# Draw Block Documentation

Jonathan Ely

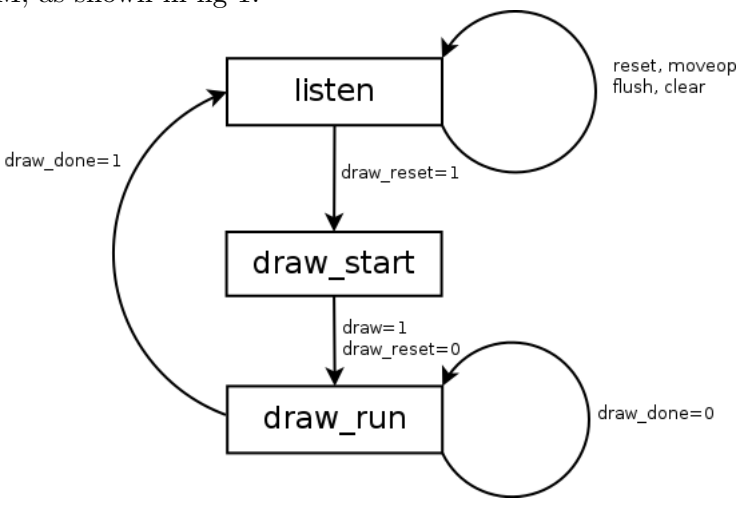
November 25, 2011

## 1 Introduction

The Draw Block is responsible for taking input commands from the host processor, decoding them and passing the pixel operations to the RAM Control Block. It is comprised of three VHDL entities: *draw_block*, *draw_any_octant* and *draw_octant*.

## 2 draw_block

This is the top entity for the draw block and supports interfaces for the host and the RCB. The main job of this block is to decode the host commands and provide pixel operations to the RCB. It is implemented as a 3 state FSM, as shown in fig 1.




In order to calculate the draw operations, the draw block creates a wrapper for the *draw_any_octant* entity from exercise 4.

### 2.1 SIGS: PROCESS

This combinational process decodes the *hdb* signal from the host processor into its more useful component parts: *xin*, *yin*, *pen* and *op*.

### 2.2 OCT: PROCESS

This process compares the x and y input positions to the pen position to work out which octant we want to draw in. It is used by the *draw_any_octant* entity.

### 2.3 STATECOMB: PROCESS

STATECOMB manages the state of the FSM. It also implements the Move Pen, Clear and Flush operations as they only take 1 cycle. For the draw command, it sets up the *draw_any_octant module* with the start position (by holding the *draw_reset* signal high) and passes to the *draw_start* state.

### 2.4 FSM: PROCESS

This clocked process increments the state with the *nstate* signal on the rising edge of the clock. There are two cases when this does not happen: when *delaycmd* is high or when *reset* is high. The *delaycmd* signal is set by the RCB and stops state progression. The *reset* signal returns the state to *listen*.

## 3 draw_any_octant - Exercise 4

This the *draw_any_octant* entity from exercise 4. It's job is to translate the input/output to *draw_octant* so that it can draw in any of the 8 octants: NNE, ENE, ESE, SSE, SSW, WSW, WNW and NNW.

It has been modified slightly to allow halting whilst the RCB is busy. This is achieved by stopping the clocked process whilst *delay* is high.

## 4 draw_octant - Exercise 1

This is the *draw_octant* entity from exercise 1. It has been modified slightly to allow halting whilst the RCB is busy. This is achieved by stopping the clocked process whilst *delay* is high.