**Chapter 4 – Arithmetic Functions**

**Practice Problems**

**Question 1:** Design a circuit that takes a 4-bit number and produces its 2’s Complement.

(**Help:** Modify Adder-Subtractor according to this requirement)

**Question 2:** Using the 4-bit binary ripple carry adder, implement increment-by-3 circuit which takes a 4-bit number A and outputs S such that S=A+3.

**Question 3:** Implement a circuit which takes two 3-bit **Unsigned Numbers** A and B and gives following outputs:

1. E (for A==B)
2. L (for A<B)
3. G (for A>B)

(**Help:** Use Adder-Subtractor and manipulate its result to give final outputs).

**Question 4:** Modify the circuit of Question 3 to give following outputs:

1. E (A==B)
2. L (A<B)
3. G (A>B)
4. LE (A<=B)
5. GE(A>=B)

**Question 5:** Following the design of Multiplier, which we studied in class, design a Square generator for 4-bit input number. Low level (detailed) logic diagram is required.

**Question 6:** Implement (make circuit of) function R = A x 2, where A is a 4-bit number. Low level logic diagram is required.

(**Help:** This circuit does not need any logic gates; it only needs wires and your observation. Make truth table and observe what this function is doing.

**Question 7:** Implement function R = A x 4, where A is a 4-bit number. Low level logic diagram is required.

**Question 8:** Guess the behavior of **R = A x 2i**.

**Question 9:** Following is the Truth Table for **D = A ÷ 4**, where A in 4-bit input (**A3A2A1A0**) and D is 6-bit output (**D3D2D1D0** . **D-1D-2**). Use link given below to fill the truth table

<http://www.exploringbinary.com/binary-converter/>

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Inputs** | | | | **Outputs** | | | | | |
| **A3** | **A2** | **A1** | **A0** | **D3** | **D2** | **D1** | **D0** | **D-1** | **D-2** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |  |  |  |  |  |  |
| 0 | 1 | 0 | 1 |  |  |  |  |  |  |
| 0 | 1 | 1 | 0 |  |  |  |  |  |  |
| 0 | 1 | 1 | 1 |  |  |  |  |  |  |
| 1 | 0 | 0 | 0 |  |  |  |  |  |  |
| 1 | 0 | 0 | 1 |  |  |  |  |  |  |
| 1 | 0 | 1 | 0 |  |  |  |  |  |  |
| 1 | 0 | 1 | 1 |  |  |  |  |  |  |
| 1 | 1 | 0 | 0 |  |  |  |  |  |  |
| 1 | 1 | 0 | 1 |  |  |  |  |  |  |
| 1 | 1 | 1 | 0 |  |  |  |  |  |  |
| 1 | 1 | 1 | 1 |  |  |  |  |  |  |

Observe the truth table and implement it.

**Question 10:** Implement Sign Extension circuit which takes a 4-bit **Signed number** and produces 8-bit number after sign bit extension.

For Example, if input is (+7) “0111”, output of the circuit will be “0000 0111” and if the input is (-7) “1001” the output will be “1111 1001”.

**Text Book Exercises:** (Moris Mano 4th Edition, Chapter 4) 4-16

**Question 11:** An Arithmetic Logic Unit (ALU) receives an instruction I (e.g. R = A+B), performs the operation specified in the instruction, and outputs its result. The instruction is received as a string of length 10 containing 0s and 1s (signals) i.e. a 10-bit string of 0’s and 1’s. It contains a 2-bit opcode (operation code) and two 4-bit operands (i.e. total 2+4+4=10 bits) in following format:

|  |  |  |
| --- | --- | --- |
| I9…I8 | I7…I4 | I3…I0 |
| Operation Code | 4-bit operand A | 4-bit Operand B |

The ALU performs following operations

|  |  |
| --- | --- |
| **Operation** | **Operation Code** |
| + | 00 |
| - | 01 |
| x | 10 |
| ÷ | 11 |

Table given below shows some examples of instructions received by the ALU and their interpretation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instruction (I)** | **Opcode** | **Operand A** | **Operand B** | **Circuit Required** |
| 01 1010 0101 | 01 | 1010 | 0101 | Adder-Subtractor |
| 10 1001 0110 | 10 | 1001 | 0110 | Multiplier |
| 11 0000 1100 | 11 | 0000 | 1100 | Divisor |

After receiving the instruction I, ALU passes the numbers to their required arithmetic block according to the OpCode received. The block produces results and ALU outputs an 8-bit result. As shown in the figure below:

ALU

R7…R0

I10…I0

Suppose all the functions listed in above table have already been implemented i.e. you have fully functional circuits (function blocks) available. Your task is to combine these function blocks and apply the concepts which we have studied so far and implement this ALU. Clearly show the flow of data from the input (instruction) wires to the output (result) wires.

**Note:** You only need to make high level diagram (using different function blocks). If any function blocks need some modifications for these requirements just mention those changes. Do not go into the details.