```
BEGIN

  fsm:
  PROCESS(clk,rst)
  BEGIN
   IF rst='1' THEN
    presente<=inicial;
   ELSIF clk='1' AND clk'event THEN
    CASE presente IS
    WHEN inicial =>
       presente<=busqueda;
    WHEN busqueda=>
       IF dat_in(2 DOWNTO 0)="111" THEN presente<=busqueda;
       ELSE presente<=ejec;
       END IF;
    WHEN ejec =>
       CASE rdat_in(2 DOWNTO 0) IS
       WHEN "000" =>
          presente<=ldaxx;
       WHEN "001" =>
          presente<=ldxxa;
       WHEN "010" =>
          presente<=anda;
       WHEN "011" =>
          presente<=adda;
       WHEN "100" =>
          presente<=suba;
       WHEN OTHERS =>      -- jz,jmp
          presente<=busqueda;
       END CASE;
    WHEN OTHERS=>
       presente<=busqueda;
    END CASE;
   END IF;
  END PROCESS fsm;

  salida:
  PROCESS(presente,pc,a,rdat_in,dat_in)
  BEGIN
    CASE presente IS
    WHEN inicial =>
      carga_pc<="00000000";      -- PC a 0
      carga_a<="00000000";       -- Acumulador a 0
      dir<=pc;                   -- Dirección a 0;
      rwaux<='1';                -- Lectura
    WHEN busqueda=>
      carga_pc<=pc+"00000001";   -- Incrementa PC
      carga_a<=a;                -- Deja A como estaba
      dir<=pc;
      rwaux<='1';
    WHEN ejec =>
      CASE rdat_in(2 DOWNTO 0) IS
      WHEN "101" =>  -- jz
         dir<=pc;
         IF a="00000000" THEN carga_pc<=dat_in;  -- Salta si A=0
         ELSE carga_pc<=pc+"00000001";
         END IF;
         rwaux<='1';
         carga_a<=a;
      WHEN "110" =>  --jmp
         dir<=pc;
         carga_pc<=dat_in;      -- Salta siempre
```