Experiment 9

Integrating Peripherals (UART) with Processor

In this lab, we are going to integrate the UART Transmitter module as an external peripheral with the core. For this purpose the UART module will be connected with the LSU (Load Store Unit) along with the data memory and the LSU will perform the selection between the UART and data memory based upon the address of the load/store instructions. (Figure 9.1).

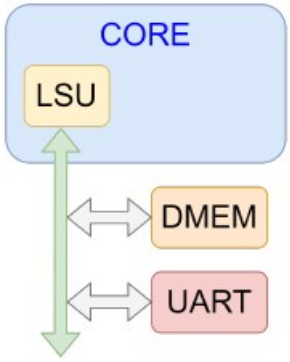

Figure 9.1: Integrating Uart Module with Processor

The LSBs of the address in the LSU are used to perform selection of bytes and halfwords. Here, we will use the MSB of the address to select peripherals. Suppose, your data memory(DMEM) is at starting address 0x0000_0000 and memory mapped UART registers start from 0x8000_0000. If the MSB of the address is 0 then the DMEM will be selected. If the MSB of the address is 1 then the UART will be selected. The LSU will set all the control signals of the UART module. The modules will only be selected in case of load/store operations. Listing 1.

```
assign dmem_sel = (LOAD OPCODE | STORE_OPCODE) && (dbus_addr[31:28] == 4`h0);
assign uart_sel = (LOAD OPCODE | STORE_OPCODE) && (dbus_addr[31:28] == 4`h8);
```

Listing 1: RTL for UART and Data Memory selection

UART peripheral will have 32-bit memory mapped registers. Storing data is equivalent to writing in these registers when their address is provided and reading is load instruction. So, for store instruction, LSU will send a *write request* along with uart-sel and data to the UART module (base adress: 0x8000_0000) via data-bus (dbus). Similarly, for load, LSU will send a *read request* along with uart-sel, and read data from UART-registers into processor registers.

- UART-Data register (**offset: 0x00**) will have the byte to be transmitted serially and full bit (31st bit) to indicate if Data register already has data which is in the process of transmission. Writing to UART-data, does not effect the full status flag, while reading it gives the current value of *full bit* and tx-byte is given as zero. So, whenever you will write to this register, full bit will be set to 1, and will be cleared after transmission has completed.
- UART-Baud register (**offset: 0x04**) will have the integer baud divisor for the baudrate. $Baudrate = SystemFrequency/(baud\_divisor + 1)$
- UART-Control register (**offset: 0x08**), for enabling serial transmission (Tx-start) when it is set to 1. This bit should be set after the baud and data registers have been written to.

Reset value of all the registers is zero.
Remember to add a Load state in your state machine, which will allow new data to be written into Data and Baud registers.
Once the transmission starts (Tx-start is set to 1), no new data can be written to these registers, i.e. writes are ignored untill the transmission completes and full bit has been cleared.

Lab Task

Integrate the UART module with the processor. Test a small assembly program for transferring data using the UART module.