ECE3073 – Lab 5 on UART
Lab Sheet
Monash University Malaysia
To be implemented in LabsLand portal

Programming the Serial Interface

Week 9

Objectives

The laboratory exercise introduces asynchronous serial communications and will give you experience of writing C programs that can handle multiple sources of interrupts. You will also investigate the interaction between interrupt code and normal code. The Qsys is able to produce Verilog code (out of your schematics) for an RS232 asynchronous serial interface. Using Qsys you will design a Nios microprocessor system incorporating an RS232 interface. This system will then be programmed to send and receive serial data.

In this exercise you will:

• Develop a Nios processor system incorporating an RS232 serial interface

• Write C code to send and receive serial data using interrupts

Prelim (10 Marks)

(1) Write down the offset numbers of following UART registers(need to refer data sheet of NIOS UART)

Receiver register(Rx)	
Transmitter(Tx)	
Status	
Control	

(2) What value in control register need to be assigned if you want UART to trigger interrupt to NIOS processor whenever a character received in UART Rx register.

(3) Where do you think you need to write the above setting code , main or exception ?

(4) We need to clear interrupt bits of UART to ensure new interrupt can be trigger for new character at Rx register. Which register and what value we need to write for this setting

(5) Which part of the code you will need to write the setting in Question 4 (Main or exception) can specify clearly

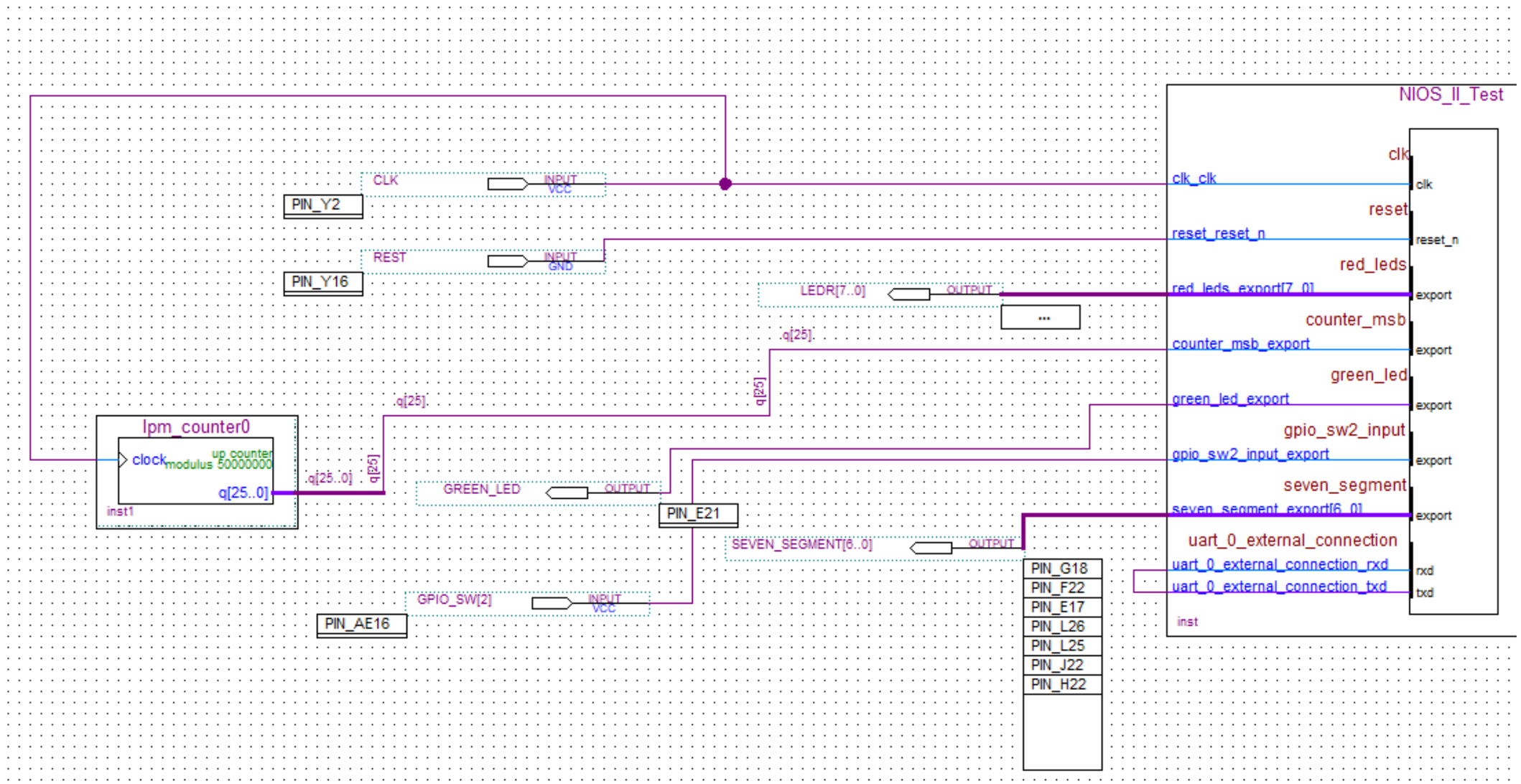
2. The computer system for this lab details

