ECE241H1
DIGITAL SYSTEMS

Final Report

Keychain Virtual Pet – DIGIMON

1999 version

James Production

This project is for the ECE241: Digital System, 2017Fall semester. All the design is implemented in DE1-SOC develop borad using Verilog. According to the course requirement, the project is finished by the group of three students, and all the code and the design are original. This is document is given in the strictest confidence and it is for the reader's knowledge only and it is not to be disclosed by the reader to any third party.

ECE241H1
DIGITAL SYSTEMS

Final Report

Keychain Virtual Pet – DIGIMON

1999 version

AT A GLANCE

The full working design is demostrated in lab session November 27. And this report is the supplement to the design demonstration

INTRODUCTION
The dsign goal and the brief description will be given

DESIGN
The Block Diagram will be provided to illustrate the function from hardware's perpective, and the Flow Chart of program as well

REPORT ON SUCCESS
The formal analysis on completeness

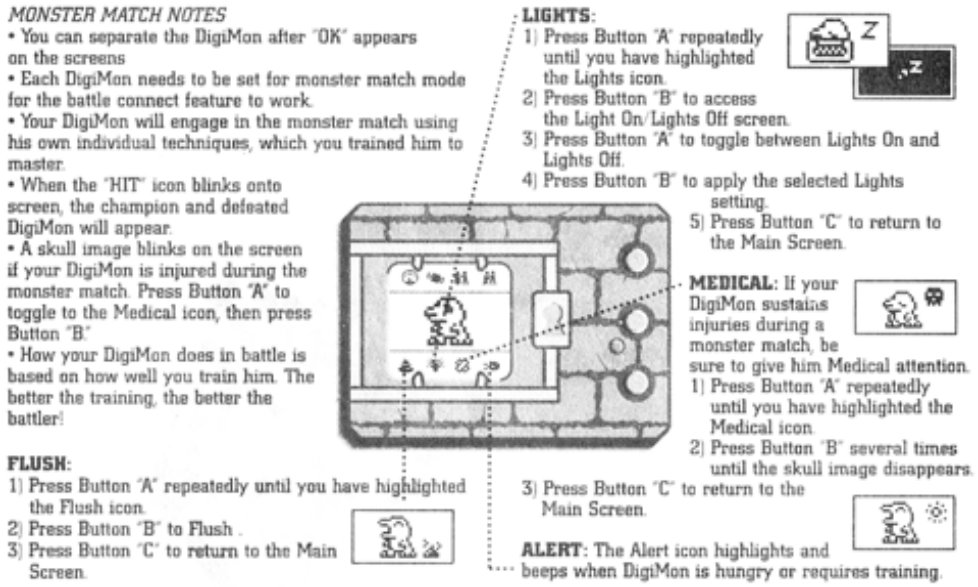
APPENDIX
The source code and the schematics

Arfa Tehreem, Yi Ashley Yuan, and Shuohe James Zhang

November 2017

Digital MonsterJPN is a Digital pet released in 1997 by Bandai. It is a virtual pet, cultivation game, for user to hatch a digital monster from egg to Baby stage, to Rookie, and to Champian stage.

As for the gamplay, the Digimon game consist of the following functions:



The original version contains many features: checking the pet's status, feeding the Digimon, training the Digimon, battle mode, cleaning up droppings, toggling the light, and healing the Digimon.

And our goal are listed as below:

- Use the video graphics adapter(VGA) as the output
- Design a Finite State Machine to control the front–end interface
- Design a Finite State Machine to store the back–end database
- Design a Finite State Machine to implement an animation effect of object's movement
- Other techniques including random access memory(RAM), register, shifter, counter, arithmetic logical unit(ALU), multiplexer, and etc, all the key concepts had been taught in ECE241.

DESIGN SPECIFICATION

1. Resolution
   320x240

2. Bits per channel
   8

3. Acutal display dimension
   64x64
   ps: The memory is limited, so the design utilizes an inner area to demonstrate the program with a smaller dimension.

