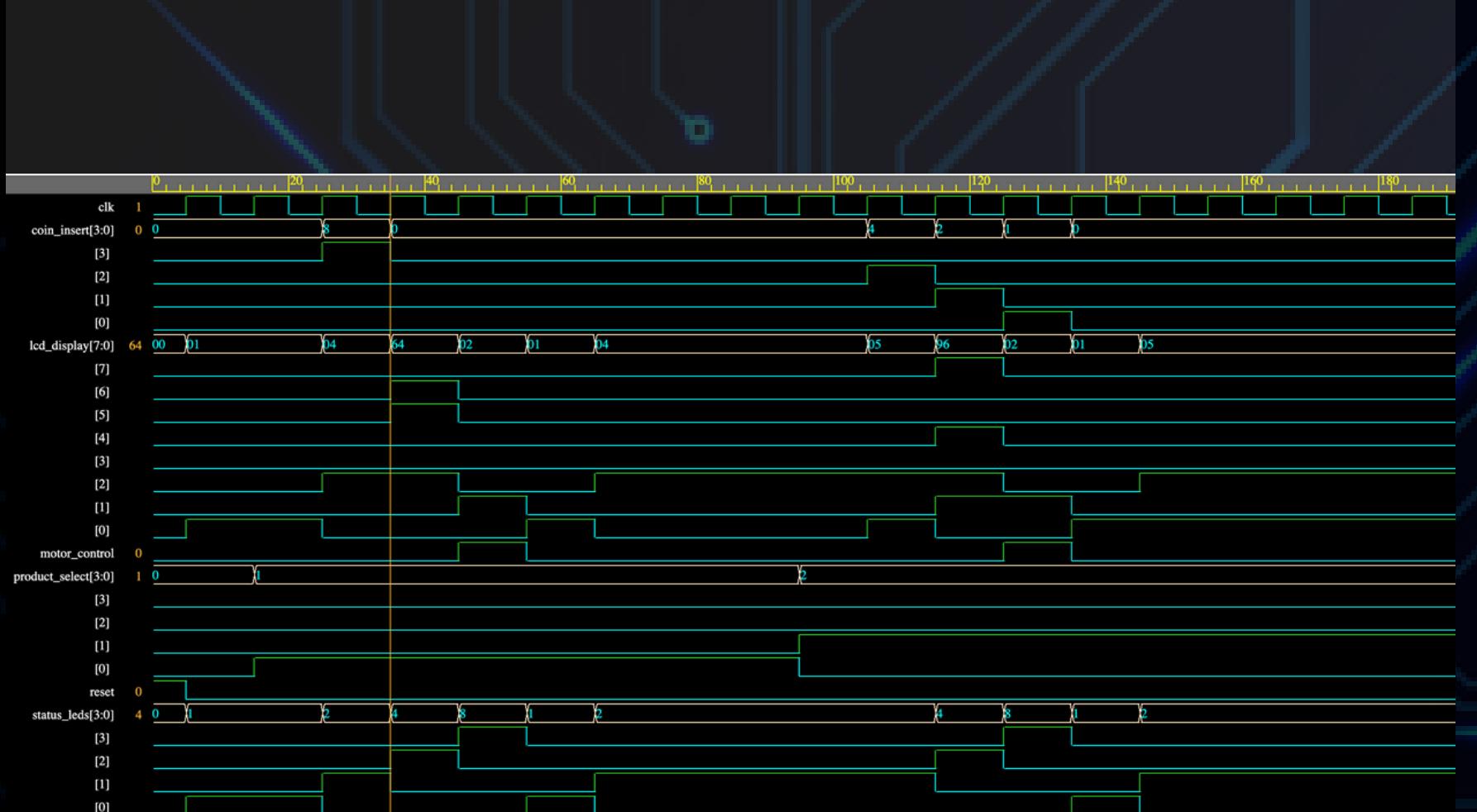
```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```

در مرحله سه یک دلار وارد ماشین میکنیم و منتظر کلاک بعدی میشویم. در این مرحله محصولی که انتخاب کردیم نمایش داده میشود.



