|  |  |
| --- | --- |
| **AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING**  **Computer and Systems Engineering**  **International Credit Hours Engineering Programs (i.CHEP)** |  |
| **Project Report**  **Design, Implementation and Test of**  **a Networks-on-Chip (NoC) Router using VHDL** | |

|  |  |  |
| --- | --- | --- |
| **Course Code**  CSE312 | **Course Name**  ElectronicDesignAutomation | |
| Dr. Haytham Azmi | **Semester**  Fall 2020 | **Date of Submission**  9/1/2021 |

Abdel-Rahman Ibrahim Megahed 18P7423

Maryam Ahmed Nouh 18P2824

Seif Ahmed Elewa 18P5662

Youssef Mohamed Mansi 18P5848

Contents

[Introduction 5](#_Toc61023023)

[Design Flow 6](#_Toc61023024)

[Dataflow Steps: 6](#_Toc61023025)

[Router Subsystems: 7](#_Toc61023026)

[Literature Review 8](#_Toc61023027)

[Design Implementation 9](#_Toc61023028)

[Block Diagram 9](#_Toc61023029)

[Design Modules 9](#_Toc61023030)

[Router 10](#_Toc61023031)

[Reports 12](#_Toc61023032)

[Register 12](#_Toc61023033)

[Demultiplexer 12](#_Toc61023034)

[RAM 13](#_Toc61023035)

[Gray Counter 14](#_Toc61023036)

[Grey to binary 15](#_Toc61023037)

[Round Robin 16](#_Toc61023038)

[FIFO 16](#_Toc61023039)

[Timing Analysis 18](#_Toc61023040)

[Register 18](#_Toc61023041)

[Demux 18](#_Toc61023042)

[RAM 18](#_Toc61023043)

[Gray Counter 19](#_Toc61023044)

[Gray to Binary 19](#_Toc61023045)

[Round Robin 20](#_Toc61023046)

[FIFO 20](#_Toc61023047)

[Schedular Design and FSM implementation 23](#_Toc61023048)

[Test and Simulation Results 25](#_Toc61023049)

[M-ROU-01 25](#_Toc61023050)

[M-ROU-02 25](#_Toc61023051)

[M-ROU-03 26](#_Toc61023052)

[M-ROU-04 27](#_Toc61023053)

[M-ROU-05 27](#_Toc61023054)

[M-ROU-06 28](#_Toc61023055)

[M-ROU-07 29](#_Toc61023056)

[M-ROU-08 29](#_Toc61023057)

[M-ROU-10 30](#_Toc61023058)

[Conclusion 31](#_Toc61023059)

[Task Distribution 32](#_Toc61023060)

[RTL Schematic 33](#_Toc61023061)

[Register 33](#_Toc61023062)

[Demultiplexer 34](#_Toc61023063)

[RAM 35](#_Toc61023064)

[Grey Generator 36](#_Toc61023065)

[Grey To Binary 36](#_Toc61023066)

[FIFO 37](#_Toc61023067)

[Round Robin 38](#_Toc61023068)

[Router 39](#_Toc61023069)

[APPENDIX A – VHDL Codes 40](#_Toc61023070)

[Register (M-ROU-01): 40](#_Toc61023071)

[Demultiplexer (M-ROU-02): 40](#_Toc61023072)

[Block RAM (M-ROU-03): 41](#_Toc61023073)

[Grey Counter (M-ROU-04): 42](#_Toc61023074)

[Grey to Binary (M-ROU-05): 43](#_Toc61023075)

[FIFO Controller (M-ROU-06): 43](#_Toc61023076)

[FIFO (M-ROU-07): 45](#_Toc61023077)

[Round Robin Scheduler (M-ROU-08): 46](#_Toc61023078)

[Router (M-ROU-09): 47](#_Toc61023079)

[APPENDIX B – VHDL Testbenches 52](#_Toc61023080)

[Register Testbench (M-ROU-01): 52](#_Toc61023081)

[Demultiplexer Testbench (M-ROU-02): 53](#_Toc61023082)

[Block RAM Testbench (M-ROU-03): 54](#_Toc61023083)

[Grey Counter testbench (M-ROU-04): 56](#_Toc61023084)

[Grey to Binary Testbench (M-ROU-05): 57](#_Toc61023085)

[FIFO Controller Testbench (M-ROU-06): 58](#_Toc61023086)

[FIFO Testbench (M-ROU-07): 61](#_Toc61023087)

[Round Robin Scheduler Testbench(M-ROU-08): 67](#_Toc61023088)

