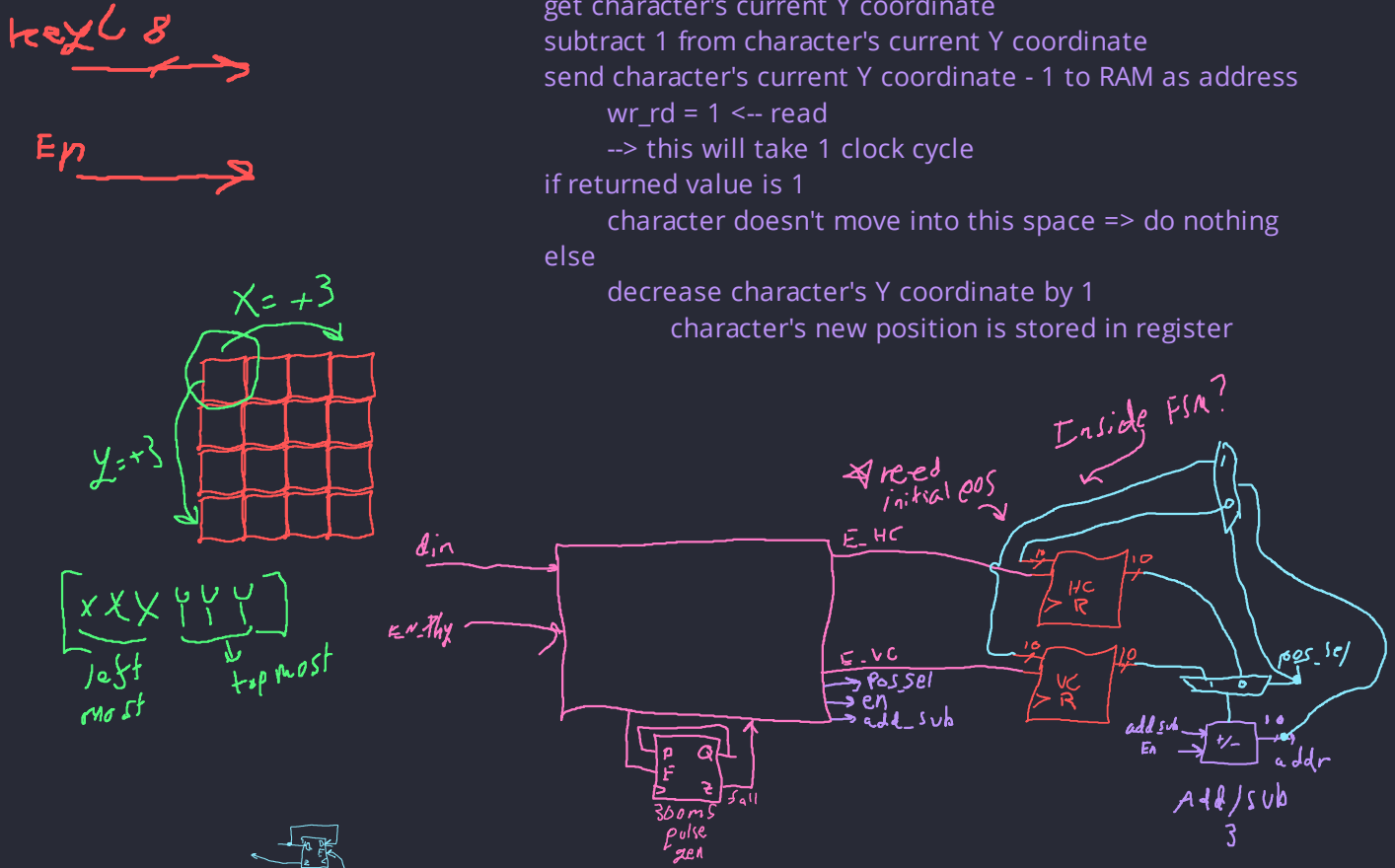
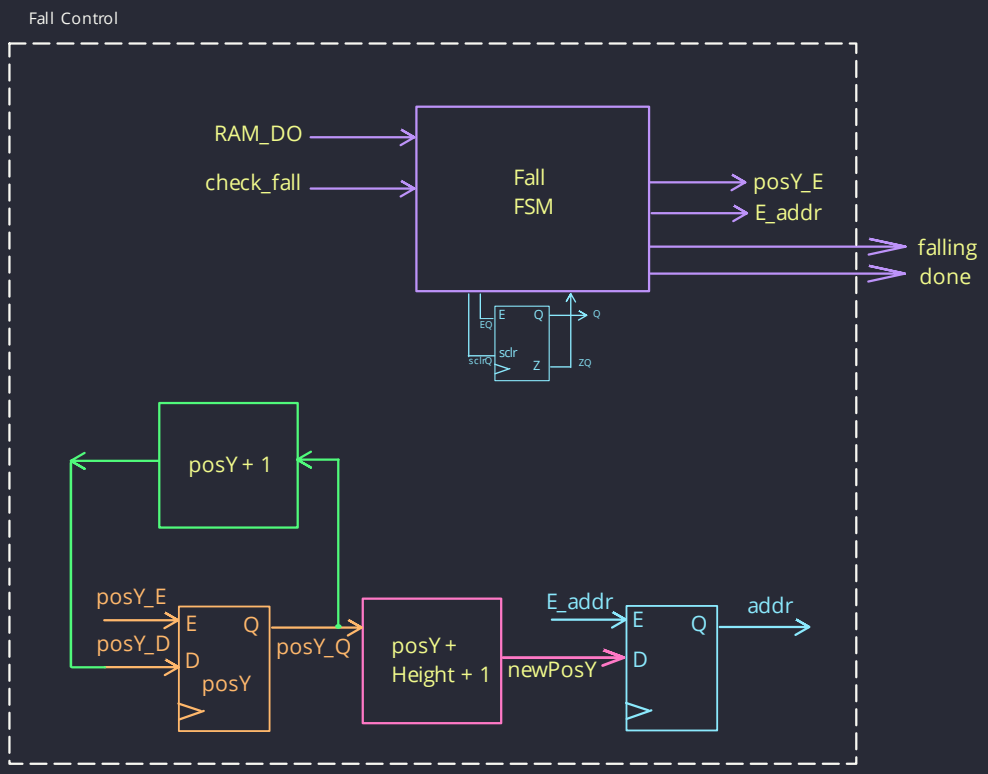
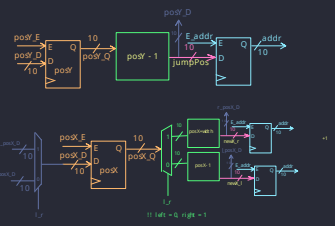
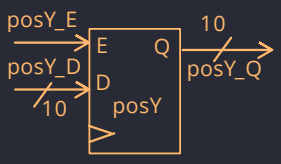
if dout == space
    get character's current Y coordinate
    subtract 1 from character's current Y coordinate
    send character's current Y coordinate - 1 to RAM as address
        wr_rd = 1 <-- read
        --> this will take 1 clock cycle
    if returned value is 1
        character doesn't move into this space => do nothing
    else
        decrease character's Y coordinate by 1
            character's new position is stored in register

