

United International University

Department of Computer Science and Engineering

**CSE 430: Digital System Design**

**Laboratory Manual**

Table of Contents

|  |
| --- |
| [**Experiment 1: Design and implementation of a 4-bit**](#page3) **Shifter** |
| [PROBLEM DESCRIPTIO](#page3)N |
| [TASK BREAKDOW](#page3)N  DESIGN AND IMPLEMENT ISSUES |
| [REPORT WRITIN](#page3)G |
| [MARK DISTRIBUTIO](#page4)N |

|  |
| --- |
| [**Experiment 2: Design and implementation of a 4-bit A**](#page5)**LU** |
| [PROBLEM DESCRIPTIO](#page5)N |
| [TASK BREAKDOW](#page5)N  DESIGN AND IMPLEMENT ISSUES  [REPORT WRITIN](#page3)G |
| [MARK DISTRIBUTIO](#page6)N |

|  |
| --- |
| [**Experiment 3: Desig**](#page7)**n and Implementation of 4 bit Microprocessor** |
| [PROBLEM DESCRIPTIO](#page5)N |
| [TASK BREAKDOW](#page5)N  DESIGN AND IMPLEMENT ISSUES  [REPORT WRITIN](#page3)G |
| [MARK DISTRIBUTIO](#page6)N |
|  |

|  |  |
| --- | --- |
| [**Experiment 4: Sequential Shift-Add Multiplier Desig**](#page7)**n** |  |
| [PROBLEM DESCRIPTIO](#page7)N |  |
| [FLOWCHAR](#page7)T  [REPORT WRITIN](#page3)G |  |
| [DESIGN AND IMPLEMENTATION ISSUE](#page8)S |  |
| [MARK DISTRIBUTIO](#page8)N |  |

**Experiment 1: Design and implementation of a 4-bit Shifter**

**PROBLEM DESCRIPTION**

In this assignment, we are going to design a 4-bit shifter. The shift unit attached to a processor transfers the output of the ALU onto the output bus. The shifter may transfer the information directly without a shift, or it may shift the information to the left or right.

For this assignment, we are going to design a shifter which can perform 8 operations on 4-bit operands. The functions performed by the shifter will be selected by means of 3 selection lines. These 3 selection lines (S2, S1, S0) will determine the shifter operation to be performed through 4-bit output (F3, F2, F1, F0).

List of Operations:

|  |  |  |
| --- | --- | --- |
| S0  S1 S2 | Operation | Function |
| 0 0 0 | F A | Transfer A to F |
| 0 0 1 | F A’ | Complement of A |
| 0 1 0 | F shr A | Shift right A into F |
| 0 1 1 | F shl A | Shift left A into F |
| 1 0 0 | F cr A | Circular right A into F |
| 1 0 1 | F cl A | Circular left A into F |
| 1 1 0 | F All 0’s | Transfer 0’s into F |
| 1 1 1 | F All 1’s | Transfer 1’s into F |

**TASK BREAKDOWN**

**Part 1 (Should be delivered within 7 days of instruction)**

Design the circuit for 4-bit shifter and show the simulation to your instructor.

**Part 2 (Should be delivered within 14 days of instruction)**

1. Prepare a report.
2. Implement the 2-bit Shifter in hardware performing the mentioned operations.

**DESIGN AND IMPLEMENT ISSUES**

● All outputs and flags must be directed to LED.

● All inputs should be given through logic switch.

● Any SSI (AND, OR, NOT, XOR etc.) and MSI (MUX) chip can be used.

**REPORT WRITING INSTRUCTIONS**

* **Problem Description**
* **Design**
* Design of 4 bit shifter with truth table
* **Test Cases**
* Show test cases for each operation you have tested your design with.
* **Circuit implementation**
* Attach the circuit diagram that you have designed using circuit maker software. It has to be neat and clearly labeled (inputs and outputs).
* **Discussion**
* Discuss the practical problems you faced while performing the experiment.

