COL215 – Digital Logic and System Design
Department of Computer Science & Engineering, IIT Delhi
Semester I, 2025-26
Lab Assignment 7

Linked List

1 Introduction

In this assignment, the objective is to implement the singly linked list data structure in Verilog HDL. As we don't have dynamic memory allocation, we would need to implement the linked list using arrays.

2 Problem Description

2.1 Switch and pin functions

The functionality of different I/O pins and switches are listed in Table 1.

Table 1: Pin and Switch Function Assignments

Pin / Switch	Function
$SW7 - SW0$	Data to be written to a node
$SW15 - SW13$	Operation: `000-011`: Idle. `100`: Insert at the head. `101`: Insert at the tail. `110`: Delete. `111`: Traverse.
$LED0$	Indicate an overflow condition
$LED1$	Indicate an underflow condition
$BTNC$	Reset. Clears all the list elements. Displays '`-rSt`' on the 7-segment display for 5 seconds.

Design a hardware module in **Verilog** to simulate a **singly linked list** with the following features:

2.2 Node Structure

Each list node should have two fields:

- **Data**: 8-bit data.
- **Next node pointer**: Index (pointer) to the next node.

Use an array to store the linked list nodes, each element containing data. You may use separate arrays to keep track of the next node pointer and free nodes.

2.3 Supported Operations

Implement the following operations in your Verilog code:

1. **insert_at_head**
   Insert a node at the beginning of the linked list.
2. **insert_at_tail**
   Insert a node at the end of the linked list.
3. **delete_node**
   Delete the first node with the given data value.
4. **traverse**
   Print the data values sequentially in the linked list on the 7-segment display0 (7SD0).

3 Assignment tasks

Implement a linked list in Verilog HDL with below features:

- Use reg arrays for the data structure.
- The number of nodes should be parameterized. The default parameter value should be 32.
- Write a testbench to:
  - Insert nodes at the head.
  - Insert nodes at the tail.
  - Delete a node.
  - Traverse the linked list and print the data on the 7-segment display.

4 Submission and Demo Instructions

1. Demo should be given in the assigned lab slot itself.
2. You are required to submit the following on Gradescope.
   - 20 points: Verilog files for all the designed modules. Only the modules written by you are needed. The IPs generated, i.e., distributed memories must not be submitted.
   - 20 points: Verilog files for the testbench and the simulation of all the operations.
   - 10 points: Warning-clean constraint file (.xdc), bit file.
   - 30 points: Questionnaire corresponding to the assignment.
   - 20 points: A short report (3-4 pages) outlining simulation snapshots and generated schematics. Explain your design decisions. Snippets in the report without explanation will result in penalty.

NOTE:

- Ensure all the files in the zip folder are correctly visible on gradescope. Later requests about technical glitches with gradescope and files being in zip folder and not visible on gradescope and being part of report wouldn't be considered anymore.
- The report must be in pdf format only (docx will not be considered any more) and should consist of the points mentioned in the Submission section. Not mentioning utilization data specifically in the report (and relying on synthesis screenshots) will not be considered anymore.
- The submission must have only the above listed items. Submitting additional files such as video of board functionality, separate utilization report, vivado xpr or dcp files, etc. will result in penalty.

We advise you to be ready with your design before the lab session, and during the session, perform validation by downloading it into the FPGA board.

5 Resources references

- IEEE document: `https://ieeexplore.ieee.org/document/1620780`
- Basys 3 board reference manual: `https://digilent.com/reference/_media/basys3:basys3_rm.pdf`
- Online Verilog simulator: `https://www.edaplayground.com`