*--Animation for a sports club*  
   
**LIBRARY** ieee;  
**USE** ieee.std\_logic\_1164.**all**;  
**use** IEEE.STD\_LOGIC\_ARITH.**ALL**;  
**use** IEEE.**STD\_LOGIC\_UNSIGNED**.**ALL**;  
   
**ENTITY** animation **IS**  
 **PORT** (  
  
 clock, reset : **IN** **std\_logic**;   
 sel : **IN** **std\_logic\_vector**(2 **downto** 0);  
 segment: **out** **std\_logic\_vector**(6 **downto** 0);  
 anods: **out** **std\_logic\_vector**(7 **downto** 0)  
   
 );  
**END** animation;  
  
  
  
*--------------------------------------------------------------------------------*  
*--Complete your VHDL description below*  
*--------------------------------------------------------------------------------*  
  
**ARCHITECTURE** TypeArchitecture **OF** animation **IS**  
  
 **type** mem **is** **array** (26 **downto** 0) **of** **std\_logic\_vector**(6 **downto** 0);  
 **type** word **is** **array** (20 **downto** 0) **of** **std\_logic\_vector**(6 **downto** 0);  
**constant** alphabet : mem :=(  
 0 =>"1011111",*--A*  
 1 =>"1110011",*--b*  
 2 =>"0110110",*--C*  
 3 =>"1111001",*--d*  
 4 =>"0110111",*--E*  
 5 =>"0010111",*--F*  
 6 =>"1101111",*--g*  
 7 =>"1010011",*--h*  
 8 =>"1001000",*--I*  
 9 =>"1111000",*--J*  
 10 =>"0110011",*--k*  
 11 =>"0110010",*--l*  
 12 =>"1011010",*--M*  
 13 =>"1010001",*--n*  
 14 =>"1111110",*--O*  
 15 =>"0011111",*--P*  
 16 =>"1111111",*--Q*  
 17 =>"0010001",*--r*  
 18 =>"1100111",*--S*  
 19 =>"0010110",*--T*  
 20 =>"1111010",*--U*  
 21 =>"1110000",*--V*  
 22 =>"1111011",*--W*  
 23 =>"1011011",*--X*  
 24 =>"0011011",*--Y*  
 25 =>"0111101",*--Z*  
 26 =>"0000000"*--space*  
 );   
 **constant** given\_word: word:=(  
 0=>alphabet(5),*--F*  
 1=>alphabet(14),*--O*  
 2=>alphabet(14),*--O*  
 3=>alphabet(19),*--T*  
 4=>alphabet(1),*--B*  
 5=>alphabet(0),*--A*  
 6=>alphabet(11),*--L*  
 7=>alphabet(11),*--L*  
 8=>alphabet(26),*--*  
 9=>alphabet(0),*--A*  
 10=>alphabet(13),*--N*  
 11=>alphabet(3),*--D*  
 12=>alphabet(26),*--*  
 13=>alphabet(15),*--P*  
 14=>alphabet(8),*--I*  
 15=>alphabet(13),*--N*  
 16=>alphabet(6),*--G*  
 17=>alphabet(15),*--P*  
 18=>alphabet(14),*--O*  
 19=>alphabet(13),*--N*  
 20=>alphabet(6)*--G*  
 );  
**signal** counter: **std\_logic**;  
**signal** counter2: **std\_logic**;   
  
**component** counter15 **is**  
**port**(  
clk :**in** **std\_logic**;  
o: **out** **std\_logic**);   
**end** **component** counter15;  
  
**component** counter25 **is**  
**port**(  
clk :**in** **std\_logic**;  
o: **out** **std\_logic**);   
**end** **component** counter25;  
  
**signal** clipping: **integer**:=0;  
**signal** l2: **integer**:= 0;  
**signal** r2: **integer**:= 7;  
**signal** close2: **integer**:= 1;  
**signal** segment3: **std\_logic\_vector**(7 **downto** 0):="01000000";  
**signal** segment4: **std\_logic\_vector**(7 **downto** 0):="11111111";  
**signal** i5:**integer**:=-7;  
**signal** i6: **integer** :=20;  
**signal** j56: **integer** :=20;  
**signal** c78: **integer** :=0;*--current*  
**signal** s78: **integer** :=0;*--state*  
**signal** bitmask7 : **std\_logic\_vector**(6 **downto** 0):="0010010";  
**signal** bitmask8 : **std\_logic\_vector**(6 **downto** 0):="0000100";  
**BEGIN**  
  
