CPS213 Summary Sheet
Printed 2012-10-08

Chapter 1 – Digital Systems & Binary

- General technique for converting from decimal to Base R
  - For decimal integers, divide the decimal number by 'R' repeatedly – the remainder get's turned into the base R number by writing it from the bottom to the top
  - For decimal part of the number being converted, multiply by 'R' repeatedly, then write the resulting number **without the decimal part** from top to bottom
- Binary to Octal/Hex
  - Octal: put bits in groups of 3, convert the group
  - Hex: put bits in groups of 4, convert the group
- 1's complement (binary): flip all the bits
- 2's complement (binary): flip all the bits & add 1
- Signed (2's, 1's, magnitudes) binary numbers

Chapter 2 – Boolean Algebra & Logic Gates

- Boolean algebra
- DeMorgan: $\overline{(xyz)} == (\overline{x} + \overline{y} + \overline{z})$; $\overline{(x + y + z)} == (\overline{x} \cdot \overline{y} \cdot \overline{z})$
- Boolean Algebra trick: multiply term by (x' + x) (where x is not part of the term)
- Error correction – XOR/Parity "Hamming Code"
  - Data Bits: 2-4 -> 3 parity, 5-11 -> 4 parity, 12-26 -> 5 parity
  - Data bits are entered, then parity is calculated and placed
  - Parity calculations are made, if C != 0, there is an error

Chapter 3 – Gate-level Minimization

- Kmap Groupings
  - The group must contain an integral of 2
  - Look for the largest groupings:
    - 4 corners
    - rows
    - columns
    - Top/Bottom of columns, left/right of rows
- Minterms -> Sum of Product (NAND) -> 1's on kmap
- Maxterms -> Product of Sums (NOR) -> 0's on kmap
  - Then after finding the function, invert each literal
- 2-level NAND & NOR
  - NAND: SOP form eg $F(A,B,C,D)=\sum(0,2,12\,13)$ => put 1's in pos 0,2,12,13
  - NOR: POS form eg $F(A,B,C,D)=\prod(0,2,12\,13)$ => put 0's in pos 0,2,12,13
- XOR (odd function) $x \oplus y = \overline{x}y + x\overline{y}$
  - $x \oplus 0 = x$
    $x \oplus 1 = \overline{x}$
    $x \oplus x = 0$
    $x \oplus \overline{x} = 1$

- XNOR (even function) $\overline{x \oplus y} = xy + \overline{x}\,\overline{y}$

Chapter 4 – Combinational Logic

- Design of Comb. Ccts:
  1. state the problem
  2. determine inputs/outputs, assign variables to them
  3. make a truth table (typically)
  4. write boolean function, then simplify
  5. draw logic diagram (circuit)

- Half Adder
  - Has 2 inputs, 2 outputs: Carry Out and Sum
- Full Adder
  - Has 3 inputs (including Carry In)
- Binary Adder (tbd)
- Carry Look-ahead (tbd)
- Tri-state buffer: on/off switch, like a transister (High Impedance when off)
- Multiplexers
  - Go from SOP to MUX using Truth Table
  - For 3-inputs, x,y are selectors, z is used to derive the output – put F rows into groups of 2 (or however many selectors there are).
- Encoders, Decoders

CPS213 Summary Sheet
Printed 2012-10-08

<u>Chapter 5</u> – Synchronouns Sequential Logic

- Latch (see NAND and NOR implementation) – building block for Flip Flop
  - RS Latch (NAND implementation)

S	**R**	**<u>Q</u>**
0	0	Undetermined
0	1	0
1	0	1
1	1	No change

- Flip Flop
  - D FF
    - "data" - Just stores data bit at output, no other special states
  - JK FF
    - J -> Set, K -> Reset

J	**K**	**<u>Q</u>**
0	0	No Change
0	1	0
1	0	1
1	1	Qbar (toggle)

  - T FF
    - "toggle" - JK FF with inputs tied together