



گزارشکار آزمایش پنجم درس آزمایشگاه معماری

اعضای گروه:

عرشیا آیین نژاد، حوریه سبزواری، الناز رضایی

هدف آزمایش:

شیفت رجیسترها با اتصال فلیپ فلاپ‌ها به یکدیگر ساخته می‌شوند. از آنجایی که فلیپ فلاپ‌ها خاصیت حافظه‌ای دارند، در يك شیفت رجیستر بسیار مورد استفاده قرار می‌گیرند. شیفت رجیسترها معمولاً برای ذخیره‌سازی اطلاعات به صورت موقت استفاده می‌شوند. يك نمونه پرکاربرد از شیفت رجیستر را میتوان در هنگام کار با ماشین حساب‌ها دید. با وارد کردن يك رقم روی صفحه کلید، عددها به سمت چپ صفحه نمایشگر انتقال می‌یابند. به عبارت دیگر برای وارد کردن عدد ۲۶۸ باید به این صورت عمل کرد که ابتدا عدد ۲ را روی صفحه کلید فشار داده و رها نمود. سپس عدد ۶ را روی صفحه کلید فشار داده و رها نمود که این کار سبب می‌شود رقم ۲ يك مکان به سمت چپ انتقال یابد و عدد ۶ میتواند در منتهی‌الیه سمت راست قرار گیرد و عدد ۲۶ روی نمایشگر ظاهر شود. در پایان با فشار دادن و رها نمودن عدد ۸ بر روی صفحه کلید، عدد ۲۶۸ روی نمایشگر ظاهر خواهد شد. این مثال دو ویژگی مهم يك شیفت رجیستر را نشان میدهد. نخست آنکه شیفت رجیستر يك حافظه موقت است و بنابراین اعداد را بر روی صفحه نمایش نگه میدارد (حتی پس از رها نمودن دکمه مورد نظر بر روی صفحه کلید). دوم آنکه با هر بار فشار دادن يك رقم جدید بر روی صفحه کلید، شیفت رجیستر، اعداد را به سمت چپ انتقال می‌دهد. این ویژگی‌های انتقالی و حافظه-ای سبب می‌شود که شیفت رجیستر در بسیاری از سیستم‌های دیجیتال مورد استفاده قرار گیرد. يك روش تشریح ویژگی‌های يك شیفت رجیستر چگونگی بارگذاری داده به رجیستر و خواندن داده از واحدهای ذخیره است. به این ترتیب، چهار دسته از شیفت رجیسترها عبارتند از:

