ESDC - Block Algorithms and State Diagrams Mastermind 2-Player Game

Contents

*The delivery of the VGA interpreter state diagram will be postponed, as a better understanding of the VGA blocks hardware is required.

1. Introduction

Specs Update

In this laboratory assignment we are not delivering any updated project SPECS, as no changes have been asked by the teacher and also at this point we have not performed any changes in the specifications of our design.

Block Diagram Update

In this laboratory assignment we are not delivering an updated project Block Diagram (this has been discussed with the teacher). However, there are some improvements related to the Block Diagram that should be taken into consideration for the understanding of this document:

- The Main Block and the Identifier Blocks (Alice and Bob Identifier Blocks) have been put together in a new block called Init Block.
- In the VGA Blocks, the first dual ram block and the write memory block have been suppressed, and the addresses will be specified by the VGA_interpreter directly.

We are waiting to develop the definitive VHDL files to be able to explain all these changes with more detail in further deliveries.

Document Description

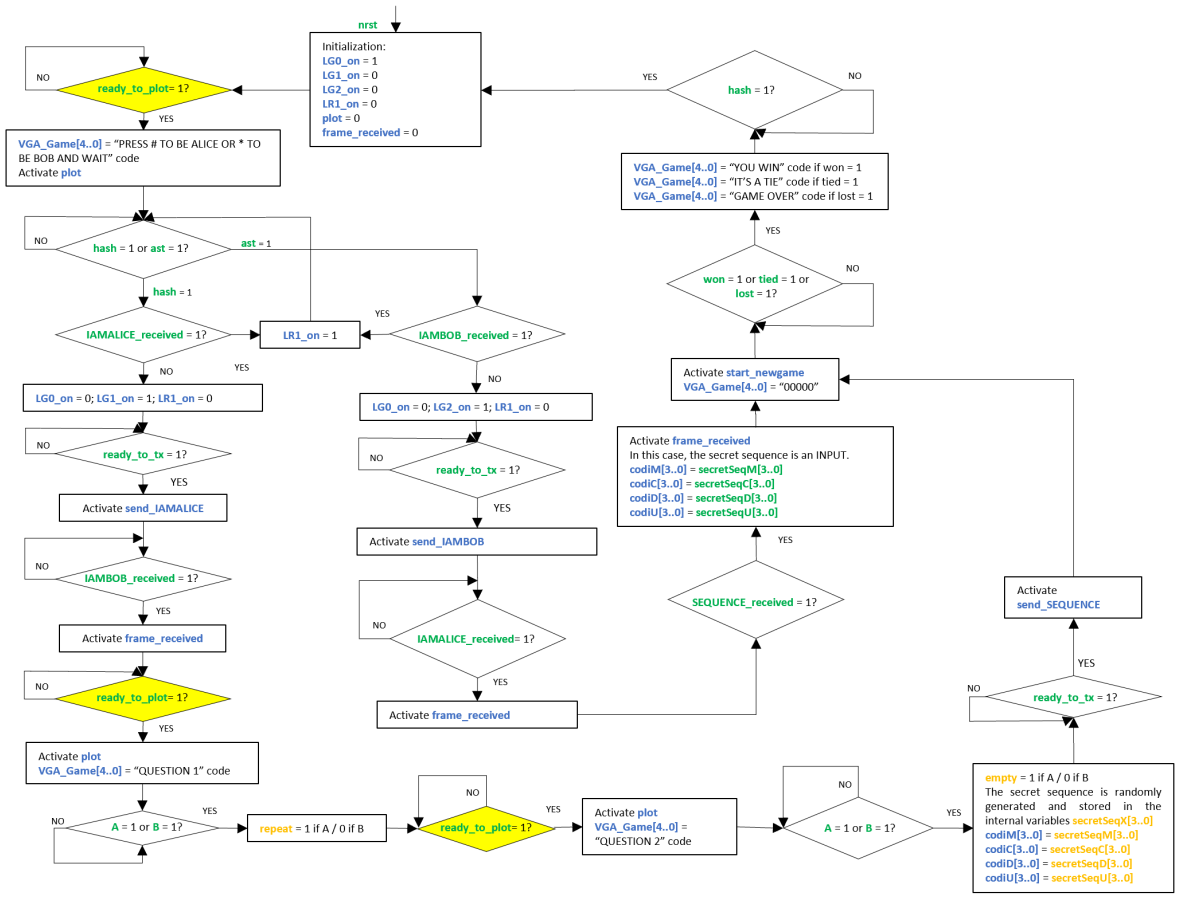
In Section 2, we will provide the reader with a detailed description of the algorithms of each of the main blocks of our design. As it is asked in the assignment description, we have chosen different colors to represent output, input and internal variables. Output variables are colored in **blue**, input variables are colored in **green** and internal variables are colored in **orange**. The state diagrams of the main blocks can be found in Section 3.

After the pasted images, we have included an URL link with a higher quality image, just in case anybody is interested in studying the algorithms/state diagrams in detail. The webpage we have used to do this is *ibb.co*.