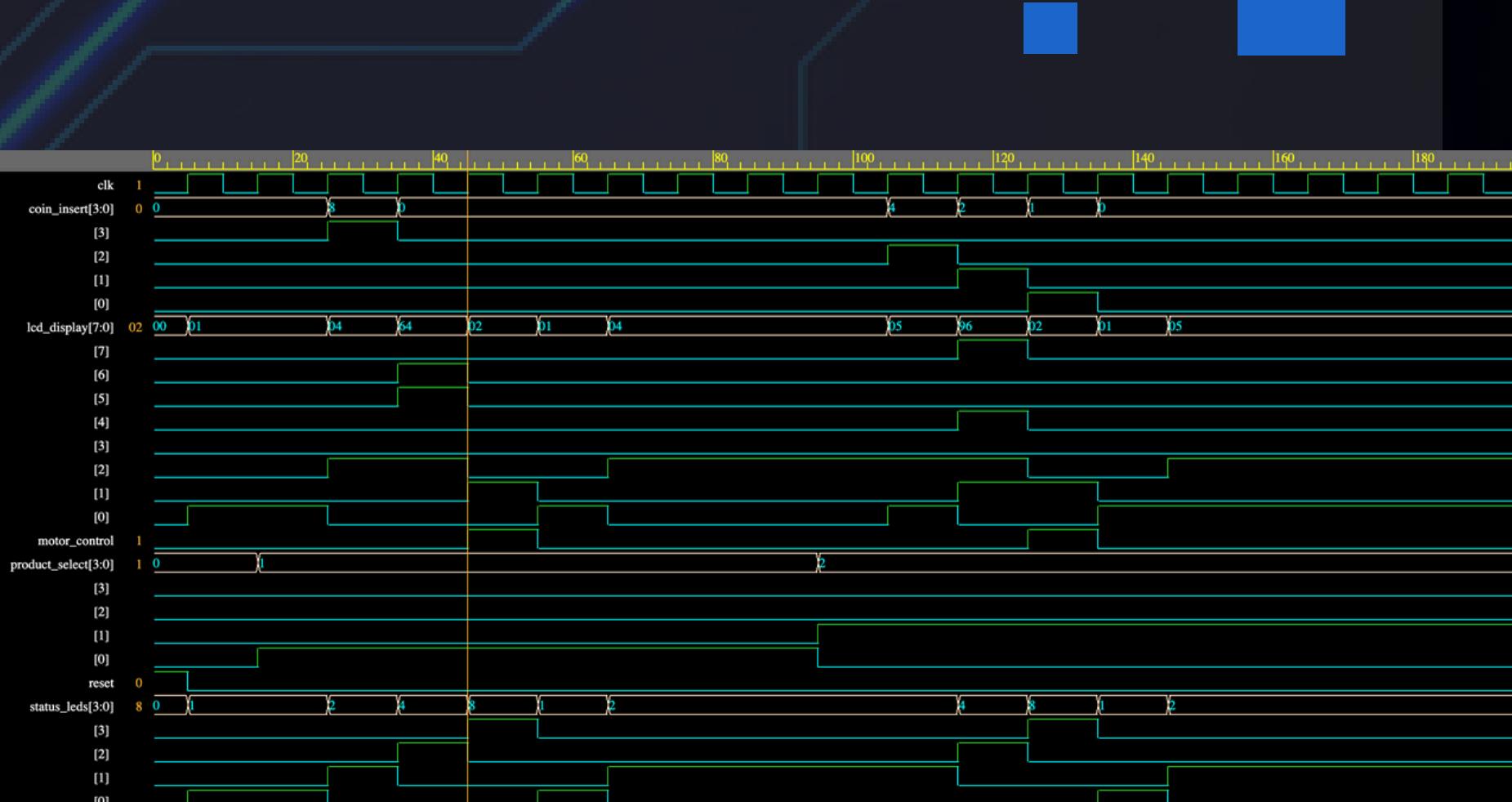
```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```

در مرحله چهار پس از خوردن کلاک dispensing قبل به حالت میرود و محصول ما را برمیگرداند. به دلیل dispensing موتور کنترل هم یک میشود.



