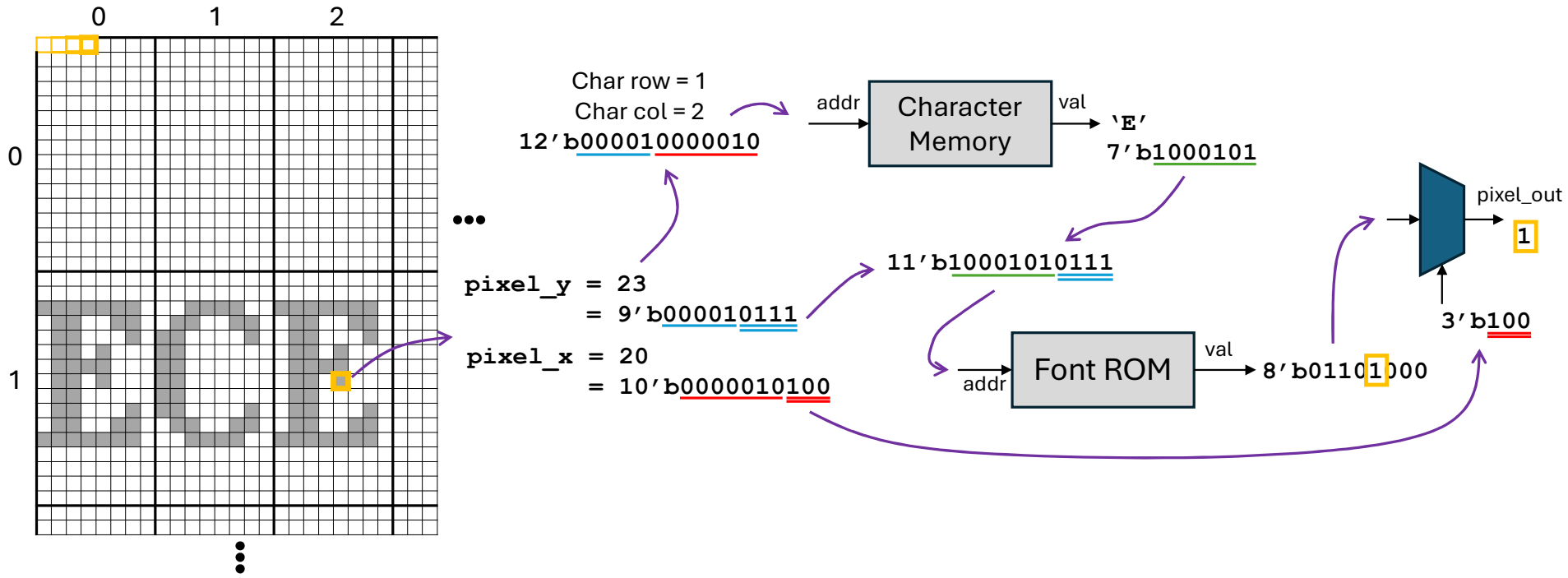
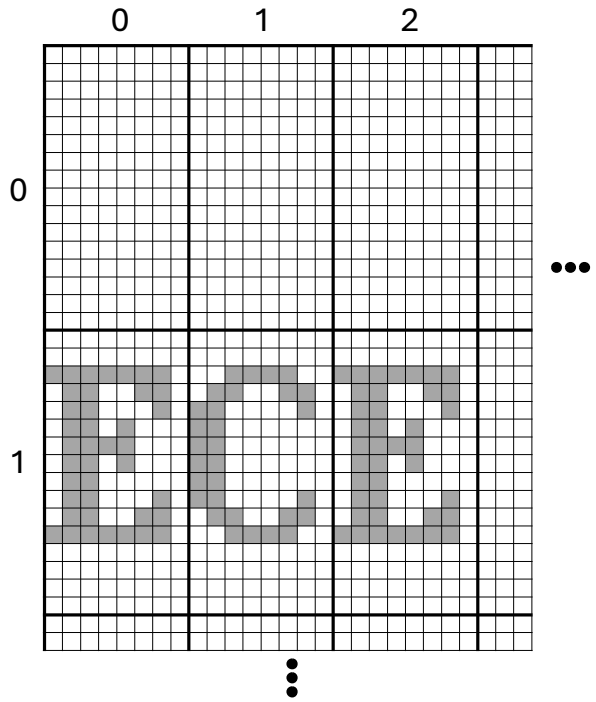
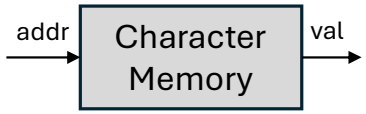