4. Main pages
   3
   ps: The program includes the opening, the gaming page, and the ending page.

5. Functions
   3
   ps: Food, Cure, and Evole.

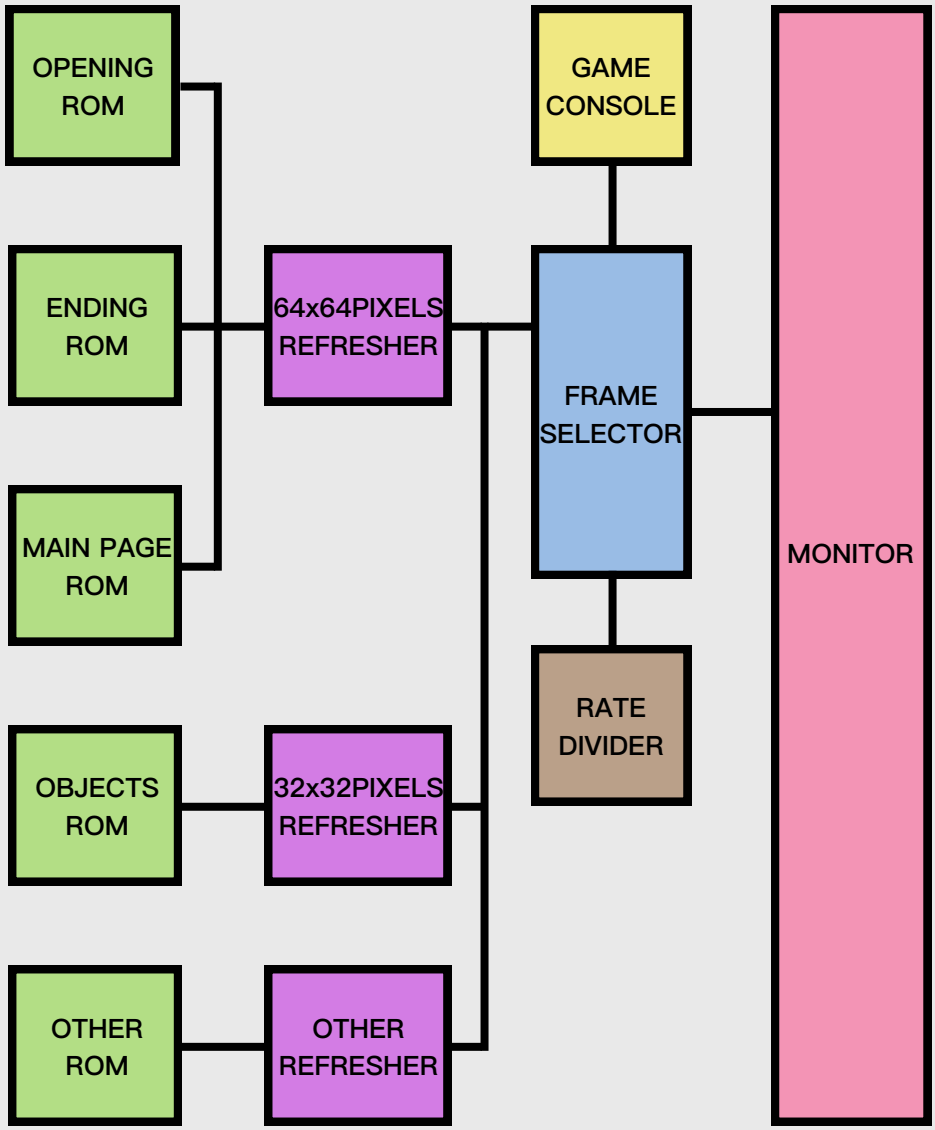
6. Stages
   4
   ps: Egg, Baby, and Rookie.

The program first shows a stop-motion animation as opening scene. After press the KEY[3], user can get into the main page of the game. The monster is hatched out of the egg, and user can feed it the food and the cure to let it grow. Once user tap the Evolve button, the monster can get to next stage.

