COL215- ASSIGNMENT 6

Memory Read and Write

COL- 215

HARSH VERMA	PARTH RATHI ARVIND
2024CS10499	2024CS50875

Table of Contents

1. Aim / Introduction

The aim of this lab assignment is to design and implement a memory control system capable of performing vector addition using three memory blocks: a Read-Only Memory (ROM) and two Random-Access Memories (RAM0 and RAM1). The control logic supports read, write, and increment operations based on user inputs through switches. The system reads two 4-bit input vectors (A and B) of size 1024 and computes their sum (C), where each element satisfies C[i] = A[i] + B[i]. The results are stored in RAM1 and displayed on a 7-segment display. The design also handles reset conditions by displaying '-rSt' on the display.

2. Design and Implementation

Top Module: Controller

The controller manages data flow between the memories, performs arithmetic operations (addition of vectors A and B), and updates the result on the seven-segment display. It also handles the reset functionality and ensures proper synchronization between read, write, and increment operations.

Following is the implementation of the code

1. ROM Instance – Stores the input vector A with 1024 4-bit elements. Initialized using a .coe file.
2. RAM0 Instance – Stores the input vector B with 1024 4-bit elements. Supports write and increment operations via switches.
3. RAM1 Instance – Stores the computed output vector C, where each element is 5 bits.
4. Control Unit – Decodes switch inputs (SW15–SW0) to perform read, write, or increment operations.
5. Seven-Segment Display – Displays ROM, RAM0, and RAM1 contents depending on mode select signals.
6. Reset Logic – Handles debouncing and triggers a display of '-rSt' for 5 seconds when BTNC is pressed.

Some decisions with respect to the design

- Switches (SW15–SW0) used for manual control of address, data, and operation.
- One-pulse logic ensures single write/increment per switch press.
- Debounced reset with “-rSt” display for clear system initialization.
- Distributed memories used for low resource usage and scalability. This works as the data is low.
- All operations are synchronous with the system clock for stable timing.
- Memory outputs registered to avoid glitches during reads.

3. Testbench and Simulation

The testbench verifies functional correctness through the following tests:

- Reset functionality: Ensures '-rSt' is displayed for 5 seconds.
- Read operation: Confirms that ROM, RAM0, and RAM1 contents are correctly read and displayed.
- Write operation: Validates that the correct 4-bit value is written to the desired address in RAM0.
- Increment operation: Ensures that stored values in RAM0 are incremented correctly and the resultant sum is reflected in RAM1.