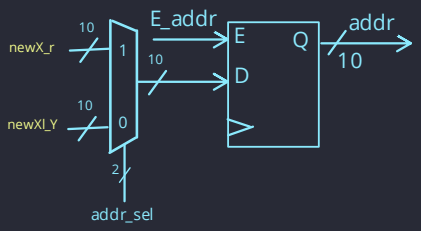


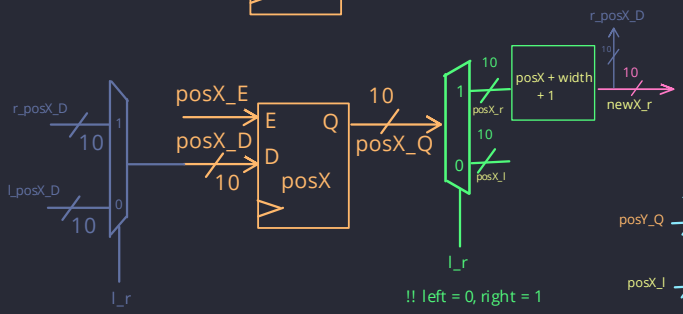


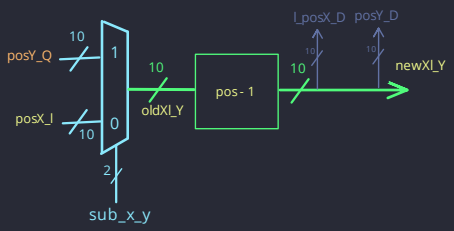

8
din
RAM_DO
ps2_done
fall_done
Physics Control
FSM
check_fall
E_addr
posY_E
posX_E