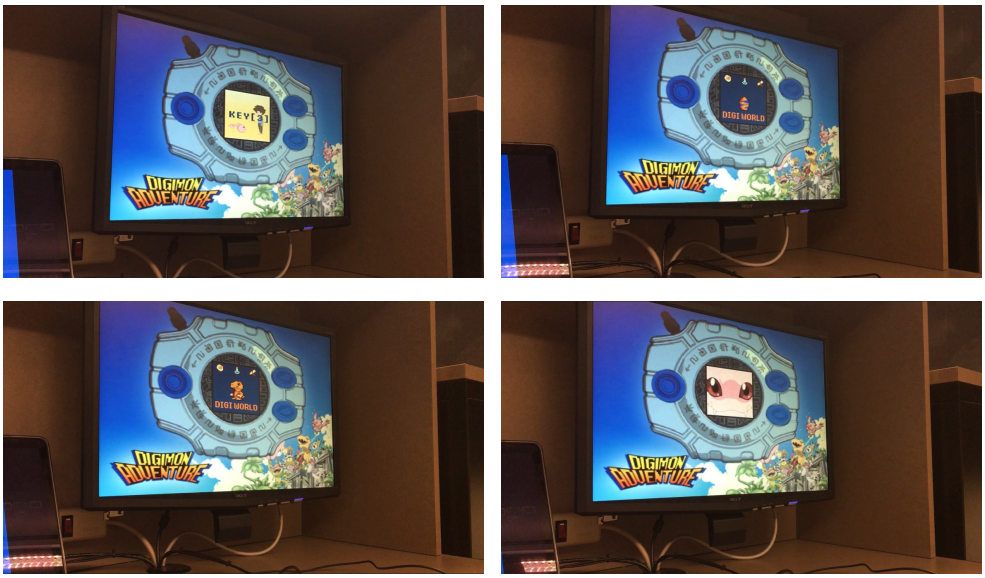
FLOW CHART

The secret to display the complicated gaming page, with several icons and objects on the screen, is the distributed display – display different frames on the screen and change from frame to frame in a high frequency. Moreover, the gaming logic is stored in the console.

BLOCK DIAGRAM



Report on success.



The program functions as designed, except for two things:

- Due to the memory limitation, only two stages of monster were stored in the ROM of the board
- Also eue to the memory limitation, the food and the cure image cannot be stored into the ROM. And the foor and the cure were displayed in HEX-display directly on the board.

Last thing, if we were going to start over, we shall control the frame number of animation. Such that, we will have enough memory space to demonstrate the whole function features, such as the apple eaten before and after.

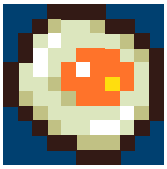
APPENDIX A
FOOTAGE

backgrounds

background.mif
320x240

inner_background
64x64

icons_fixed



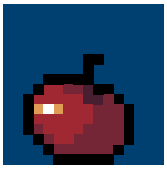
icon_food
11x11

icon_cure
11x11

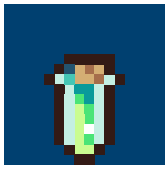
icon_seal
11x11

icon_footer
64x16

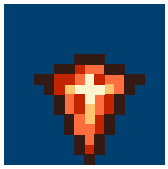
ovjects_dynamic



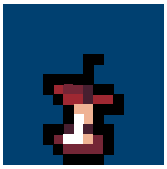
food_before
16x16



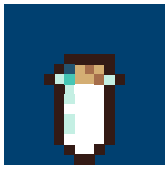
cure_before
16x16



seal
16x16



food_after
16x16



cure_after
16x16

cursor
11x4

monster_stages

egg_1
32x32

botamon_1
32x32

koromon_1
32x32

egg_2
32x32

botamon_2
32x32

koromon_2
32x32

monster_stages

agumon_right_1
32x32

agumon_left_1
32x32

agumon_right_2
32x32

agumon_left_2
32x32

evolutions

egg_evolve_1
32x32

egg_evolve_2
32x32

egg_evolve_3
32x32

egg_evolve_4
32x32

baby_evolve_1
32x32

baby_evolve_2
32x32

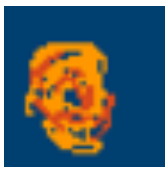
baby_evolve_3
32x32

baby_evolve_4
32x32

rookie_evolve_1
32x32

rookie_evolve_2
32x32

rookie_evolve_3
32x32



rookie_evolve_4
32x32

animation_scenes

opening_1
64x64

opening_2
64x64

opening_3
64x64

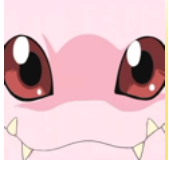
opening_4
64x64

opening_5
64x64

opening_6
64x64

opening_7
64x64

opening_8
64x64



ending_1
64x64

ending_2
64x64

ending_3
64x64