Active-HDL PDF Export
cheraghe_hoshmand workspace

Contents

2 cheragh_hoshmand

2.1 cherag_hoshmand.vhd

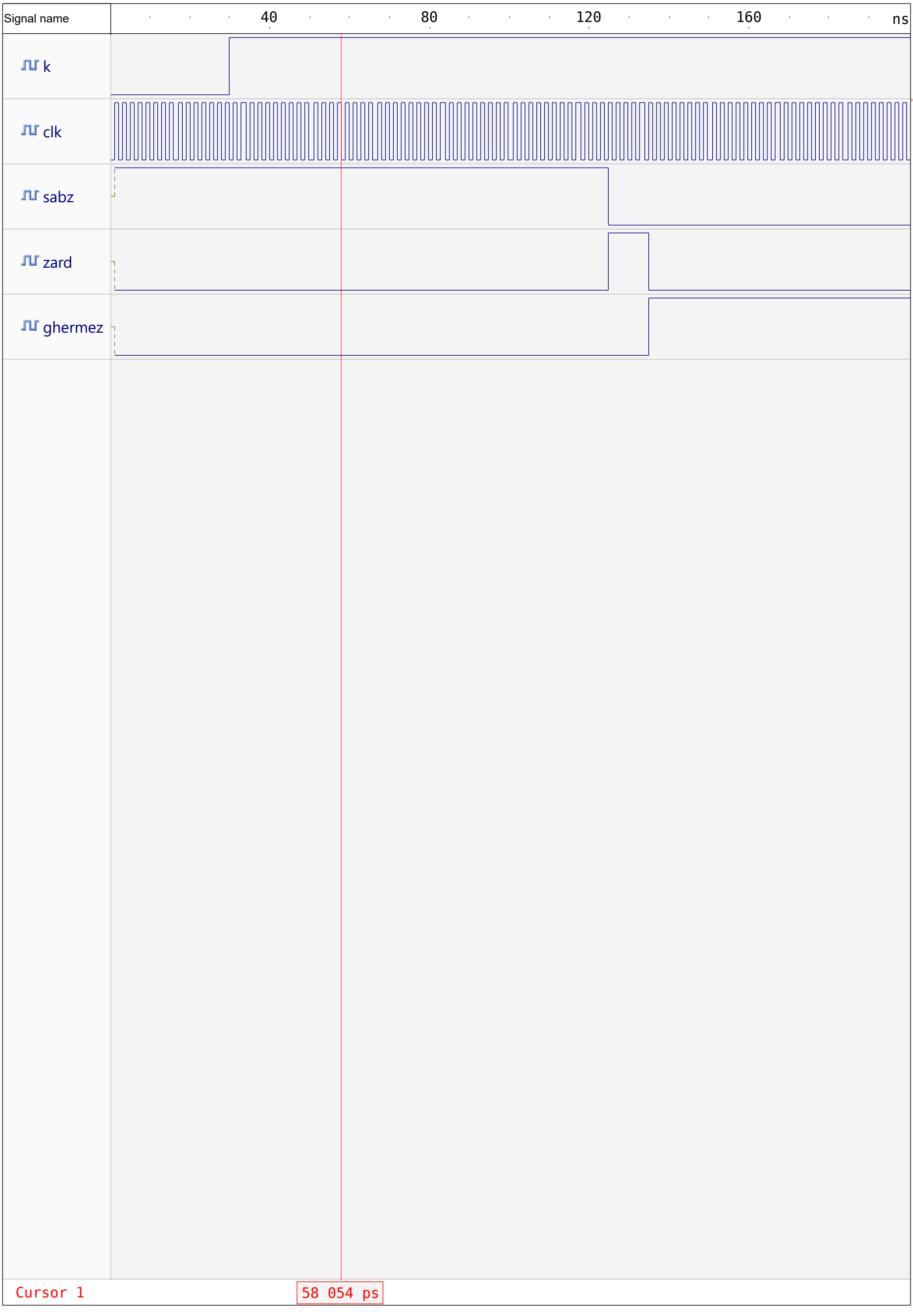
```
LIBRARY IEEE ;
USE IEEE.STD_logic_1164.ALL ;
USE IEEE.STd_logic_UNSIGNED.ALL;


entity koli is
    port(k :in std_logic ;
    clk :in std_logic;
    sabz :out std_logic;
    zard :out std_logic;
    ghermez :out std_logic);
end entity;

architecture behav of koli is
type stat is (s1,s2,s3,s4);
signal next_stat,hal_stat :stat ;

begin
    process (clk)
    begin
            hal_stat<=next_stat;
    end process;

    process (clk)
    variable C :integer:=0;
    begin
        if clk'event and clk='1' then
        case hal_stat is
when s1 =>

        sabz <= '1';
        zard <= '0';
        ghermez <='0';

        if c<60 and k='0' then

            c:=c+1 ;
            next_stat<=s1;

        elsif c<60 and k='1' then

            next_stat <= s2;

        elsif c>=60 and k='1' then

            next_stat<=s3;
            c:= 0;

        end if;

when s2 =>

         sabz <= '1';
        zard <= '0';
        ghermez <='0';

        if c<60 then

            c:=c+1;
            next_stat<=s2;

        elsif c>= 60 then

            next_stat<=s3;
            c:=0;

        end if;

when s3 =>

        sabz <='0';
        zard <='1';
        ghermez <='0';

        if c<4 then

            next_stat<=s3;
            c:=c+1;

        else

            next_stat<=s4;
            c:=0;

        end if;

when s4=>

        sabz <='0';
        zard <='0';
        ghermez <='1';

        if c<60 then

            next_stat<=s4;
            c:=c+1;
```

```
                    else

                        next_stat<=s1;
                        c:=0;

                    end if;

       end case;
        end if;
end process;
end architecture;
```