[Router Testbench (M-ROU-10): 68](#_Toc61023089)

[APPENDIX C – Waveform Screenshots 74](#_Toc61023090)

[Register Waveform (M-ROU-01): 74](#_Toc61023091)

[Demultiplexer Waveform (M-ROU-02): 74](#_Toc61023092)

[Block RAM Waveform (M-ROU-03): 75](#_Toc61023093)

[Grey Counter Waveform (M-ROU-04): 75](#_Toc61023094)

[Grey to binary Waveform (M-ROU-05): 76](#_Toc61023095)

[FIFO Controller Waveform (M-ROU-06): 76](#_Toc61023096)

[FIFO Waveform (M-ROU-07): 77](#_Toc61023097)

[Round Robin Scheduler Waveform (M-ROU-08): 77](#_Toc61023098)

[Router Waveform (M-ROU-10): 78](#_Toc61023099)

[References 79](#_Toc61023100)

# Introduction

The following report describes the full implementation of the NOC using the router module, the report contains the design flow of the module, the background of the implemented module, the code and the tests conducted of each submodule. It also contains the block diagram and the wave form of each module. The main building blocks of our routers are the input buffer , switch fabric , the output buffer, and their controller.

The router main function is to take input of 8 bits and routs it to different output depending on the address consisting of the last 2 bits of the input while having the ability to reset, write data, and read data. It is formed from 8 modules split into 4 different subsystems that group certain modules together. Our objective is to see the router take various inputs through different ports and routing the inputs to their correct output port according to the address (last 2 bits of the input) without any errors or delays in the process.

We need the NOC router because it helps support the high communication requirements of many modern architectures where they include routers and links to enable the communication on-chip. The NOCs provide parallel communication with high bandwidth and modularity. Their high communication bandwidth and processing power helps the designers implement intensive computation functions easily. Example for the application of the NOC in the modern technology is the drone application which will need flight control software along with the computational strength to process images and videos.

The router is not only limited to the NOC it has various types which we can see around us such as the wired router which has one connection port to connect the router to the modem while another set of ports that connect the router to the different devices. There is also the wireless router used in most offices spaces where the output ports don’t exist but instead it transmits the data to the devices wirelessly and usually digitally. There are also other types of routers that are not widely used such as the core router, edge router and virtual router.

This chip design is built according to specifications using the ModelSim for first design steps and checking the syntax along with the functionality of each module. After successfully completing this step we will conduct the second step of applying the design to the ISE Xilinx where we will check our timing report and the design specification along with the RTL schematic.

# Design Flow

For this project we have chosen FPGA over ASIC because the chip is programmable in the field as it is a set of gates and other sequential elements combined in configurable logic blocks. Although it has some limitations compared to ASIC but it has more advantages as the setup cost is lower and the testing becomes easier as a lot of tools are provided by the chip manufacturer to help automate the tests but for ASIC the chip is customized and the design flow is very specific for a target technology and requires more knowledge in the field and ASIC is more oriented toward mass production while FPGA is for smaller production. FPGA is configured according to the software application so that the internal circuit is connected in a technique to create a hardware implementation of that software. FPGA based systems can rewire their internal logic circuit to allow configuration after the deployment of the control system to the field. FPGA can substitute thousands of components by integrating millions of logic gates in an IC chip.

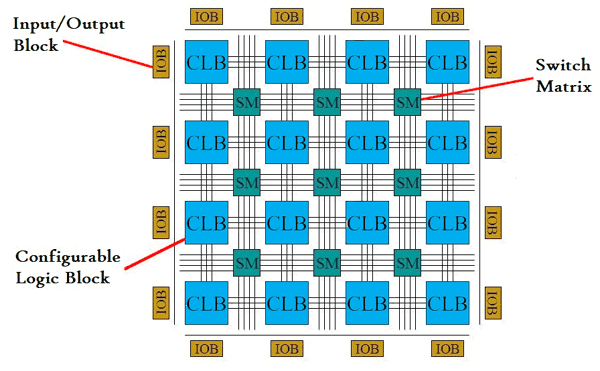


Figure 1: FPGA Module

Configurable Logic Block (CLB): The basic repeating logic resource on FPGA. It executes complex logic functions, synchronize code, and implement memory functions.

Configurable I/O Block: Composed of an input buffer and an output buffer with a three and an output buffer. It manages the signals in and out of the chip.

Switch Fabric: It interconnects the gates and the CLB together and manages the signals within the chip.

