Accelerometer Controls

SPI

Accelerator Arithmetic

ADXL362

Score/Timer

7 Seg Display

Accelerator ticker for ball control

Ball behavior

Maze wrapper

Maze ROM

Binary to BCD

VGA

dtg

Colorizer

Ball icon ROM wrapper

Ball ROM

img ROM wrapper

Img ROM

If game is won

Stop accepting new input

Else, game is still being played.

If there is new movement

If movement is validated

If move is valid -> Increase the ball position

Reset movement registers

Else, need to check the movement direction for obstacles

If wall ==1

Move is not valid, abandon movement

else if hole == 1

move is not valid

signal for game position reset

reset ball position to INITIAL

else if ball reaches ‘win zone’

assert ‘game won’

else

move is valid

Movement is validated

Else, keep checking for new movement

The images used in the program are adapted from stock images or drawn by hand. The images is edited using Microsoft Paint to make an image with a somewhat solid lines and color. This is especially important for the maze since the bot uses the color to check for obstacles and requires sharp edges with little or no anti-alias. The images are save in 256 colors BMP file format.

We utilized a simple Matlab script to generate the initial draft for the ROMs’ coefficient (coe) files (http://www.mathworks.com/matlabcentral/fileexchange/12437-bmptocoe). This script generates coefficient files which use 8-bit pixel width (8 bit RBG color). Bit [7:6] are blue, [5:3] are green and [2:0] are for red. The resulted coefficient files are then edited to give a more accurate colors on the Nexys4 12bit color scheme.

The ball icon is drawn in the same fashion as in Project 2. Whenever the vertical sync and the horizontal sync signals are matched those of the ball’s x-y coordinates, the wrapper would index in to the icon ROM to get the RBG value for the VGA.

The congratulation image is also work similar to the ball icon. The image is 240x 240 pixels store din a single port ROM. Instead of checking for a match of the horizontal and vertical sync signals as in the ball icon (since the image does not move), the image is only displayed when the ball reaches the ‘win’ condition and asserts the ‘win’ signal. When the wrapper receives this signal, it’s index into the ROM and sent out the pixel information as a function of sync signal + offset (to center the image)