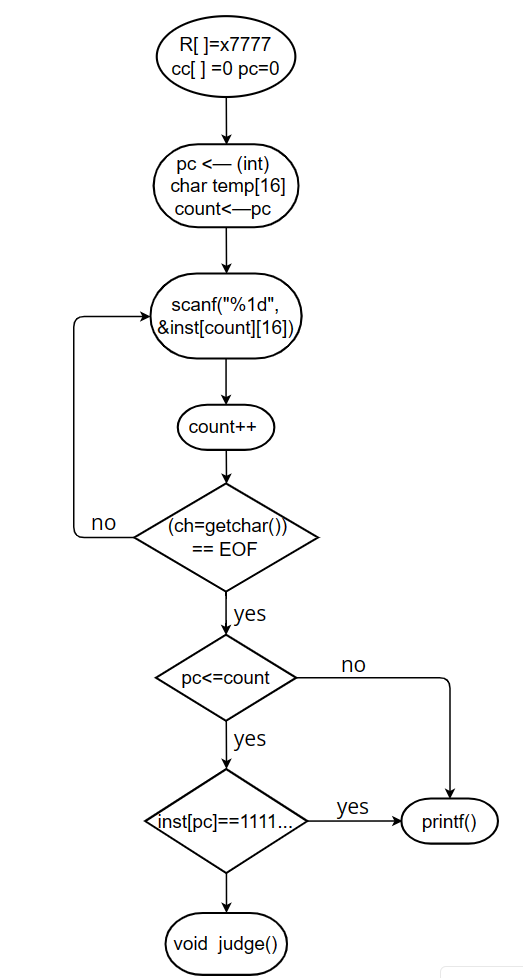
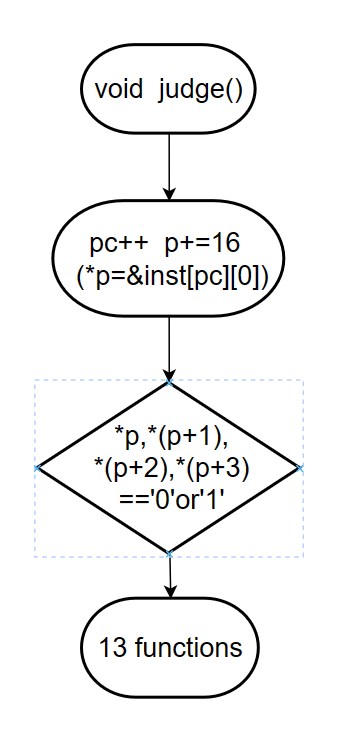
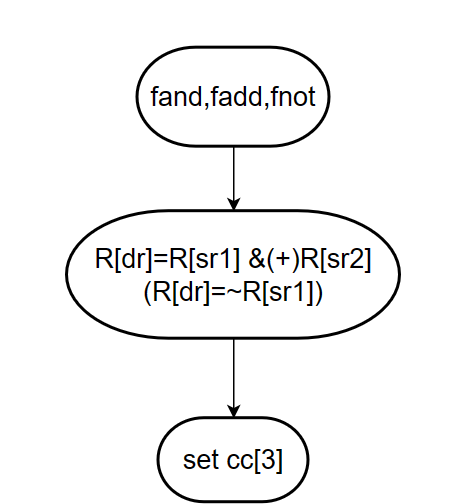
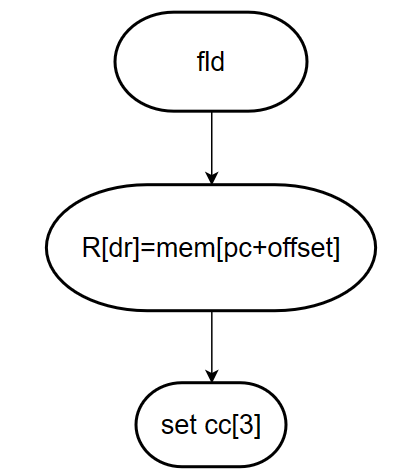
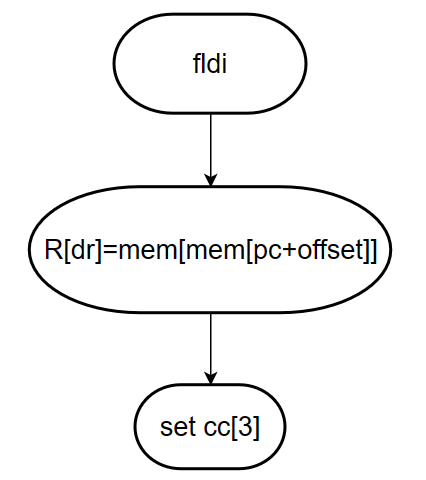