We have also added some comments in the text that might help the reader to understand the general behaviour of our system. To see the text with the whole detailed description, please go to our already delivered and graded document *BD_Albert_Helena*.

2. Block Algorithms

Algorithm of the Init Block



Init Block Algorithm: LINK to see with higher quality (please ask the students if the hyperlink is broken)

This Block (Init Block) is a mix of the Main, Alice Identifier and Bob Identifier Blocks. We have finally decided to put these blocks together in order to simplify the interconnection of input/output variables in our schematic (*.bdf* file).

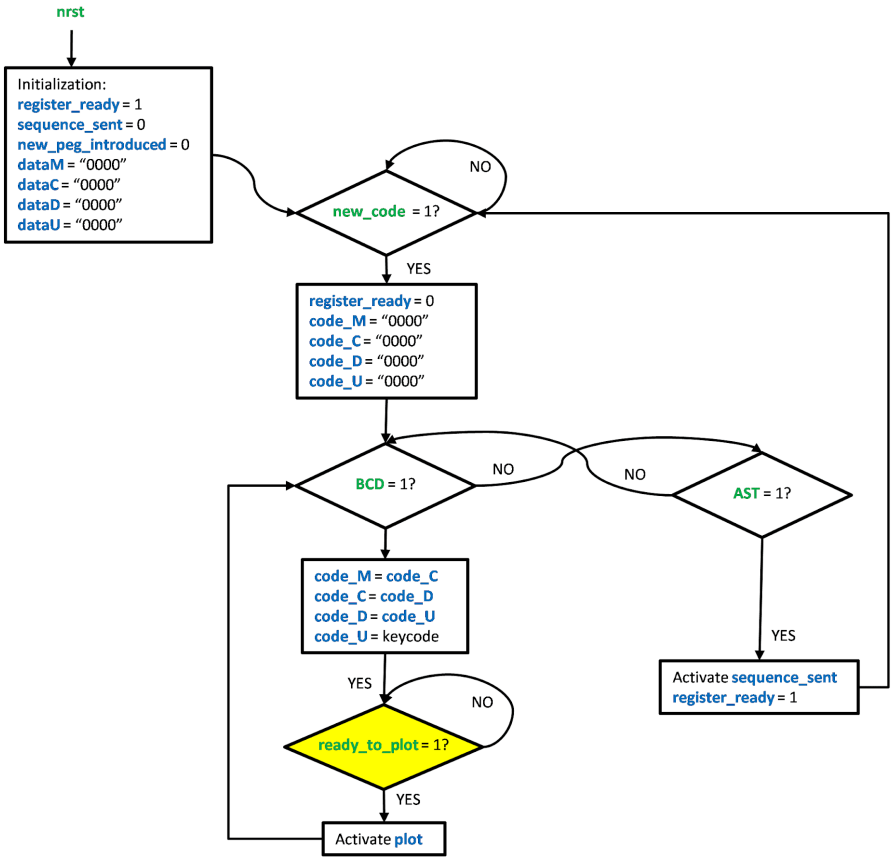
Internal Variables:

repeat: True if Alice has decided that the secret sequence has repeated colors.
empty: True if Alice has decided that the secret sequence has empty pegs.

SecretSeqX[3..0] stores the secret sequence.
secretSeqM[3..0]; secretSeqC[3..0]: secretSeqD[3..0]: secretSeqU[3..0]

For each output variable, and with the objective of making decisions/calculations, we will create an internal equivalent variable:
output_variable_IN

Algorithm of the Guessing Register



Guessing Register Block Algorithm: LINK to see with higher quality (please ask the students if the hyperlink is broken)