0
1
2
0
1
Char row = 1
Char col = 2
12'b000010000010
addr
Character Memory
val
'E'
7'b1000101
pixel_y = 23
= 9'b000010111
pixel_x = 20
= 10'b0000010100
11'b10001010111
addr
Font ROM
val
8'b01101000
pixel_out
1
3'b100

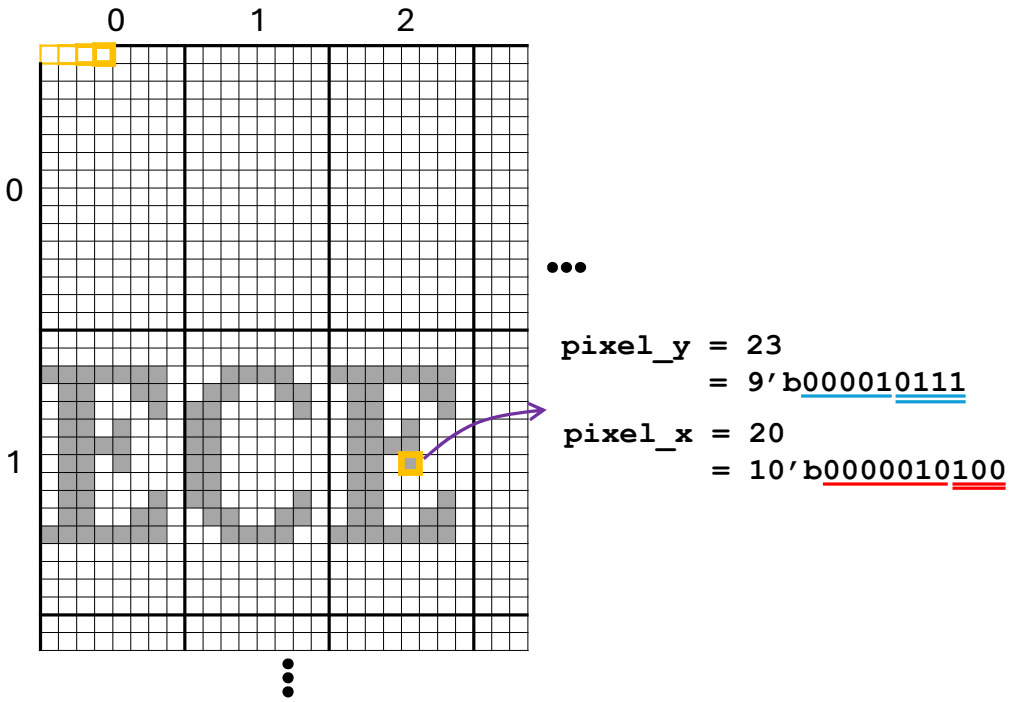


