**One-player Blackjack**

A project plan for ECE 383

by

C2C Dananga K. Agalakotuwa

April 22, 2024

* 1. Objective Statement

This will be a single-player game of Blackjack against the computer. The computer will act as the dealer.

* 1. Requirements

Basic functionality

* A player must be able to play a complete game of Blackjack against the computer.
* Use keyboard inputs to implement game inputs.
* The HDMI Monitor must be used to draw the "cards" and simulate gameplay. Card dimension would be 32x32 pixels.
* For this level, the deck of cards can be fixed instead of being random.
* *The UART feedback may be used for debugging and echoing gameplay.*

B-functionality

* Complete Basic functionality.
* A mouse must be used for game control.
* For this level, the deck of cards needs to be random, and cards cannot be repeated (2 Jack of Clubs for example).

A-functionality

* Complete both basic functionality and B-functionality.
* Include audio sound outputs for start of game and end of game.
  1. Level-0 Description & Top-Level Design

Overall Inputs:

Keyboard via USB

Mouse vis USB

Switch

Overall Outputs:

Display on Monitor

Speaker

Overall Behavior:

A switch on the FPGA would be used to start the game. This switch will act as an interrupt, if turned off game will end regardless of progress. The game will initially be played using the keyboard (“H” key to hit and “S” key to stand). A new input device will be introduced later(mouse) to control game inputs (left mouse button to hit and right mouse button to stand). The display will show cards as they are dealt (dealer’s initial card will not be revealed till end of game). Once the end of game has been reached (dealer is done hitting) the result of the game will be shown on the display (a green screen for player win, red screen for player loss).

Keyboard

Mouse

*\*B-functionality*

FPGA

Custom Hardware

Monitor

Speaker

*\*A-functionality*

UART

Microblaze

HDMI out

Audio out

USB

PS2

HDMI in

Audio in

1. Plan

The project plan defines how you are going to go about implementing the design set forth in your proposal. The plan should then go on to include the following sections.

* 1. Proposal

I have corrected the mistakes pointed out to me by Dr York for Section 1.

* 1. Detailed Architecture and Sub-System Design

You need to provide the detailed design of your system. A detailed design should be split into level-1 subsystems, such as Datapath and control.