C15:counter15 **port** **map**(clk=>clock,o=>counter2);  
C25:counter25 **port** **map**(clk=>clock,o=>counter);  
*------*  
*---ANIMATION PROCESS*  
  
fordisplay: **process**(counter2,reset)  
**variable** j: **integer**:=0;  
**variable** anod:**std\_logic\_vector**(7 **downto** 0):="10000000";  
**begin**  
**if**(reset='1') **then**  
anod:="10000000";  
anods<=anod;  
segment<=(**others**=>'0');  
**elsif**(counter2='1' **and** counter2'event) then  
*---display time*  
*---constructing j-the memory index*  
**case** anod **is**  
**when** "00000001"=>  
j:=1;  
**when** "00000010"=>  
j:=2;  
**when** "00000100"=>  
j:=3;  
**when** "00001000"=>  
j:=4;  
**when** "00010000"=>  
j:=5;  
**when** "00100000"=>  
j:=6;  
**when** "01000000"=>  
j:=7;  
**when** "10000000"=>  
j:=0;  
**when** **others**=>  
j:=1;  
**end** **case**;  
*--- done constructing j*  
**case** sel **is**  
**when** "000"=>  
**if**(clipping=0) **then**  
segment<=given\_word(j);  
**else**  
segment<=(**others**=>'0');  
**end** **if**;  
**when** "001"=>  
**if**(j<=l2 **or** j>=r2) **then**  
segment<=given\_word(j);  
**else**  
segment<=(**others**=>'0');  
**end** **if**;  
**when** "010"=>  
**if**(segment3(j)='1') **then**  
segment<=given\_word(j);  
**else**  
segment<=(**others**=>'0');  
**end** **if**;  
**when** "011"=>  
**if**(segment4(j)='1') **then**  
segment<=given\_word(j);  
**else**  
segment<=(**others**=>'0');  
**end** **if**;  
**when** "100"=>  
j:=j+i5;  
**if**(j>=0 **and** j<=j56) **then**  
segment<=given\_word(j);  
**else**  
segment<="0000000";  
**end** **if**;  
**when** "101"=>  
j:=j+i6;  
**if**(j<=j56 **and** j>=0) **then**  
segment<=given\_word(j);  
**else**  
segment<="0000000";  
**end** **if**;  
**when** "110"=>  
 **if**(j<c78) **then**  
 segment<=given\_word(j);  
 **elsif** (j=c78) **then**  
 segment<=given\_word(j) **and** bitmask7;  
 **else**   
 segment<="0000000";  
 **end** **if**;  
**when** "111"=>  
 **if**(j<c78) **then**  
 segment<=given\_word(j);  
 **elsif** (j=c78) **then**  
 segment<=given\_word(j) **and** bitmask8;  
 **else**   
 segment<="0000000";  
 **end** **if**;  
**when** **others**=>   
segment<=(**others**=>'0');  
**end** **case**;  
 **if**(anod(0)='1') **then**  
 anod(0):='0';   
 anod(1):='1';  
 **elsif** (anod(1)='1') **then**   
 anod(1):='0';  
 anod(2):='1';  
 **elsif** (anod(2)='1') **then**   
 anod(2):='0';  
 anod(3):='1';  
 **elsif** (anod(3)='1') **then**   
 anod(3):='0';  
 anod(4):='1';  
 **elsif** (anod(4)='1') **then**   
 anod(4):='0';   
 anod(5):='1';  
 **elsif** (anod(5)='1') **then**   
 anod(5):='0';  
 anod(6):='1';  
 **elsif** (anod(6)='1') **then**   
 anod(6):='0';  
 anod(7):='1';  
 **elsif** (anod(7)='1') **then**   
 anod(7):='0';  
 anod(0):='1';   
 **end** **if**;  
 anods<=anod;  
**end** **if**;  
  
  
**end** **process** fordisplay;  
  
framechange: **process**(counter,reset)  
  
**begin**  
**if**(reset='1') **then**  
clipping<=0;  
l2<=0;  
r2<=7;  
close2<=1;  
segment3<="01000000";  
segment4<="11111111";  
i5<=-7;  
i6<=j56;  
c78<=0;*--current*  
s78<=0;*--state*  
bitmask7<="0010010";  
bitmask8<="0000100";  
  
**elsif**(counter='1'**and** counter'event) then  
**case** sel **is**  
**when** "000"=>  
clipping<=clipping+1;  
**if**(clipping=2) **then**  
clipping<=0;  
**end** **if**;  
**when** "001"=>  
**if**(close2=1) **then**  
l2<=l2+1;  
r2<=r2-1;  
**if**(l2>r2) **then**  
l2<=l2-2;  
r2<=r2+2;  
close2<=0;  
**end** **if**;  
  
**elsif**(close2=0) **then**  
l2<=l2-1;  
r2<=r2+1;  
**if**(l2<0 **and** r2>7) **then**  
l2<=l2+1;  
r2<=r2-1;  
close2<=1;  
**end** **if**;  
  
