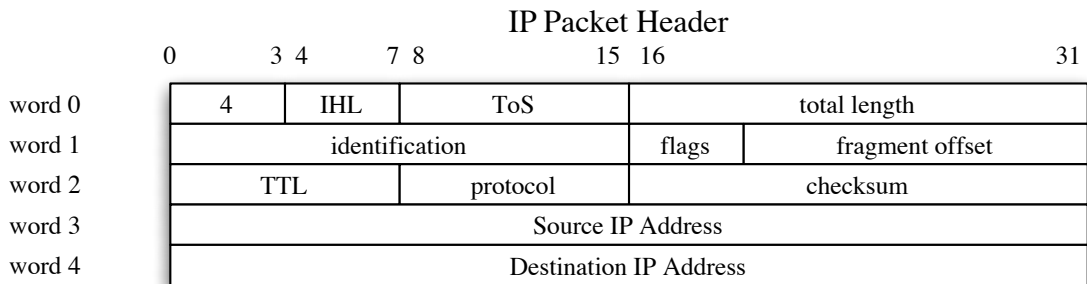


CmpSci 201

Homework 3

1. Given $A = 11011010$ and $B = 11100010$,
 - (a) Compute $A \text{ AND } B$
 - (b) Compute $A \text{ OR } B$
 - (c) Compute $A \text{ XOR } B$
2. Given $A = 10010000$ and $B = 10100111$,
 - (a) Compute $A \text{ AND } B$
 - (b) Compute $A \text{ OR } B$
 - (c) Compute $A \text{ XOR } B$
3. Given $A = 11100101011$ and $B = 10100111100$,
 - (a) Compute $A \text{ AND } B$
 - (b) Compute $A \text{ OR } B$
 - (c) Compute $A \text{ XOR } B$
4. Assuming 16-bit registers, and $A = 1010111000101010$
 - (a) Compute $A \ll 5$ (logical left)
 - (b) Compute $A \gg 6$ (logical right)
 - (c) Compute $A \ggg 6$ (arithmetic right)
 - (d) Compute $(A \ll 2) \gg 5$
 - (e) Compute $(A \gg 5) \ll 2$
5. Masking and shifting 32-bit values, where bit 0 is LSB and bit 31 is the MSB. Assume X is a 32-bit value,
 - (a) Using AND, how would you mask out all but the low 4 bits of X ?
 - (b) Using AND and \gg , how would you get the value for bits 7 - 3 of X ?
 - (c) Using OR, how would you set the value of X 's bit 11 to be high?

6. The following problems will refer to this diagram:



This diagram shows the layout of various fields in an IPv4 packet header. The various fields conveniently lie along 4, 8, or 16-bit boundaries. For instance, the IHL field begins at bit 4 of the first word of the header and ends at bit 7 of that same word (therefore IHL is four bits wide). Packet processing is big business and is an application where shifting and masking are used heavily. Assuming that the code has already loaded words 0 - 4 in registers 0 - 4 (that is, word 0 is in R0, word 1 in R1, and so forth),

- (a) Write ARM code to extract the total length from the header and write it into R6
- (b) Write ARM code that extracts the protocol field from the header and writes it into R6
- (c) Write ARM code to set the protocol field to TCP (TCP = 6)
- (d) Remember that the network is big-endian. If the value of the source IP address (word 3) was 0x7F000001, how would the bytes have to be laid out in the memory of a little endian machine?
- (e) You're writing code on an Intel box (which is little-endian), and you need to set the destination IP address. The value in memory (by byte) is: 0x80 0x1F 0x0A 0x0A, write out the bytes as they would need to appear in the packet header.