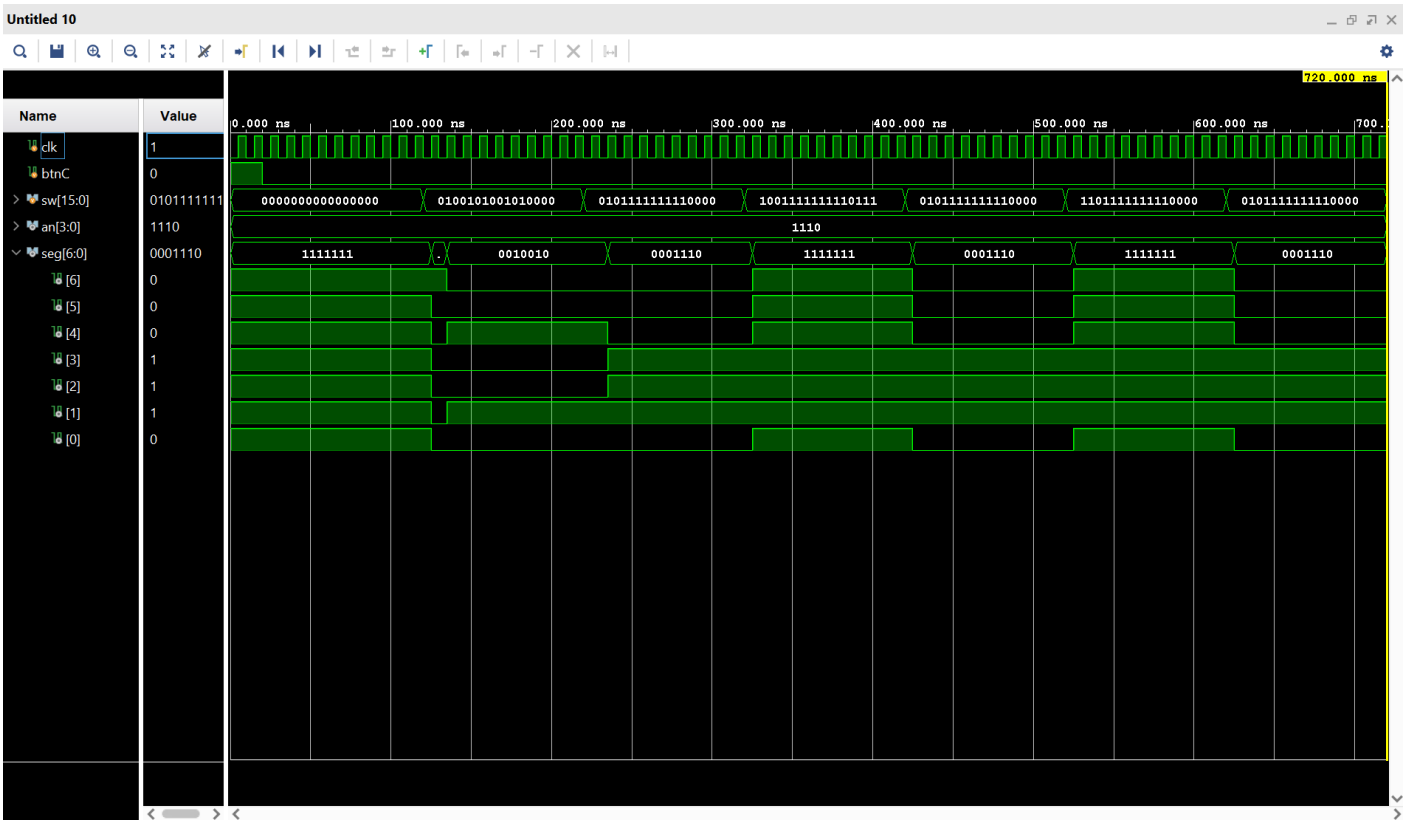