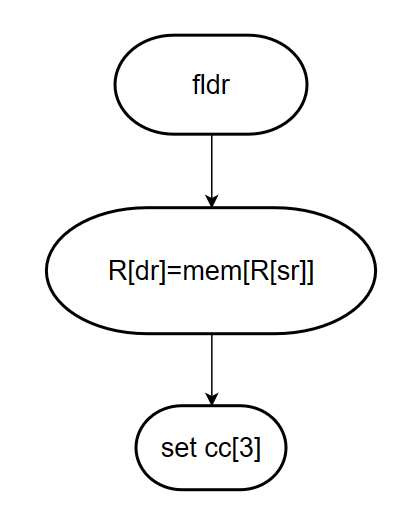
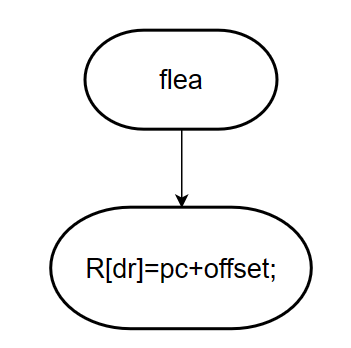
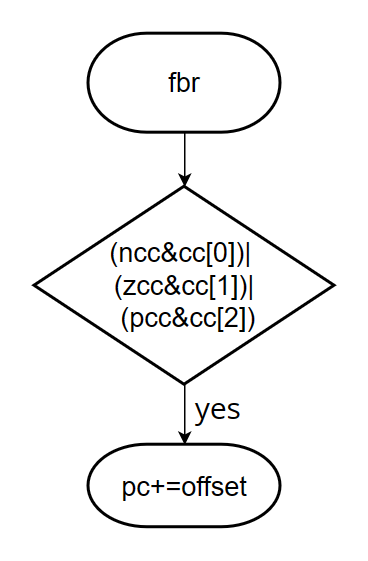
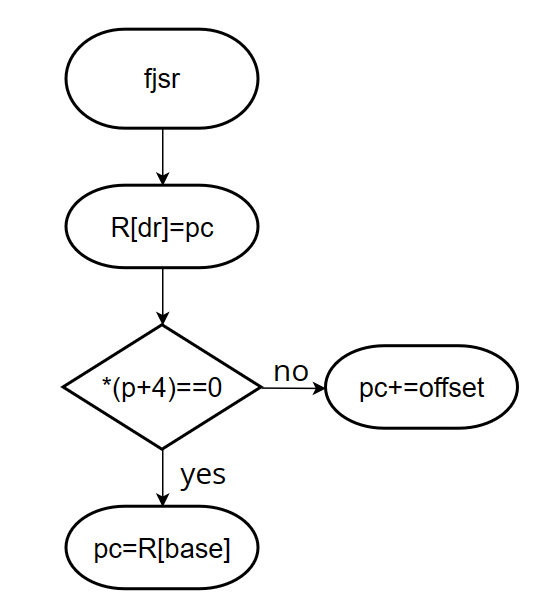
Lab6

