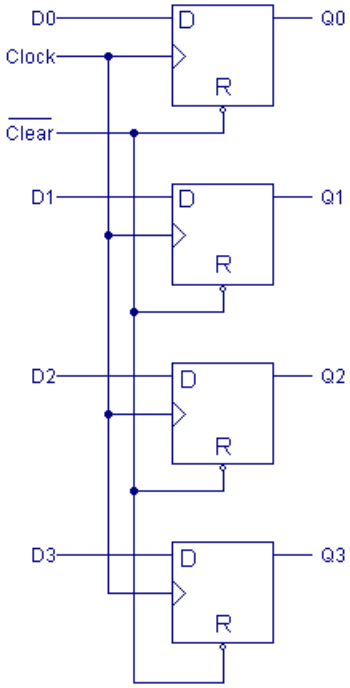
Лабораторийн ажил №4

Регистр

Зорилго: *Тоон системд өргөн хэрэглэгддэг санах төхөөрөмж регистрийн дизайныг хэрхэн VHDL код дээр хийж гүйцэтгэхийг судална.*

Регистр

Цуваа (sequintial) логик схемд мэдээлэл хадгалах элементээр төрөл бүрийн флип-флопууд, лач-ийг хэрэглэдэг бөгөөд ерөнхий клоктой хоёр буюу түүнээс дээш тооны флип-флопуудын цуглуулгыг регистр гэж нэрлэдэг. Тоон системд төрөл бүрийн регистрүүд өргөн хэрэглэгддэг ба тэдгээр нь өөрсдийн ажиллагаа, сигналуудаараа ялгагддаг. Тухайлбал: клок импульсээр урьдчилан тодорхойлсон төлвүүдийн дарааллыг цувуулан гаргадаг регистрийг тоолуур гэдэг.



Зураг 1.

Хэдийгээр тоолуур нь нэг төрлийн тусгай регистр боловч түүнийг регистрээс тусад нь ялгаж авч үздэг. Иймээс бид дараагийн лабораторийн ажлаар тоолуурын тухай үзэх болно.

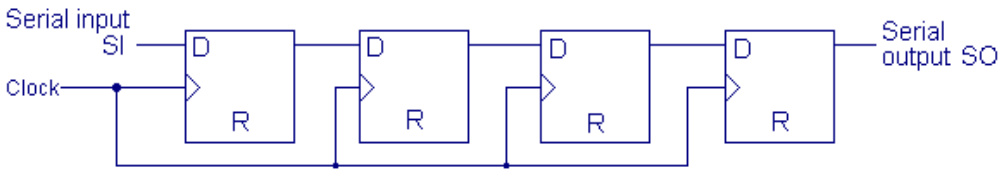
4 ширхэг D төрлийн флип-флопоос тогтсон хамгийн хялбар регистрийг зураг 1-д үзүүлэв. Ерөнхий Clock оролтонд ирэх импульс бүрийн өсөх фронтоор бүх флип-флопууд триггер хийгдэх ба D оролтууд дээрх өгөгдлүүд 4 бит регистр рүү шилжинэ. Clear ерөнхий оролт нь регистрийг тэглэхэд хэрэглэгдэх ба логик ‘0’-ээр идэвхжих асинхрон сигнал байна. Энэхүү регистрийн VHDL кодыг дараахь байдлаар бичиж болно.

```
library ieee;
use ieee.std_logic_1164.all;
entity rg_4 is
  port(clk, clear : in std_logic;
       D : in  std_logic_vector(3 downto 0);
       Q : out std_logic_vector(3 downto 0));
end rg_4;

architecture behavioral of rg_4 is
signal Q_i : std_logic_vector(3 downto 0);
begin
 process(clk, clear)
 begin
   if clear='0' then Q_i <= (others => '0');
   elsif (clk'event and clk='1') then Q_i <= D;
   end if;
  Q <= Q_i;
 end process;
end behavioral;
```

