

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



شبیه سازی شی گرای سیستم های الکترونیکی

CA3

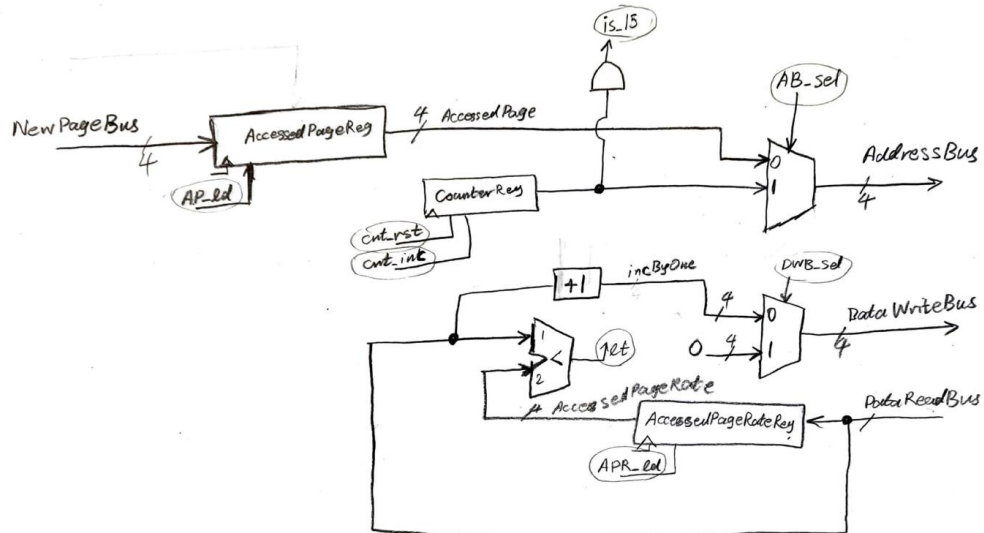
علیرضا جابری راد

810196438

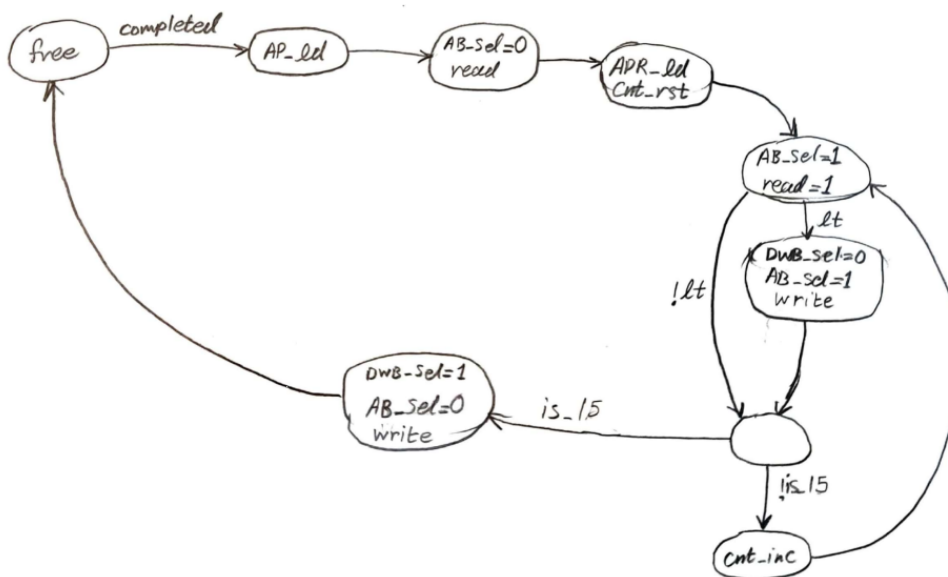
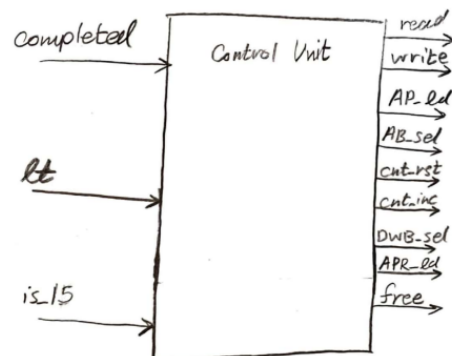
بهار 1401