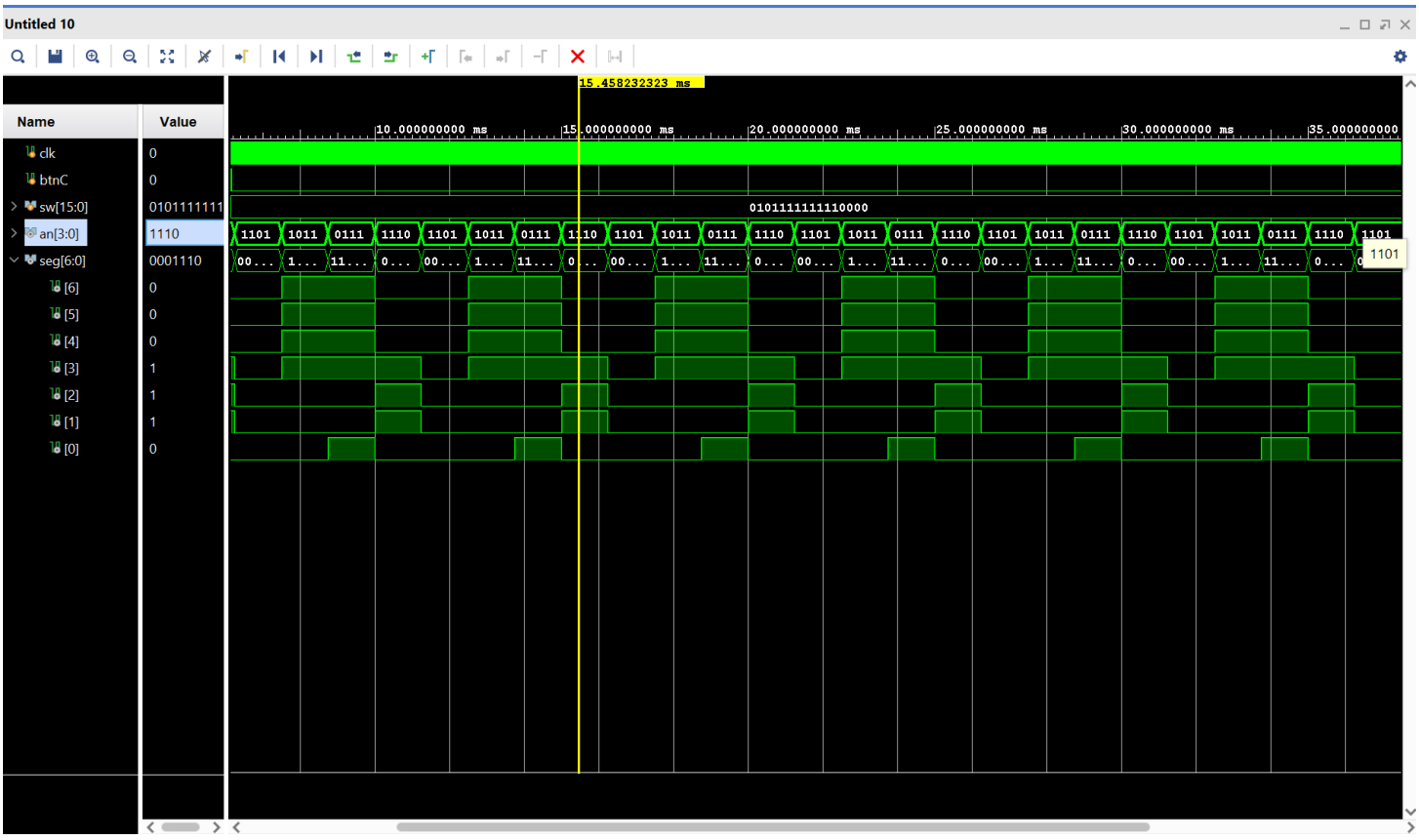