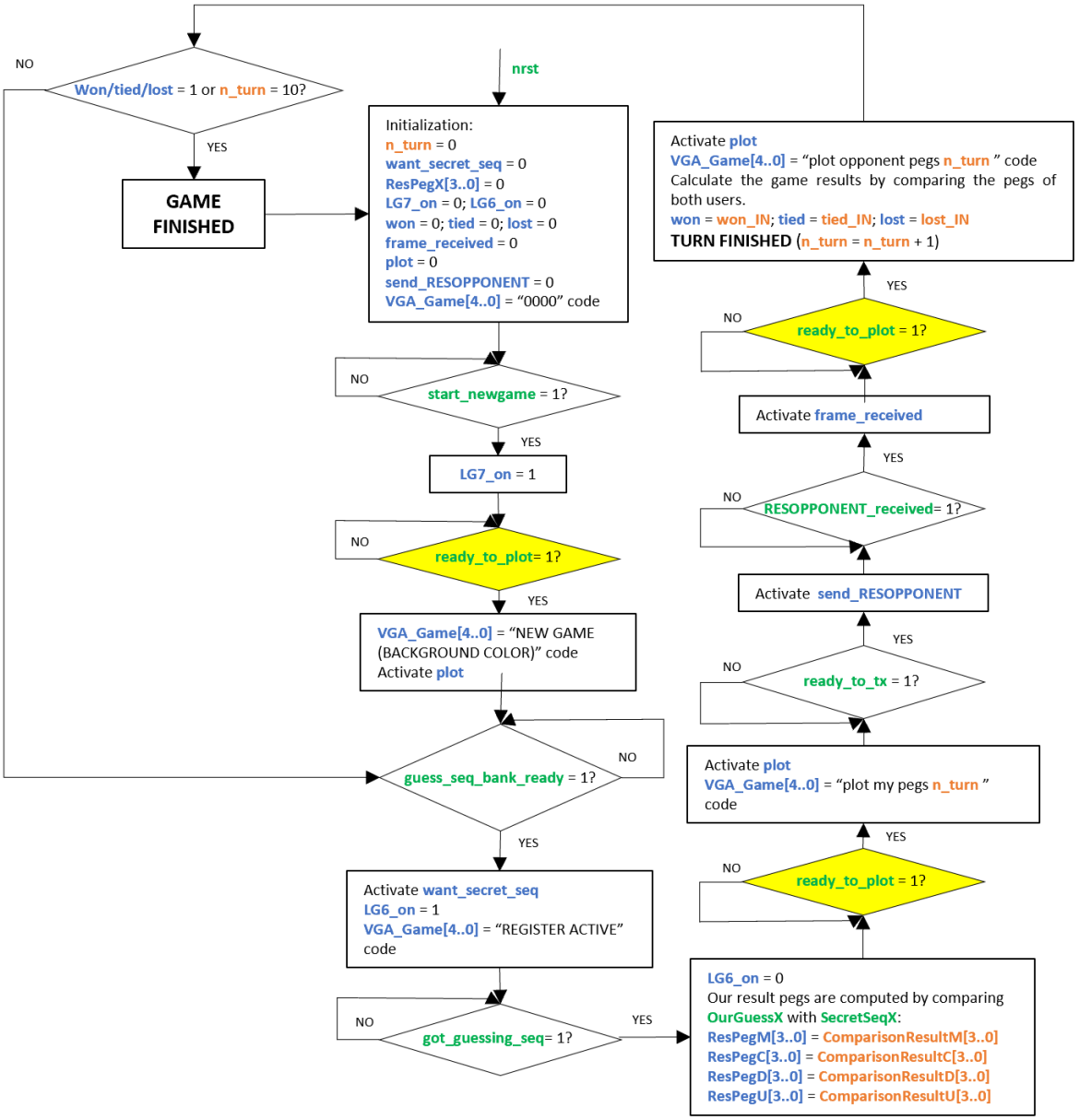
This register waits until the VGA Interpreter says (by activating the ready_to_plot register input) that the VGA is free and can plot something new. The register does this everytime the user has introduced a new peg and therefore there is the need to plot this new peg on the screen. Therefore, the procedure is the following one:

1) User introduced a new peg.
2) The register waits until it is possible to plot on the screen
3) The register output ***plot*** is activated and the peg is plotted.
4) We return to 1) unless user presses the * key.

Internal Variables:

For each output variable, and with the objective of making decisions/calculations, we will create an internal equivalent variable:
output_variable_IN

Algorithm of the Mastermind (Game) Block



Mastermind (Game) Block Algorithm: [LINK to see with higher quality] (please ask the students if the hyperlink is broken)

We could say that this is the most important block of the system because the main processes that happen when the game is ON happen in this block. The interconnection between this and the other blocks is the key to understand our design.

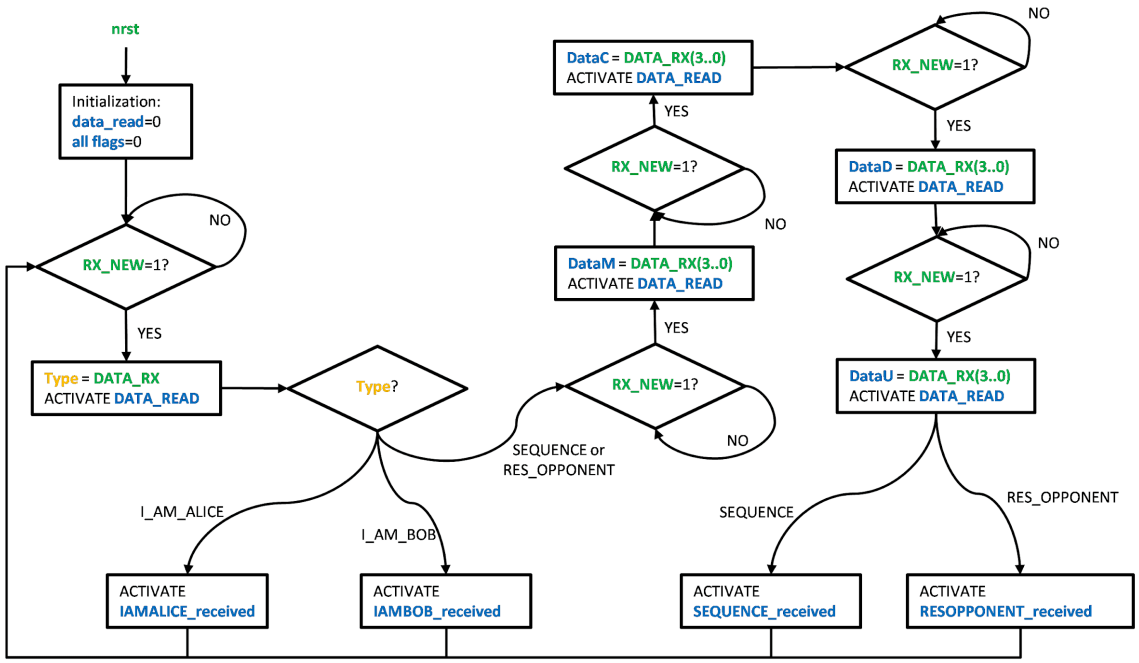
Internal Variables:

n_turn: this variable stores the value of the current turn (it has a maximum of 10). The maximum is 10 because in the Mastermind Classical Game definition, the challenge is to guess the secret sequence in 10 turns.

ComparisonResultX[3..0] stores the result of the comparison between the secret sequence and the user guessing sequence.
ComparisonResultM[3..0]; ComparisonResultC[3..0]; ComparisonResultD[3..0]
ComparisonResultU[3..0]

For each output variable, and with the objective of making decisions/calculations, we will create an internal equivalent variable:
output_variable_IN

Algorithm of the Receiver Block



Receiver Block Algorithm: [LINK to see with higher quality] (please ask the students if the hyperlink is broken)

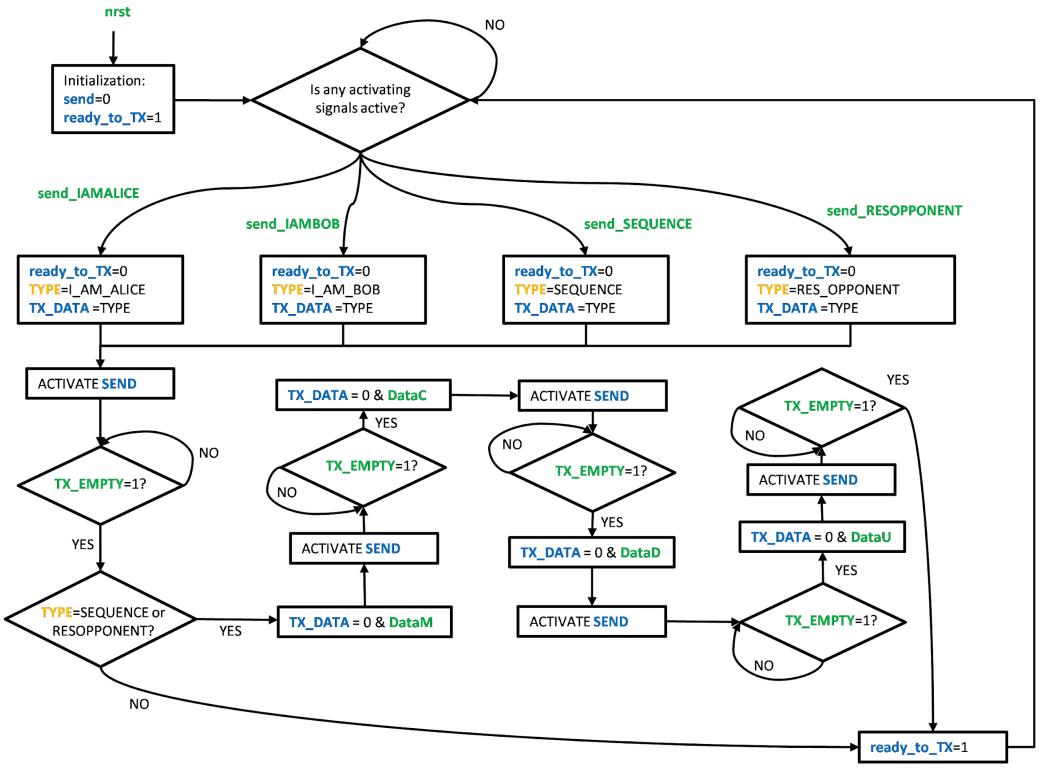
The algorithm of our system Receiver Block is quite similar to the one in the Design 1 from the Laboratory. The possibilities that we have (taking into account the FRAMES that we have in our system) are the following ones:

- IAMALICE_received
- IAMBOB_received
- SEQUENCE_received
- RESOPPONENT_received

Internal Variables:

Type: As in the Laboratory Design 1 design, this variable stores the FRAME TYPE that is being received by our system.