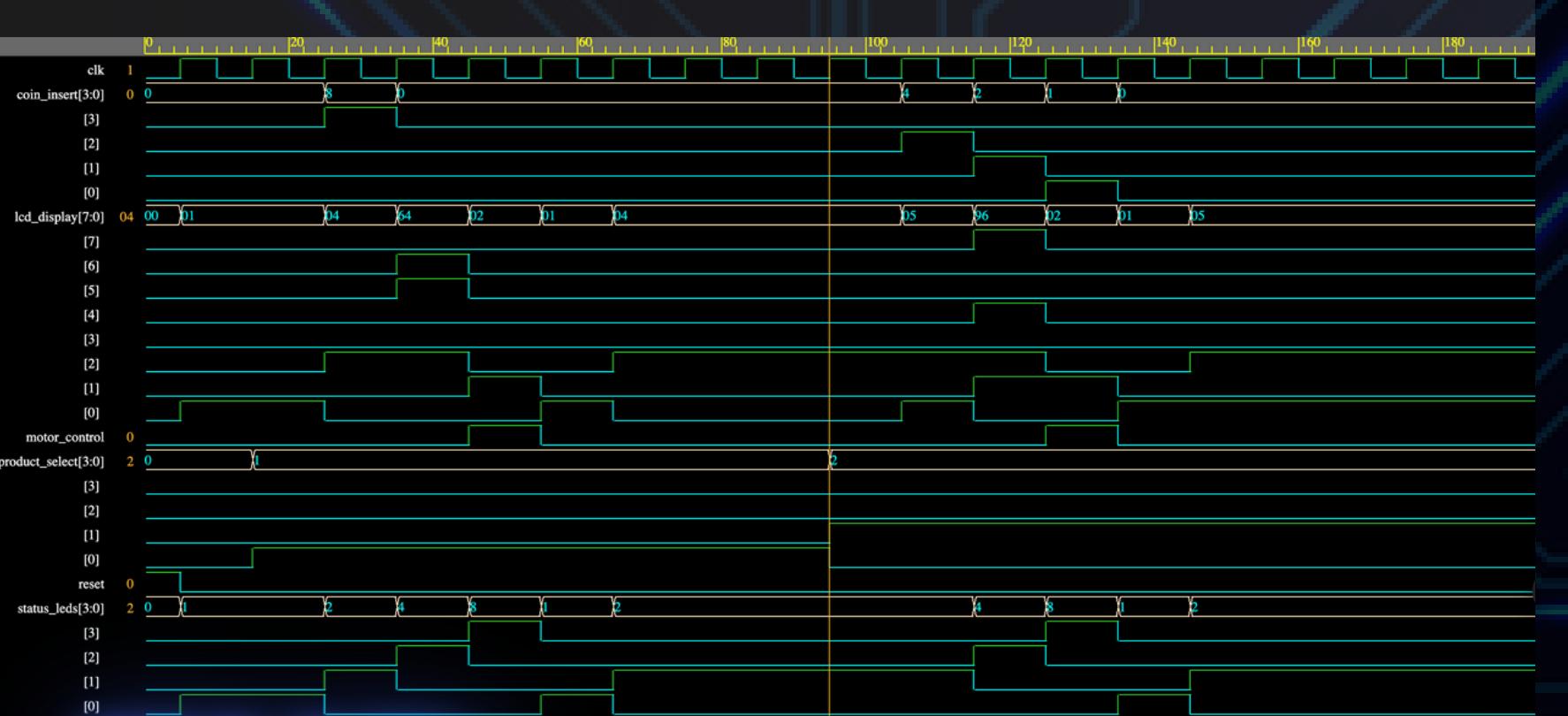
```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```

در مرحله پنجم یک محصول دیگر (محصول شماه دو) را انتخاب میکنیم و سه سکه را به ترتیب وارد میکنیم (مرحله شش) و همان نتایج مراحل قبل را مشاهده میکنیم.

