VGA Controller

Our VGA controller consists of a top module which takes in the 100MHz system clock and divides it down to 25MHz and feeds that to a VGA controller module which does the horizontal and vertical pixel counting. Both the pixel counters are triggered on the positive edge of the 25MHz clock. The counter is used to identify vertical and horizontal position and to ensure the pixels being generated are within the active pixels or drawing area. Once the counter moves past the drawing area, there is logic to restart the counter and to zero out the pixels while it restarts the count. Within the top module, the inputs are the system clock, a reset signal as well as the contents of a user defined register. To display those contents, there is a finite state machine which runs within a loop which extracts the bits of the register contents four bits at a time so that they can be matched up with a hex value. Once the equivalent hex is identified, a starting address for that hex digit is placed into a 32-bit variable used for containing the starting addresses for all four hex values to be displayed. These initial addresses are used by an additional nested font ROM module which identifies the bits to be used by the VGA controller to represent a particular ASCII character. The top module takes the counters as the output of the controller and on every system clock edge enters a block where the first step is to identify which hex character, we are displaying based on the horizontal position as defined by the counter. Knowing that the register data is 16 bits, we split the horizontal range into quadrants to display the four hex characters required to represent that data. In addition, the ASCII characters are separated into 16 lines of binary digits and so we divide the vertical range by 16 and the next step is to identify which segment of the vertical drawing range we are in and set the address which will be extracted from the font ROM module appropriately (i.e if we are in the fifth of the sixteen vertical segments, the address will be four past the initial address of the appropriate hex digit). Finally, by taking the modulus of the horizontal counter, we identify where within the given hex digit we are and pull the particular bits from the ASCII line for display.