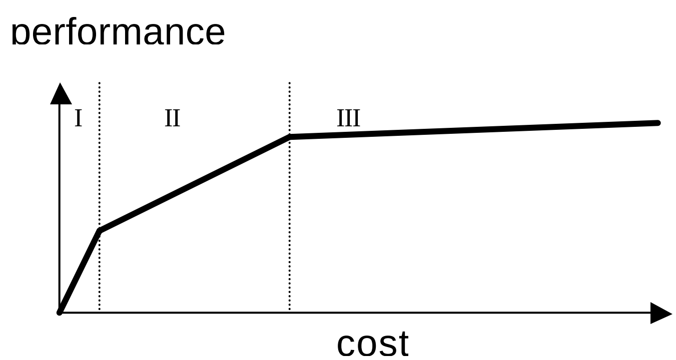
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS150 J. Wawrzynek
Spring 2000 E. Caspi

Homework #1 – Solution

1. Recall, in a performance-cost tradeoff, we typically seek to maximize performance and/or minimize cost.

   - Region I: The system is optimized for low cost. The high slope indicates that reducing cost incrementally further requires a big loss in performance
   - Region II: The system is balanced. Cost and performance trade off well.
   - Region III: The system is optimized for high performance. The low slope indicates that increasing performance incrementally further requires a large additional cost.

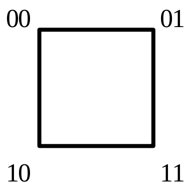


2. Describe an object hierarchically. This is fairly open ended. Be creative.

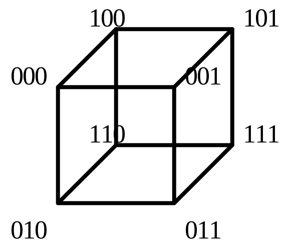
3. There are 12 3-bit gray codes.

   Recall that an n-bit gray-code is a cyclic sequencing of all n-bit numbers such that successive numbers differ by exactly one bit. You should be able to list and count all such sequences for n=3.

   There is a nice geometric interpretation for gray codes. For a given bit width n, the set of all n-bit numbers corresponds to the corners of an n-dimensional hypercube (e.g. 2-bit numbers are the corners of a 2D square; 3-bit numbers are the corners of a 3D cube). Each axis of the cube represents one bit, with coordinates valued 0 or 1.



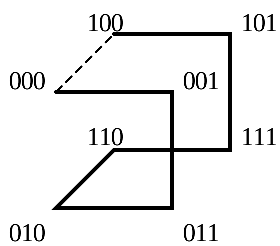
2-bit numbers



3-bit numbers

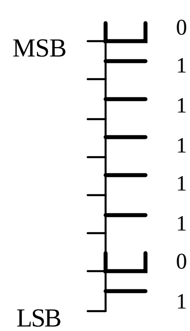
An n-bit gray code is a cyclic sequence of all n-bit numbers such that successive numbers differ in exactly 1 bit. Geometrically, such a sequence is a path along the legs of the hypercube, visiting every corner, and ultimately returning to the starting point (such a path is called a "Hamilton cycle"). The number of unique n-bit gray codes is the number of unique paths around an n-bit hypercube. Keep in mind that rotations, mirror images, and direction reversal may transform a path into a multitude of unique paths.

There are twelve (12) unique paths which cover the 3D cube. The basic topology is the same for all paths. The twelve-fold multiplicity arises from rotating about the 000-111 axis (3x), mirroring about the vertical 000-101 plane (2x), and reversing the path direction (2x).



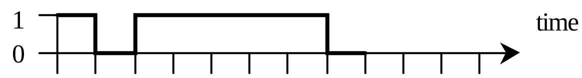
4. 125 decimal is "1111101" in binary, or padded to 8 bits: "01111101".
   25 decimal is "11001" in binary, or padded to 8 bits: "00011001".
   In all examples, we assume that a wire carries a value "1" when not in use.

   a. The "bit-parallel" waveform transmits one bit on each of 8 wires.



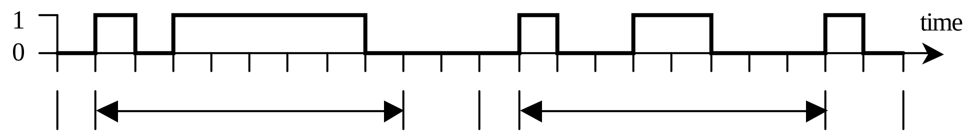
Bit-parallel transmission of 125

b. The “bit-serial” waveform transmits each bit in sequence on 1 wire, LSB first, MSB last, in 1 second intervals.



Bit-serial transmission of 125

c. With parity and start/stop bits, each byte transmission look like: {start (0), LSB … MSB, parity, stop (0)}.



5. Convert the following numbers from base ten to two’s complement, one’s complement and sign magnitude (use 16 bits for each number). -1011, -32752, 1500

a) Decimal -1011
- Positive decimal 1011 = binary “1111110011”
- To negate into one’s complement, pad with a “0” sign bit, then invert all bits: binary “10000001100”.
- Two’s complement = one’s complement +1: binary “10000001101”.
- Sign magnitude = sign bit adjoined to positive value: binary “11111110011”.

b) Decimal –32752
- Positive decimal 32752 = binary “111111111101111”
- To negate into one’s complement, pad with a “0” sign bit, then invert all bits: binary “1000000000010000”.
- Two’s complement = one’s complement +1: binary “1000000000010001”.
- Sign magnitude = sign bit adjoined to positive value: binary “1111111111101111”.

c) Decimal 1500
- Positive decimal 1500 = binary “100011001010”.
- For one’s complement representation of a positive number, simply pad with a “0” sign bit: binary “0100011001010”.
- For two’s complement representation of a positive number, do same: binary “0100011001010”.
- For sign-magnitude representation of a positive number, do same: binary “0100011001010”.