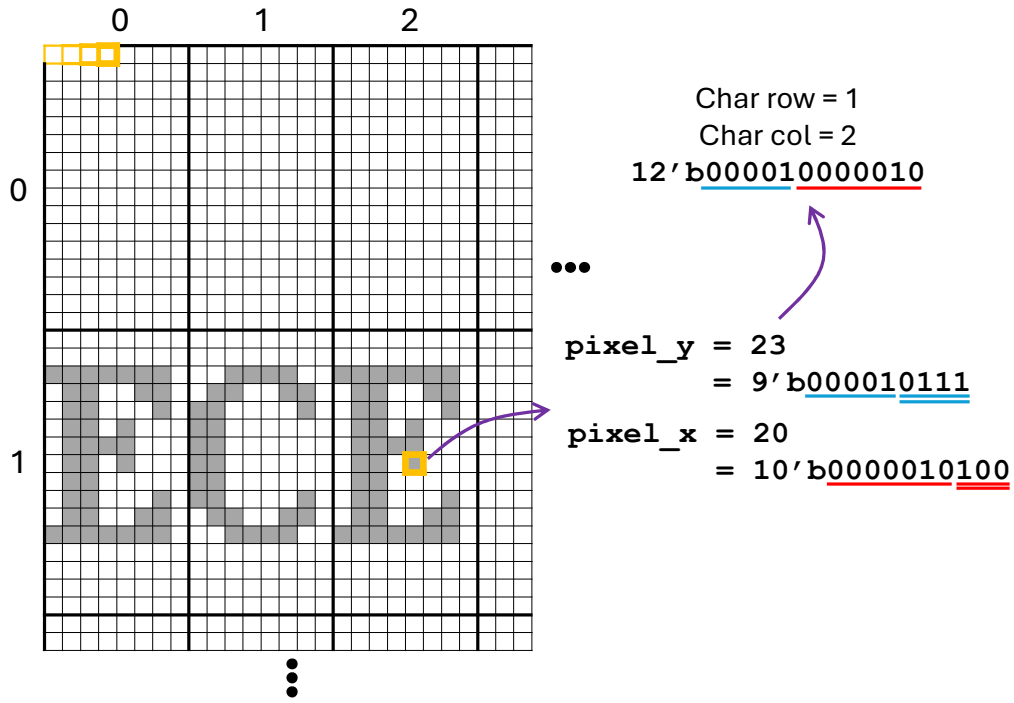


The Character memory stores which character (e.g. ‘A’, ‘b’, ‘C’) is at each character location on the screen

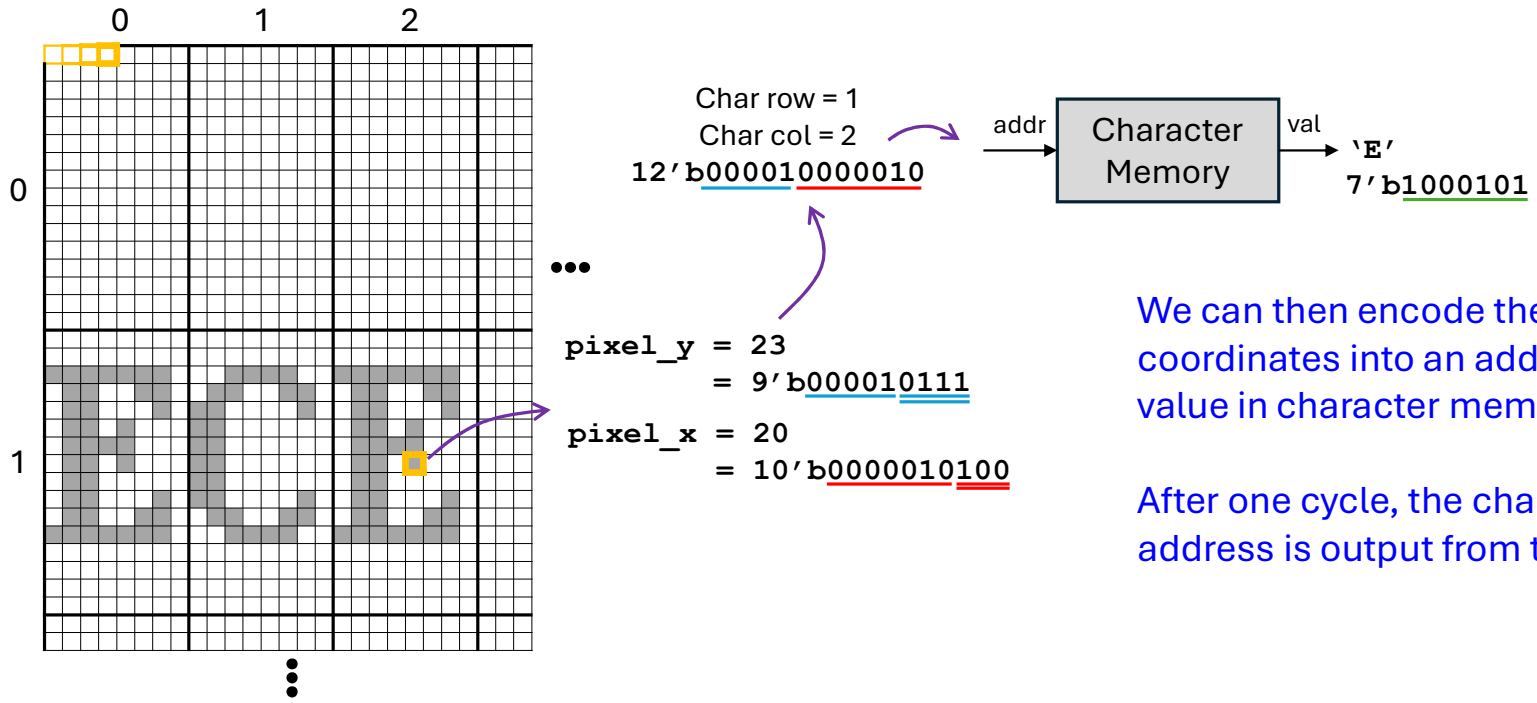
The goal is to draw all of these characters out to the VGA display.