(a) Using Qsys to create a Nios system with the following components:

- a Nios II/e processor,
- 16384 bytes of RAM
- one 1-bit PIO input port (to read SW[2] of labsland – same as lab 4)
- one 1-bit PIO input port configured to produce an interrupt on an input rising edge ((MSB) of 1 Hz counter)
- 7 - bit PIO input port configured to first seven segment display (refer to pin numbers in screenshot)
- an RS232 UART configured for odd parity, 8 data bits, 1 stop bit and a fixed Baud rate of 9600
- a JTAG UART Note: you will find the UARTs in the left hand window of Qsys if you expand Interface protocols and then Serial.

Click on the bubbles in the IRQ column of Qsys to connect interrupts to counter PIO together with the RS232 and JTAG UARTs. You should make a note of the interrupt number assigned to each device as well as all of the I/O base addresses. (refer to the QYSYS screen shot)

(b) Following are the screen shot of the schematics



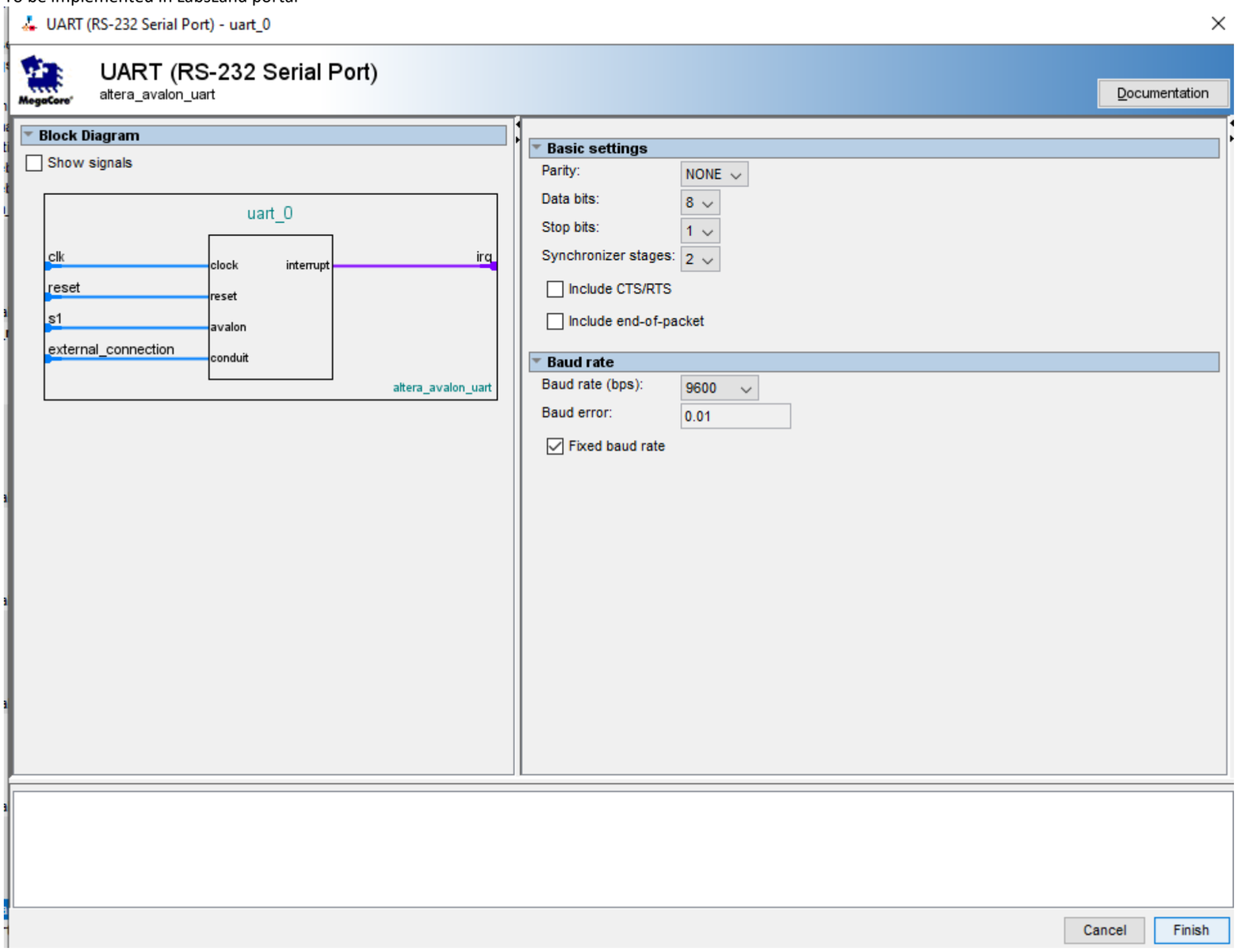
(Note the green Led just there but not used in LAB5)