* + 1. Level-1 Design

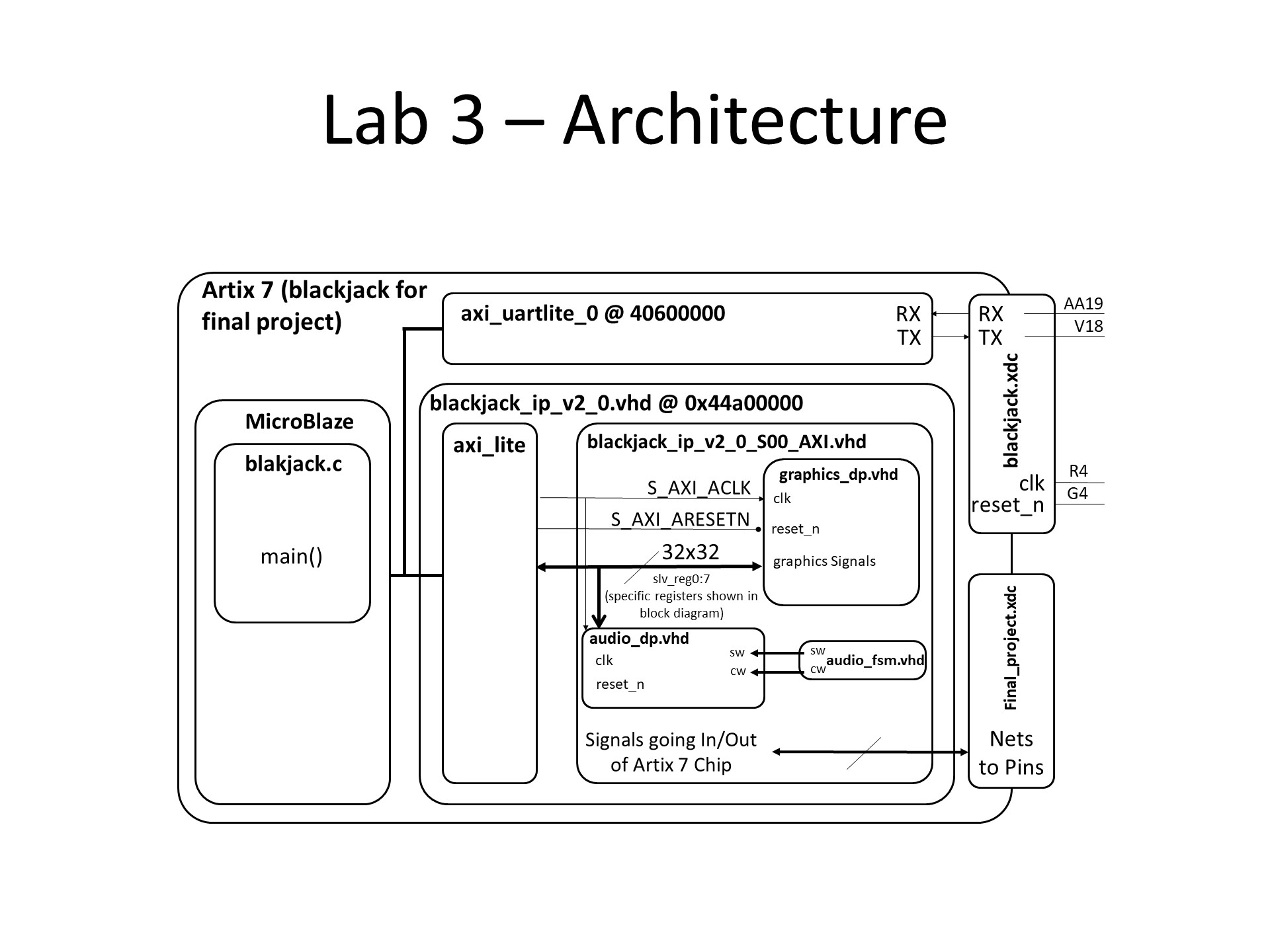


Figure 1: Microblaze Level-1 Design

Figure 1 shows the microblaze design which will be used for my final project. The specific registers written and read using microblaze will be shown below.

A close-up of a diagram

Description automatically generated

Figure 2: Level-1 Design

Figure 2 shows the main three components that I will be using to achieve the final project. I rely on the microblaze registers to run my graphics datapath, therefore, most of my “game” code will be in C-code.