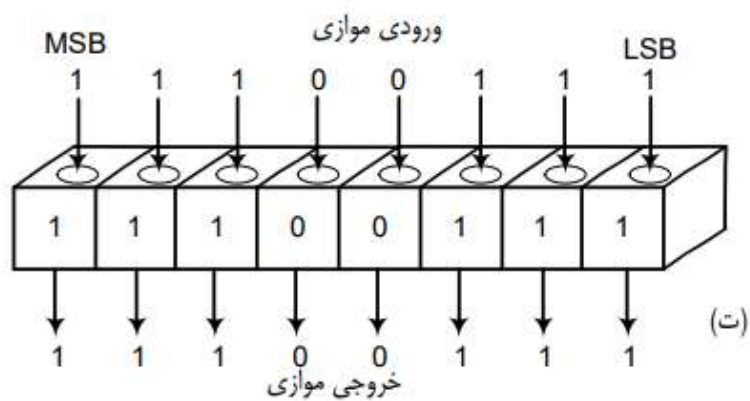
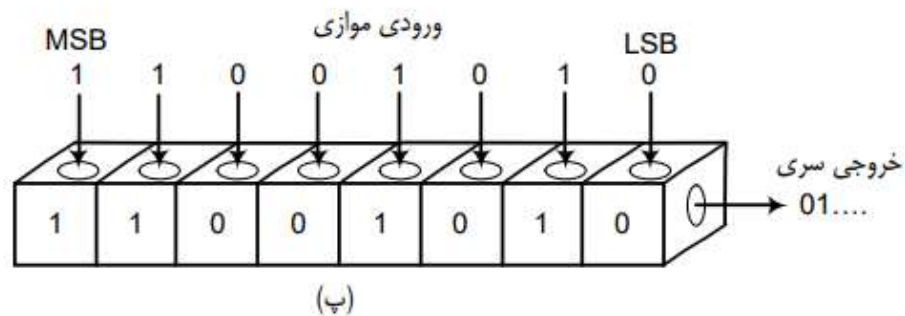
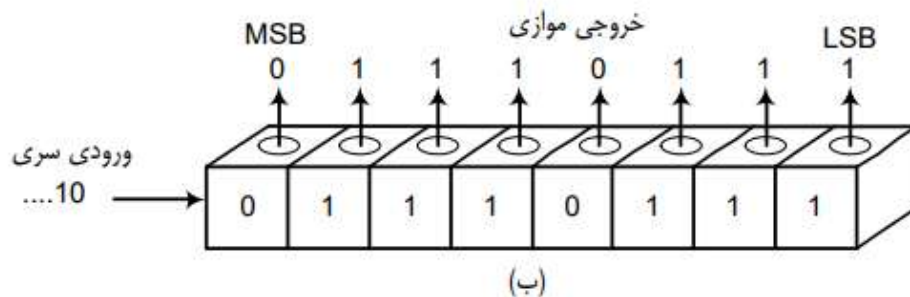
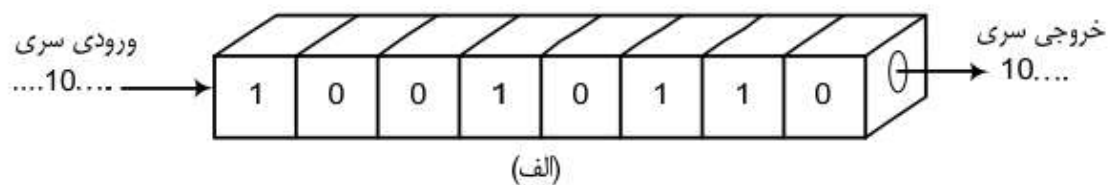
(۱) ورودی سری – خروجی سری

(۲) ورودی سری – خروجی موازی