1. Algorithm



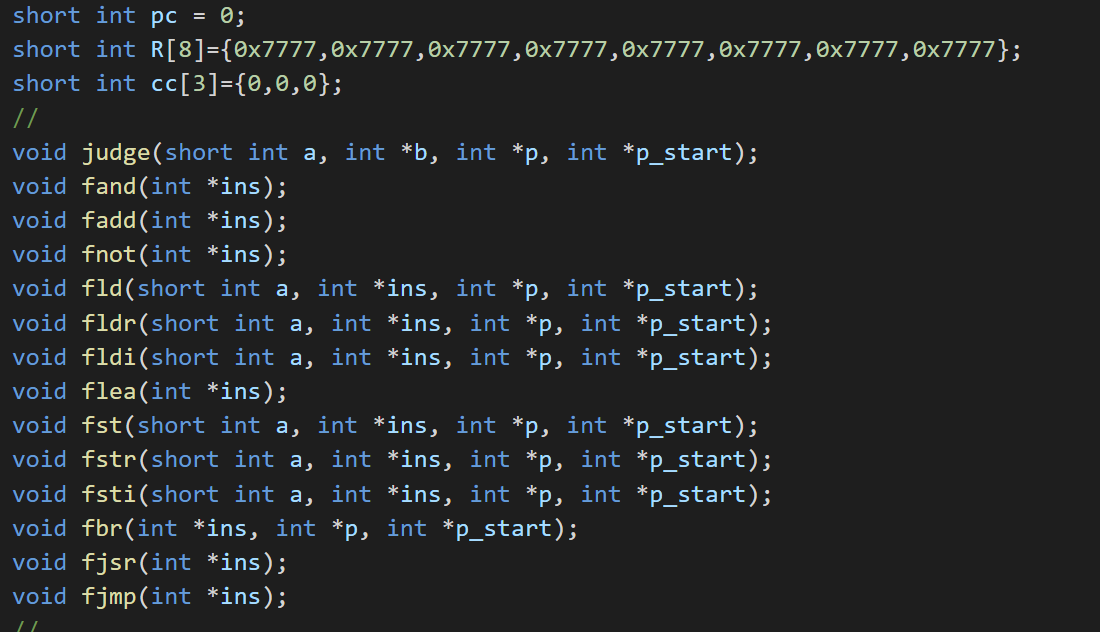




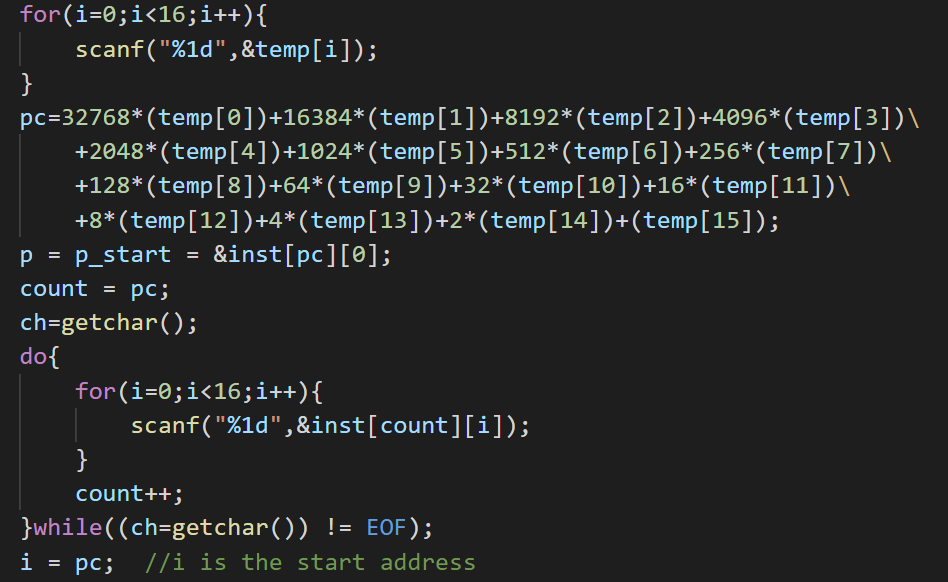


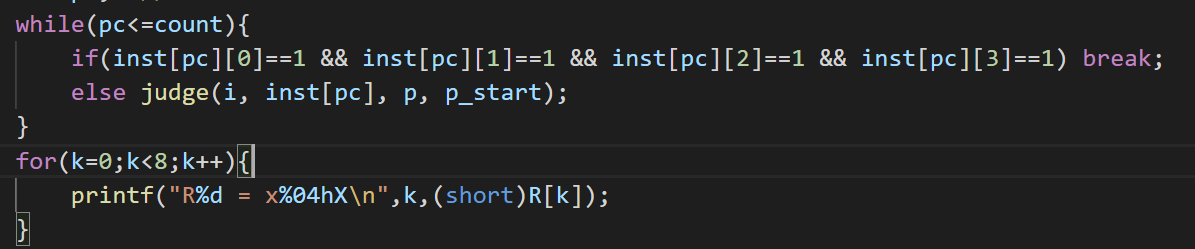
1. Essential parts

(1)Defines a short global variable register array R[8], pc and conditional code cc[3], and declares the judge function and 13 instruction functions.