**Marks Distribution (50)**

|  |  |
| --- | --- |
| Report | 10 |
| Viva | 10 |
| Software Simulation | 20 |
| Hardware Simulation | 10 |
| Total | 50 |

**Experiment 2: Design and implementation of a 4-bit ALU**

**PROBLEM DESCRIPTION**

In this assignment, we are going to design an arithmetic and logic unit (ALU). An ALU is a combinational circuit, capable of performing arithmetic and logic operations on a pair of n-bit operands. For this assignment, we are going to design an ALU which can perform 12 operations on a pair of 4-bit operands. The functions performed by the ALU will be selected by means of 3 selection lines. These 3 selection lines (S2, S1, S0), along with a carry input Cin will determine the ALU operation to be performed through 4-bit output (F3, F2, F1, F0). 4 bit status register containing Zero, Carry, Sign and Overflow flag will be designed too.

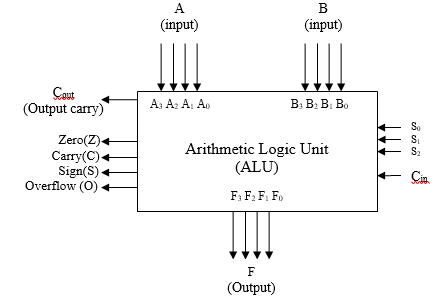


Figure 1: Block diagram of 4-bit ALU with status register

**List of Operations:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S2** | **S1** | **S0** | **Cin** | **Output** | **Function** |
| 0 | 0 | 0 | 0 | A | Transfer |
| 0 | 0 | 0 | 1 | A+1 | Increment A |
| 0 | 0 | 1 | 0 | A+B | Addition |
| 0 | 0 | 1 | 1 | A+B+1 | Add with carry |
| 0 | 1 | 0 | 0 | A-B-1 | Subtract with borrow |
| 0 | 1 | 0 | 1 | A-B | Subtraction |
| 0 | 1 | 1 | 0 | A-1 | Decrement A |
| 0 | 1 | 1 | 1 | A | Transfer A |
| 1 | 0 | 0 | X | A or B | OR |
| 1 | 0 | 1 | X | A xor B | XOR |
| 1 | 1 | 0 | X | A and B | AND |
| 1 | 1 | 1 | X | A’ | COMPLEMENT A |

**TASK BREAKDOWN**

**Part 1 (Should be delivered within 7 days of instruction)**

Design ONLY the circuit for 2-bit arithmetic operations and show the simulation to your instructor.

**Part 2 (Should be delivered within 14 days of instruction)**

1. Prepare a report.
2. Show the simulation of a 4-bit ALU performing the mentioned operations. You must show Carry Flag (C), Sign Flag (S), Zero Flag (Z) and Overflow Flag (V) separately.

**Part 3 (Should be delivered within 14 days of instruction)**

1. Implement the 3-bit ALU in hardware performing the mentioned operations. Also show all the flags.

**DESIGN AND IMPLEMENT ISSUES**

● All outputs and flags must be directed to LED.

● All inputs should be given through logic switch.

● Only basic gates (SSI- AND, OR, NOT, XOR etc.) chip can be used

**REPORT WRITING INSTRUCTIONS**

* **Problem Description**
* **Block Diagram**
* Show 4 stage full adder design with status register.
* **Design**
* Design of Arithmetic Unit with detail function specification, truth table and clearly indicate input and output columns.
* Design of Logic Unit with detail function specification, truth table and clearly indicate input and output columns.
* Combined equation of Arithmetic and Logic unit.
* Mention the flag details.
* **Test Cases**
* Show test cases for each operation you have tested your design with.
* **Circuit implementation**
* Attach the circuit diagram that you have designed using circuit maker software. It has to be neat and clearly labeled (inputs and outputs).
* **Discussion**
* Discuss the practical problems you faced while performing the experiment.

**Marks Distribution (50)**

|  |  |
| --- | --- |
| Report | 10 |
| Viva | 10 |
| Software Simulation | 20 |
| Hardware Simulation  Bonus (Efficient design) | 10  5 |

**Experiment 3: Design and implementation of a 4-bit Microprocessor (SAP-1)**

**PROBLEM DESCRIPTION**

We are going to design a 4-bit microprocessor (SAP-1) which introduces the basic necessities for a functional microprocessor. It is combination of Program counter, Input and MAR, RAM, Instruction Register, Accumulator, ALU, B Register, C Register, 2:1 MUX, Output Register, Ring Counter, Instruction Decoder, Power Supply, clock circuit, clear circuit, and Control –matrix. It has 8 bit wired bus (4 bit address, 4 bit data, 12 bit control). We’ve designed it to run 9 instructions (LDA, ADD, SUB, MOV, AND, JMP, JZ, OUT, HLT). Many types ICs, Switches, Registers, PCB etc. are used which give us different knowledge about computer circuitry.