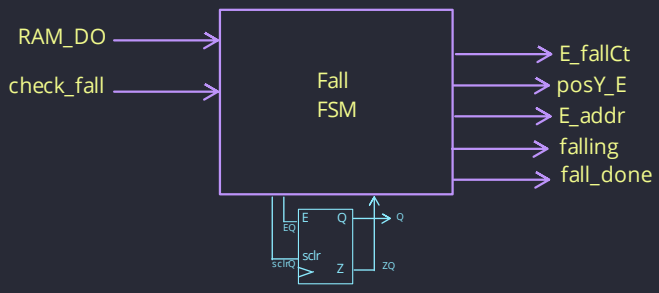


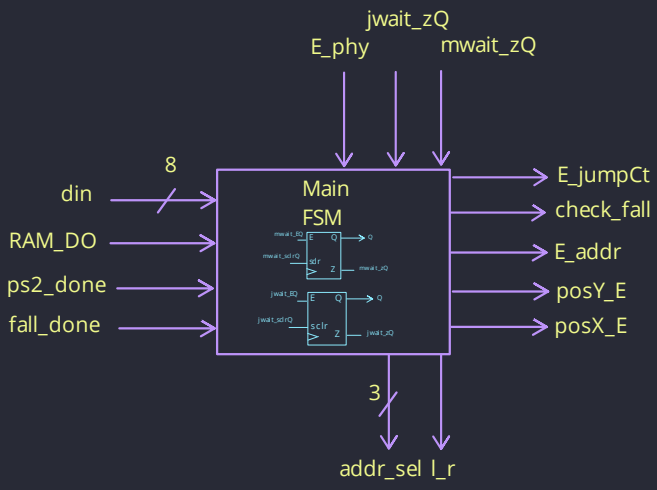

posY_E
posY_D
10
E
Q
D
posY
10
posY_Q

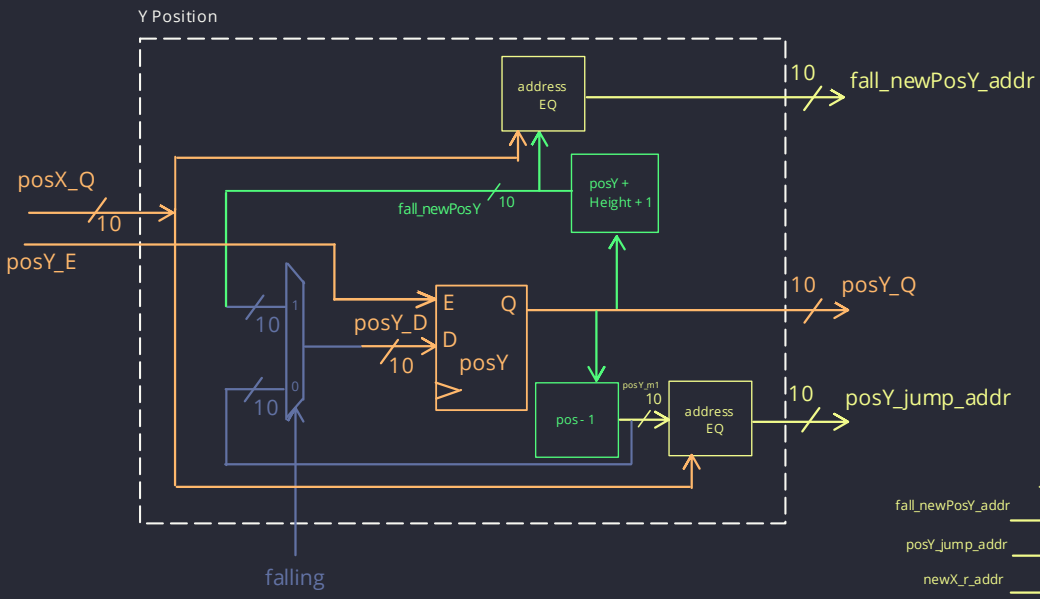

newX_r
newXl_Y
10
1
0
2
E_addr
addr
addr_sel

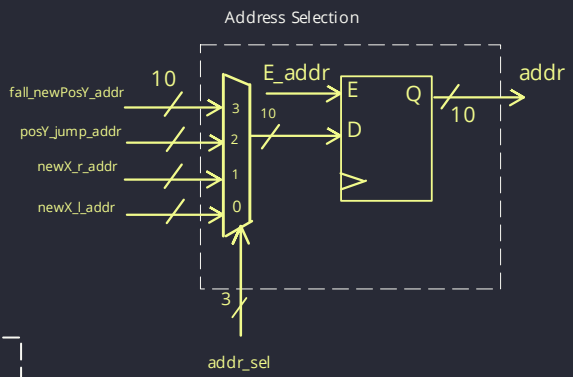

r_posX_D
l_posX_D
posX_E
posX_D
posX
posX_Q
posX + width
+ 1
newX_r
l_r
!! left = 0, right = 1