2.2 test.vhd

```
LIBRARY IEEE ;
USE IEEE.STD_logic_1164.ALL ;
USE IEEE.STd_logic_UNSIGNED.ALL;

entity test is
end entity;

architecture behav of test is

signal k : std_logic:='0' ;
signal clk : std_logic:='0';
signal sabz : std_logic;
signal zard : std_logic;
signal ghermez : std_logic;

begin
    u1:entity work.koli port map(k,clk,sabz,zard,ghermez);
    CLK<= NOT CLK AFTER 1 NS WHEN NOW<= 200 NS ELSE '0';

    k<='1' after 30 ns;
end;
```

2.3 1.awc


Signal name
40
80
120
160
ns
k
clk
sabz
zard
ghermez
Cursor 1
58 054 ps

3 cheragh_hoshmand_part2

3.1 part2.vhd

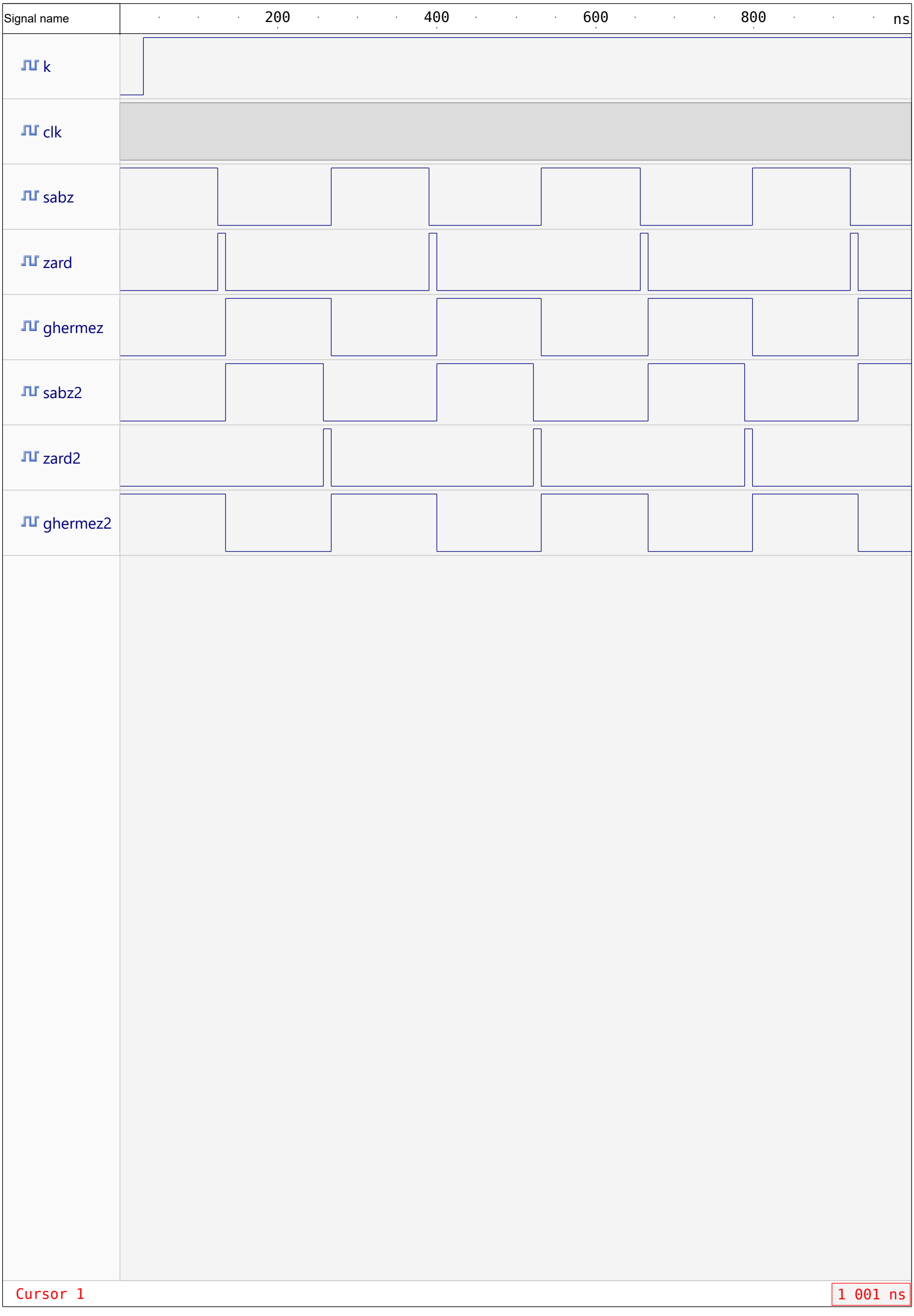
```
LIBRARY IEEE ;
USE IEEE.STD_logic_1164.ALL ;
USE IEEE.STd_logic_UNSIGNED.ALL;


entity koli is
    port(k :in std_logic ;
    clk :in std_logic;
    sabz :out std_logic;
    zard :out std_logic;
    ghermez :out std_logic;
    sabz2 :out std_logic;
    zard2 :out std_logic;
    ghermez2 :out std_logic);
end entity;

architecture behav of koli is
type stat is (s1,s2,s3,s4,s5);
signal next_stat,hal_stat :stat ;

begin
    process (clk)
    begin
            hal_stat<=next_stat;
    end process;

    process (clk)
    variable C :integer:=0;
    begin
        if clk'event and clk='1' then
case hal_stat is

    when s1 =>

            sabz <= '1';
            zard <= '0';
            ghermez <='0';
            sabz2 <= '0';
            zard2 <= '0';
            ghermez2 <='1';
            if c<60 and k='0' then

                c:=c+1 ;
                next_stat<=s1;

            elsif c<60 and k='1' then

                next_stat <= s2;

            elsif c>=60 and k='1' then

                next_stat<=s3;
                c:= 0;

            end if;

    when s2 =>

             sabz <= '1';
            zard <= '0';
            ghermez <='0';
            sabz2 <= '0';
            zard2 <= '0';
            ghermez2 <='1';