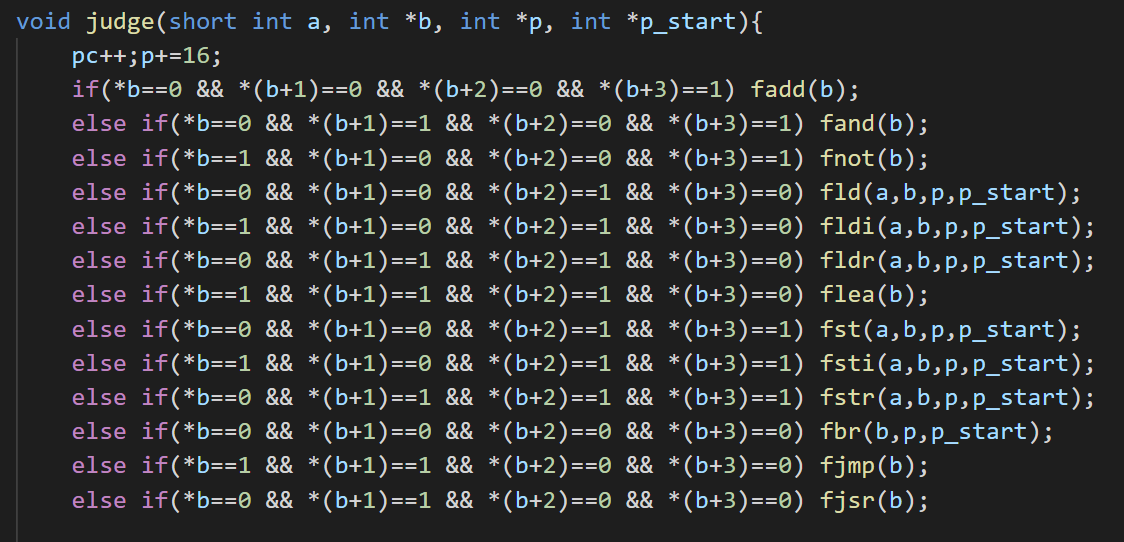


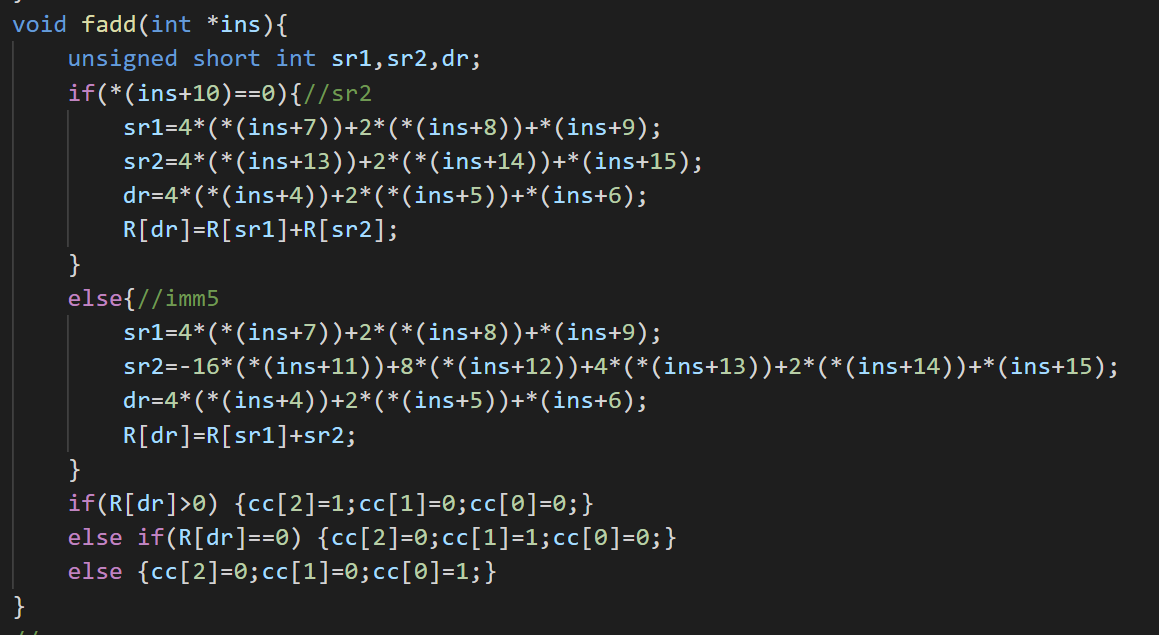
(2)The first line of input is the initial value of the pc, and then each 16-bit instruction is stored in inst[][16]. Halt exits when the first four bits of an instruction are 1111, otherwise enter the judge function.



