Lab 3 - Timer Peripheral

Introduction

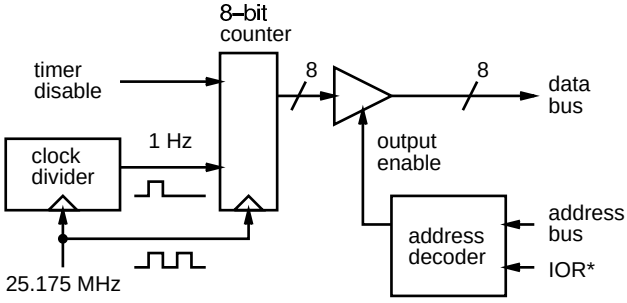
In this lab you will design a timer peripheral that can be read by the SBC and write an assembly-language utility to display the elapsed time in hex.

The timer circuit is an 8-bit register that is incremented once per second. The clock input for the timer is a 25.175 MHz oscillator on the FPGA board. The timer also has an external disable input which is controlled by a push-button on the FPGA board.

Your utility program continuously reads the contents of the timer register and displays the value on the SBC console as 2-digit hex character.

Hardware Description

The diagram below shows the internal structure of the timer:



Clock Divider

The clock divider circuit is a counter that counts from 0 to 25,174,999 (the clock frequency-1) and then starts at 0 again. An output is asserted during the clock cycle where the count is equal to 0. The frequency of the clock divider output is 1 Hz and the period is 1/25,175,000 s. Design this part of the circuit in VHDL as an entity which can be included as a component in the rest of the design.

Timer

The timer circuit is an 8-bit counter that is incremented when the clock divider output is asserted and the disable input is not asserted. It thus counts up at 1 Hz.

CPU Interface

The count held by the timer circuit can be read at location 220H. The processor interface places the contents of this register on the data bus in response to read cycles to I/O memory (port) address 220H as indicated by the address bus and the IOR* signal. The PC-104 bus IOR* signal is generated by combining the CPU's M/IO* and W/R* signals to create a signal (strobe) that is low only during a read cycle to the I/O space.

Your circuit must tri-state it's data bus pins except during an I/O read to memory location 220H to avoid contention between the different devices that share the data bus.

Note that the IOR* signal is not treated as a clock. Unlike the the CPU interface to write to a register, the CPU interface to read a register requires only combinational logic.

Your design should be *synchronous* (use only one clock).

The table below shows the new PC-104 signals. The rows are numbered starting at 1 for the top row.

	Left Board		Right Board	
Row	PC-104 Signal	FPGA Pin	PC-104 Signal	FPGA Pin
4			IOR*	158

The FPGA connections to the data bus, address bus, LED and pushbutton are given in previous labs. The 25.175 MHz clock is permanently connected to the FPGA chip on pin 91.

Pre-Lab Assignment

Before the lab you must write, assemble and test (to the extent possible) the utility program. You must also design the circuit and test it by simulating its

operation. The TA will ask to see your assembler and VHDL code and the simulation waveforms at the start of the lab.

Assembly Language Program

Write an 8088 assembly-language program that does the following:

- inputs a byte from memory location 220H
- prints the value as two hex characters
- prints a carriage return (ASCII decimal 13)
- loops back to the beginning

You will be able to re-use most of the code from assignment 2. Create an executable .COM file as described in the previous lab. When you run this program it should repeatedly print a 2-digit hex number at the left margin. Press control-break to interrupt the program.

VHDL Description

Write, compile and test by simulation a VHDL description for the timer circuit described above. Create simulation test waveforms that demonstrate the following[1]:

- a read from address 221H leaves the bus in a high-impedance state
- a read from address 220H puts a value of zero on the data bus only when IOR* is asserted

The instructions for compiling and simulating your VHDL description are given in the previous lab.

Print and Copy Files

Save the files *projectname*.`asm` (assembly language source code), *projectname*.`com` (DOS executable), *projectname*.`acf` (device and pin assignments), *projectname*.`vhd` (VHDL code), and *projectname*.`scf` (test waveforms) to a floppy disk to bring it with you to the lab. Print out the assembler and VHDL code and the simulator output waveforms.

[1] Unfortunately, it is not practical to simulate the operation of the device over approximately 25.175 million clock cycles.

Lab Procedure

Connect the PC-104 address and data bus signals and IOR* to the FPGA pins on the interconnect board as described in the previous lab. Double-check your connections and turn on the power.

Compile your VHDL code if you haven't already done so, and configure the FPGA as described in the previous lab.

Assemble and link your assembly code if you haven't already done so. All of your files should be stored in the `c:\max2work` directory.

Run the Windows Hyperterm program. Reset the SBC and download your program.

Run your program on the SBC. It should continuously display the timer contents which should be incrementing once per second. Hold down the pushbutton button to disable counter and verify that it is no longer incrementing.

When your device is working properly ask the TA to check your work. He will make sure your device works as required and ask you one or two questions to verify your understanding of the material.

Report

Submit a short report with a written description of your circuit. Include a block diagram showing the connections between the PC-104 bus, the FPGA the LED and the pushbutton, a listing of your assembly-language program, the VHDL code and a printout of the simulation waveforms that demonstrate correct operation of your device.