ECE 120 Midterm 1
HKN Review Session

Exam Time: Tuesday, September 20 (7:00-8:30pm)
Logistics

Exam: Tuesday, September 20, 7pm-8:30pm

Conflict exam: 5pm-6:30pm

Location: Check Compass for room assignment

UA Review Session: Sunday, September 18, 2pm-4pm
Abstraction/Levels of Transformation

- Abstraction - the means to simply events without going into heavy specifics, reducing information to the essentials

- Levels of Transformation
  - Problem Statement
  - Algorithm
  - Program
  - Instruction Set Architecture
  - Microarchitecture
  - Logic gates
  - Devices
Binary Types

- **Unsigned**
  - Can only represent nonnegative integers
  - $K =$ number of bits
  - Total unique representations $\Rightarrow 2^k$
  - Range $\Rightarrow 0$ to $(2^k-1)$

- **Signed - Magnitude**
  - First bit determines if positive or negative $\Rightarrow 1 =$ negative, $0 =$ positive
  - Rest of bits determine magnitude
  - Range $\Rightarrow -(2^{(k-1)}-1)$ to $(2^{(k-1)}-1)$

- **1’s Complement**
  - Positive numbers represented just like unsigned
  - Negative numbers represented by flipping all the bits
  - Range $\Rightarrow -(2^{(k-1)}-1)$ to $(2^{(k-1)}-1)$
Binary Types

- **2’s complement**
  - Positive numbers lead with “0”, negative numbers lead with “1”
  - \( K \) bits \( \rightarrow \) can represent \( 2^k \) total numbers, half being positive and half being negative
  - Can represent positive numbers from range \( -(2^{(k-1)}) \) to \( (2^{(k-1)}-1) \)
  - Procedure for 2’s complement (PFA):
    - PLACEHOLDER BITS
    - FLIP ALL BITS
    - ADD 1
Hexadecimal *

- Base 16, Uses 0-9 and A-F
- Takes groups of 4 bits and represents them as symbols
  - Ex: 0011 1101 0110 1110 → 3 D 6 E
- To go from hex to binary, write out each hex value into 4 bit binary
  - Ex: 4E7F → 0100 1110 0111 1111
- Shortens binary representation by a factor of 4
Overflow in Operations

● 2 primary operations: addition and subtraction

● Checking for Overflow
  ○ Unsigned operations
    ■ There is a nonzero carry bit (bit carries out of bit range)
  ○ 2’s Complement operations
    ■ Result has wrong sign if
      ● 2 positive numbers sum to negative number
      ● 2 negative numbers sum to positive number
      ● NOTE: in 2’s complement, a positive and negative number added never results in overflow
    ■ Quick Check- For MSB, does carry in bit = carry out bit (i.e. $C_n = C_{n-1}$)?
      ● If not, overflow has occurred
Floating Point

- Use IEEE 754 standard (32 total bits)
  - 1 sign bit
  - 8 exponent bits
  - 23 mantissa bits
- Increased precision
  \[ \Rightarrow \text{decreased range} \]

\[
\text{IEEE 754 Floating Point Standard}
\]

\[
\begin{array}{ccc}
\text{s} & \text{e=exponent} & \text{m=mantissa} \\
1 \text{ bit} & 8 \text{ bits} & 23 \text{ bits}
\end{array}
\]

\[
\text{number} = (-1)^s \times (1.m) \times 2^{e-127}
\]
Floating Point (cont.)

- Special Cases
  - Denormalized representation
    - Exponent = 0
    - Mantissa takes any value
    - Formula: \((-1)^s \times 0.\text{Mantissa} \times 2^{-126}\)
  - Exponent is all 1s
    - Mantissa = 0
      - \((-1)^s \times \text{infinity}\)
    - Mantissa not equal to 0
      - NaN
Boolean Operators *

- NOT
- AND, NAND
- XOR, XNOR
  - $A \oplus B = A \overline{B} + \overline{A} B$
- OR, NOR
- Note:
  - Order of precedence:
    - (), NOT, AND, OR
  - AND, NOT, and OR are logically complete
C Programming

- Basic Characteristics
  - High level/independent (of ISA), procedural, expressive

- Variables
  - Int, double, float, char
  - Note that result is truncated during integer division!

- Operators
  - Order of precedence: *, /, % and then +, -

- C Constructs
  - Sequential
  - Conditional (if, else if)
  - Iterative (for, while loops)
Flow Chart Components

Start/Stop

Input/Output

General Work

Decision

T

F
Cheat Sheet: Recommendations

- Common powers of 2
- 2’s Complement
  - Converting to/from (PFA)
  - Representable range with K bits
- Floating Point
  - Formula for general case
  - Special cases
- Overflow Conditions (both unsigned and 2’s complement)
- Harder boolean operators
  - XNOR, XOR, NAND, NOR
- Basic C syntax
General Advice

● Use your Cheat Sheet!
  ○ Don’t memorize
● Read the directions carefully!!!!
● Don’t be afraid to ask questions
● Relax and trust what you’ve learned :)