Шифт регистр

Хадгалж байгаа битүүдийнхээ байрыг нэг чиглэлд эсвэл хоёр чиглэлд хоёуланд нь шилжүүлэх чадвартай регистрийг шифт регистр гэдэг. Шифт регистрийн бүтэц нь нэг флип-флопийн гаралт нь дараагийн флип-флопийн оролтонд холбогдох маягаар шатлан холбогдсон флип-флопуудын гинж хэлбэртэй байна. Бүх флип-флопууд нь ерөнхий клок импульсээр идэвхжин, нэг шатаасаа дараагийнх руугаа шифт хийгдэнэ. Хялбар шифт регистрийн жишээг зураг 2-д харуулав.



Зураг 2.

Шифт регистрүүдийг хоорондоо зайтай байгаа тоон системүүдийн интерфейс-д өргөн хэрэглэдэг. Жишээ нь хоёр цэгийн хооронд n-бит тоог дамжуулах болжээ. Хэрэв хоорондох зай нь хол бол n-битийг n-бит шугам хэрэглэн параллелиар дамжуулах нь үнэтэй болно. Тэгвэл энэхүү мэдээллийг ганц шугам хэрэглэн бит битээр нь цуваагаар дамжуулбал илүү хэмнэлттэй болно. Энэ тохиолдолд өөрөөр хэлбэл өгөгдлийг параллелиас цуваа, цуваагаас параллель болгон хувиргахад шифт регистрийг хэрэглэдэг.
Өгөгдлөө зүүн тийш шифт хийдэг, асинхрон reset-тэй шифт регистрийн behavioral VHDL кодыг дараахь байдлаар бичиж харууллаа. Энд шифт үйлдэл хийхдээ concatenation оператор & -ийг хэрэглэсэн байгааг харж болно.

```
-- Асинхрон Reset-тэй 4 бит зүүн шифт регистр
library ieee;
use ieee.std_logic_1164.all;
entity srg_4_r is
  port(CLK, RESET, SI: in std_logic;
       Q : out std_logic_vector(3 downto 0);
       SO: out std_logic);
end srg_4_r;

architecture behavioral of srg_4_r is
signal shift : std_logic_vector(3 downto 0);
begin
process (RESET, CLK)
begin
  if (RESET='1') then
    shift <= "0000";
  elsif (CLK'event and (CLK='1')) then
    shift <= shift(2 downto 0) & SI;
  end if;
end process;
  Q <= shift;
  SO <= shift(3);
end behavioral;
```

Даалгавар

1. Дээрх хоёр жишээ кодыг ModelSim симулятор дээр шалгах testbench кодыг бичиж үр дүнг тэмдэглэж ав.
2. Нэмэлт параллель load оролттой 8 бит регистрийн VHDL кодыг бич. Өөрөөр хэлбэл load сигналаар зөвшөөрөгдсөн үед клок сигналын фронтоор оролтон дахь өгөгдлийг регистрт хадгалдаг гэсэн үг.
3. Дараахь хүснэгтэд нэмэлт ажиллагаатай 8 бит шифт регистрийн функцүүдийг харуулжээ. Үүнд харгалзах регистрийн кодыг зохиож түүний ажиллагааг шалгах testbench кодыг мөн бичиж үр дүнг харуул.

Функц	***Оролт***			***Дараагийн төлөв***							
	S2	**S1**	**S0**	**Q7***	**Q6***	**Q5***	**Q4***	**Q3***	**Q2***	**Q1***	**Q0***
Hold	0	0	0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
Load	0	0	1	D7	D6	D5	D4	D3	D2	D1	D0
Shift right	0	1	0	RIN	Q7	Q6	Q5	Q4	Q3	Q2	Q1
Shift left	0	1	1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	LIN
Shift circular right	1	0	0	Q0	Q7	Q6	Q5	Q4	Q3	Q2	Q1
Shift circular left	1	0	1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Q7
Shift arithmetic right	1	1	0	Q7	Q7	Q6	Q5	Q4	Q3	Q2	Q1
Shift arithmetic left	1	1	1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	0