

## Design Project 2: Design and Implementation of an Arithmetic Logic Unit on the DE0 Nano board

### Validation Sheet – Page 1

The Validation Sheet for Project 2 is two pages long. Make sure that you include both pages as the last two pages of your project report. You should complete page 1 and leave page 2 blank for the TA to complete.

No *GTA* or *student* should discuss any aspect of a student's design with another student. Among other things, this includes the manner in which a student implements specific operations.

Student Name: Jacob Abel

Last four Digits of your student ID: 6453

#### GTA Validation Instructions

1. Program the FPGA on the DE0 Nano board using the Start button on the programmer window.
2. When the programming has successfully completed, reset the design by pressing and releasing the KEY0 pushbutton.
3. Set the DIP switches to 1110 and record the value of the LEDs as two hex digits in Table 1 on the next page. (The DIP switches are 0 on the side labeled "ON".)
4. Set the DIP switches to 1111 and record the value of the LEDs as two hex digits in Table 1.
5. Compare the four digits from steps 3 and 4 to the last four digits of the student's ID number. **If the four digits do not match the last four digits of the student's ID number on their ID card, STOP THE VALIDATION. DO NOT CONTINUE.**

For the remaining steps of the validation, the values of the switch settings in Gray code order will allow the requested items to be checked in the order in the table more quickly. **All values should be recorded in the table as two hexadecimal digits.** For each address, verify that the opcode, operand A and operand B digits match the value in the corresponding address in the rom.txt file on the student's computer.

6. Fill in the row of Table 2 for the "sub" operation. Verify that the address is 0.
7. Press and release KEY1.
8. Fill in the row of Table 2 for the "add" operation. Verify that the address is 1.
9. Press and release KEY1.
10. Fill in the row of Table 2 for the "and" operation. Verify that the address is 2.
11. Press and release KEY1.
12. Fill in the row of Table 2 for the "xnor" operation. Verify that the address is 3.
13. Press and release KEY1.
14. Fill in the row of Table 2 for the "not" operation. Verify that the address is 4.
15. Press and release KEY1.
16. Fill in the row of Table 2 for the "mult8" operation. Verify that the address is 5.
17. Press and release KEY1.
18. Fill in the row of Table 2 for the "inc" operation. Verify that the address is 6.
19. Press and release KEY1.
20. Fill in the row of Table 2 for the "neg" operation. Verify that the address is 7.
21. Press and release KEY1.
22. Fill in the row of Table 2 for the "dec" operation. Verify that the address is 8.
23. Press and release KEY1.
24. Fill in the row of Table 2 for the "mod4" operation. Verify that the address is 9.
25. Press and release KEY1.
26. Fill in the row of Table 2 for the "asr" operation. Verify that the address is A.
27. Press and release KEY1.
28. Fill in the row of Table 2 for the "lsl" operation. Verify that the address is B.

**Design Project 2: Design and Implementation of an Arithmetic Logic Unit on the DE0 Nano board**  
**Validation Sheet – Page 2**

Table 1: Checking BCD equivalent to student ID

	Switch setting, SW[3:0]	
	1110	1111
	— — — —	— — — —
LED value in hexadecimal	— — — —	— — — —

Table 2. Checking the operation of the ALU. All values of the LEDs should be recorded as two digit hexadecimal numbers.

SW[3:0] →	1000	1001	1010	1011	1100	1101
operation	address	opcode	operandA	operandB	result	status
sub	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
add	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
and	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
xnor	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
not	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
mult8	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
inc	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
neg	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
dec	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
mod4	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
asr	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —
lsl	— — — —	— — — —	— — — —	— — — —	— — — —	— — — —

Comments (only required if something is unusual or wrong):