Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
  port(C, SI : in  std_logic;
       SO : out std_logic);
end shift;

architecture archi of shift is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='1') then
      for i in 0 to 6 loop
        tmp(i+1) <= tmp(i);
      end loop;
      tmp(0) <= SI;
    end if;
  end process;
  SO <= tmp(7);
end archi;
```

Following is the VHDL code for an 8-bit shift-left register with a negative-edge clock, clock enable, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
  port(C, SI, CE : in  std_logic;
       SO : out std_logic);
end shift;

architecture archi of shift is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='0') then
      if (CE='1') then
        for i in 0 to 6 loop
          tmp(i+1) <= tmp(i);
        end loop;
        tmp(0) <= SI;
      end if;
    end if;
  end process;
  SO <= tmp(7);
end archi;
```
Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, asynchronous clear, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
    port(C, SI, CLR : in std_logic;
        SO : out std_logic);
end shift;

architecture archi of shift is
    signal tmp: std_logic_vector(7 downto 0);
    begin
        process (C, CLR)
        begin
            if (CLR='1') then
                tmp(0) = (others => '0');
            elsif (C'event and C='1') then
                tmp(0) = tmp(6 downto 0) & SI;
            end if;
        end process;
        SO <= tmp(7);
    end archi;
```

Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, synchronous set, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
    port(C, SI, S : in std_logic;
        SO : out std_logic);
end shift;

architecture archi of shift is
    signal tmp: std_logic_vector(7 downto 0);
    begin
        process (C, S)
        begin
            if (C'event and C='1') then
                if (S='1') then
                    tmp <= (others => '1');
                else
                    tmp <= tmp(6 downto 0) & SI;
                end if;
            end if;
        end process;
        SO <= tmp(7);
    end archi;
```
Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
    port(C, SI : in  std_logic;
            PO : out std_logic_vector(7 downto 0));
end shift;

architecture archi of shift is
    signal tmp: std_logic_vector(7 downto 0);
    begin
        process (C)
        begin
            if (C'event and C='1') then
                tmp <= tmp(6 downto 0) & SI;
            end if;
        end process;
        PO <= tmp;
    end archi;
end archi;
```

Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, asynchronous parallel load, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
    port(C, SI, ALOAD : in std_logic;
            D   : in std_logic_vector(7 downto 0);
            SO  : out std_logic);
end shift;

architecture archi of shift is
    signal tmp: std_logic_vector(7 downto 0);
    begin
        process (C, ALOAD, D)
        begin
            if (ALOAD='1') then
                tmp <= D;
            elsif (C'event and C='1') then
                tmp <= tmp(6 downto 0) & SI;
            end if;
        end process;
        SO <= tmp(7);
    end archi;
end archi;
```
Following is the VHDL code for an 8-bit shift-left register with a positive-edge clock, synchronous parallel load, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
  port(C, SI, SLOAD : in std_logic;
       D  : in std_logic_vector(7 downto 0);
       SO : out std_logic);
end shift;

architecture archi of shift is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='1') then
      if (SLOAD='1') then
        tmp <= D;
      else
        tmp <= tmp(6 downto 0) & SI;
      end if;
    end if;
  end process;
  SO <= tmp(7);
end archi;
```

Following is the VHDL code for an 8-bit shift-left/shift-right register with a positive-edge clock, serial in, and serial out.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity shift is
  port(C, SI, left_right : in std_logic;
       PO : out std_logic_vector(7 downto 0));
end shift;

architecture archi of shift is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='1') then
      if (left_right='0') then
        tmp <= tmp(6 downto 0) & SI;
      else
        tmp <= SI & tmp(7 downto 1);
      end if;
    end if;
  end process;
  PO <= tmp;
end archi;
```