## Dataflow Steps:

* Design: There are multiple ways to illustrate the design. We have chosen to demonstrate the design as VHDL code using MODELSIM. All the modules were created and found to be fully functional.
* Simulation: The simulation was done by creating testbenches for each module and verifying they are working separately and generating their waveforms and generating the expected results according to the tables using MODELSIM.
* Synthesis: The modules were synthesized using Xilinix ISE generate the gate level netlist and reports finding the timing analysis was correct and there were no errors found for the desired technology and the reports were generated correctly
* Layout and Manufacturing: This step could not be achieved for this project. It is used to show the masking and the routing between the modules of the chip after manufacturing and checking all the connections are functioning correctly.

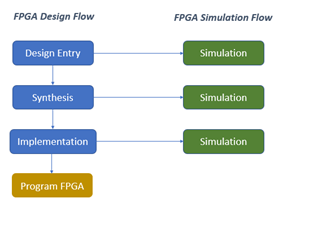


Figure 2: Dataflow Steps

## Router Subsystems:

The router consists of four main modules input buffer (4 registers), switch fabric (4 demultiplexers), output queue (16 FIFO), and output buffer (4 round robin schedulers). The input buffer simply is a register consists of 8 flip-flops. The switch fabric provides full connectivity between the inputs and outputs. Output queue is implemented as FIFO buffers. The output buffer is just like the input buffer consisting of 8 flip-flops.

The input buffer is a temporary memory that stores the packet before sending to the router according to its 2 least significant bits that reference to the correct output port. The demultiplexer takes the 2 least significant bits as input to select the correct output port from the available four to send the data to the FIFO.

The FIFO controller and RAM block are connected in the FIFO. The two modules take the inputs from the router as clocks to commence their job.

The RAM block is used to store the data in the form of arrays. Multiple reads or writes can happen concurrently. The FIFO controller manages when to read or write from the RAM block by sending read and write signals by integrating the Gray counter and Gray to Binary converter with each other; The Gray counter counts from 0 to 7 and then converted by the Gray to Binary converter that allows the controller to point to the location where it is going to read from or write into. The FIFO controller is also responsible for determining whether the RAM block is empty or full.

The round robin module uses the round robin algorithm, where each clock cycle it takes a new input. There are 16 instances of the FIFO that yield 8-bit data inputs that are directed to four instances of the round robin, where it gives the router four packets as a final output.

# Literature Review

Recently, there have been many advancements in hardware technologies. These advancements enabled designers to add several various transistors on a single chip, in turn creating complex systems on chips (SoC). This study ‎1 shows how the Quality-of-Service Network on Chip (QNoC) produces a power efficient SoC and low latency transfers for an asynchronous NoC router. The router is based on the Speed Independent Transition Graph model and is implemented in 0.35 µm CMOS technology.

Advancements in chip technology, where many cores are integrated into one chip, have been aided by Very-Large-Scale Integration (VSLI) as stated by the study ‎‎2. This paper uses a novel router that is dynamically reconfigurable on chip-systems. On runtime, the router is able to detect errors in packet forwarding. There is another router designed with an error detection system for adaptive networks.

This paper ‎‎3 showcases the effect of reducing the network occupancy of the routers to a minimum in turn, making the logic use area on the chip efficiently. This allows power consumption to be reduced. The router implemented in the paper is a parallel router that supports five simultaneous routing requests. XY routing and decoding logic optimizations are also introduced.

This paper ‎‎4 designs and implements a configurable, high speed router. The router’s buffer slots are allocated dynamically, which leads to an increase in productivity when the network deals with heavy loads. The depth of each input buffer is chosen at design time. The chip consumes 5mW of power. The timing constraints have been enhanced to make the router high speed.

The following paper ‎5 combines two different router architecture concepts: the CDVC and EDVC. Also, an innovative writing mechanism is being utilized. The architecture also uses latch-based pipelining and a Clock gating technique to increase the operating frequency of the system whilst maintaining a relatively small power consumption. The result of using this router model achieved a power consumption decrease of 47% and 16% with respect to the CDVC and EDVC architectures, respectively.