Figure 1 This shows the different cases for the switching read and write enable.



This shows the values displayed on the anodes

Note that the simulation had to be run for longer time to display all the results of the anode.

Starting at the yellow line shows an[0]=F, an[1]=8, an[2]=7 and an[3]=1 which is the correct sum.

4. Constraints and Hardware Testing

The design was synthesized and implemented on the Basys3 FPGA board. Switches SW15–SW0 were mapped according to the assignment specifications. During hardware testing, the following were verified:

- Proper reset indication on the 7-segment display.
- Correct data read and display from ROM, RAM0, and RAM1.
- Successful write and increment operations without glitches.

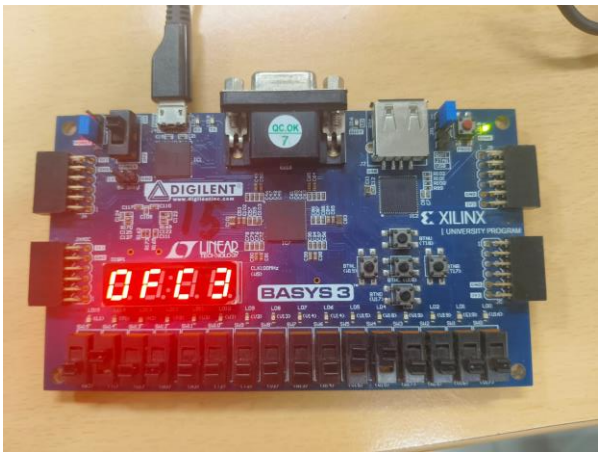
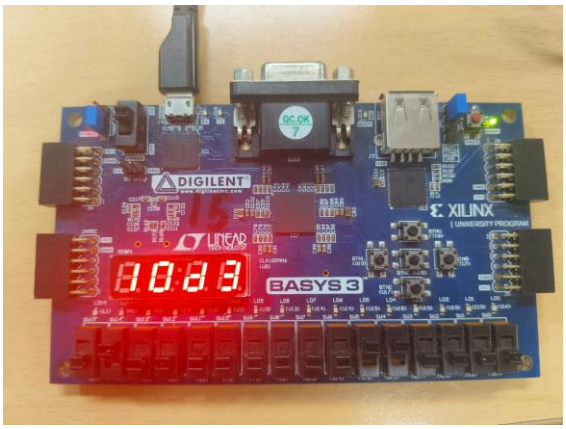
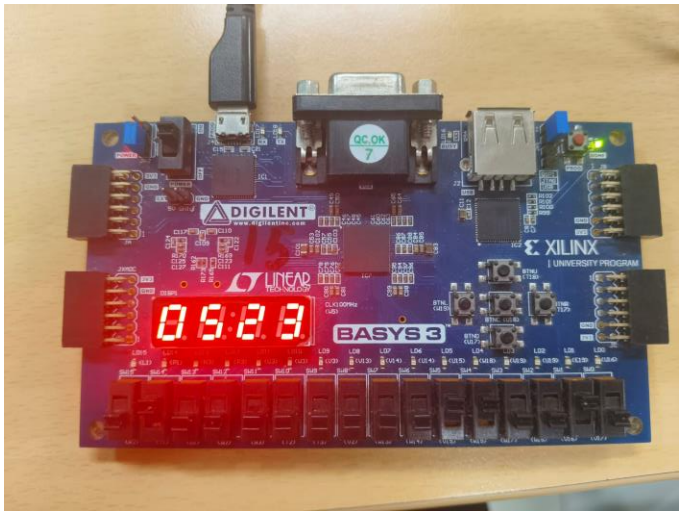


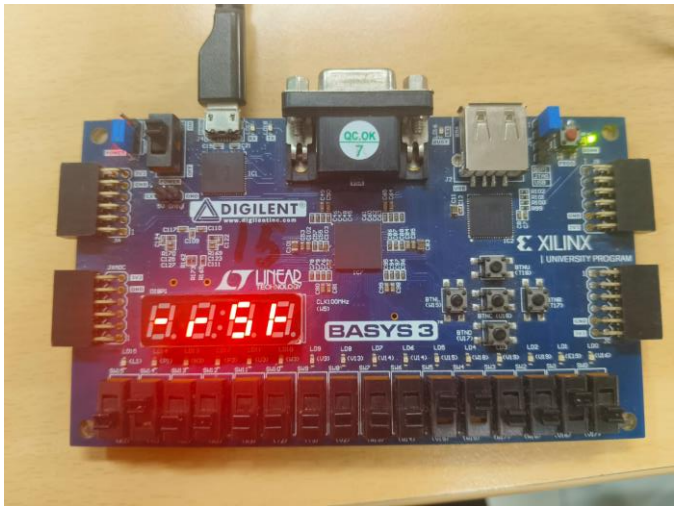
Figure 2 Displaying initial A[3], B[3] and C[3]



Inreasing the value of the B[3] and displaying the Sum



Written the value of B[3] and displayed the sum



Reset function using the push button

5. Resource Utilization

Resource utilization after synthesis and implementation.

- No Block RAMs or DSPs are used, reflecting the assignment's focus on small vector memory and basic arithmetic only.
- Utilization for all main resources (LUT, LUTRAM, Flip-Flop) is very low, confirming the design's efficiency.
- Flip-Flops usage is minimal, indicating few sequential registers needed.

