Prepared by:	Date:	Document and Filename:	Revision:
Kian Souresrafil	November 6, 2018	Project 2	1

CECS 460

Project 2: Transmit Engine

Professor: John Tramel

By

KIAN SOURESRAFIL

11 - 5 - 2018

Prepared by:	Date:	Document and Filename:	Revision:
Kian Souresrafil	November 6, 2018	Project 2	1

Table of Contents

Introduction

In this project we take our initial completed project of the basic tramelblaze program counter from earlier in the semester design course and add a functional UART. We only add one half of the UART, that half being the transmit engine. The transmit engine is responsible for serially transmitting data we feed it. Our loadable data for our project is 8 bits wide and our design will communicate with a terminal interface on our computers. The data transmitted can be sent out at different speeds due to our control of the 4-bit user inputted baud rate. We use the transmit engine to print out the statement "CSULB CECS 460 - <count> along with a carriage return. The interrupt used to come from an onboard button but now is fed from the TX_RDY signal directly interrupting the tramelblaze when data is ready or not ready to be transmitted. The interrupt is then cleared by activating the interrupt acknowledge and sending the active signal to the same flop (RS FLOP) to clear the signal going feeding into the previously mentioned trameblaze interrupt.

Prepared by: Kian Souresrafil	Date: November 6, 2018	Document and Filename: Project 2	Revision: 1

3. Requirements

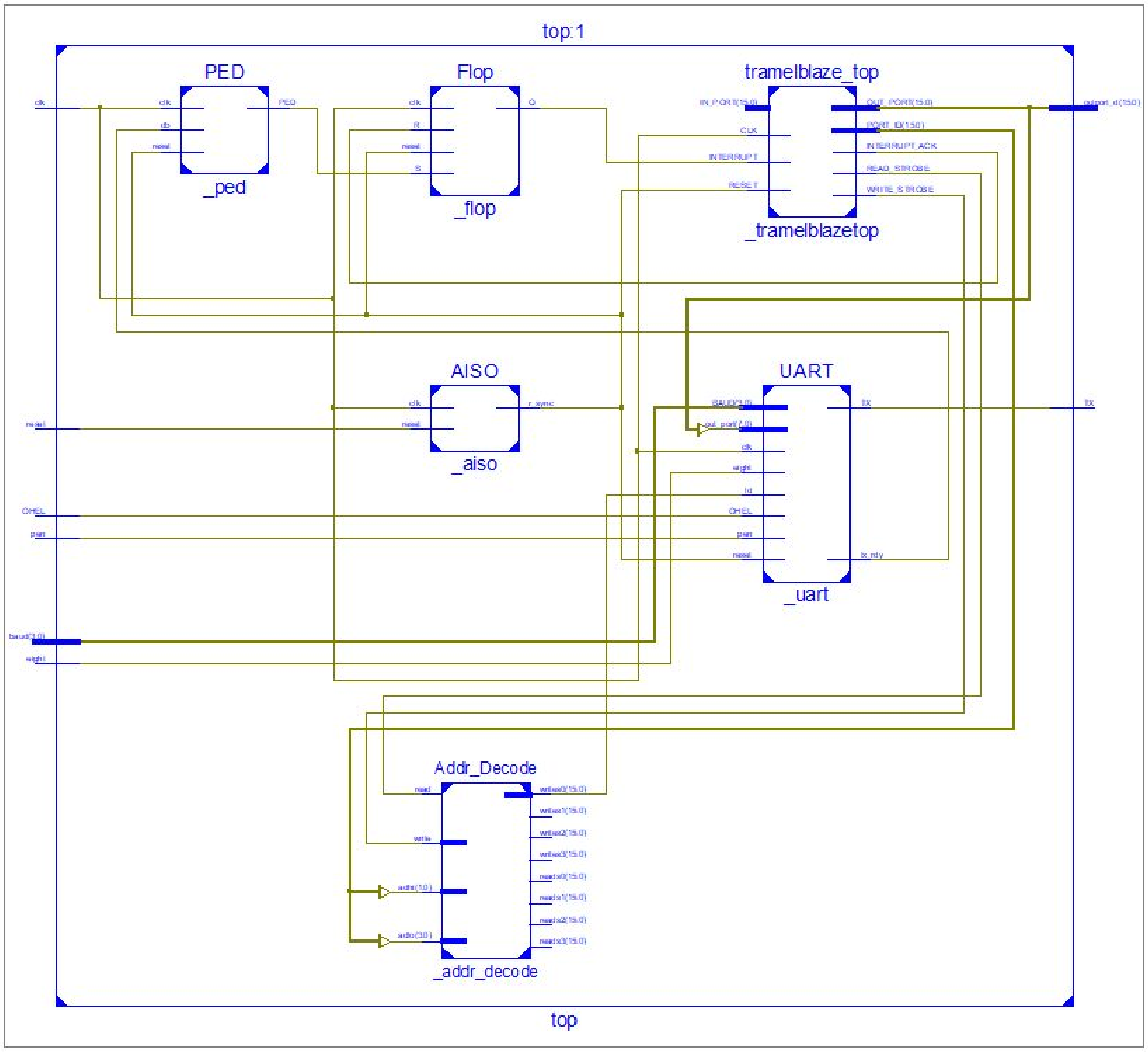
4. **Physical Requirements**
5. Nexys4 FPGA Board
6. Laptop/Desktop Computer
7. Micro-USB Cable
8. Interface/Software Requirements
    a. Digilent Adept
    b. RealTerm: Serial/TCP Terminal

Design Description

1. PED - Positive edge detector to only read and send in the positive edge of a signal into the program.
2. FLOP(RS) - Original signal from the button is sent in to the S input of the flop and will be then sent on the posedge of the clk into the tramelblaze as an interrupt. If the interrupt is acknowledged, then it will feed into the R and then into the interrupt to to clear it.
3. TRAMELBLAZE - Processor given by Professor Tramel with the use of writing and reading to and from specific memory blocks. Also has the ability to have interrupts and an input of port_id to give specific instructions. Acts as an emulator of a microcontroller specifically the picoblaze.
4. AISO - Creates a synchronous reset by taking in the reset value from the button press and delivering it synchronously with the clock to all blocks within the program.
5. UART - This is communications device used to talk with I/O devices in general. For our project specifically we used it to transmit a message onto a terminal. This module has two separate counters that keep track of the amount of bits transferred given a certain baud rate. It also implements a shift register module to serially transfer each bit of data and to set the transmit signal on or off.
6. Addr_Decode - Takes in the 16 bit wide PORT_ID wire coming out of the tramelblaze and decodes specific sections of it to figure out where to read or write the data being transmitted or received.

Top Level Block Diagram

Prepared by:	Date:	Document and Filename:	Revision:
Kian Souresrafil	November 6, 2018	Project 2	1


top:1
PED
_ped
Flop
_flop
tramelblaze_top
_tramelblazetop
AISO
_aiso
UART
_uart
Addr_Decode
_addr_decode
top
clk
db
reset
PED
R
S
Q
IN_PORT(15:0)
CLK
INTERRUPT
RESET
OUT_PORT(15:0)
PORT_ID(15:0)
INTERRUPT_ACK
READ_STROBE
WRITE_STROBE
outport_d(15:0)
r_sync
BAUD(3:0)
out_port(7:0)
eight
ld
OHEL
pen
tx
tx_rdy
baud(3:0)
read
write
addra(1:0)
addrb(3:0)
writex0(15:0)
writex1(15:0)
writex2(15:0)
writex3(15:0)
readx0(15:0)
readx1(15:0)
readx2(15:0)
readx3(15:0)