


```
xi => xi, yi => yi, zi => zi,  
xo => xo, yo => yo, zo => zo );
```

```
clk <= not clk after half_period;
```

```
rst <= '0', '1' after 2* half_period;
```

```
process  
begin
```

```
wait until rst = '1';
```

```
-----  
-- printf ("\nGrinding on [K, 0, 0]\n");  
-- Circular (X0C, 0L, 0L);  
-----
```

```
for i in 0 to 4 loop  
wait until clk = '1';  
end loop; -- i
```

```
xi <= Conv2fixedPt(K, nBit);  
yi <= Conv2fixedPt(0.0, nBit);  
zi <= Conv2fixedPt(pi/6.0, nBit);  
DispReg(xi, yi, zi, 0);  
load <= '1', '0' after clk_period;
```

```
wait until (ready'event and ready='1');  
wait until clk = '1';  
DispReg(xo, yo, zo, 1);
```

```
-----  
-- printf ("\nGrinding on [K, 0, pi/6] -> [0.86602540, 0.50000000, 0]\n");  
-- Circular (X0C, 0L, HalfPi / 3L);  
-----
```

```
for i in 0 to 4 loop  
wait until clk = '1';  
end loop; -- i
```

```
xi <= Conv2fixedPt(K, nBit);  
yi <= Conv2fixedPt(0.0, nBit);  
zi <= Conv2fixedPt(pi/6.0, nBit);  
DispReg(xi, yi, zi, 0);  
load <= '1', '0' after clk_period;
```

```
wait until (ready'event and ready='1');  
wait until clk = '1';  
DispReg(xo, yo, zo, 1);
```

```
-----  
-- printf ("\nGrinding on [K, 0, pi/4] -> [0.70710678, 0.70710678, 0]\n");  
-- Circular (X0C, 0L, HalfPi / 2L);  
-----
```

```
for i in 0 to 4 loop  
wait until clk = '1';  
end loop; -- i
```

```
xi <= Conv2fixedPt(K, nBit);  
yi <= Conv2fixedPt(0.0, nBit);  
zi <= Conv2fixedPt(pi/4.0, nBit);  
DispReg(xi, yi, zi, 0);  
load <= '1', '0' after clk_period;
```

```
wait until (ready'event and ready='1');  
wait until clk = '1';  
DispReg(xo, yo, zo, 1);
```

```
-----  
-- printf ("\nGrinding on [K, 0, pi/3] -> [0.50000000, 0.86602540, 0]\n");  
-----
```

```
-- Circular (X0C, 0L, 2L * (HalfPi / 3L));
```

```
-----  
    for i in 0 to 4 loop  
        wait until clk = '1';  
    end loop; -- i  
  
    xi <= Conv2fixedPt(K, nBit);  
    yi <= Conv2fixedPt(0.0, nBit);  
    zi <= Conv2fixedPt(pi/3.0, nBit);  
    DispReg(xi, yi, zi, 0);  
    load <= '1', '0' after clk_period;  
  
    wait until (ready'event and ready='1');  
    wait until clk = '1';  
    DispReg(xo, yo, zo, 1);  
  
    for i in 0 to 4 loop  
        wait until clk = '1';  
    end loop; -- i  
end process;  
  
process  
begin  
    wait for 2000* clk_period;  
    assert false report "end of simulation" severity failure;  
end process;  
  
XXXXXXX XXXXXX XXXXXX XXXXXX XXXXXXXX XXXXXX XXXXX  
  
end beh;
```