(۳) ورودی موازی – خروجی سری

(۴) ورودی موازی – خروجی موازی

نحوه کارکرد این چهار دسته از ثبات‌ها در شکل ۱ نشان داده شده‌اند.

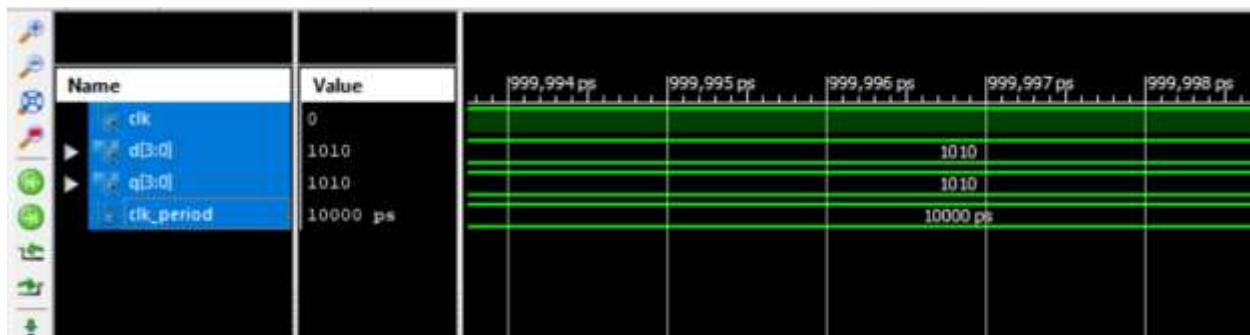
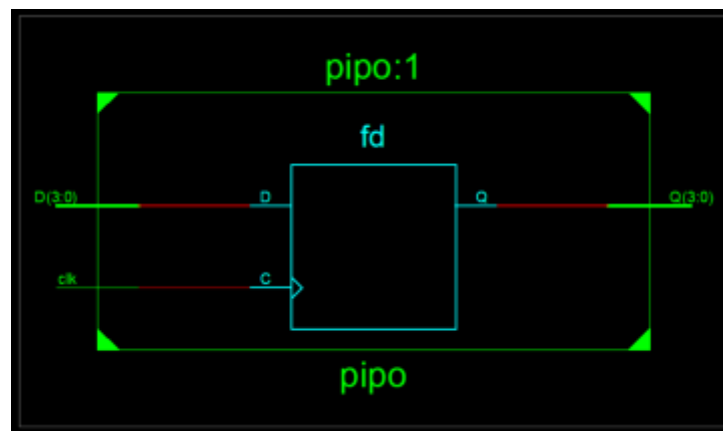