ECE3073 – Lab 5 on UART
Lab Sheet
Monash University Malaysia
To be implemented in LabsLand portal

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Opc
☑		**clk_0**	Clock Source						
		clk_in	Clock Input	**clk**					
		clk_in_reset	Reset Input	**reset**					
		clk	Clock Output	*Double-click to export*	clk_0				
		clk_reset	Reset Output	*Double-click to export*					
☑		**nios2_qsys_0**	Nios II Processor						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset_n	Reset Input	*Double-click to export*	[clk]				
		data_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]			IRQ 31	
		instruction_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]				
		jtag_debug_module_re...	Reset Output	*Double-click to export*	[clk]				
		jtag_debug_module	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x4800**	0x4fff		
		custom_instruction_m...	Custom Instruction Master	*Double-click to export*					
☑		**pio_0**	PIO (Parallel I/O)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5030**	0x503f		
		external_connection	Conduit	**red_leds**					
☑		**onchip_memory2_0**	On-Chip Memory (RAM or ROM)						
		clk1	Clock Input	*Double-click to export*	**clk_0**				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk1]	0x0000	0x3fff		
		reset1	Reset Input	*Double-click to export*	[clk1]				
☑		**pio_1**	PIO (Parallel I/O)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5060**	0x506f	0	
		external_connection	Conduit	**counter_msb**					
☑		**pio_2**	PIO (Parallel I/O)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5050**	0x505f		
		external_connection	Conduit	**green_led**					
☑		**pio_3**	PIO (Parallel I/O)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5040**	0x504f		
		external_connection	Conduit	**gpio_sw2_input**					
☑		**pio_4**	PIO (Parallel I/O)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5020**	0x502f		
		external_connection	Conduit	**seven_segment**					
☑		**uart_0**	UART (RS-232 Serial Port)						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5000**	0x501f	1	
		external_connection	Conduit	**uart_0_external_connecti...**					
☑		**jtag_uart_0**	JTAG UART						
		clk	Clock Input	*Double-click to export*	**clk_0**				
		reset	Reset Input	*Double-click to export*	[clk]				
		avalon_jtag_slave	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	**0x5070**	0x5077	2	

(c) The QYSYS

ECE3073 – Lab 5 on UART
Lab Sheet
Monash University Malaysia
To be implemented in LabsLand portal



UART Settings as above.

NOTE: Even though we include JTAG UART here, we do not use . This can be used if you want to print statement in console window, unfortunately we could not do that yet in LabsLand. So just include JTAG UART and ignore while program, you will work only with the above UART-RS232 (screenshot)

Exercise (1) (5 marks)

Complete the Lab5 computer system schematics and qysys without errors in your computers , and Paste screenshots here (its same as above- you could develop from Lab4)

Exercise (2) (10 Marks)

Write simple code to display “1” and then “2” in seven segment display. Paste your screen shot here (basically 2 screen shots)

Displaying “1”	Displaying “2”

Exercise (3) (35 Marks)

Write suitable C – code to transmit your student ID (which are numbers but you will transmit them as characters, in the sample output video I have used alphabets for illustration purpose) serially one by one character to NIOS UART Transmitter register for every rising edge of 1 Hz MSB counter (trigger interrupts), as the transmitter and receiver are shorted , display the received characters (trigger interrupt) to 7 – segment display. Using LabsLand portal :Record your output using screen recording tool with your name and student ID typed in notepad are captured . You may refer the recorded sample video that I used to transmit a string :" ABCD". You need to post your screen capture video in the moodle submission link (20 marks) – Refer to **sample solution for Exercise 3** video in moodle.

Paste the screen shot of your main code from labsland (5 Marks)

Paste the screen shot of interrupt handler code from labsland (10 marks) – Note that I need to see the codes only for the above function explained in the exercise (3)

Exercise 4 (25 marks)

Now further write codes such that when all characters (your student numbers one by one as char) is transmitted and received, transmit character "E" continuously . That means I should see after your ID numbers, E will be always displayed (Refer to **Sample output of Exercise 4 video** in moodle). Ensure you keep the Notepad indicating your student ID and name aside in your recording Screen record submit to exercise 4 moodle link. (15 Marks)

Paste here the additional code from Exercise 3 to achieve the output (10 marks)

Exercise 5 (15 Marks)

Now further include in your code such that when SW[2] of LabsLand is kept ON the cycle repeats . (Refer the **sample Exercise 5 output video**). Ensure you keep the Notepad indicating your student ID and name aside in your recording (10 marks)

Paste the additional code you wrote to achieve the output (5 Marks)