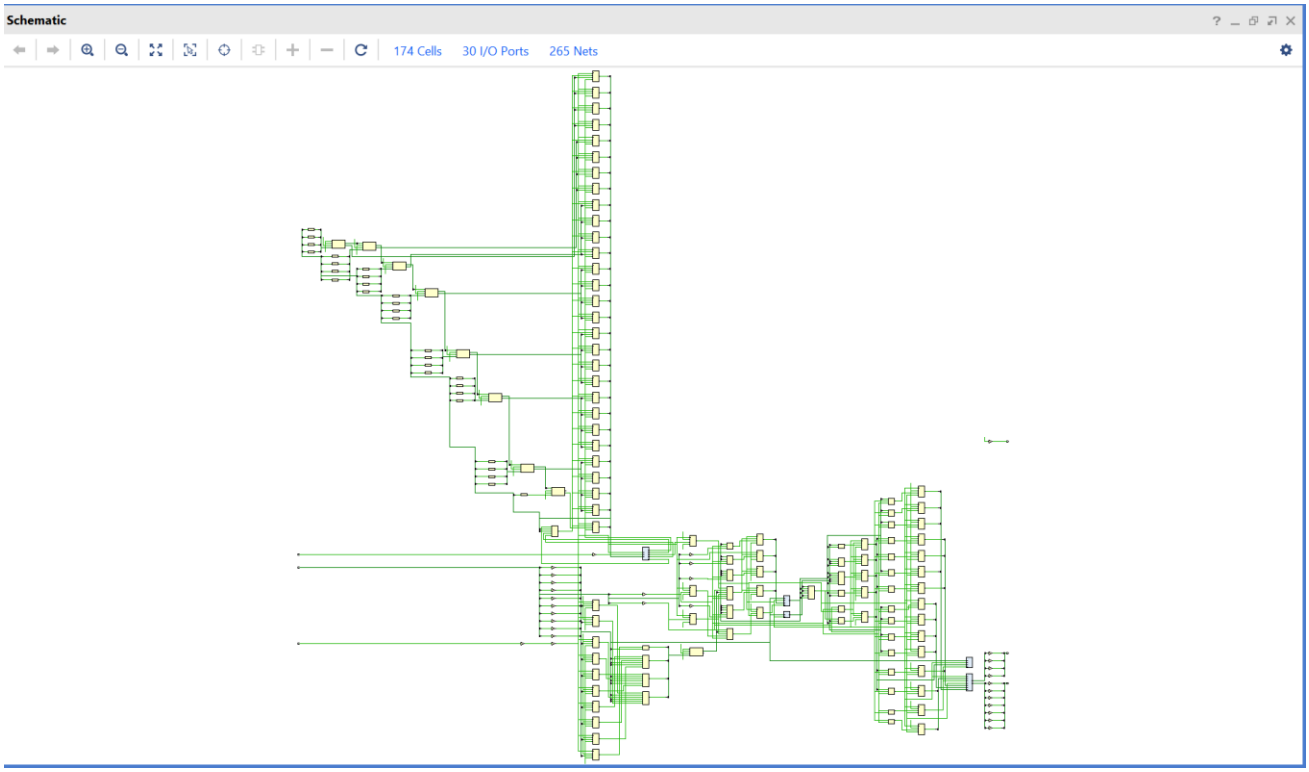
Resource Type	Used	Available	Utilization (%)
LUTs	174	20800	0.84
LUT RAM	64	9600	0.67
Block RAMs	0	50	0.00
DSPs	0	90	0.00
Flip Flops	115	41600	0.28

LUT	FF	BRAM	URAM	DSP	S
105	**107**	**0**	**0**	**0**	
174	115	0	0	0	

Resource	Utilization	Available	Utilization %
LUT	174	20800	0.84
LUTRAM	64	9600	0.67
FF	115	41600	0.28
IO	30	106	28.30

6. Schematics

The synthesized schematic consists of interconnected blocks representing ROM, RAM0, RAM1, the control unit, and the display controller. The memory instances are generated using Vivado's Distributed Memory Generator IP core, configured with 1024 addresses and respective bit widths.



Schematics of the design

7. Conclusion

The Memory Read and Write system was successfully implemented and verified through simulation and hardware testing. It demonstrated correct functionality across all modes — read, write, and increment. The reset logic and seven-segment display enhanced usability and clarity of operations. Resource utilization remained low, confirming design efficiency and scalability for larger memory sizes or extended functionalities.