LAB #9,10,11 "Computer Design Competition"

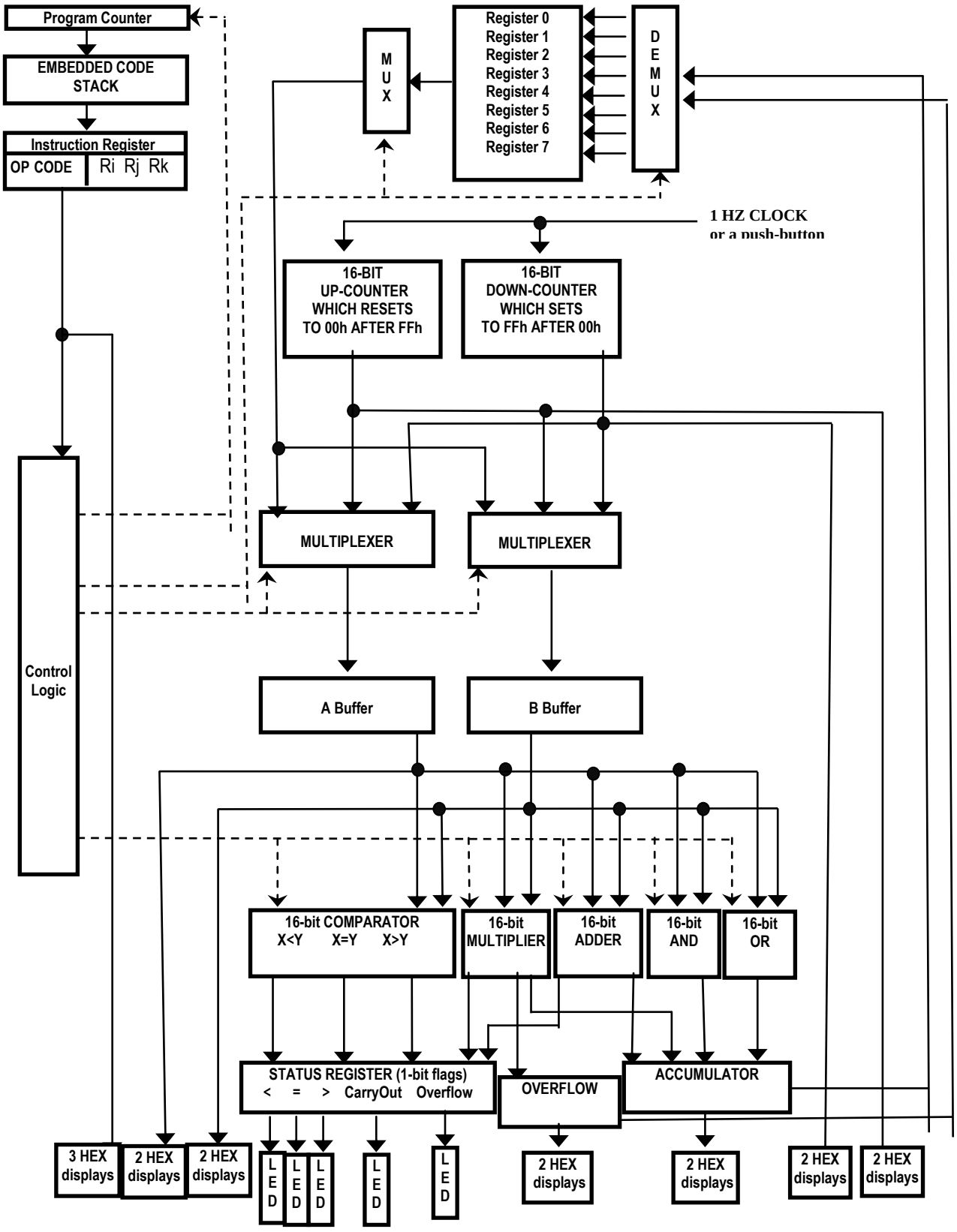
COURSE: EGR/CS433 **"Advanced Computer Engineering"** Lecture and Lab
SYLLABUS: http://users.etown.edu/w/wunderjt/syllabi/CS433%20Wunderlich,Joseph.htm
INSTRUCTOR: J. Wunderlich PhD
LATE PENALTY: Minus **33.3%** per class period for each late item
LAST REVISED:

1) Use Logisim to Implement the instruction set and circuit below including a pipeline for Fetching, Decoding, and Executing your instructions; and including the design of a Finite State Machine in your control unit logic to drive your pipeline.
2) Design a piece of software to embed using your instruction set **(You will be graded on a curve for the complexity of your code, and you may add instructions to your instruction set if you wish)**
3) Use an 8051 microcontroller simulator to perform the exact same task as your machine's embedded code; or some alternative assembly language execution (ARM. Motorola, etc.) Microcontroller or Microprocessor.
4) Compare the performance of your machine to that of at least one existing assembly language

```
        00h (OP-CODE = 000000 00) Ri + counter#1 → Rk
        01h (OP-CODE = 000000 01) Ri + counter#2 → Rk
        02h (OP-CODE = 000000 10) Ri + Rj → Rk
        03h (OP-CODE = 000000 11) counter#1 + counter#2 → Rk

 04h to 07h (OP-CODE = 000001 XX) Reserved for subtraction instructions

        08h (OP-CODE = 000010 00) Ri x counter#1 → Rk, overflow → Rk+1
        09h (OP-CODE = 000010 01) Ri x counter#2 → Rk, overflow → Rk+1
        0Ah (OP-CODE = 000010 10) Ri x Rj → Rk, overflow → Rk+1
        0Bh (OP-CODE = 000010 11) counter#1 x counter#2 →Rk, overflow→Rk+1

 0Ch to 0Fh (OP-CODE = 000011 XX) Reserved for division instructions

        10h (OP-CODE = 000100 00) Compare Ri with counter#1 → Rk
        11h (OP-CODE = 000100 01) Compare Ri with counter#2 → Rk
        12h (OP-CODE = 000100 10) Compare Ri with Rj → Rk
        13h (OP-CODE = 000100 11) Compare Counters

        14h (OP-CODE = 000101 00) Ri AND counter#1 → Rk
        15h (OP-CODE = 000101 01) Ri AND counter#2 → Rk
        16h (OP-CODE = 000101 10) Ri AND Rj → Rk
        17h (OP-CODE = 000101 11) AND counters → Rk

        18h (OP-CODE = 000110 00) Ri OR counter#1 → Rk
        19h (OP-CODE = 000110 01) Ri OR counter#2 → Rk
        1Ah (OP-CODE = 000110 10) Ri OR Rj → Rk
        1Bh (OP-CODE = 000110 11) OR counters→ Rk

 1Ch to 1Fh (OP-CODE = 000111 XX) Reserved for future instructions
 20h to FFh                       Reserved for future instructions
```

GRADE PERCENTAGES and DUE DATES:

- Demonstrate **New PAD-234 Circuit Trainer** [primarily for **5 volt** TTL chips]**:**
- Demonstrate **Old Circuit Trainer** [primarily for **5 volt** TTL chips]**:**
- Demonstrate **RadioShack Circuit Trainer** [primarily for **~3 volt** CMOS chips]**:**
- Demonstrate **NanoLC PLC simulations:**
- Demonstrate **NanoLC PLC**:
- Demonstrate **IEC-61131 PLC simulations**:
- Demonstrate **IEC-61131 PLC:**
- Demonstrate **Logisim simulations (MSI):** (**40%) Monday 4/28/14 AT 3:30PM ON YOUTUBE and presented to class**
- Demonstrate **FPGA simulation AND prototyping board (MSI):**
- Demonstrate **8051/80251 microcontroller simulations:** (**20%) Monday 4/28/14 AT 3:30PM ON SAME YOUTUBE video. and presented to class**
- Demonstrate **8051/80251 microcontroller boards:**
- Demonstrate **Arduino Controller**
- Demonstrate **Basic-Stamp microcontroller interface:**
- Demonstrate **Direct PC-port interface:**
- Demonstrate **Remote mobile-device interface:**
- Demonstrate **LabView:**
- Demonstrate **Isolated high-voltage "bench-test"** (i.e., sensitive <5 Volt electronics disconnected)
- Demonstrate **Other**:
- Written Report: (**20%) Monday 4/28/14 AT 3:30PM**
- **POSTER of your design (PRINTED THE WEEK BEFORE) and presented to the class (20%) Monday 4/28/14 AT 3:30PM**


Program Counter
EMBEDDED CODE STACK
Instruction Register
OP CODE
Ri Rj Rk
M U X
Register 0
Register 1
Register 2
Register 3
Register 4
Register 5
Register 6
Register 7
D E M U X
1 HZ CLOCK
or a push-button
16-BIT UP-COUNTER WHICH RESETS TO 00h AFTER FFh
16-BIT DOWN-COUNTER WHICH SETS TO FFh AFTER 00h
MULTIPLEXER
MULTIPLEXER
Control Logic
A Buffer
B Buffer
16-bit COMPARATOR X<Y X=Y X>Y
16-bit MULTIPLIER
16-bit ADDER
16-bit AND
16-bit OR
STATUS REGISTER (1-bit flags) < = > CarryOut Overflow
OVERFLOW
ACCUMULATOR
3 HEX displays
2 HEX displays
2 HEX displays
L E D
L E D
L E D
L E D
L E D
2 HEX displays
2 HEX displays
2 HEX displays
2 HEX displays

RULES FOR ALL LABS

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 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). **INCLUDE PHOTO'S OF ALL CIRCUITS (AND TEST-SET-UP'S) BUILT**

REPORTS must include:

- 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**)
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). 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 with the Teaching Assistant. Try to stick with same person for this role.
TEACHING ASSISTANT will guarantee all working hardware and software when needed and will facilitate acquisition of needed parts.

DEMONSTRATIONS Alternate team members demonstrate lab to me (partner must be present). We do this for previous week's assignment while you begin the next assignment

GRADING **for both demonstrations and reports, a 92 is for everything done very well and professional; to get up to 100, enhance things in creative way**