

# AXI Video Graphics Array (v1.00a)

## Specification

### Introduction

This document describes the specifications of the Video Graphics Array (VGA) core for the On-Chip AXI Bus.

The VGA is a 32-bit master-slave module that attaches to the AXI.

### Features

- Supports 32-bit AXI v2.0 bus interface
- Monochrome (1 bit per pixel, 32-pixel per word)
- Supports 640 x 480 video resolution
- Generates 60 Hz vertical synchronization
- Configurable start address
- Support for interruptions and polling

### Functional Description

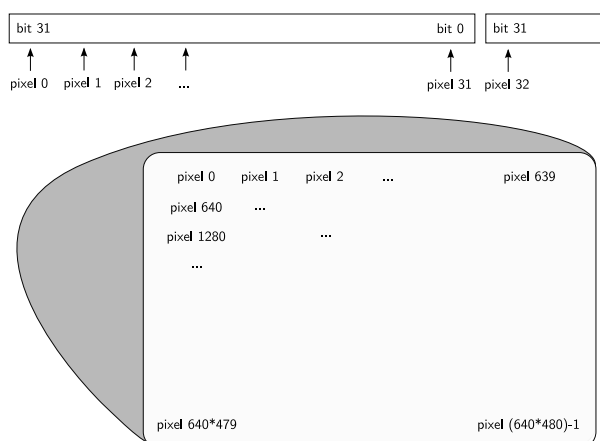


Figure 1: Video buffer data format

After reset, the VGA is initially idle. User sets the start address by writing to the **CFG** register. If the value written to this register is different than **0x0**, then the VGA becomes enabled.

When enabled, the VGA continuously reads 32-bit words, starting from the start address (value of the **CFG** register), and incrementing until the end of the video buffer. It drives the color signals **RED**, **GREEN** and **BLUE** and the horizontal synchronization **HSYNC**.

Each time the VGA reaches the end, it drives the vertical synchronization signal **VSYSN** and sends an interruption using the **IP2INTC\_Irpt** signal. It also sets the **INT** register to **0x1**. The **INT** register can be cleared by writing **0x1** to it.

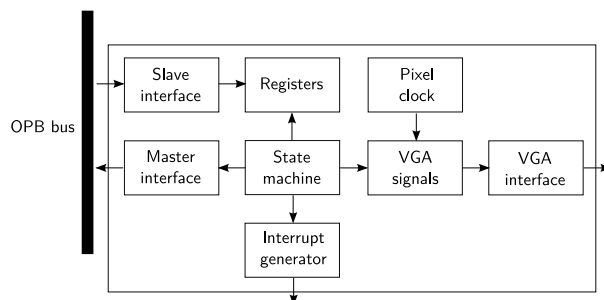


Figure 2: VGA Block diagram

The VGA can be idled by writing **0x0** to the **CFG** register.

### Programming Model

#### Modes

The VGA provides the following modes:

- **IDLE**: when **CFG** register is **0x0**. In this mode, a test pattern is displayed (vertical lines).
- **ENABLED**: otherwise

#### Register offsets

	Offset	Size	Type	Description
<b>CFG</b>	<b>0x00</b>	Word	R/W	Configuration reg.
<b>STT</b>	<b>0x04</b>	Word	R	Status register
<b>INT</b>	<b>0x08</b>	Word	R/W	Interrupt register

### Registers descriptions

#### Configuration register

The configuration registers holds the start address of the video buffer. The value should be modified soon after the vertical synchronization to give best results.

#### Status register

The status register is not implemented.

#### Interrupt register

The interrupt register is set to **0x1** after a vertical synchronization and can be cleared by writing **0x1** to it.