Algorithm of the Transmitter Block



Transmitter Block Algorithm: [LINK to see with higher quality] (please ask the students if the hyperlink is broken)

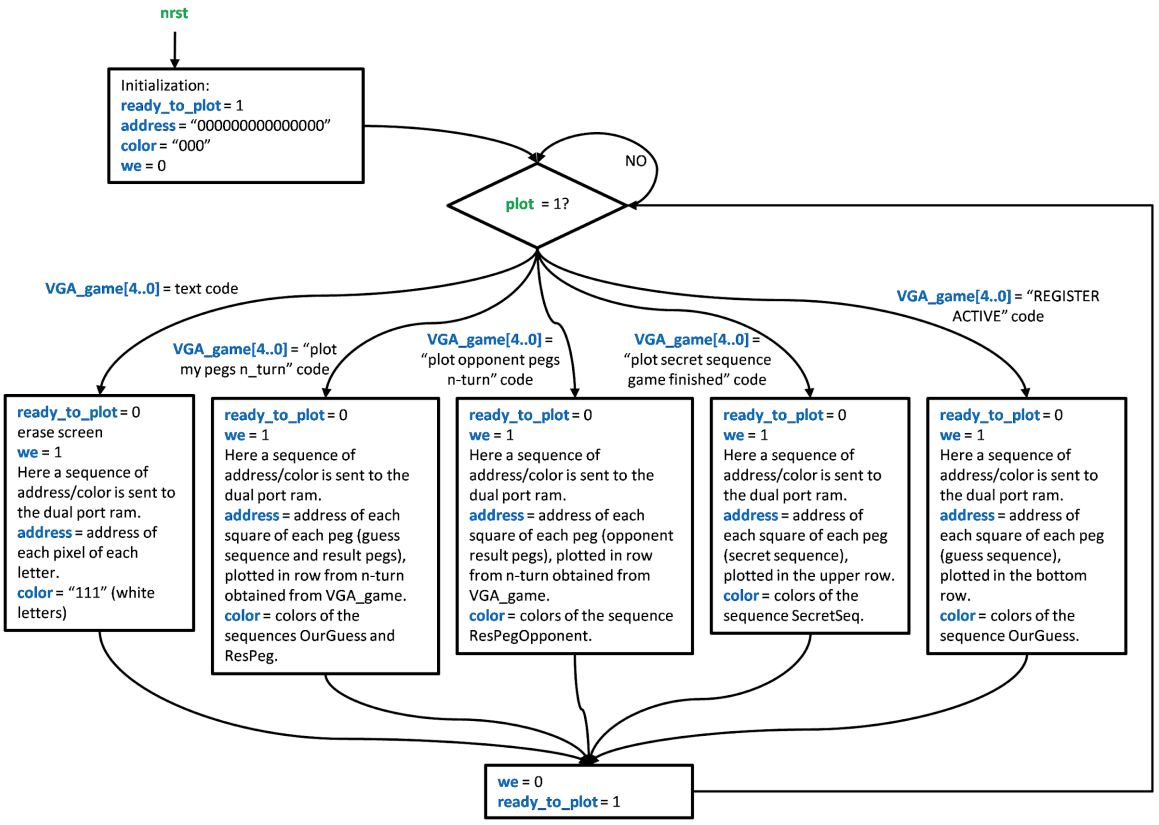
The algorithm of our system Transmitter Block is quite similar to the one in the Design 1 from the Laboratory. The possibilities that we have (taking into account the FRAMES that we have in our system) are the following ones:

- SEND_IAMALICE
- SEND_IAMBOB
- SEND_SEQUENCE
- SEND_RESOPPONENT

Internal Variables:

Type: As in the Laboratory Design 1 design, this variable stores the FRAME TYPE that is being received by our system.

Algorithm of the Video Graphics Array (VGA) Block



VGA interpreter Block Algorithm: LINK to see with higher quality (please ask the students if the hyperlink is broken)

Basically, the behaviour of this algorithm is the following. It enters in a waiting state until the activating signal ***plot*** is high. This activating signal can come from the following blocks:

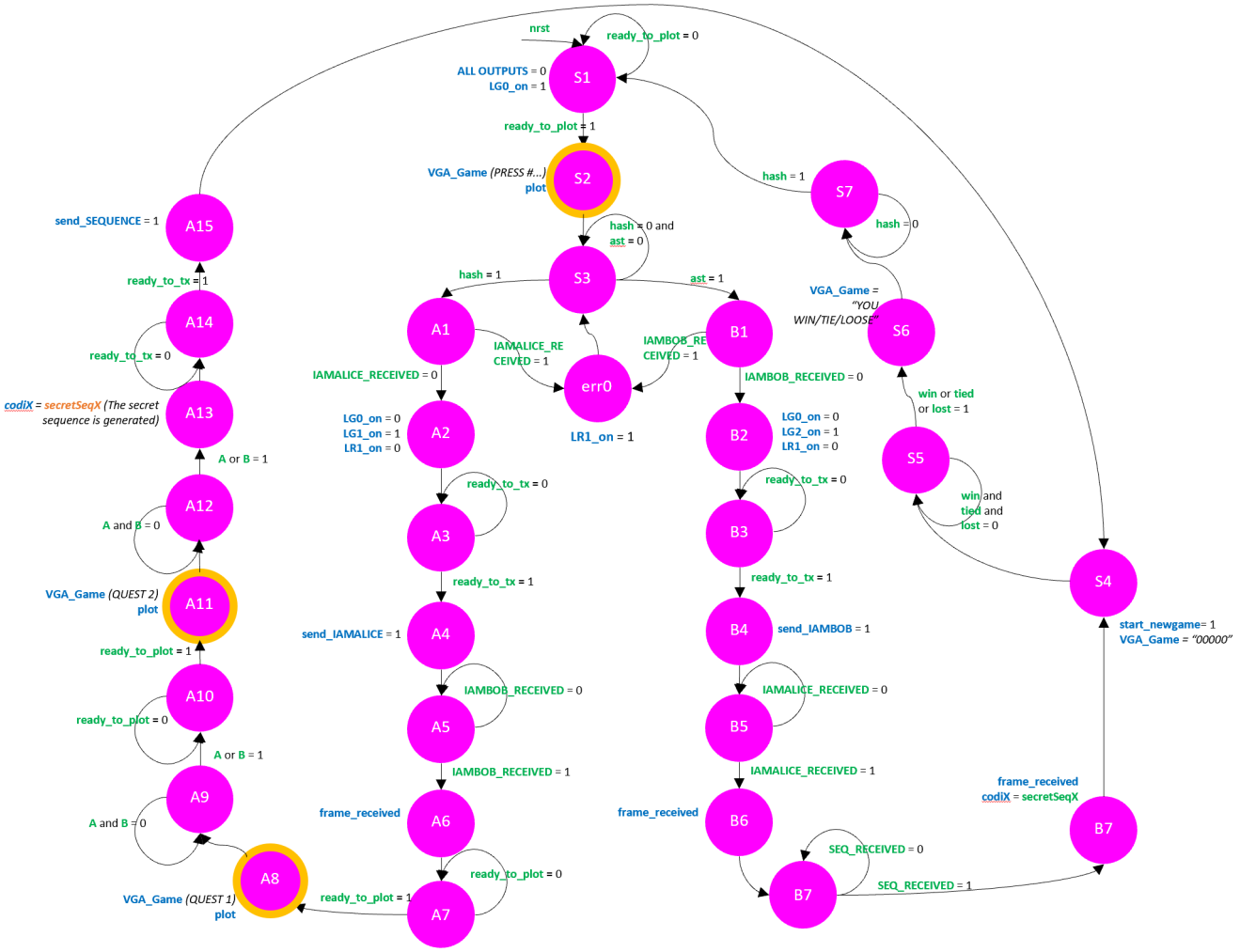
- Init Block
- Game (Mastermind) Block
- Guessing Register

When the activating signal is activated, the output ready_to_plot turns to 0 in order to inform the blocks that the VGA Interpreter is busy (all the blocks will enter in a wait state until this signal is high if what they want to do is plotting). What the VGA Interpreter does is reading the value of the signal VGA_Game and, depending on the value of this signal, plotting one thing or another.

Internal Variables to determine, as the current description we have of this block is 100% hardware and the algorithm/state diagram are hard to define at this project stage.