شکل ۱: مشخصه‌های یک شیفت رجیستر. (الف) ورودی سری - خروجی سری. (ب) ورودی سری - خروجی موازی. (پ) ورودی موازی - خروجی سری. (ت) ورودی موازی - خروجی موازی

بخش اول:

(۱) شیفت رجیستر ۴ بیتی در مد PIPO :

```
20 library ieee;
21 use ieee.std_logic_1164.all;
22
23 entity pipo is
24   port(
25     clk : in std_logic;
26     D: in std_logic_vector(3 downto 0);
27     Q: out std_logic_vector(3 downto 0)
28   );
29 end pipo;
30
31 architecture arch of pipo is
32
33   begin
34
35     process (clk)
36     begin
37       if (CLK'event and CLK='1') then
38         Q <= D;
39       end if;
40     end process;
41
42 end arch;
```

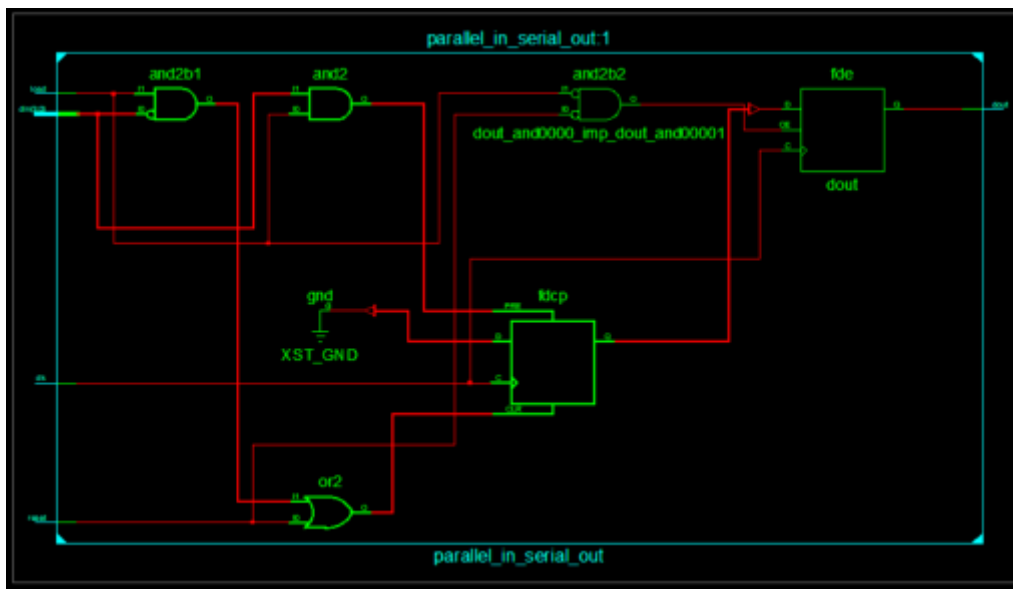


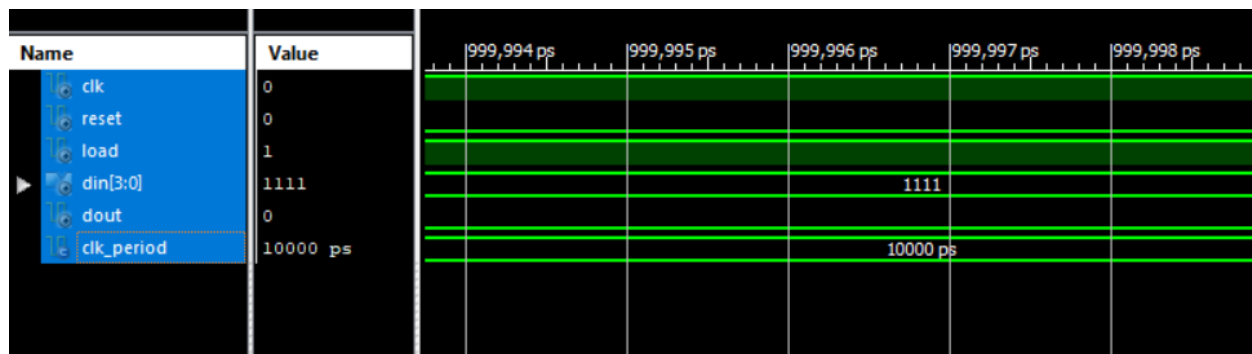
(۲) شیفت رجیستر ۴ بیتی در مد PISO :