To do this, the VGA timing module will loop over every x,y location and feed them in this module



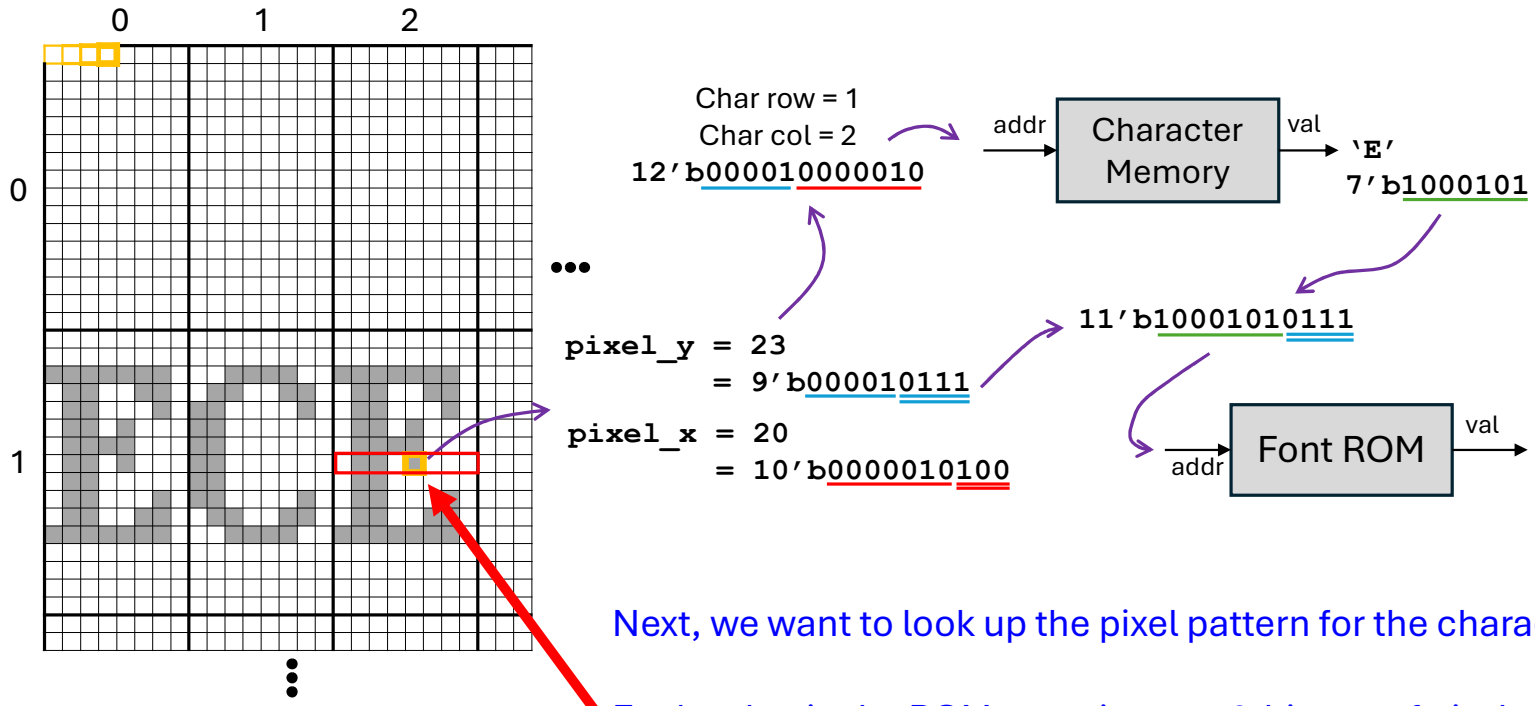


From a pixel_x and pixel_y location, we can extract out the character coordinates