3. Block State Diagrams

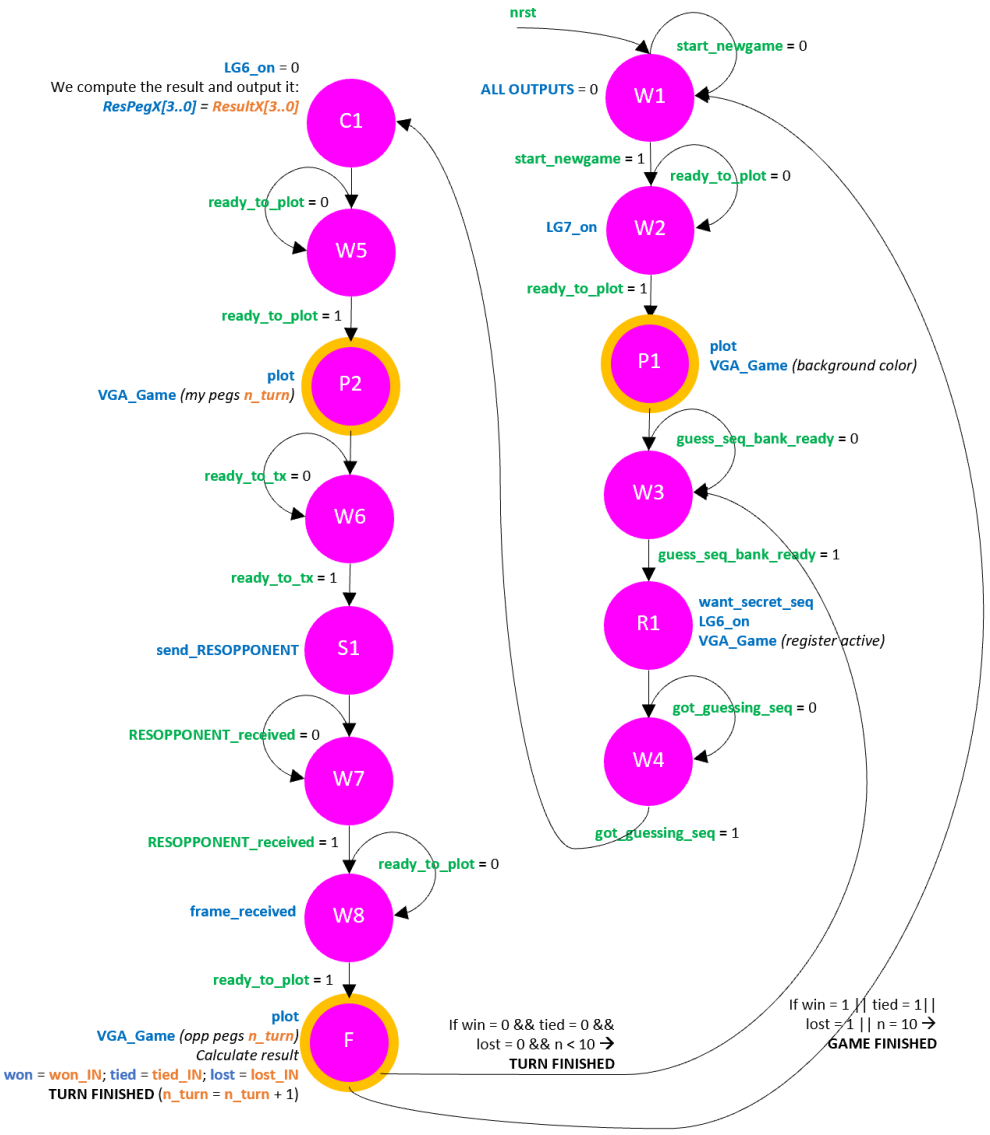
State Diagram of the Init Block



Init Block State Diagram: [LINK to see with higher quality] (please ask the students if the hyperlink is broken)

States highlighted with an orange line are the ones where the VGA plots on screen what is indicated in the VGA_Game signal. As **our design is symmetric**, we have defined the Init Block State Diagram like this (we have two possible paths to follow depending on whether we are Alice or Bob). States in the Alice path are named with a letter A and an ID of the state. States in the Bob path are named with a letter B and an ID of the state. General states (common states for both characters) are named with a letter S (of 'State') and an ID of the state.

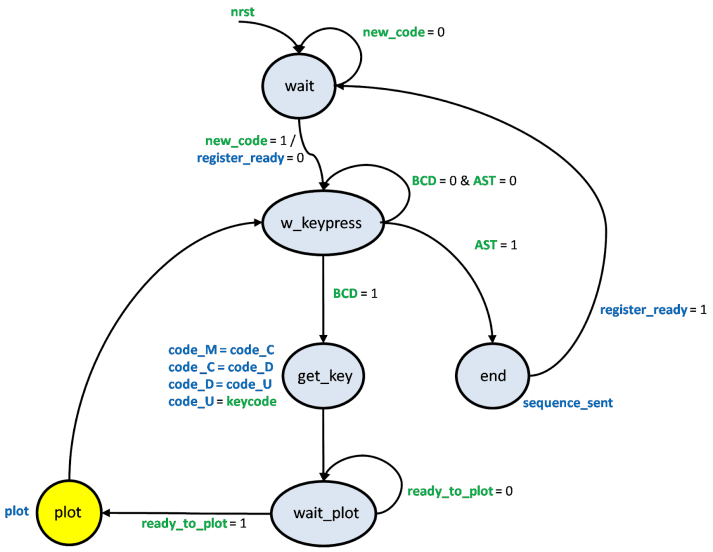
State Diagram of the Mastermind (Game) Block



Mastermind (Game) Block State Diagram: LINK to see with higher quality (please ask the students if the hyperlink is broken)

Again, states highlighted with an orange line are the ones where the VGA plots on screen what is indicated in the VGA_Game signal. Waiting states are named with letter W and a state ID. States that interact with the register are named with letter R and a state ID. States where something is plotted on screen (VGA plotting) are named with letter P and a state ID. The final state is named with letter F. Also, the name of the state where the secret sequence is compared to the guessing sequence is ‘C1’.

