**FINAL EXAM, 10:00–12:00 noon**

---

### Day/Event | Tuesday | Lab topic | Thursday
---|---|---|---
1/9 | Syllabus/Binary counting<sup>a</sup> | 1/10 Intro. to Unix/ Create HTML page | 1/11 Representation of Text/Images, intro. of Disc. Board
1/16 | Assign HW 1 over-did representations, badly covered binary | 1/17 Binary | 1/18 Re-did binary, bin. add, covered 2's complement conversion
1/23 | Covered 2's comp., Excess-C notation, and 8-bit FP → base-10 | 1/24 Binary | 1/25 base-10 → 8-bit FP, 2's comp., Excess-C review, Ch 1.4-1.7 done.
1/30 | Bool logic, truth tables, AND, OR, XOR, NOT, circuits w/ AND, OR & NOT | 1/31 Binary | 2/1 – HW 1 due; Assign HW 2 2- / 3-bit adders, circuits review, flip-flops, data storage, memory.
2/6 | Mass Storage, hard drive, tape drive, on-line / off-line, review for test. | 2/7 Digital Circuits | 2/8 – MIDTERM 1 XXXXXXXX
2/20 | HW 2 due; Assign HW 3 ML, RTN, how to do a program (setup, work, finish). Started 3 × 4. | 2/21 Digital Circuits | 2/22 Program 3 × 4, example of jump<sup>b</sup>; Did prog. showed on Super Machine (SM).
2/27 | RTN on 3 × 4, rotation, masking, communication between objects. | 2/28 CPU Simulation | 3/1 XXXXXXXXX - Snow. Class Cancelled.
3/6 | | 3/7 CPU Simulation | 3/8 – HW 3 due; Assign HW 4
3/20 | XXXXXXXX | XXXXXXXX | XXXXXXXX
4/3 | HW 4 due; Assign HW 5 | 4/4 | 4/5
4/10 | | 4/11 | 4/12
4/17 | | 4/18 | 4/19 – HW 5 due
4/24 | Dead Week | 4/25 – Dead Week | 4/26 – Dead Week
5/4 | | | | 

---

<sup>a</sup> Chose 4 students, who lined up in front of class. Each one represented one bit (crouching = 0, standing = 1). Trying to make the smallest possible pattern beyond the previous pattern (starting with all crouching, or 0000), we made a number of patterns. After doing this for some time, I assigned “values” to each of the students names by examining the times when only one person was standing. When this made sense, I called out a number, and the students would tell which people needed to stand to make the corresponding pattern.

<sup>b</sup> Decision process using a pepsi machine: Steps were (1) Walk to Pepsi machine, (2) Kick machine, (3) Put a quarter in the machine, (4) Kick the machine. Showed we want to do step 1, but want to skip step 2 (hurt toe). But, we want to repeat step 3 five times (drew arrows for each draw/skip type operation), no more, no less (otherwise, waste money). Related this back to a jump operation: If less than 5, go back to step 3. Otherwise, skip to next instruction (B212, B00A ← always jump back). If Pepsi got stuck in machine, then we would want to kick the machine. Otherwise, not (another decision). In this fashion, I related the for loop and if stmt to them.