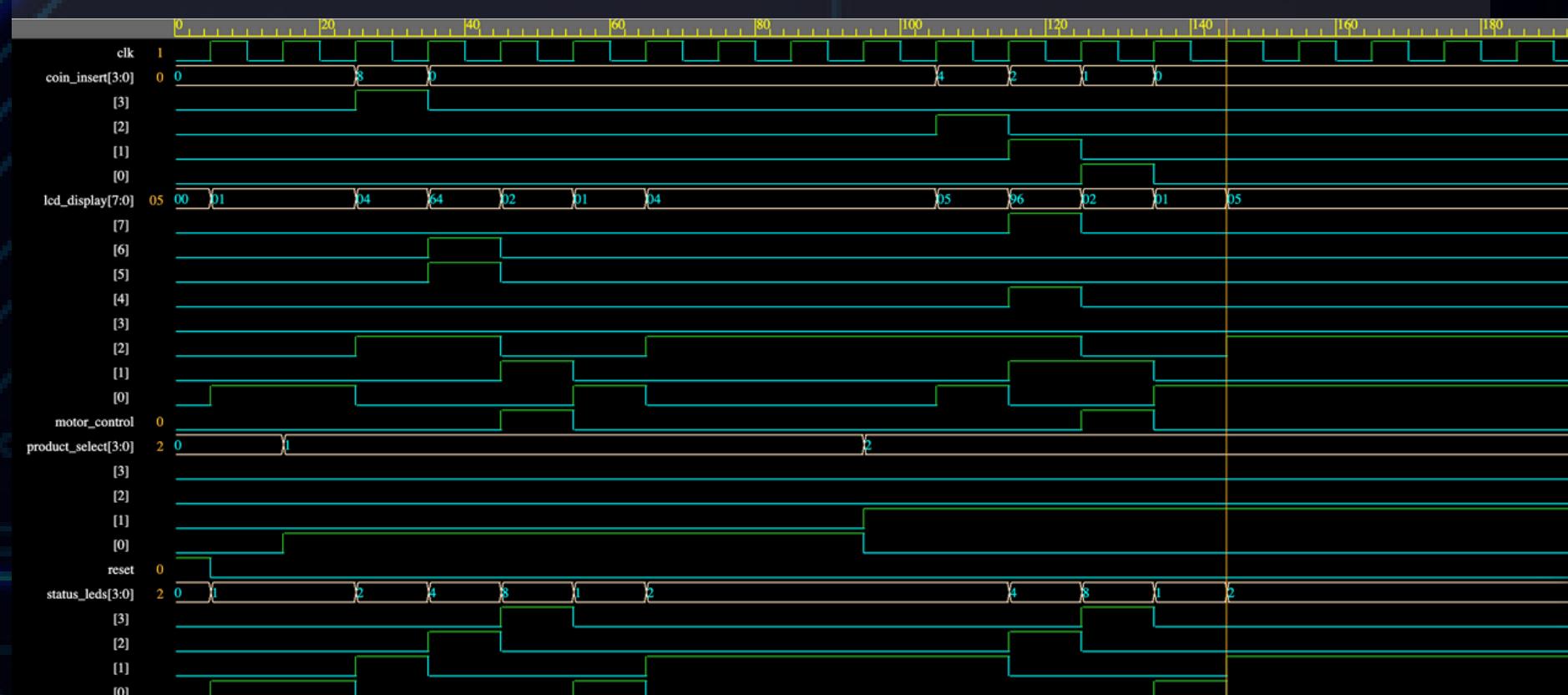


```

initial begin
    // VCD Dump
    $dumpfile("dump.vcd");
    $dumpvars(0, vending_machine_tb);

    clk = 0;
    reset = 1;
    product_select = 0;
    coin_insert = 0;
    1 #5 reset = 0;
    2 #10 product_select = 4'b0001; // Select product P1
    3 #10 coin_insert = 4'b1000; // Insert $1 coin
    4 #10 coin_insert = 0;
    #50; // Wait for dispensing
    5 #10 product_select = 4'b0010; // Select product P2
    6 #10 coin_insert = 4'b0100; // Insert 50¢ coin
    #10 coin_insert = 4'b0010; // Insert 20¢ coin
    #10 coin_insert = 4'b0001; // Insert 10¢ coin
    #10 coin_insert = 0;
    7 #50; // Wait for dispensing
    #10 $finish;
end

```



مرحله هفتم dispensing رد شده و دوباره به حالت ریست و IDLE میرویم.

**END**

3UN 2024

توضیحات خط به خط کد به صورت کامنت در خود فایل کد...