LAB 3 – Arithmetic Circuits

Learning Objectives

- Use top-down design approach with sub-systems
- Implement basic arithmetic circuits on FPGA
- Use the radix complement method to perform subtraction

There is no separate tutorial part for this lab as most of the examples are covered in lectures with VHDL code examples given on lecture slides. The radix complement based subtraction was covered in tutorials.

The lab task is to design and implement a binary coded decimal (BCD) adder/subtractor and test on FPGA.

An Overview of BCD Addition

The binary coded decimal representation is based on representing each decimal digit (from 0 to 9) with 4 binary digits. Such representation of decimal data makes conversion from decimal to relevant BCD format easy, hence used in display applications. However, BCD makes arithmetic operations such as addition more complicated.

The BCD representation uses only ten out of the 15 possible combinations of a 4-bit binary number to represent valid BCD digits – i.e., 0000 is for 0, 0001 for 1, and up to 1001 for 9. The bit patterns 1010, 1011, 1100, 1101, 1110, 1111 are invalid in BCD. Consider the addition of two BCD digits $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$ to produce a sum $S = S_3S_2S_1S_0$ and a carry-out C_{out}. Note that, the ***carry-out should be set to 1 when the sum is greater than 9*** rather than 15 as it would be the case in a normal 4-bit binary adder. The following observations can be helpful when designing this single digit BCD adder:

- CASE1: If $S = A + B \leq 9$, then the 1-digit BCD addition is the same as 4-bit (unsigned) binary addition, where you will get the 4-bit sum and no carry-out.
- CASE2: If $9 < (S = A + B) \leq 15$, then the 4-bit (unsigned) binary addition will generate a sum and carry-out of 0. In this case, a correction must be applied to obtain the correct BCD result, where the sum is made within 0-9 and a carry-out is generated.
- CASE3: If $(S = A + B) > 15$, then the 4-bit (unsigned) binary addition will generate a sum and a carry-out. The sum will be incorrect in BCD format (because 4-bit binary addition is modulo-16 whereas BCD is modulo-10) but the carry-out is correct. In this case, a correction must be applied to obtain the correct BCD result, where the sum is made modulo-10.

The correction required in both cases 2 and 3 is adding 6 to the result, which is coming from the fact that 4-bit binary addition is modulo-16 whereas BCD is modulo-10. Therefore, the following pseudocode can be used to describe the 1-digit BCD addition operation:

```
If A + B > 9: -- cases 2&3 above
      Carry = 1
      Output = A + B + 6
Else -- case 1 above
      Carry = 0
      Output = A + B
```

Some additional explanations, a block diagram, and numerical examples will be presented during the lab class.

Design Task

Your task is to design, implement and test a 2-digit BCD adder/subtractor. The following steps are suggested to implement the 2-digit BCD adder/subtractor as incremental steps. These steps are suggested so that, in case you don't achieve the full functionality at the end, then you can still report evidence of partial functionality in the intermediate steps and receive partial marks for the lab report.