We can then encode the character coordinates into an address to lookup a value in character memory.

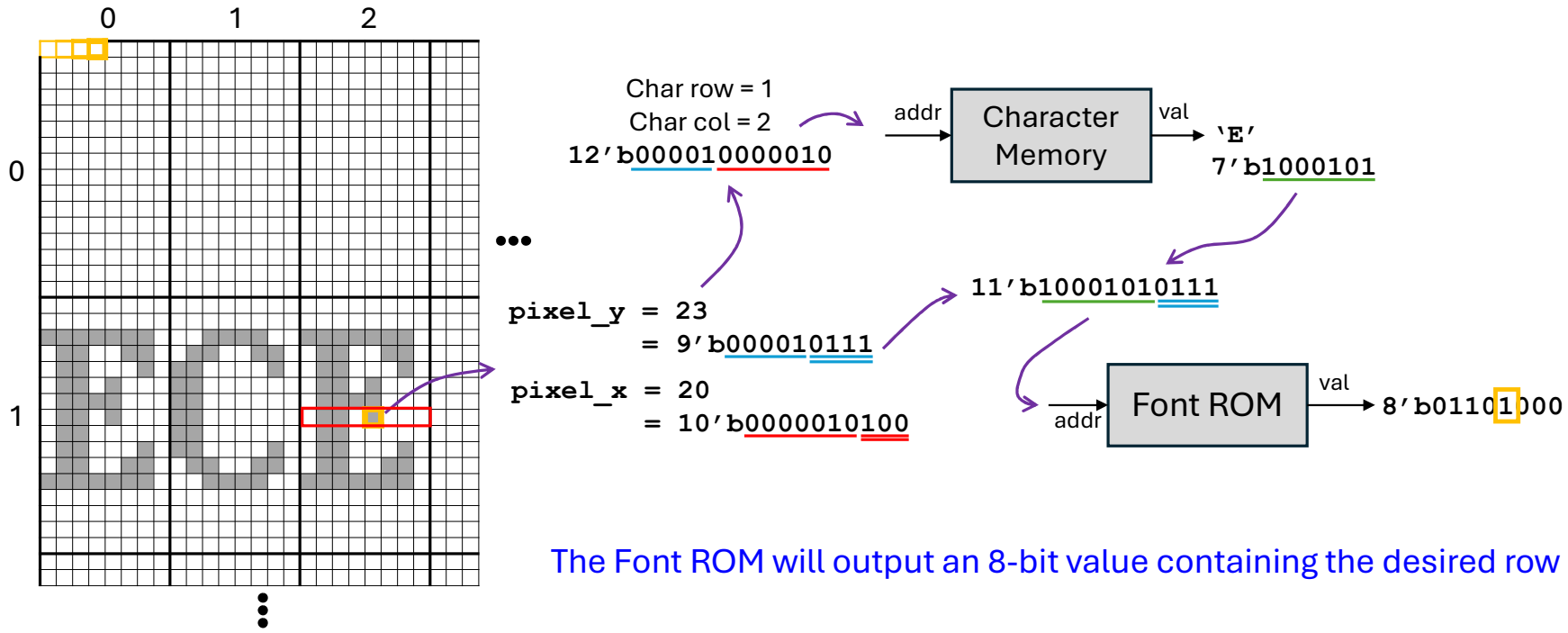
After one cycle, the character at that address is output from the memory.



