ASSIGNMENT:  LAB #2 **COMPETING PARALLEL COUNTERS (Real-Time)**

COURSE: EGR433*(430)* Advanced Computer Engineering *(Parallel Processing)* Lecture & Lab

INSTRUCTOR:  J T Wunderlich PhD

LATE PENALTY:  Minus 33.3% per class period for each late item

LAST REVISED: -

**PRE-LAB**

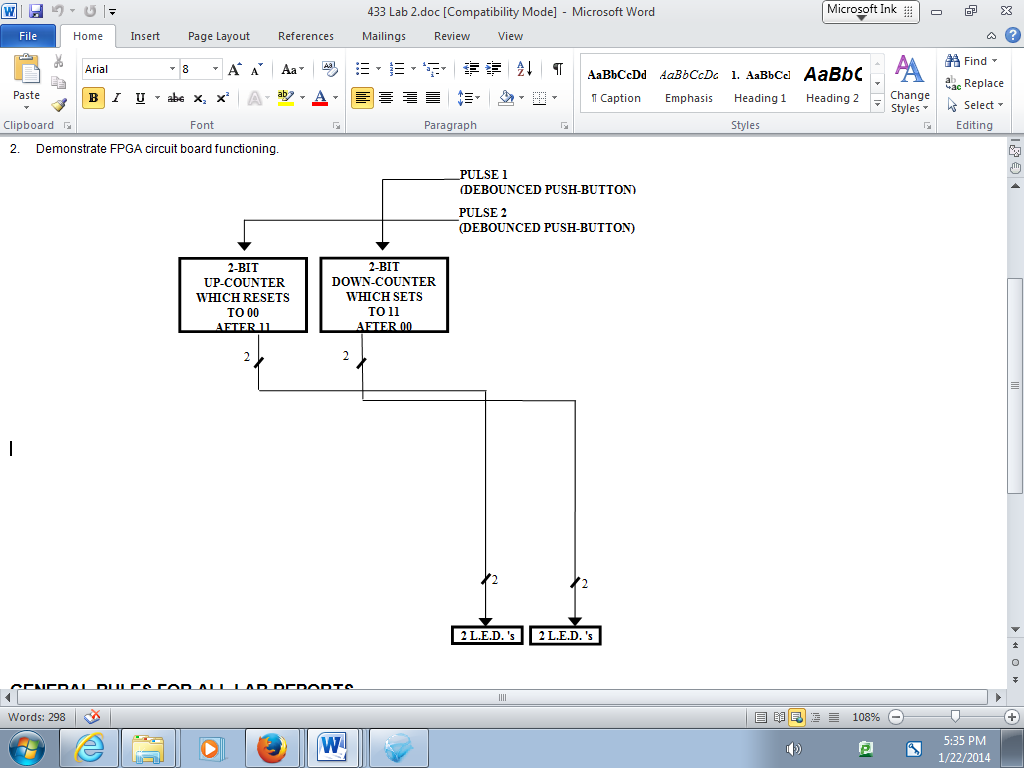
Read our Custom User Manuals:

1. [**2018 IC's, Circuit Trainer, and Power Supply**](http://users.etown.edu/w/wunderjt/Chips_PSU_Trainer_Lab%20Manual%20Flores_Csongradi_Schick_Wunderlich%20VERSION%201.2.pdf)<http://users.etown.edu/w/wunderjt/Chips_PSU_Trainer_Lab%20Manual%20Flores_Csongradi_Schick_Wunderlich%20VERSION%201.2.pdf>
2. [**2019 Relays**](http://users.etown.edu/w/wunderjt/LAB%20MANUALS%20post-2018/2019%20MANUAL%20Relays.pdf)  <http://users.etown.edu/w/wunderjt/LAB%20MANUALS%20post-2018/2019%20MANUAL%20Relays.pdf>
3. [**2019 NanoLC**](http://users.etown.edu/w/wunderjt/LAB%20MANUALS%20post-2018/2019%20MANUAL%20NanoLC%20PLC.pdf)  <http://users.etown.edu/w/wunderjt/LAB%20MANUALS%20post-2018/2019%20MANUAL%20NanoLC%20PLC.pdf>

**DURING LAB**

Design and implement the circuit below in two ways:

1. TTL SSI chips on circuit trainers using only FLIP-FLOP’s, INVERTORS, AND’s, OR’s, and/or NAND’s
2. NanoLC Programmable Logic Controllers (PLC’s) with **two separate base units** mounted on the same rail.
   * Improvise inputs with any physical switches you can find or make (debouncing not required)
   * Improvise outputs with any LED’s or light-bulbs you can find

****

**GRADE PERCENTAGES:**

(**30%)** Demonstrate in a video (3 minutes max) **PAD-234 Circuit Trainer** **circuits** [primarily for **5 volt** TTL chips]

none Demonstrate in a video (3 minutes max) **Old Circuit Trainer** **circuits** [primarily for **5 volt** TTL chips]

none Demonstrate in a video (3 minutes max) **RadioShack Circuit Trainer circuits** [primarily for **~3 volt** CMOS chips]

none Demonstrate in a video (3 minutes max) **Logisim Circuit simulations**

(**30%)** Demonstrate in a video (3 minutes max) **Phoenix Contact** **NanoLC PLC** **simulations** AND