بخش A تا C

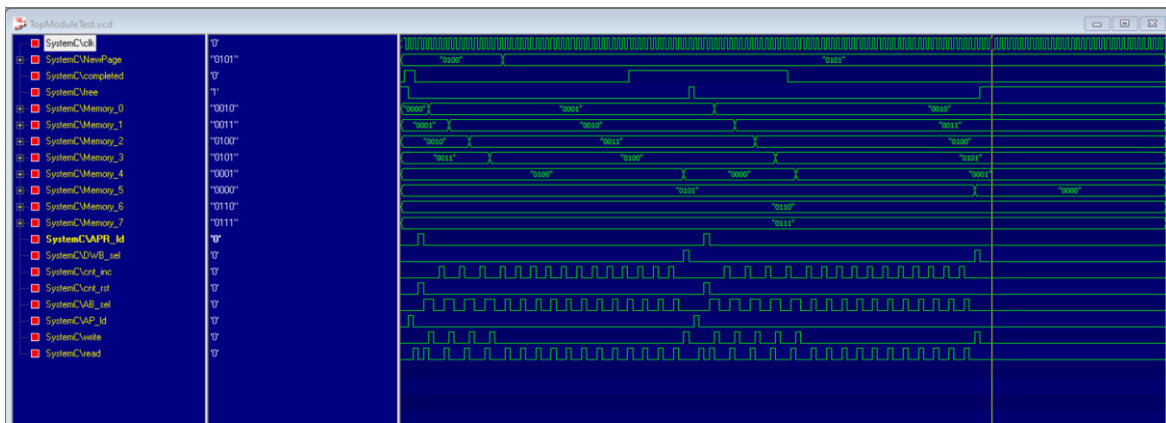
Datapath



Controller



در کد نوشته شده، ابتدا مموری با مقادیر اولیه i به ازای آدرس i ام پر میشود و سپس به ترتیب دو پیچ آدرس 4 و 5 مورد دسترسی قرار میگیرند و مقادیر مموری مطابق خروجی زیر به درستی تغییر میکند. (برای آنکه خروجی VCD File خیلی شلوغ نشود فقط خطوط آدرس 0 تا 7 از مموری را نمایش دادم)