Next, we want to look up the pixel pattern for the character in the Font ROM

Each value in the ROM contains one 8-bit row of pixels.

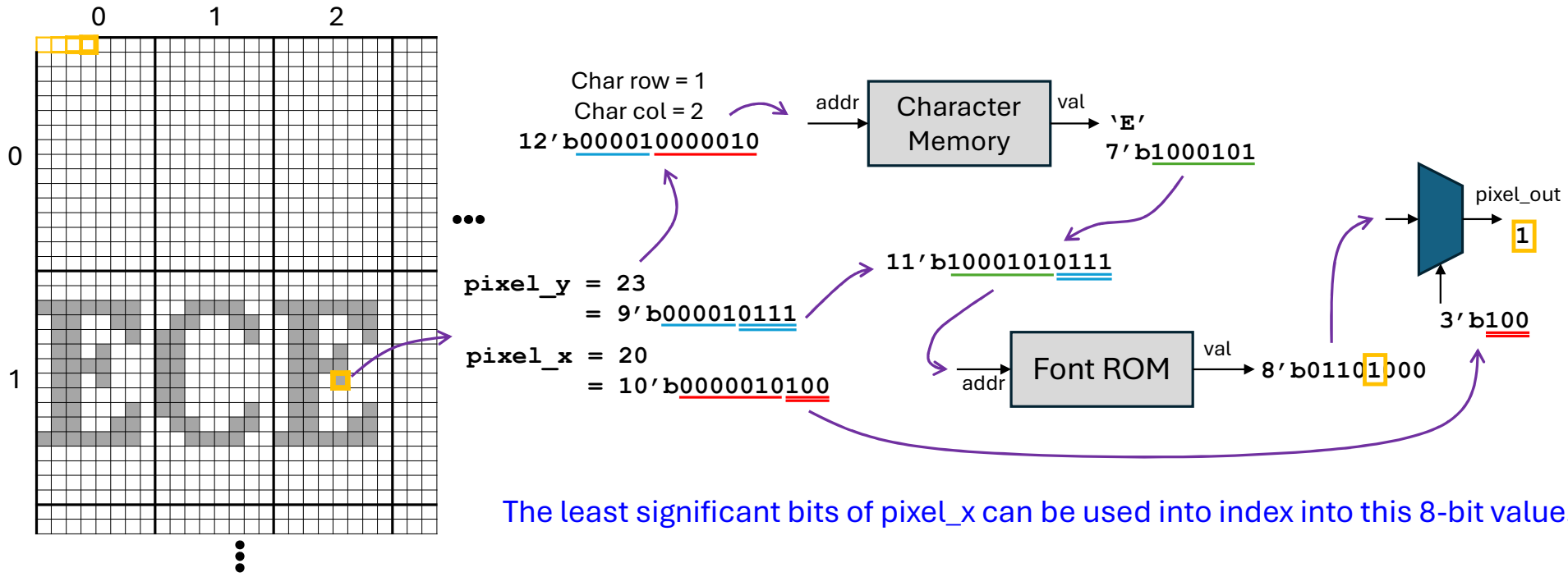
The address to the Font ROM contains both the character and the row number.
The row number comes from the least significant bits of pixel_y



The Font ROM will output an 8-bit value containing the desired row of pixels.

This also comes with 1 cycle of delay.

Next, we need to get the exact pixel.



The least significant bits of pixel_x can be used into index into this 8-bit value.

Feed this value into a multiplexer to output the desired pixel.