ECE 477 SAMPLE HOMEWORK

STUDENT NAME

|  |  |
| --- | --- |
| **Module Name:** | 2-to-1 Multiplexer with enable |
| **Module Category:** | Digital Logic |
| **Module Description:** | Module selects one of two digital input signals and forwards the selected input into a single line. |
| **Module Features:** | Module has 3 inputs and 1 output. |
|  | |
| **Input Output Block Diagram**  **C:\Users\cagri_000\Downloads\Untitled Diagram.png** | **Truth Table** |
|  | |
| **FPGA Implementation Results:** | |
| **FPGA Implementation VHDL codes:**  **library** IEEE**;**  **use** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **entity** two\_to\_one\_mux **is**  **Port** **(** I0**,** I1 **:** **in** STD\_LOGIC**;**  E **:** **in** STD\_LOGIC**;**  S0 **:** **in** STD\_LOGIC**;**  F **:** **out** STD\_LOGIC  **);**  **end** two\_to\_one\_mux**;**  **architecture** Behavioral **of** two\_to\_one\_mux **is**  **begin**  F **<=** I0 **WHEN** **(**S0**=**'0' AND E**=**'1'**)** **ELSE**  I1 **WHEN** **(**S0**=**'1' AND E**=**'1'**)** **ELSE**  'Z'**;**    **end** Behavioral**;** | |
| **Simulation codes:**  **LIBRARY** ieee**;**  **USE** ieee**.**std\_logic\_1164**.ALL;**    **ENTITY** MUX\_TEST **IS**  **END** MUX\_TEST**;**    **ARCHITECTURE** behavior **OF** MUX\_TEST **IS**    **COMPONENT** two\_to\_one\_mux  **PORT(**  I0 **:** **IN** std\_logic**;**  I1 **:** **IN** std\_logic**;**  E **:** **IN** std\_logic**;**  S0 **:** **IN** std\_logic**;**  F **:** **OUT** std\_logic  **);**  **END** **COMPONENT;**    --Inputs  **signal** I0 **:** std\_logic **:=** '1'**;**  **signal** I1 **:** std\_logic **:=** '0'**;**  **signal** E **:** std\_logic **:=** '0'**;**  **signal** S0 **:** std\_logic **:=** '0'**;**  --Outputs  **signal** F **:** std\_logic**;**  **BEGIN**  -- Instantiate the Unit Under Test (UUT)  uut**:** two\_to\_one\_mux **PORT** **MAP** **(**  I0 **=>** I0**,**  I1 **=>** I1**,**  E **=>** E**,**  S0 **=>** S0**,**  F **=>** F  **);**  -- Stimulus process  stim\_proc**:** **process**  **begin**  -- hold reset state for 100 ns.  E **<=** '0'**;**S0 **<=** '0'**;**  **wait** **for** 100 ns**;**  E **<=** '0'**;**S0 **<=** '1'**;**  **wait** **for** 100 ns**;**  E **<=** '1'**;**S0 **<=** '0'**;**  **wait** **for** 100 ns**;**  E **<=** '1'**;**S0 **<=** '1'**;**  **wait** **for** 100 ns**;**  **end** **process;**  **END;** | |