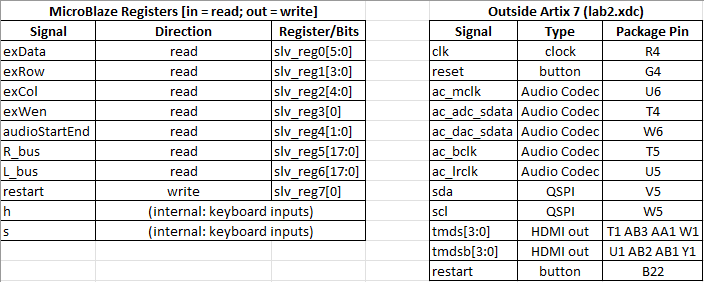


Figure 3: Datapath signals to Microblaze registers

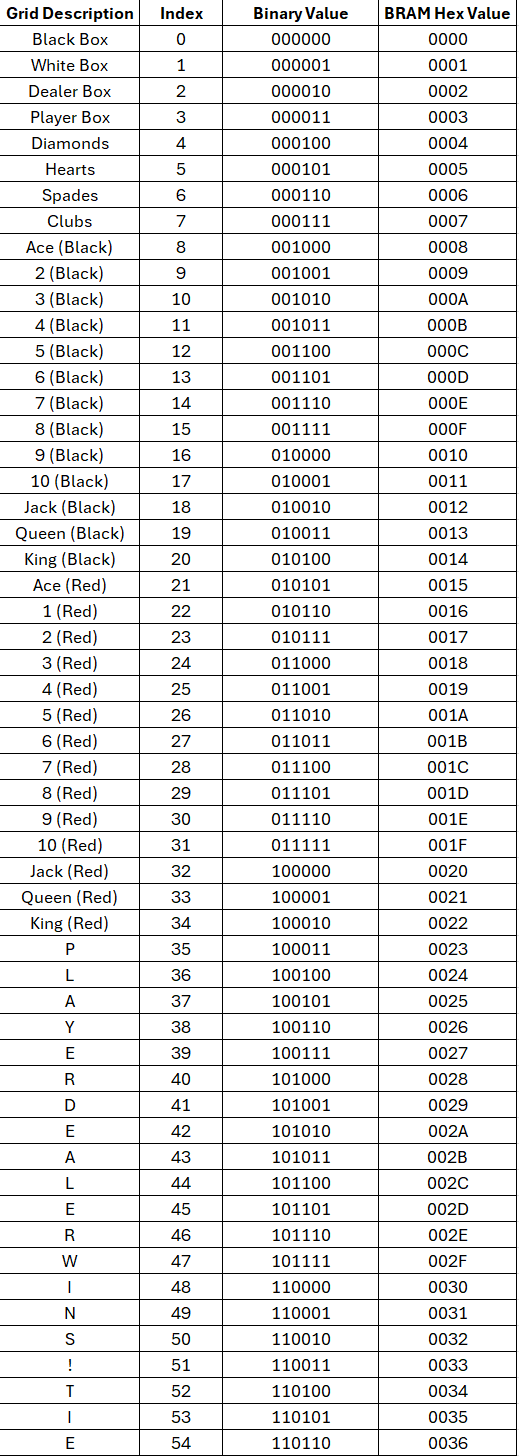


Figure 4: Grid Memory

* 1. Calculations/Analysis/Drawings

Grid Memory Calculations

* I am implementing a 20x15 grid with each cell containing 32x32 pixels.
* My cell rows would be represented with 4-bits and cell columns would be represented with 5-bits. From this,
* This means that my BRAM would have 300 entries.