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.all;
22
23 entity parallel_in_serial_out is
24     port(
25         clk : in STD_LOGIC;
26         reset : in STD_LOGIC;
27         load : in STD_LOGIC;
28         din : in STD_LOGIC_VECTOR(3 downto 0);
29         dout : out STD_LOGIC
30     );
31 end parallel_in_serial_out;
32
33
34 architecture piso_arc of parallel_in_serial_out is
35 begin
36
37     piso : process (clk,reset,load,din) is
38         variable temp : std_logic_vector (din'range);
39     begin
40         if (reset='1') then
41             temp := (others=>'0');
42         elsif (load='1') then
43             temp := din ;
44         elsif (rising_edge (clk)) then
45             dout <= temp(3);
46             temp := temp(2 downto 0) & '0';
47         end if;
48     end process piso;
49
50 end piso_arc;

```



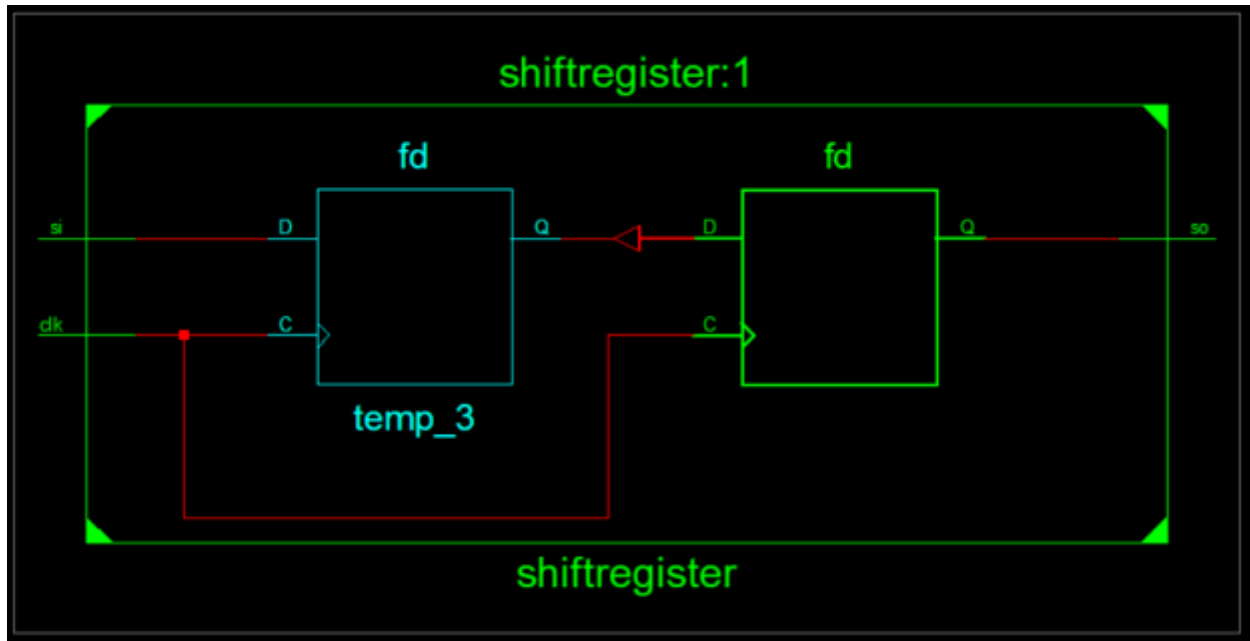


۳) شیفت رجیستر ۴ بیتی در مد SISO :

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_ARITH.ALL;
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25
26 entity shiftregister is
27     Port ( si : in  STD_LOGIC;
28           clk : in  STD_LOGIC;
29           so : out  STD_LOGIC);
30 end shiftregister;
31
32 architecture Behavioral of shiftregister is
33
34     signal temp : std_logic_vector(3 downto 0);
35
36 begin
37     process(clk)
38     begin
39         if(clk'event and clk='1')then
40             temp <= si & temp(3 downto 1);
41         end if;
42     end process;
43     so <= temp(0);
44 end Behavioral;

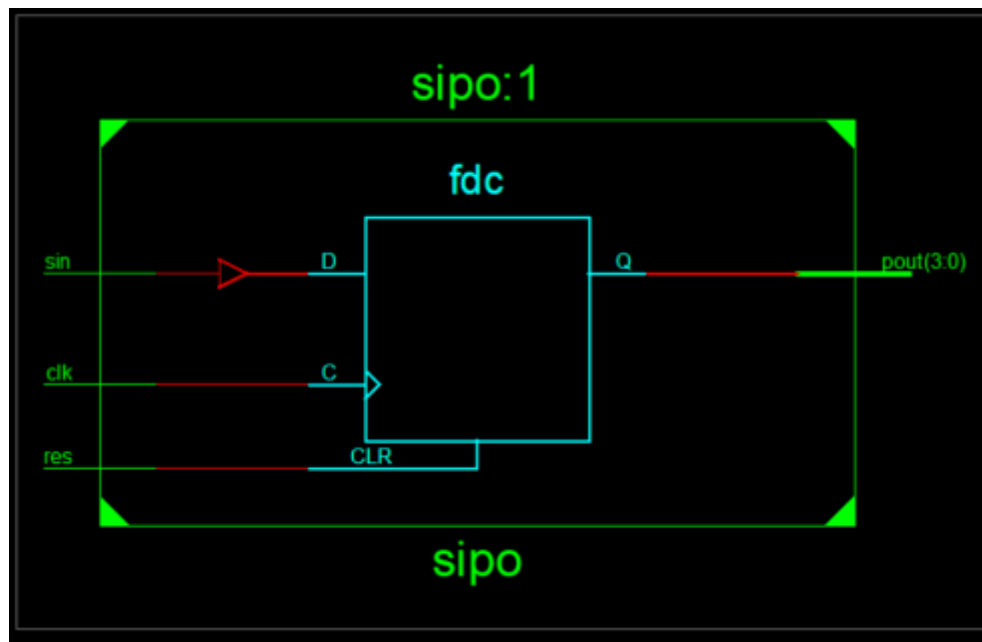
```