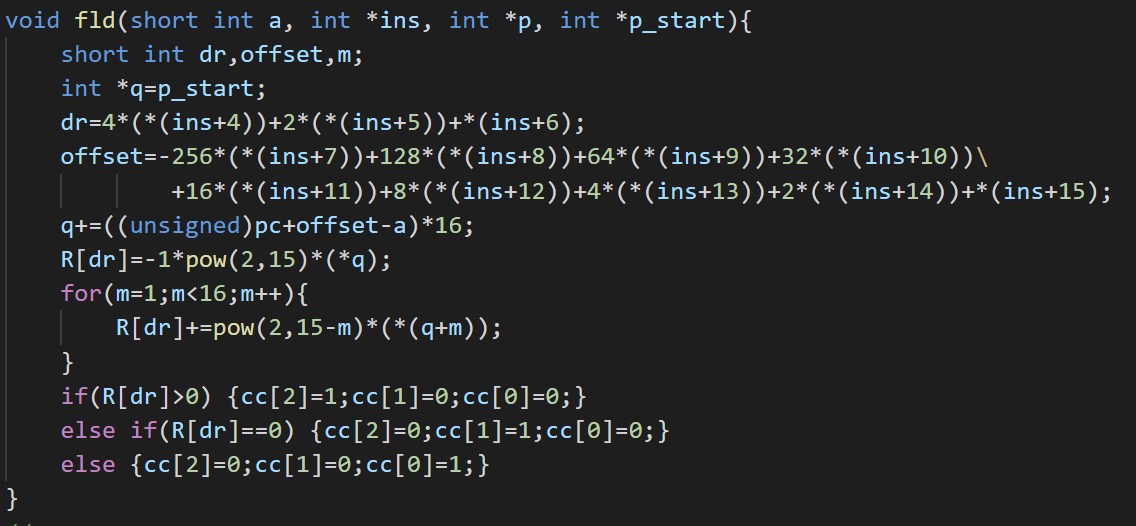


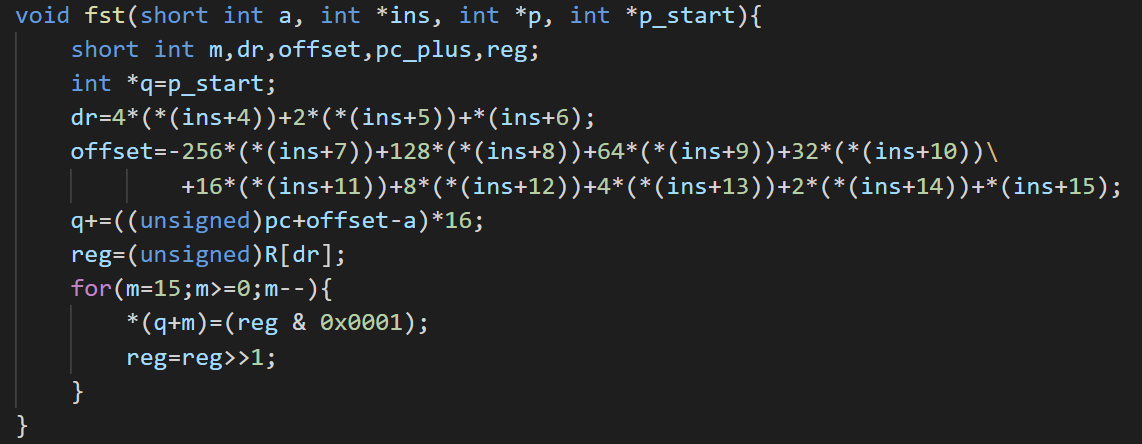
(3)The first step is to let pc ++, and then enter different functions according to the first four digits of the instruction



(4)The fadd, fand, fnot functions are similar in that the decoded value is stored in the R register.

(5)fld, fldr, fldi functions are similar in that they fetch the value from memory and assign it to the register, while fst, fstr, fsti stores the value of the register into memory. The L-type instruction also changes cc.





(6)Jump instruction mainly deals with pc changes, fbr function determines whether to jump according to cc, and fjsr function saves the return address in R[7].