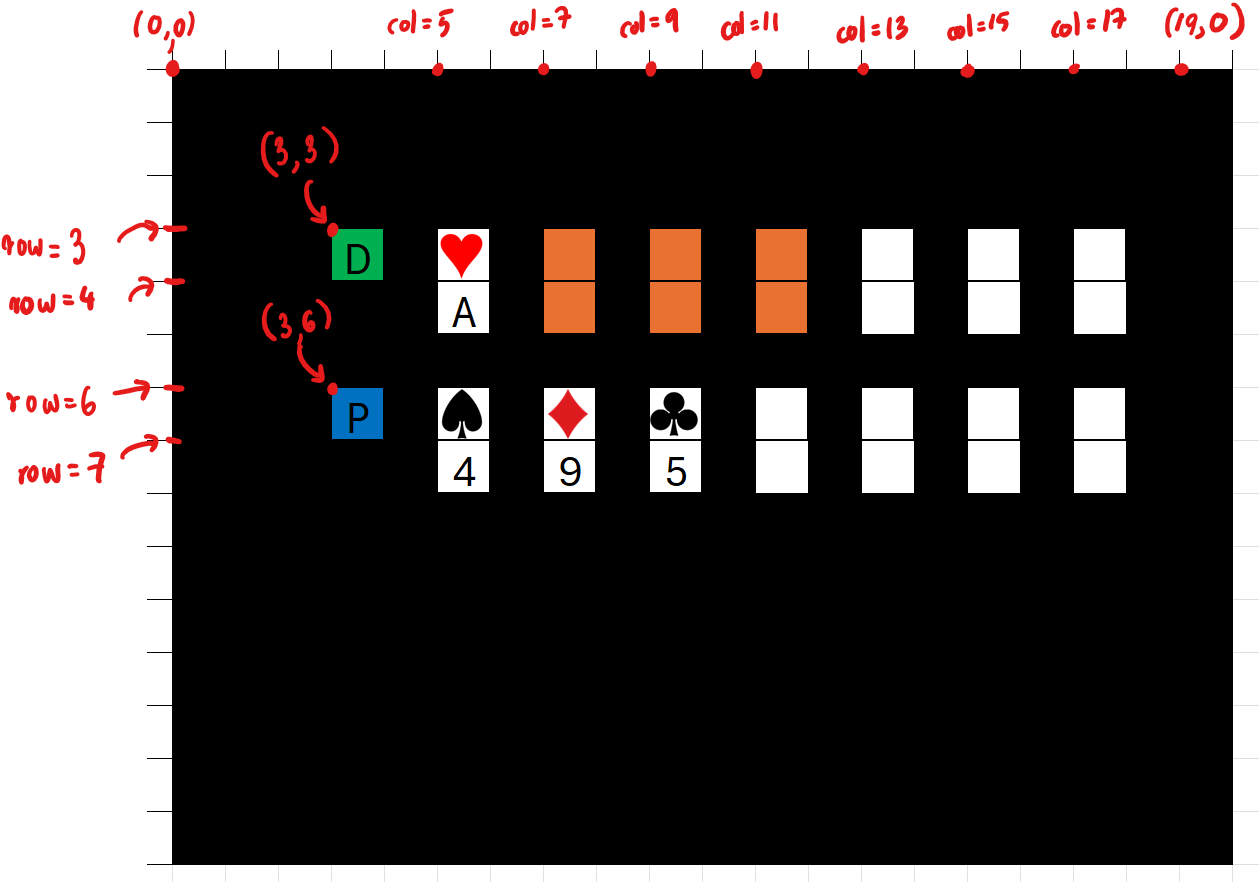


Figure 5: Gameplay Screenshot

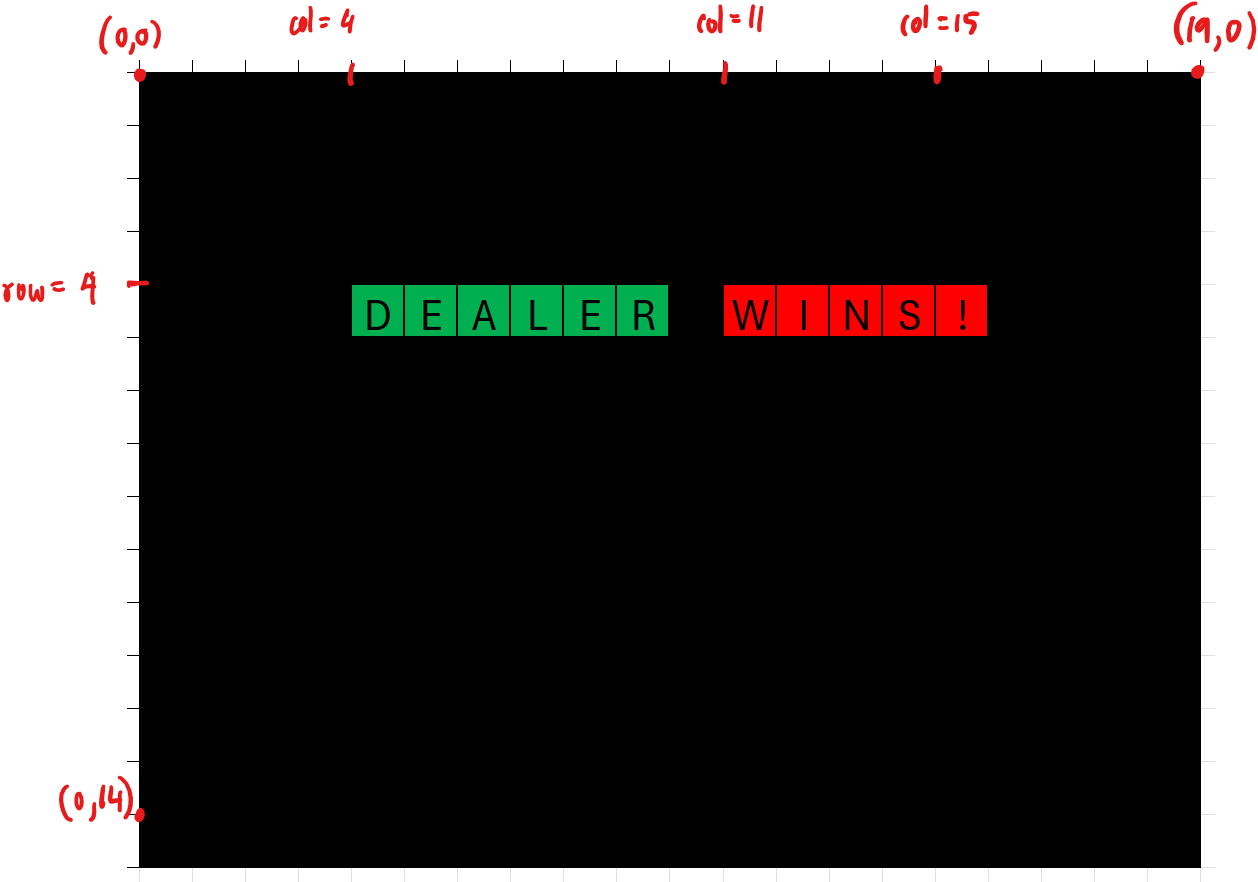


Figure 6: End result screenshot

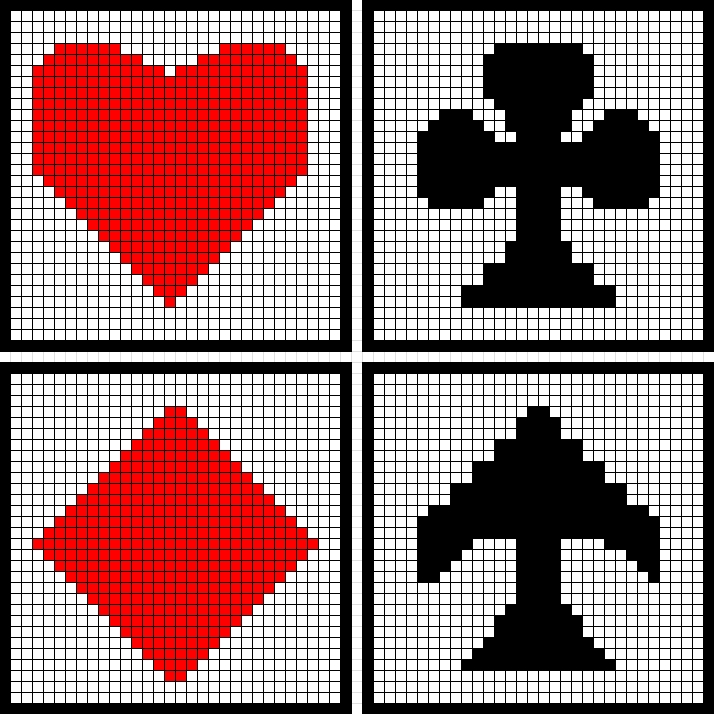


Figure 7: 32x32 pixel cells containing suits

Figure 5 shows the expected output of the display when the game is being played. The orange cards shown on the dealers hand are cards that are in the dealers hand (but the player cannot see them).

Figure 6 shows one of the result possibilities, this case where the Dealer wins.

Figure 7 shows the 32x32 pixel cells that would contain the suits of the deck of cards.

On figures 5, 6, and 7, I have only noted the most important coordinates that would be needed when interfacing this.

* 1. Milestone I

This milestone will mostly consist of putting all the necessary files together and setting up microblaze. After setting up microblaze and other files, as seen in the level-1 design, set up an array of a deck of cards and use SDK to test different combinations of cards being printed to the screen. You do not have to implement the game play yet, although encouraged.

* 1. Milestone II

Finish the gameplay code and ensure the game runs properly by testing with an unrandomized deck of cards and see if behavior is as expected. After this, set up gameplay to work with a random deck of cards, and ensure no cards will be repeated. You can use print statements on your .c file to see which card is used every time a “hit” is made by the player.

Ensure you have set up the mouse circuit needed and know how to implement it. If time permits, set up PS2 mouse to implement “hit” and “stand” by the player.

* 1. Updated Functionality and Requirements

Major corrections included in Dr York’s feedback were updated in the Section 1.

Documentation

None.