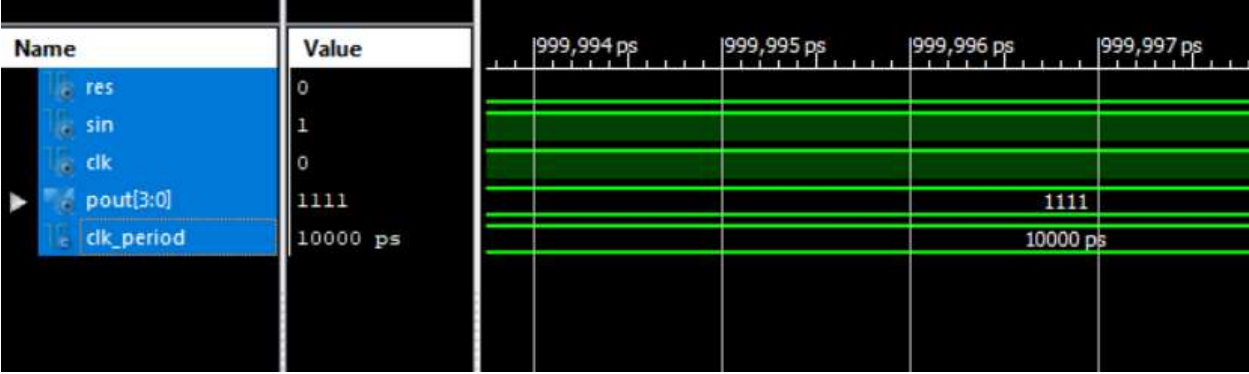


Name	Value	1999,994 ps	1999,995 ps	1999,996 ps	1999,997
si	0				
clk	0				
so	0				
clk_period	10000 ps			10000 ps	

۴) شیفت رجیستر ۴ بیتی در مد SIPO :

```
20 library ieee;
21 use ieee.std_logic_1164.all;
22 entity sipo is
23 port( res: in std_logic;
24       sin: in std_logic;
25       clk: in std_logic;
26       pout: out std_logic_vector(3 downto 0));
27 end sipo;
28
29 architecture beh of sipo is
30     signal temp: std_logic_vector( 3 downto 0);
31     begin
32     process( clk, res)
33     begin
34         if(res='1') then
35             temp<="0000";
36         elsif (clk'event and clk ='1') then
37             temp(3)<=sin;
38             temp(2)<=temp(3);
39             temp(1)<=temp(2);
40             temp(0)<=temp(1);
41         end if;
42     end process;
43     pout<=temp;
44 end beh;
```





بخش دوم: کاربرد ثبات‌ها در کامپیوتر را بنویسید.

پاسخ:

یک ثبات یا رجیستر حاوی مجموعه‌ای از فلیپ‌فلاپ‌ها است. یک فلیپ‌فلاپ برای ذخیره داده‌های تک بیتی استفاده می‌شود. به منظور ذخیره تعداد بیت بالاتر، ظرفیت ذخیره‌سازی با افزایش فلیپ‌فلاپ‌ها بیش‌تر می‌شود. برای ذخیره یک کلمه n بیتی نیاز به استفاده از یک ثبات n بیتی با n عدد فلیپ‌فلاپ است. از رجیستر برای اجرای عملیات مختلفی استفاده می‌شود. CPU برای اجرای عملیات، از ثبات‌ها استفاده می‌کند. سه نوع عملیات مختلف توسط ثبات‌ها انجام می‌شود:

- **واکشی (Fetch):** واکشی برای دریافت دستورات ارسال شده توسط کاربر و همچنین بیرون کشیدن دستورات ذخیره شده در حافظه اصلی استفاده می‌شود.
 - **کدگشایی (Decode):** عملیات کدگشایی برای تفسیر و تعبیر دستورات صورت می‌گیرد. در کدگشایی، عملیاتی که باید روی دستورات انجام شود، به وسیله CPU تعیین و مشخص می‌شود.
 - **اجرا (Execute):** عملیات اجرا در واقع ذخیره جواب تولید شده توسط CPU در حافظه است. پس از ذخیره شدن، نتایج در صفحه نمایش کاربر نشان داده می‌شوند.
- می‌توان گفت ثبات‌ها مهم‌ترین و کاربردی‌ترین حافظه در امر پردازش هستند؛ زیرا بسیاری از دستورات نمی‌توانند به صورت مستقیم توسط واحد پردازشی CPU پردازش شده و خروجی را تحویل دهند. این دستورات نیازمند پردازش دستور یا دستوراتی دیگر قبل از ارائه خروجی هستند. در این حالت، CPU با استفاده از واحد اسمبلر (Assamblar Unit)، به معنای واحد تبدیل‌کننده به اسمبلی (دستور اصلی را به مجموعه‌ای از دستورات تبدیل می‌کند. مجموعه دستورات بدست آمده، دستورات فرعی نام دارند.
- از جمله ثبات‌های معروف میتوان به Memory address register ، PC ، Accumulator register ، Index register ، Memory data register و .. اشاره کرد.