

## VGA Verification Plan

This plan tries to verify the partial functionality of VGA to display text and uses dual-lock step method to flag errors that occur in only one of the instances.

The following functions of the VGA are to be verified:

1. Displaying text that conforms to the font\_rom file
2. Correct operation of HSYNC and VSYNC
3. Reset operation
4. Correct storage of characters – ram can store 900 characters, to store more, it needs to be reset or backspace functionality has to be used
5. Functionality of scroll, backspace and return keys (will not be verified in this coursework, but are part of the text displaying functionality)
6. Nothing is being displayed in the front and back porches

Assertions in the VGA:

1. RGB is 0 when pixel\_x and pixel\_y are 0, which verifies nothing is displayed in front/back porches
2. Assertions for scroll, backspace and return key (not done in this coursework, but a useful way to verify part of their functionality)
3. Console\_wdata has the correct value – 0 if the write conditions are not met and HWDATA on next cycle if write conditions are met

Driving the VGA; output of VGA is evaluated with a monitor and scoreboard to check whether symbols are displayed correctly:

1. Drive random symbols
2. Inject bug in top-level when HWDATA = 07
3. Test with random resets
4. Drive certain symbols to verify their position on the display

Random stimulus:

1. HWDATA\_data -> data that will be put into console\_wdata to display the corresponding symbol
2. command\_signals[2:0] -> [0]->HREADY, [1]->HSEL, [2]->HTRANS[1]
3. HWDATA\_upper\_bits -> upper 24 bits are not used in VGA, so most of the time they are zero, but sometimes they are not zero
4. inject\_wrong\_address[1:0] -> LSB for direction phase, MSB for data phase,
5. HADDR\_inject -> value to inject if either bits of inject\_wrong\_address are high

Finally, functional coverage will be sampled in the interface and code coverage is automatically generated by QuestaSim. Functional coverage is checked through the following covergroups:

1. Separate coverpoints (no cross coverage) for HWDATA, RGB, HADDR, console\_wdata
  2. Cross coverage of HSYNC and VSYNC and function that checks if RGB is 0 when HSYNC and VSYNC are low
- Sampling of HWDATA and HADDR (has illegal bin for invalid VGA addresses) is done only when HRESETn and HSEL is high because that is when the peripheral is selected.
  - Sampling of console\_wdata happens on the next cycle after a value is written to it.
  - Sampling of HSYNC, VSYNC and RGB happens every cycle if HRESETn is high