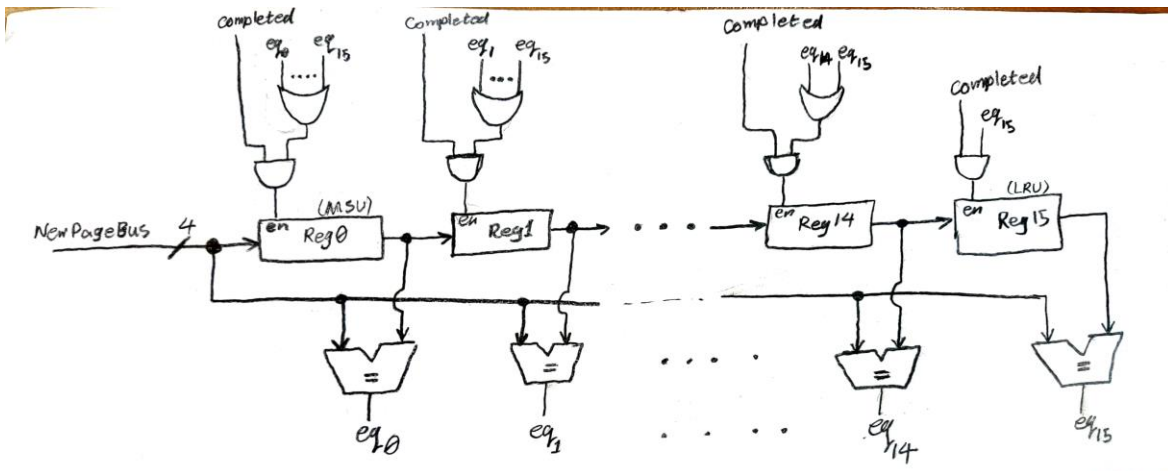
بخش اصلی کد تست بنچ نوشته شده که منجر به تولید خروجی بالا شده است نیز به ترتیب زیر است:

```

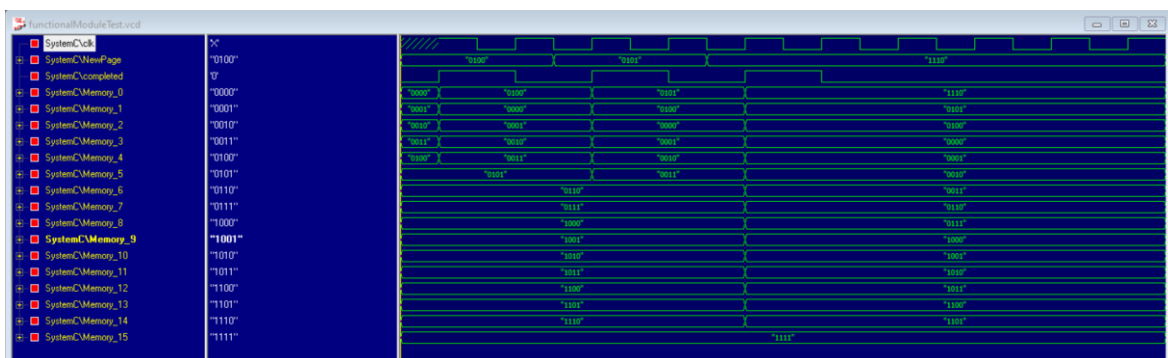
27 void testBench::newPageInsertion(){
28     while(true){
29         NewPageBus = sc_lv<4>("0100");
30         wait(200, SC_NS);
31         NewPageBus = sc_lv<4>("0101");
32         wait();
33     }
34 }
35
36 void testBench::start(){
37     while(true){
38         completed = sc_logic('0');
39         wait(7, SC_NS);
40         completed = sc_logic('1');
41         wait(21, SC_NS);
42         completed = sc_logic('0');
43         wait(420, SC_NS);
44         completed = sc_logic('1');
45         wait(310, SC_NS);
46         completed = sc_logic('0');
47         wait();
48     }
49 }
50
51 void testBench::clocking(){
52     while(true){
53         wait(5, SC_NS);
54         clk = sc_logic('1');
55         wait(5, SC_NS);
56         clk = sc_logic('0');
57     }
58 }

```

بخش D و E



همانطور که در تصویر زیر میبینید، کد functional نوشته شده بر اساس طراحی بالا، اگر سیگنال completed یک باشد و لبه ی بالارونده کلاک دیده شود، مقادیر همه رجیسترهای شیفت رجیستر بر اساس مقدار NewPageBus به طور همزمان آپدیت میشود. پس طبق طراحی جدید، در هر لبه بالارونده کلاک میتوان یک بار مقادیر رجیستر ها را آپدیت کرد.



بر اساس کد این بخش، آپدیت کردن رجیسترها فقط یک کلاک طول میکشد، در حالی که در طراحی بخش c آپدیت کردن مموری حدود 60 سیکل (در بدترین حالت 68 سیکل) طول میکشید.

بخش اصلی کد تست بنچ نوشته شده برای این بخش به ترتیب زیر است:

```
25 void testBench::newPageInsertion(){
26     while(true){
27         NewPageBus = sc_lv<4>("0100");
28         wait(20, SC_NS);
29         NewPageBus = sc_lv<4>("0101");
30         wait(20, SC_NS);
31         NewPageBus = sc_lv<4>("1110");
32         wait();
33     }
34 }
35
36 void testBench::start(){
37     while(true){
38         completed = sc_logic('0');
39         wait(5, SC_NS);
40         completed = sc_logic('1');
41         wait(10, SC_NS);
42         completed = sc_logic('0');
43         wait(10, SC_NS);
44         completed = sc_logic('1');
45         wait(10, SC_NS);
46         completed = sc_logic('0');
47         wait(10, SC_NS);
48         completed = sc_logic('1');
49         wait(10, SC_NS);
50         completed = sc_logic('0');
51         wait();
52     }
53 }
54
55 void testBench::clocking(){
56     while(true){
57         wait(5, SC_NS);
58         clk = sc_logic('1');
59         wait(5, SC_NS);
60         clk = sc_logic('0');
61     }
62 }
```