CSE 2441 **DIGITAL LOGIC DESIGN** **FALL 2021**

TERM DESIGN PROJECT

Objective: The term project consists of a design and implementation task that is based on knowledge you gained throughout the semester in the lecture, lab sessions, and homework. It is meant to challenge your ability to apply your understanding and knowledge of logic design on a practical problem. This is an **individual** project, so group design and reporting is neither appropriate nor acceptable.

Grading: The project will be graded as follows and will count 10% toward your final CSE 2441 course grade. Failure to complete the project will result in a course grade of Incomplete or F depending upon your standing in the course.

> **90 points** – design completed, simulated, implemented, tested and meets all specifications.
> **10 points** – project report completed and submitted and meets all requirements.
> **Bonus points** (30 maximum) – meet early completion deadlines. Your report must be complete, and your machine must properly execute the test program to be eligible for bonus points.

ABET outcome assessment: ABET, Inc. is the organization that accredits engineering programs in the United States and in many other countries around the world. One of the accreditation criteria requires documentation that students have demonstrated an ability to design a system, component, or process to meet desired needs within realistic constraints. The CSE 2441 term design project has been selected as one source of this documentation for the Computer Engineering Program at UT Arlington.

Project timeline

> **11/16/2021** – Project assigned.
> **12/3/2021** (11:59 PM) – First early completion deadline. (15 points maximum)
> **12/6/2021** (11:59 PM) – Last early completion deadline. (15 points maximum)
> **12/7/2021** (11:59 PM) – Project report due.

Project description – Integrate, on the DE10-Lite, the TRISC components that have been completed in the laboratory exercises and homework to implement a functioning processor that can execute programs consisting of INC, CLR, JMP, LDA, STA, and ADD instructions. Only use the components necessary as shown in Figure 1. This will simplify the integration and testing.

- Inputs – Start/Stop (KEY0), SysClock (KEY1)
- HEX outputs – PC (HEX5), MAR (HEX4), MDOut (HEX3 & HEX2), MDIn (HEX1 & HEX0)
- LED outputs – C0 (G0), C1 (G1), C2 (G2), C3 (G3), C4 (G4), C5 (G5), C7 (G6), C8 (G7), C9 (G8), C10 (G9), C11 (R6), C12 (R7), C13(R8), and C14 (R9). You may use the remaining LEDs for debugging purposes in any way you see fit.
- Must properly execute the program stored in triscRAM (fall 2021) to earn full credit. The RAM is posted on CANVAS.

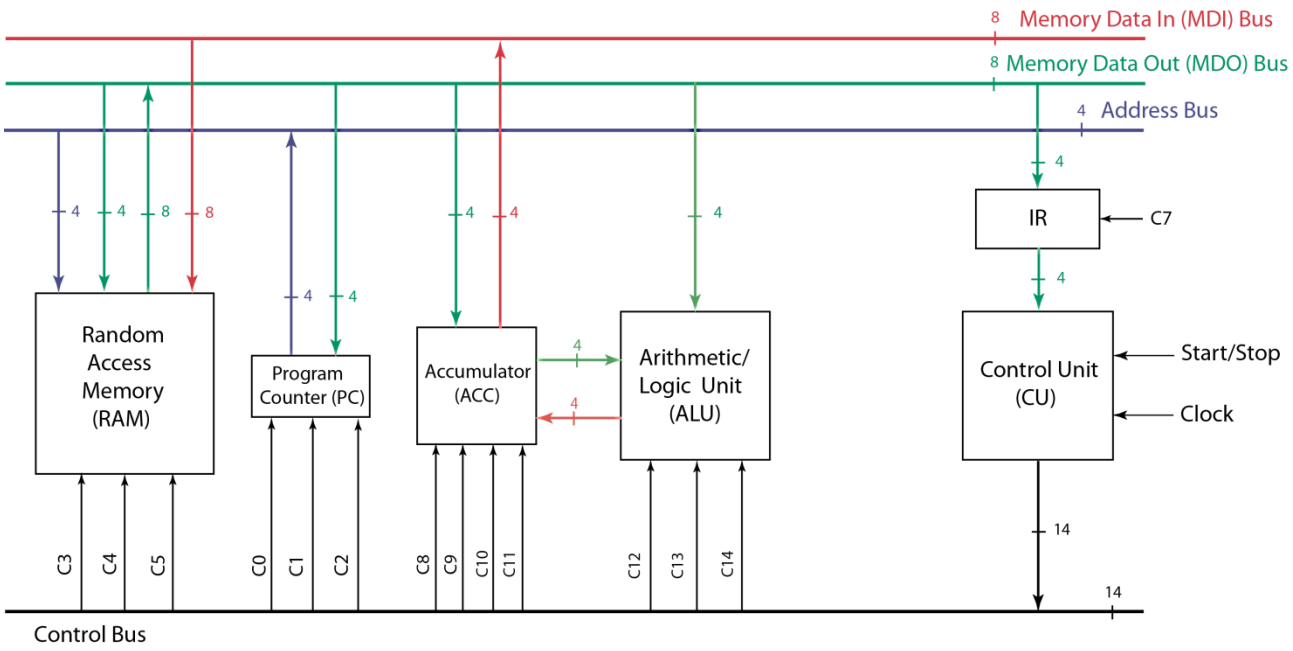

Figure 1 – TRISC for INC, CLR, LDA, STA, ADD, and JMP.

Project Report (File name—*yourname*2441ProjectReport.pdf) The report should be structured as follows.

- Cover sheet
- Table of contents (with page numbers)
- Introduction
  - Project overview including requirements
  - Project status including any unresolved problems
- System design
  - System-level description and diagrams
  - Verilog code
  - Test results
- Controller design details
  - Functional description and diagram showing I/O
  - State diagram
  - Verilog code
- Test results
  - Summary
  - Photos or video of program execution
- Conclusion
  - Resolution of design and/or implementation issues
  - Lessons learned

APPENDIX

Control Signal Definitions (Active High)

C0 – *Clear* Program Counter (PC)

C1 – *Load* PC

C2 – *Increment* PC

C3 – *Select* Memory Address Register (MAR) source (C3=0: PC, C3=1: MDO)

C4 – *Execute* RAM READ/WRITE cycle

C5 – *Enable* RAM WRITE cycle

C7 – *Load* Instruction Regiser (IR)

C8 – *Clear* Accumulator (ACC)

C9 – *Increment* ACC

C10 – *Select* ACC source (C10=0: ALU, C10=1: MDO)

C11 – *Load* ACC

C12, C13 – Arithmetic Logic Unit (ALU) Function Code (00: ADD, 10: SUB, 01: AND, 11: XOR)

C14 – *Load* Buffer Register (BR)

C15 – *Load* Flag Register (FR)