2008 Paper 6 Question 2

Computer Design

(*a*) What do *RISC* and *CISC* stand for and what are the differences in practice?

[6 marks]

(*b*) Some instruction sets use a register file and others use an operand stack for intermediate storage. How does the code density compare between these two approaches? [4 marks]

(*c*) Early computers used just an accumulator rather than a register file or stack. How does the code density compare between accumulator and stack machines?

[4 marks]

(*d*) Compact RISC instruction sets typically use a fixed 16-bit instruction size. If three operands are to be specified, each of n bits, then $(16 - 3 \times n)$ bits are left for the opcode. If $n = 3$ then we cannot access enough registers and if $n = 4$ we do not have enough opcode bits. In practice we usually need 5 bits for the opcode leaving 11 bits to specify three registers. Using 11 bits to store three operands, how many registers can be specified and how might these three operands be decoded? [6 marks]