Demonstrate in a video (3 minutes max) **Phoenix Contact** **NanoLC PLC real-time systems**

none Demonstrate in a video (3 minutes max) **Phoenix Contact** **Advanced Axioline PLC real-time systems (PCworks software)**

none Demonstrate in a video (3 minutes max) **Phoenix Contact** **Advanced PLCnext PLC real-time systems**

none Demonstrate in a video (3 minutes max) **2019 Field Programmable Gate Array (FPGA) simulations and Real-time systems (in HDL Hardware Descriptive Language)**

none Demonstrate in a video (3 minutes max) **2018 Field Programmable Gate Array (FPGA) simulations and Real-time** **systems**

none Demonstrate in a video (3 minutes max) **INTEL 8051/80251 or ARM microcontroller simulations**

none Demonstrate in a video (3 minutes max) **INTEL 8051/80251 or ARM microcontroller real-time systems**

none Demonstrate in a video (3 minutes max) **Raspberry Pi real-time systems**

none Demonstrate in a video (3 minutes max) **Arduino real-time systems**

none Demonstrate in a video (3 minutes max) **Basic-Stamp real-time systems**

none Demonstrate in a video (3 minutes max) **Direct PC-port real-time systems**

none Demonstrate in a video (3 minutes max) **Remote mobile-device real-time systems**

none Demonstrate in a video (3 minutes max) **LabView real-time systems**

none Demonstrate in a video (3 minutes max) **Isolated high-voltage bench-test** (with low Voltage electronics disconnected)

none Demonstrate in a video (3 minutes max) **Other**:

(**40%)** Written Report

**LAB RULES & PROCEDURES**

**IF LABS ARE BUILT (AND POSSIBLY REBUILT), BUT DON'T FULLY FUNCTION:** For **demonstrations** and **reports**, deduct depends on how adequately you identify problems. For example, make test set-ups to verify functionality of isolated simulation sub-parts, chips, circuit trainer elements, software, relays, other electronics, motors or other higher-voltage circuits and devices. PROVE THAT NO EASY FIX OR SUBSTITUTION WAS POSSIBLE or EASILY IDENTIFIABLE AT THE TIME. Discuss (1) How you identified problems, and (2) How you tried to fix them. Include evidence that you fully understand and have properly connected all pins on a given chip (including considering floating-pins, powering the chip, needed pull-up resistors, proper voltage levels, etc.), and that you have exhausted much time attempted to solve all problems.

**DEMONSTRATIONS**

Alternate team members demonstrating lab in video.

**GRADING**

For both demonstrations and reports, a **92** is for everything done very well and professional; to get more points, enhance things in creative ways

**REPORTS** must include:

* **PHOTO’S OF ALL CIRCUITS AND TEST SET-UP’S BUILT**
* Title Page with lab number, name of lab, your names, Majors, Year (e.g., Junior), who is demonstrating, and who is the designated TEAM LEADER
* Sections numbered and tilted as follows (always list all of these, and simply put “NA” if not applicable):

1. **“Assignment”** (An exact copy of everything in this document -- exactly how it looks here)
2. **“Equipment Used”** (A list of hardware and software) **INCLUDE PHOTO’S OF ALL EQUIPMENT**
3. **“Methodology”** (including all design steps, analysis, **DECISIONS MADE**, etc.) **INCLUDE PHOTO’S OF ALL CIRCUITS BUILT**
4. **“Options”** (if applicable, a comparison of each method used)
5. **“Problems Encountered”** (including any debugging methodology) **INCLUDE PHOTO’S OF ANY TEST-CIRCUITS BUILT**
6. “**Testing Methodology**” (including timing traces, test-vectors, and **RATIONALE FOR HOW YOUR METHODOLOGY ASSURES QUALITY**)
   * **Including estimated** **PROBABILITY of satisfactory coverage by chosen test vectors**
7. **“References”** (in standard IEEE format)
8. **“Appendices”** (for spec sheets, etc.)

* **ALL DESIGN PROCESS STEPS** **MUST BE INCLUDED** for Digital Logic designs (**NUMBERED** as in EGR/CS 332/330). If design step not done, list as “N.A.”

*For Combinational Digital Logic Design:*

Step 1: Define problem

Step 2: Encode variables

Step 3: Create truth table

Step 4: Find simplified function(s)

Step 5: Draw logic circuit

Step 6: Convert to NAND’s

Step 7: Check assumptions

Step 8: Chip circuit diagram

*For Sequential Digital Logic Design::*

Step 1: Define problem

Step 2: Create state diagram

Step 3: Encode variables

Step 4: Minimize machine

Step 5: Create state table

Step 6: Append flip-flop inputs

Step 7: Find simplified function(s)

Step 8: Draw logic circuit

Step 9: Convert to NAND’s

Step 10: Analyze any unused states

Step 11: Revise state diagram

Step 12: Check Assumptions

Step 13: Chip circuit diagram

* **COLOR-CODED LOGIC DIAGRAMS** are required for any digital circuit (Breadboard, FPGA, etc.)
* **COLOR-CODED CIRCUIT SCHEMATICS** are required for any circuit implemented (Breadboard, PLC, ladder logic, etc.), color is a must, hand-colored is ok
* **FLOW CHART** is required for any program
* **COMMENT EVERY LINE OF CODE**
* **TEAM LEADER** has responsibility of coordinating all equipment problems and acquisition of needed parts with TA. Try to stick with same person for this role.

*Remember to include all of this document in the beginning of your report*