**end** **if**;  
**when** "010"=>  
  
**case** segment3 **is**  
**when** "01000000"=>  
segment3<="01000010";  
**when** "01000010"=>  
segment3<="01001010";  
**when** "01001010"=>  
segment3<="01011010";  
**when** "01011010"=>  
segment3<="11011010";  
**when** "11011010"=>  
segment3<="11011110";  
**when** "11011110"=>  
segment3<="11111110";  
**when** "11111110"=>  
segment3<="11111111";  
**when** "11111111"=>  
segment3<="00000000";  
**when** "00000000"=>  
segment3<="01000000";  
**when** **others**=>  
segment3<="00000000";  
**end** **case**;  
  
**when** "011"=>  
**case** segment4 **is**  
**when** "11111111"=>  
segment4<="11101110";  
**when** "11101110"=>*--*  
segment4<="11011101";  
**when** "11011101"=>  
segment4<="10111011";  
**when** "10111011"=>  
segment4<="01110111";*--*  
**when** "01110111"=>  
segment4<="11111111";  
**when** **others**=>  
segment4<="11111111";  
**end** **case**;  
  
**when** "100"=>  
i5<=i5+1;  
**if**(i5>j56) **then**  
i5<=-7;  
**end** **if**;  
**when** "101"=>  
i6<=i6-1;  
**if**(i6<-7) **then**  
i6<=j56;  
**end** **if**;  
**when** "110"=>  
s78<=s78+1;  
**if**(s78>2) **then**  
s78<=0;  
c78<=c78+1;  
**if**(c78>7) **then**  
c78<=0;  
**end** **if**;  
**end** **if**;  
**case** s78 **is**  
**when** 0 =>  
bitmask7<="0010010";  
**when** 1=>  
bitmask7<="0110111";  
**when** 2=>  
bitmask7<="1111111";  
**when** **others**=>  
bitmask7<="0000000";  
**end** **case**;  
  
**when** "111"=>  
s78<=s78+1;  
**if**(s78>4) **then**  
s78<=0;  
c78<=c78+1;  
**if**(c78>7) **then**  
c78<=0;  
**end** **if**;  
**end** **if**;  
 **case** s78 **is**  
 **when** 0 =>  
 bitmask8<="0000100";  
 **when** 1=>  
 bitmask8<="0001110";  
 **when** 2=>  
 bitmask8<="0001111";  
 **when** 3=>  
 bitmask8<="1011111";  
 **when** 4=>  
 bitmask8<="1111111";  
 **when** **others**=>  
 bitmask8<="0000000";   
 **end** **case**;  
**when** **others**=>  
**end** **case**;  
**end** **if**;  
**end** **process** framechange;  
**end** Typearchitecture;  
*-------*  
*---COUNTER 25*  
*-------*  
**LIBRARY** ieee;  
**USE** ieee.std\_logic\_1164.**all**;  
**use** IEEE.STD\_LOGIC\_ARITH.**ALL**;  
**use** IEEE.**STD\_LOGIC\_UNSIGNED**.**ALL**;  
   
**entity** counter25 **is**  
**port**(  
clk :**in** **std\_logic**;  
o: **out** **std\_logic**);   
**end** counter25;  
**Architecture** count1 **of** counter25 **is**  
**signal** c: **std\_logic\_vector**(25 **downto** 0):=(**others**=>'0');  
**begin**  
  
counting : **process**(clk)  
**begin**  
**if**(clk='1' **and** clk'event) then  
c<=c+1;  
**if**(c="10111110101111000010111111") **then**  
o<='1';  
**end** **if**;  
**if**(c="10111110101111000011000000") **then**  
c<=(**others**=>'0');  
o<='0';  
**end** **if**;  
**end** **if**;  
  
**end** **process** counting;  
  
**end** count1;   
  
*---COUNTER 15*  
**LIBRARY** ieee;  
**USE** ieee.std\_logic\_1164.**all**;  
**use** IEEE.STD\_LOGIC\_ARITH.**ALL**;  
**use** IEEE.**STD\_LOGIC\_UNSIGNED**.**ALL**;  
   
**entity** counter15 **is**  
**port**(  
clk :**in** **std\_logic**;  
o: **out** **std\_logic**);   
**end** counter15;  
**Architecture** count2 **of** counter15 **is**  
**signal** c: **std\_logic\_vector**(15 **downto** 0):=(**others**=>'0');  
**begin**  
  
counting : **process**(clk)  
**begin**  
**if**(clk='1' **and** clk'event) then  
 c<=c+1;  
**if**(c="1111000000000000") **then**  
o<='1';  
**end** **if**;  
**if**(c="1111000000000001") **then**  
c<=(**others**=>'0');  
o<='0';  
**end** **if**;  
**end** **if**;  
  
**end** **process** counting;  
  
**end** count2;