COL215
Digital Logic & System Design

Assignment 4

Submitted by:
Dhairya Kuchhal (2024CS50396)
Siddhant Agrawal (2024CS50469)

August 28, 2025

1 Task

The objective of this lab was to design, simulate, and implement Verilog files for a MAC (Multiply-Accumulate) to take in inputs from the switches and display summed output via LEDs. The case of Overflow (number larger than our LED's can represent) is handled with Overflow and Reset being displayed on 7-segment display when required.

2 Design Decisions

We kept a modular design for clarity and ease of debugging. There were total 4 auxiliary modules and one Main one.

Module Name	Funtion
eight bit multiplier	inputs b,c and outputs product
accumulator	accumulates the product and checks overflow
rising edge detector	ensures accumulation happens only once on turning switch on
display	handles display of OFLO and -rst
main	brings together the logic and maps to basys ports

Table 1: Modules in code

INPUTS TO B AND C: Inputs to b and c were governed by an activation based logic. since we reuired seven inputs for each b and c, instead of alloting all of sw[15:0] with 8 switches each for b and c, we

1. used only sw[7:0] to represent a magnitude
2. used the state of sw[10] and sw[11] to decide who to allot to

CHECK FOR OVERFLOW: we used a register fallback of 17 bits to keep checking the value of our accumulated ouptut a, and report an overflow when the MSB of our fallback turns to 1. A parametricized value was used to set the default value of our accumulator to 0

DEBOUNCER: we used an intermediate temp variable and implemented a counter counting the number of clock cycles passed with noisy input and its previous value (allotted to temp) remains the same. hence it checks for consistency of output signal. once it remains constant for a parametricised while, the main (clean) signal is updated

DISPLAY: Display of reset for 5 seconds is implemented with a counter, with value calculated taking into account Basys3's internal clock frequency of 100MHz

2.1 Simulation using Testbench

We wrote a dedicated testbench module (seven_tb.v) for simulation, checking the following conditions:

1. Display and re-initialisation of values in case of Reset button
2. Overflow and resetting of Accumulator to zero
3. Calculations with the MAC module

2.2 Mapping of Pins on Basys3

The constraint file (basys3.xdc) was edited to bind logical inputs and outputs to physical switches and the display segments in a user-friendly, simple left to right manner. The following table summarizes the mapping:

Signal Type	Logical Name	Function
Input	$sw[7:0]$	take inputs b,c
Input	$sw[11:10]$	choose between b, c
Input	$sw[12]$	Accumulate
Input	$seg[6:0]$	7-segment display cathodes
Input	$anode[3:0]$	7-segment display anodes
Input	$led[15:0]$	Accumulated value
Input	$btnc$	reset button

Table 2: Mapping of pins and I/O

3 Synthesis Report

The following tables show the main resource counts. Other details are present in the synthesis report.

1. Slice Logic

Site Type	Used	Fixed	Prohibited	Available	Util%
Slice LUTs*	120	0	0	20800	0.58
LUT as Logic	120	0	0	20800	0.58
LUT as Memory	0	0	0	9600	0.00
Slice Registers	111	0	0	41600	0.27
Register as Flip Flop	111	0	0	41600	0.27
Register as Latch	0	0	0	41600	0.00
F7 Muxes	0	0	0	16300	0.00
F8 Muxes	0	0	0	8150	0.00

Figure 1: Flip-Flops and LUT

3. DSP

Site Type	Used	Fixed	Prohibited	Available	Util%
DSPs	0	0	0	90	0.00

Figure 2: DSP usage

2. Memory

Site Type	Used	Fixed	Prohibited	Available	Util%
Block RAM Tile	0	0	0	50	0.00
RAMB36/FIFO*	0	0	0	50	0.00
RAMB18	0	0	0	100	0.00

Figure 3: BRAM usage

4 Simulation Snapshots

Figure shows the behavioural simulation results on running tb_main.v . the varying values of input and corresponding outputs are tracked.

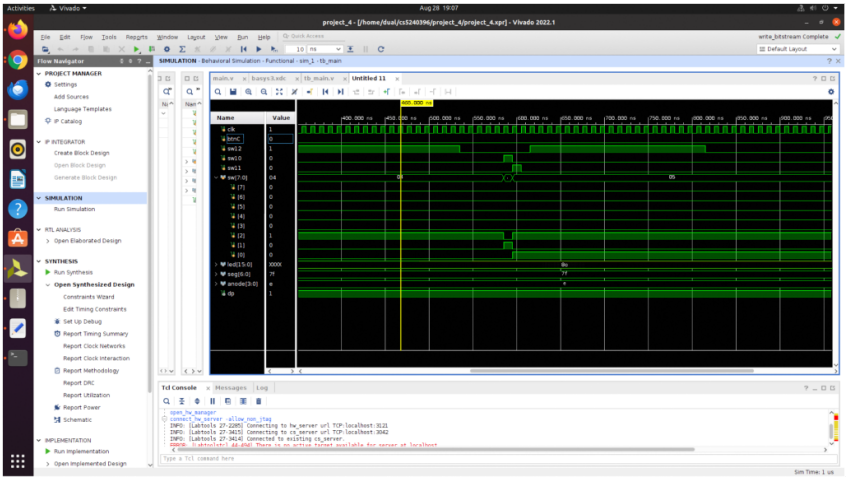

Figure 4: Simulation

5 Generated Schematics

This makes us see beyond the code and into the actual fpga boards wiring between the various combinational gates and ports

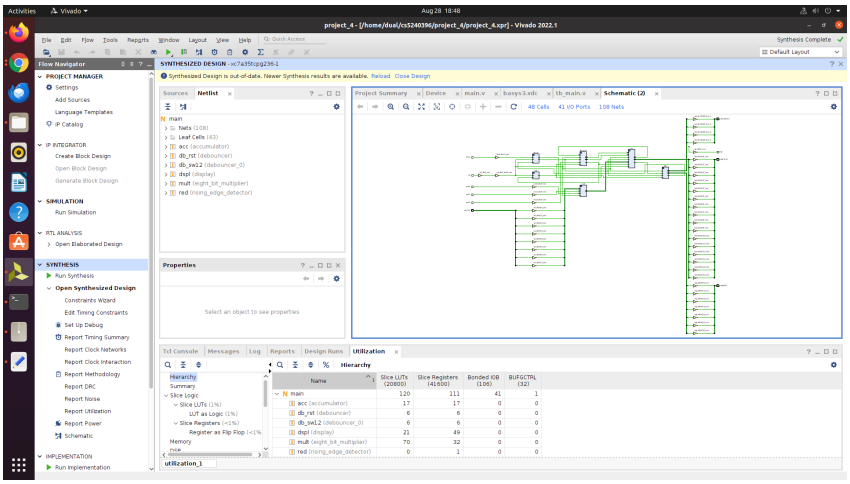

Figure 5: Generated Schematic design