Embedded System Design (CS60087)

Assignment - 4

Submitted to:
Dr. Anupam Basu
Professor
Dept. of Computer Science & Engineering
IIT Kharagpur

Submitted by:
Bhiman Kumar Baghel
17CS60R74
M. Tech 1st Year CSE
IIT Kharagpur

Objective: Design an ASIP to find MAX, MIN and AVG of ‘n’ given numbers

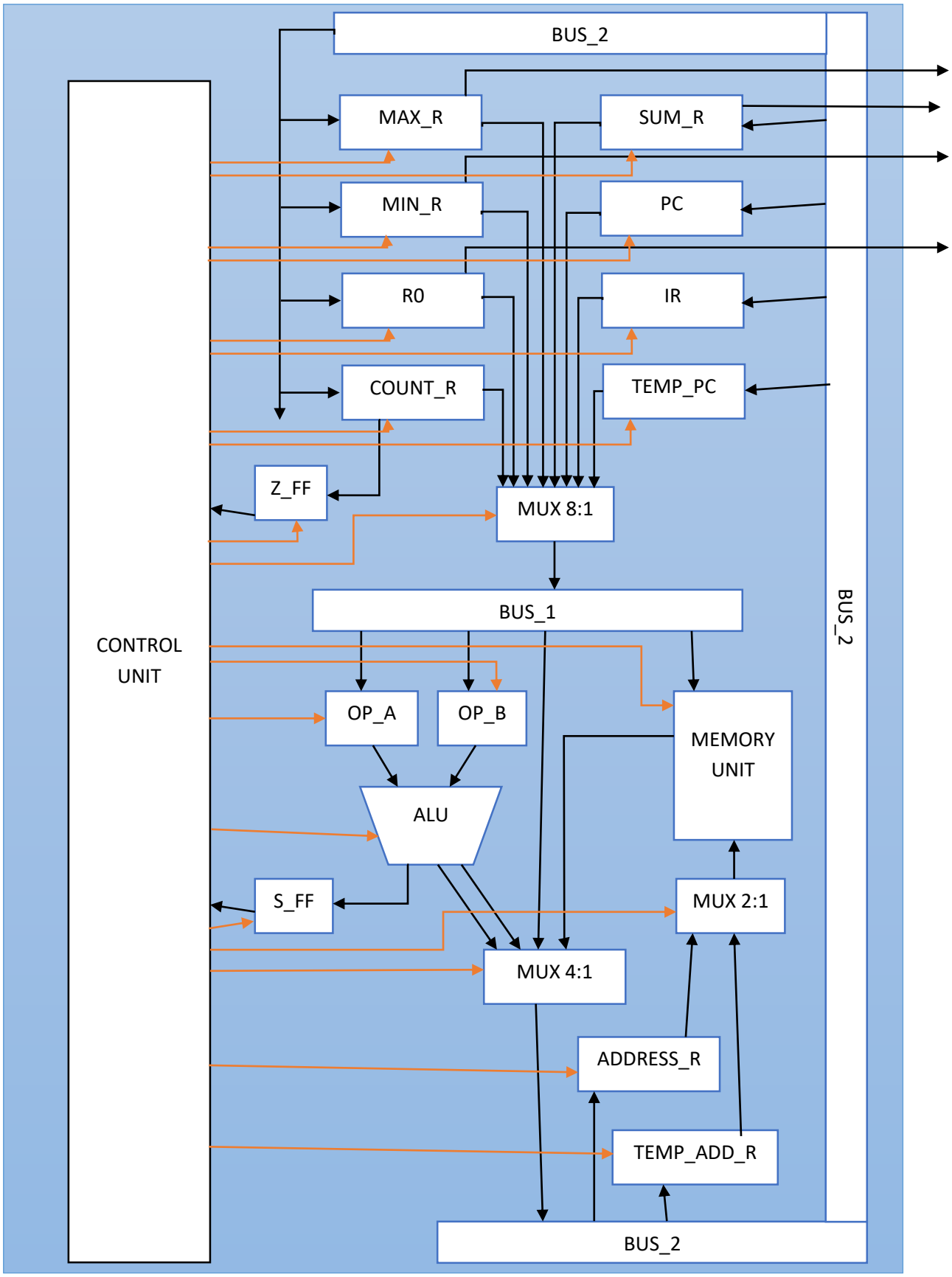
Solution:

I have designed an 8-bit ASIP which perform the MAX, MIN and AVG operation. The needs to write ‘n’ and that many numbers to the memory provided and then start the ASIP. It will generate the desired output.

The ASIP consists of

- Control Unit
- ALU : Size – 8 bits, Operation – Addition, Comparison and Division
- Memory Unit : Size – 8x256 bits
- MAX_R : Type – Register, Size – 8 bits
- MIN_R : Type – Register, Size – 8 bits
- R0 : Type – Register, Size – 8 bits
- COUNT_R : Type – Register, Size – 8 bits
- SUM_R : Type – Register, Size – 8 bits
- PC : Type – Register, Size – 8 bits
- TEMP_PC : Type – Register, Size – 8 bits
- ADDRESS_R : Type – Register, Size – 8 bits
- TEMP_ADD_R : Type – Register, Size – 8 bits
- OP_A : Type – Register, Size – 8 bits
- OP_B : Type – Register, Size – 8 bits
- MUX 8:1
- MUX 4:1
- MUX 2:1
- BUS_1
- BUS_2
- Z_FF
- S_FF

High Level Diagram



Instruction Set:

- LDA : Opcode – 00000000
- INC_ADD : Opcode – 00000001
- MOV : Opcode – 00000010
- MAX : Opcode – 00000011
- MIN : Opcode – 00000100
- SUM : Opcode - 00000101
- DIV : Opcode - 00000110
- LOOP : Opcode - 00000111
- LOOPEND : Opcode - 00001000
- END : Opcode – 00001001

Program:

The program to find MAX, MIN and AVG is:

1. LDA 16
2. LOOP
3. INC_ADD
4. MOV
5. MAX
6. MIN
7. SUM
8. LOOPEND
9. LDA 16
10. DIV
11. END

Memory:

The above program is loaded in memory as:

- Memory[0] = 00000000;
- Memory[1] = 00010000;
- Memory[2] = 00000111;
- Memory[3] = 00000001;
- Memory[4] = 00000010;

- Memory[5] = 00000011;
- Memory[6] = 00000100;
- Memory[7] = 00000101;
- Memory[8] = 00001000;
- Memory[9] = 00000000;
- Memory[10] = 00010000;
- Memory[11] = 00000110;
- Memory[12] = 00001000;

The numbers are loaded from memory location 16

- Memory[16] = 00000101; //Count of Numbers
- Memory[17] = 00000111;
- Memory[18] = 00001000;
- Memory[19] = 00000010;
- Memory[20] = 00000011;
- Memory[21] = 00000101;