            if c<60 then

                c:=c+1;
                next_stat<=s2;

            elsif c>= 60 then

                next_stat<=s3;
                c:=0;

            end if;


when s3 =>

            sabz <='0';
            zard <='1';
            ghermez <='0';
            sabz2 <= '0';
            zard2 <= '0';
            ghermez2 <='1';
            if c<4 then

                next_stat<=s3;
                c:=c+1;

            else

                next_stat<=s4;
                c:=0;

            end if;

when s4=>
```

```
        sabz <='0';
        zard <='0';
        ghermez <='1';
        sabz2 <= '1';
        zard2 <= '0';
        ghermez2 <='0';
        if c<60 then

            next_stat<=s4;
            c:=c+1;

        else

            next_stat<=s5;
            c:=0;

        end if;
    when s5=>
        sabz <='0';
        zard <='0';
        ghermez <='1';
        sabz2 <= '0';
        zard2 <= '1';
        ghermez2 <='0';

        if c<4 then

            next_stat<=s5;
            c:=c+1;
        else

            next_stat<=s1;
            c:=0;
                    end if ;


       end case;
        end if;
end process;
end architecture;
```

3.2 test.vhd

```
LIBRARY IEEE ;
USE IEEE.STD_logic_1164.ALL ;
USE IEEE.STd_logic_UNSIGNED.ALL;

entity test is
end entity;

architecture behav of test is
signal k : std_logic:='0' ;
signal clk : std_logic:='0';
signal sabz : std_logic;
signal zard : std_logic;
signal ghermez : std_logic;
signal sabz2 : std_logic;
signal zard2 : std_logic;
signal ghermez2 : std_logic;
begin
    u1:entity work.koli port map(k,clk,sabz,zard,ghermez,sabz2,zard2,ghermez2);
    CLK<= NOT CLK AFTER 1 NS WHEN NOW<= 1000 NS ELSE '0';

    k<='1' after 30 ns;
end;
```

3.3 2.awc


Signal name
200
400
600
800
ns
k
clk
sabz
zard
ghermez
sabz2
zard2
ghermez2
Cursor 1
1 001 ns