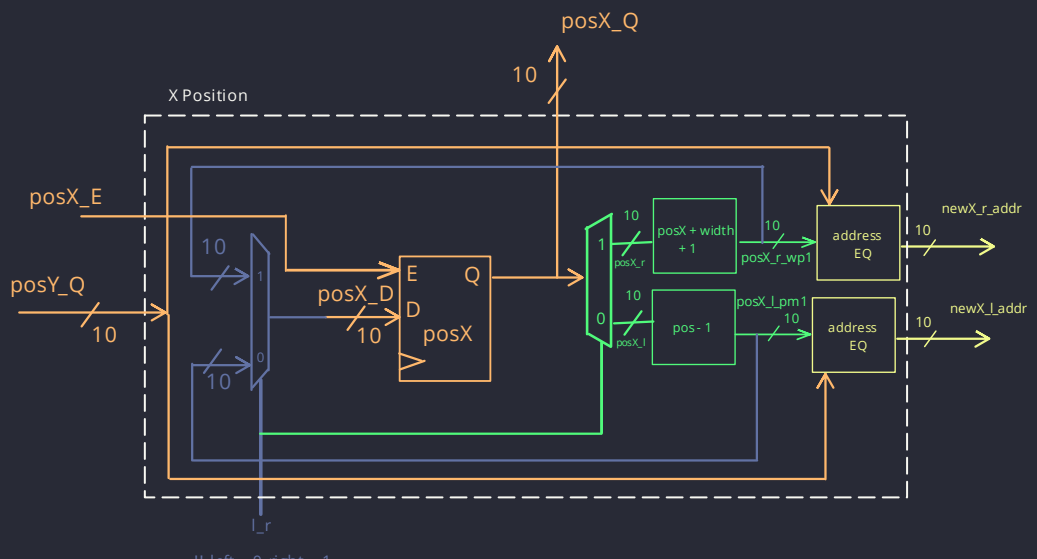

posY_Q
posX_l
oldXl_Y
pos- 1
l_posX_D
posY_D
newXl_Y
sub_x_y











address EQ definitions: { address = VC * 640 + HC }

falling: address = fall_newPosY * 640 + posX_Q
jumping: address = posY_m1 * 640 + posX_Q
moving right: address = posY_Q * 640 + newX_r
moving left: for moving left): address = posY_Q * 640 + newXl_Y