##### 

Program:

|  |  |
| --- | --- |
| Course Number | 1444 |
| Section Number | 1 |
| Course Title | EMBEDDED SYSTEMS ARCHITECTURE 1 |
| Semester/Year | WINTER/2019 |

|  |  |
| --- | --- |
| Instructor | **Mohsen Salahi** |

|  |  |
| --- | --- |
| **Lab No.** | **3** |

|  |  |
| --- | --- |
| Submission Date | **31/04/2019** |
| Due Date | **31/04/2019** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student Name | **FRANCIS VIJITH** | **NIBIN SOBY** | **AKHIL GEORGE** | **VIJAYAPRAKASH** |
| Student ID | C0742980 | C0741660 | C0748466 | C0744450 |
| Signature\* | **FV** | **NS** | **AG** | **VI** |

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties.*

**VHDL CODE:**

Entity Adv\_Game is

PORT(clock: IN STD\_ LOGIC;

n,s,e,w,v: IN STD\_ LOGIC;

reset: IN STD \_LOGIC;

sw,win,die,s0,s1,s2,s3,s4,s5,s6 : OUT STD \_LOGIC);

architecture Behavioral of Adv\_Game is

TYPE state\_type IS (Caveof\_cacophonhy,TTunnel,RRiver,SSword\_Stash,DDen,GGraveyard,VVault,OutGAME);

SIGNAL State : State\_Type;

BEGIN

PROCESS(clock,reset,n,s,e,w)

Begin

If (reset=’1’) THEN

State<= caveof\_Cacophony;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

ELSEIF rising\_edge(clock) THEN

CASE state IS

WHEN Caveof\_Cacophony=>

IF (e=’1’) and (n=’0’) and (s=’0’) and (w=’0’) THEN

State<=TTunnel;

S0<=’0’;

S1<=’1’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

else

State <= Caveof\_Cacophony;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

end if

WHEN TTunnel =>

IF (S=’1’) and (n=’0’) and (e=’0’) and (w=’0’) THEN

S0<=’0’;

S1<=’0’;

S2<=’1’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’

elseif ( W=’1’) and (n=’0’) and (s=’0’) and (e=’0’) then

State <=Caveof\_Cacophony;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

else

State <= TTunnel;

S0<=’0’;

S1<=’1’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

END IF:

WHEN RRiver =>

IF (e=’1’) and (n=’0’) and (s=’0’) and (w=’0’) THEN

State<=DDen;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’1’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

elseif (n=’1’) and (s=’0’) and (e=’0’) and (w=’0’) then

state<=TTunnel;

S0<=’0’;

S1<=’1’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

Elseif (w=’1’) and (n=’0’) and(s=’0’) and (w=’0’) then

State <= SSword\_Stash;

S0<=’0’;

S1<=’0’;

S2<=’0’;

S3<=’1’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’1’;

else

State <= RRiver;

S0<=’0’;

S1<=’0’;

S2<=’1’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

END IF;

When DDen=>

IF (e=’1’) and (n=’0’) and (w=’0’) and (s=’0’) and (v=’0’) THEN

State <=GGraveyard;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’1’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’1’;

Sw<=’0’;

ELSEIF (e=’1’) and (n=’0’) and (s=’0’) and and (w=0’’) and (v=’1’) THEN

state<=VVault;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’1’;

Win<=’1’;

Die<=’0’;

Sw<=’0’;

END IF;

WHEN SSword Stash=>

IF (e=’1’) and (n=’0’) and (w=’0’) and (s=’0’) THEN

State <=RRiver;

S0<=’1’;

S1<=’0’;

S2<=’1’;

S3<=’0’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’0’;

else

State <= SSword¬\_Stash;

S0<=’1’;

S1<=’0’;

S2<=’0’;

S3<=’1’;

S4<=’0’;

S5<=’0’;

S6<=’0’;

Win<=’0’;

Die<=’0’;

Sw<=’1’;

END IF;

WHEN Others =>

END CASE;

END IF;

END PROCESS;

End Behavioral;

**Sequence for win:**



**Sequence for die:**