1. Draw a block diagram for 1-digit BCD adder using various combinational blocks you know and verify with teaching staff that your block diagram is correct. The system has two 1-digit BCD inputs, 1-bit carry-in, and 1-digit BCD output and 1-bit carry-out. You can assume that the user input is in valid BCD format.
2. Describe your design in step 1 in VHDL and ensure the correct operation of the 1-digit BCD adder in simulation. You should use behavioural/dataflow abstraction for basic combinational blocks within the design and use structural approach at the top-level. You may (optionally) test this intermediate design on board before moving onto step 3. Your simulation and testing should cover all three cases: $A + B \leq 9$, $9 < A + B \leq 15$, and $A + B > 15$ to ensure the correct operation of the BCD adder.
3. Extend your previous block diagram in step 1 to make it a 2-digit BCD adder having two 2-digit BCD inputs and three BCD outputs, where the most significant output digit is the carry-out from the 2nd output digit. You can use a ripple carry architecture for this.
4. Describe in VHDL, simulate, synthesise, implement, and test the 2-digit BCD adder in step 3 on the FPGA board. You can use 16 slide switches to provide the two 2-digit BCD inputs and the 3-digit BCD output should be displayed on 3 consecutive seven segment displays with appropriate display multiplexing. Alternatively, you can use 8 slide switches and registers to provide the two 2-digit BCD inputs, in a digit-serial manner. You can decide the approach for providing user input and describe it in the report. For the output on SSD, you can reuse your own design for SSD multiplexing from prac 2 with a citation added in the report to reflect this. You can assume that the inputs are in BCD format.
5. Extend your previous block diagram in step 3, now to facilitate BCD subtraction as well. i.e., the system should implement 2-digit BCD addition and subtraction based on another control signal. The control signal to select between addition and subtraction can be provided using either a slide switch (in case you do not use all the 16 switches to provide BCD inputs) or a push button on the board. The BCD subtraction should be performed as addition using the appropriate radix complement. For the subtraction operation via radix complement, you can simply keep the output in the radix complement format and display the carry-out as the third digit (i.e., you do not need to do additional manipulation of carry-out from the most significant digit, and you may ignore that when interpreting the result).
6. Describe in VHDL, simulate, synthesise, implement, and test the 2-digit BCD adder/subtractor in step 5 on the FPGA board with appropriately selected inputs and outputs on the board. Your report should clearly mention the inputs and outputs at your top-level diagram.

Submission and Assessment

You will be assessed based on a report submission followed by oral assessment. Submit a short report (electronically typeset) in PDF format via Blackboard containing the following. If you achieve the full functionality (i.e., step 6 above) then you don't need to report the intermediate steps and you can only provide the below items pertaining to your final design. However, if you don't achieve the full functionality, the steps 1-6 above are suggested so that you can report evidence of partial functionality.

- Short introduction to the task **[1 mark]**
- Block diagram(s) **[4 marks]**
  - Step 1 (1-digit BCD adder, 2 marks)
  - Step 3 (2-digit BCD adder, 3 marks)
  - Step 5 (2-digit BCD adder/subtractor, 4 marks)
- Functional Simulation **[5 marks]**
  - Step 2 (1-digit BCD adder covering all 3 test cases specified, 3 marks)
  - Step 4 (2-digit BCD adder covering at least 2 different test cases, including at least 1 case with intermediate carry propagation, 4 marks)
  - Step 6 (2-digit BCD adder/subtractor covering at least 2 different test cases in each mode. The subtraction must include a case where P > Q and P < Q where P, Q denote 2-digit BCD numbers, 5 marks)
- On-board testing (photo/mentioning in report and demo during oral assessment) **[5 marks]**
  - Step 1 (1-digit BCD adder, 1 mark)
  - Step 2 (2-digit BCD adder covering at least 2 different test cases, also with intermediate carry propagation, 3 marks)
  - Step 6 (2-digit BCD adder/subtractor covering at least 2 different test cases in each mode. The subtraction must include a case where P > Q and P < Q where P, Q denote 2-digit BCD numbers, 5 marks)
- VHDL descriptions **[2 marks]**
  - Step 1 (1-digit BCD adder, 0.5 mark)
  - Step 2 (2-digit BCD adder, 1 mark)
  - Step 3 (2-digit BCD adder/subtractor, 2 marks)
- RTL schematic **[1 mark]**
- FPGA resources **[1 mark]**
- Conclusions **[1 mark]**
- Oral assessment **[4 marks]**

The report should not be just a collection of screenshots, but with proper explanation of figures/results with some discussion. The report is **due on Monday 02/09/2024 4:00 pm AEST**.

Oral assessment: Conducted during scheduled lab sessions in week 7. You need to explain your design to teaching staff and ***demonstrate on-board testing of your design*** followed by some Q&A to check the understanding of your work. Marks for the oral assessment will be distributed as follows:

0: No knowledge of the design **(total mark capped at 0)**
1: Very little knowledge of the design **(total mark capped at 50%)**
2: Reasonable knowledge of the design
3: Good knowledge of the design
4: Excellent knowledge of the design

Total mark for this lab is out of 24, subject to oral assessment caps as above and any late penalties as outlined in the ECP. If you do not attend/complete the oral assessment, your marks for this lab will be capped at 0, as indicated above.