**List of Operations:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction** | **Operation** | **Function** | **OPCODE** |
| LDA [M] | A=[M] | Load RAM data into accumulator | 0000 |
| ADD B | A=A+B | Add B with accumulator | 0001 |
| SUB B | A=A-B | Subtract B from the accumulator | 0010 |
| AND B | A=A^B | AND the accumulator with the designated register | 0011 |
| MOV A,B | A=B | Move B register to the accumulator | 0100 |
| JMP [M] | PC=[M] | Jump to designated memory location | 0101 |
| JZ [M] | PC=[M] | Jump to designated memory location if z=1 | 0110 |
| OUT | OUT=A | Load accumulator data into output register | 1110 |
| HLT | No Operation (NOP) | Stop all the process | 1111 |

**TASK BREAKDOWN**

**Part 1 (Should be delivered within 7 days of instruction)**

Design the control matrix circuit for 4-bit microprocessor performing the mentioned operations and show the simulation to your instructor.

**Part 2 (Should be delivered within 14 days of instruction)**

1. Prepare a report.
2. Show the simulation of a 4-bit microprocessor performing the mentioned operations. Show each T state clearly.

**DESIGN AND IMPLEMENT ISSUES**

● All outputs and flags must be directed to LED.

● All inputs should be given through logic switch.

● Any SSI (AND, OR, NOT, XOR etc.) and MSI (MUX, Decoder, Adder etc.), ALU chip can be used.

**REPORT WRITING INSTRUCTIONS**

* **Problem Description**
* **Architecture of SAP-1**
* Show block diagram
* **Components details of SAP-1**
* **Required IC’s List**
* **Instructions with T states**
* **Control word equations**
* **Discussion**
* Discuss the practical problems you faced while performing the experiment.

**Marks Distribution (50)**

|  |  |
| --- | --- |
| Report | 10 |
| Viva | 10 |
| Design | 10 |
| Software Simulation  T state minimization | 20  5 |
| Total | 50(+5) |

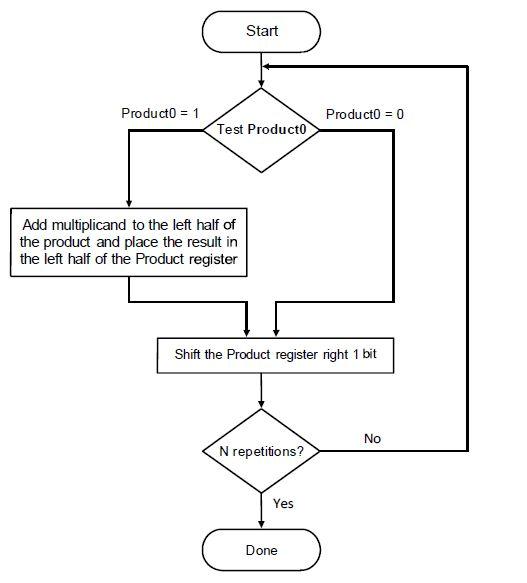
**Experiment 4: Sequential Shift-Add Multiplier**

**Design**

**PROBLEM DESCRIPTION**

You have to design a **4**​ **bit** multiplier​ digital system that will perform the multiplication of two binary numbers represented in **sign**​ **magnitude** representation. The multiplier will resemble the algorithm of elementary school’s sequential shift-add approach.

**FLOWCHART**



7

**DESIGN AND IMPLEMENTATION ISSUES**

* All outputs must be directed to LED.
* Multiplication will start only if an input signal **qm**​ =​ 1
* Use **labels**​ for​ connection rather than placing wire.
* Design the control unit using **one**​ **flip-flop per state** method​.

**REPORT WRITING INSTRUCTIONS**

* Show block diagram of Equipment​Configuration.
* Flowchart​ of​ the Algorithm
* State​ Diagram
* Data​ processor
* ​Control​ State Design
* Block​ Diagram of Circuit

**Marks Distribution (30)**

|  |  |
| --- | --- |
| Report | 10 |
| Viva | 10 |
| Software Simulation | 10 |
| Total | 30 |