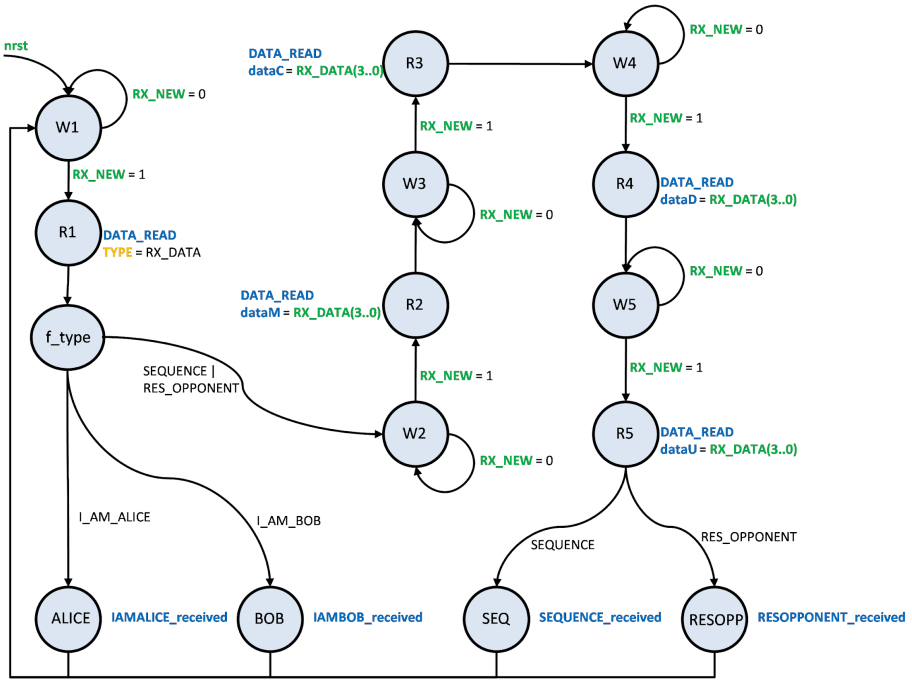
State Diagram of the Guessing Register



Guessing Register State Diagram: [LINK to see with higher quality] (please ask the students if the hyperlink is broken)

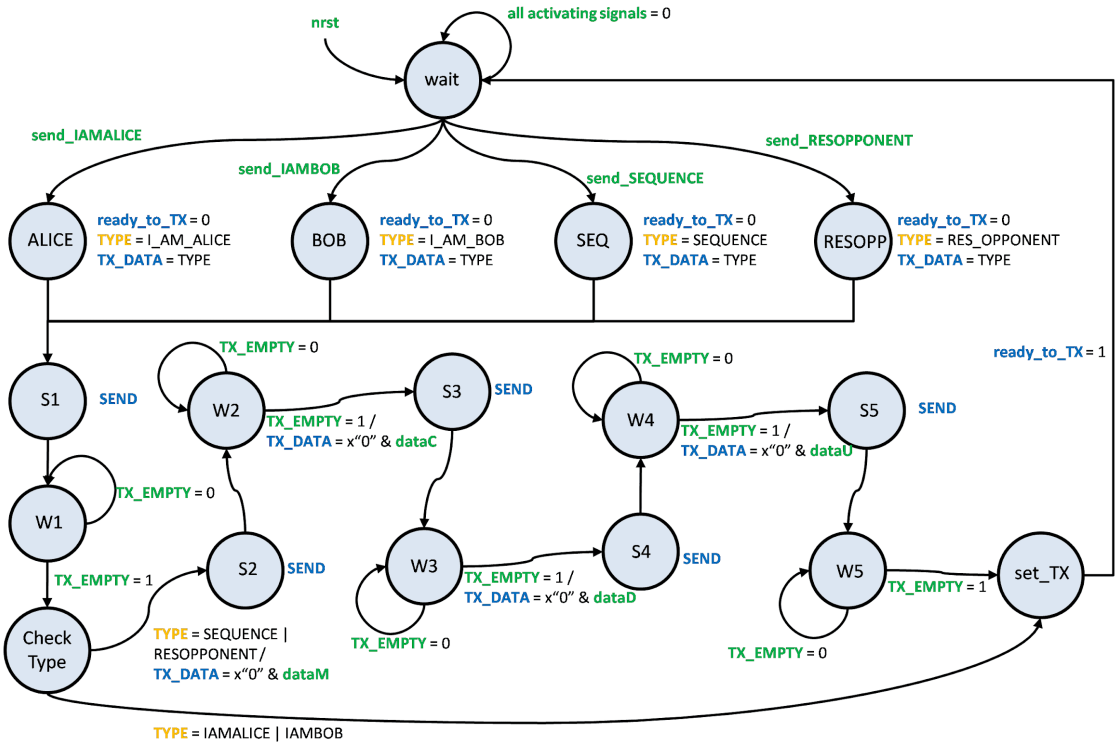
The state “plot”, filled with yellow, is the one where the VGA plots on screen what is indicated in the VGA_Game signal, which in this case will be the current guess.

State Diagram of the Receiver Block



Receiver Block State Diagram: LINK to see with higher quality (please ask the students if the hyperlink is broken)

State Diagram of the Transmitter Block



Transmitter Block State Diagram: LINK to see with higher quality (please ask the students if the hyperlink is broken)

State Diagram of the VGA Block

*The delivery of the VGA interpreter state diagram will be postponed, as a better understanding of the VGA blocks hardware is required.