# Design Implementation

## Block Diagram

Figure 3: Full Block Diagram of the router

## Design Modules

|  |  |
| --- | --- |
| **Module** | **Function** |
| **M-ROU-01**  **8-Bit register** | This module is used to store data and instructions in and out of the router. The Register was simply implemented using one process. If the clock enable has a value of one and the clock has a rising edge, the data is stored in the register, otherwise, data remains as its last value. If reset has a value of one, the output value is “0000 0000”. |
| **M-ROU-02**  **1x2² 8-bit demultiplexer** | This module is responsible for connecting each input register coming from the input buffer to its corresponding output queue. It was implemented with one process, inside which the enable is checked to have a value of one, if that is true, depending on the selector, the input data will be displayed on the corresponding output. When enable has a value of zero, the output data does not change. |
| **M-ROU-03**  **Block RAM** | This module is used as a memory component to store data with a size of eight bits, while allowing reading and writing having Dual-Port to allow the data to be access according to different addresses input. It initiates the writing or reading operation based on both their clock and the clock of the port that will indicate the accessed address. |
| **M-ROU-04**  **Grey Counter** | This module is used as an incremental counter (in ascending order) made of four bits. It is implemented using a process with a switch case, that checks the current value of the counter, and depending on that value, it enters a specific branch of the case, where it increases its value by one in grey code. This is done with a signal that updates the actual counter at the end of the process. Once the counter reached its maximum, equivalent to the number seven, it resets automatically to zero. |
| **M-ROU-05**  **Grey to Binary Converter** | This module takes the counter implemented in the previous module and converts it into binary code. This component was implemented via simple RTL. The most significant bit is the same in the conversion. The second is the result of an exclusive OR operation between the first and second bits of the grey counter. The third bit is the result of an exclusive OR operation between the first, second and third bits of the grey counter. Lastly, the least significant bit is the result of an exclusive OR operation between all bits of the grey counter. |
| **M-ROU-06**  **FIFO Controller** | This module is one of the main components in the router, as it is responsible for verifying any read or write requests. It manages the validity of the write and read operations based on the clock, request and the empty/full flags to avoid loss of data and unexpected errors. To set the write validity to high, the write request must be high, the full flag must be low and write clock must be high. Meanwhile, to set the read validity to high, the empty flag must be low, the read request must be high, and the read clock must be high. |
| **M-ROU-07**  **FIFO** | This module is responsible for creating a queue that follows the first in first out strategy to organize the data in the RAM with the help of the FIFO controller. It helps organize the data input and output operations based on the requests and the clock. |
| **M-ROU-08**  **Round Robin Scheduler** | This module implements the round robin algorithm to schedule the inputs and taking a new input every clock cycle. This component was implemented via a finite state machine, discussed precisely in the next section. |
| **M-ROU-09**  **Router** | This module encompasses all the previous components by taking an input of specific size and connecting it to its corresponding output, depending on the address. This component represents the main objective of this project. Its implementation is discussed in detail below. |

## Router

The Router is mainly made up of four different parts, which work jointly to form the router. The first component is the input buffer, which consists of registers (M-ROU-01). The second part of the router is made up of demultiplexers (M-ROU-02), forming the switch fabric. Following that is the output queues, containing numerous FIFOs (M-ROU-07). Lastly, there’s an output buffer, which comprises the Round Robin schedulers (M-ROU-08). The router has four output ports and four corresponding input ports of a fixed size of 8 bits, with a packet availability indicator, along with a reset and synchronization clocks for reading and writing signals. Assuming that read requests and write requests come from outside the router, those two signals were added.

The first step is marked as the data entering the input buffer region. Upon data arrival at one of the input ports, the corresponding packet availability indicator, which is simply a bit, changes its value from 0 to 1. This availability indicator remains high for one clock cycle, during which, the relative register stores this data, and at its turn, at a rising clock edge, feeding it to the demultiplexer. However, the transition of the data to the register will only take place if, and only if, the clock enable of that register is high. This ensures that packets are inserted in the router when they are labeled as ready, via the signals wr1, wr2, wr3 or wr4.

Moving on to the next part of the router, which is the switch fabric, holding the demultiplexers. The demultiplexers are implemented without a clock, but to compensate this, an enable is added to the module. Data moves from the register to its corresponding demultiplexer as data input, while extracting the 2 least significant bits (last two bits in each 8-bit data) serving as the address in which the input data will be stored. Data is displayed as the output of the demultiplexer only when the enable is high. This enable needs to be high when a packet arrives and is ready as input for the router. For that reason, as demonstrated in the block diagram above, the packet availability indicator serves also as the enable for the demultiplexer. The demultiplexer’s role is to route each data input to a FIFO queue, depending on the data input’s least significant bit, which is the group of FIFOs. Meaning that given a data input such as “1111 1100”, the least significant bits that are extracted are “00”; thus, this data input is associated to the first group of FIFOs. Similarly, the data input “1011 1001” is routed to the second output port, which will be distributed to the second group of FIFO queues.

The router’s third part, labeled as the output queue, consists of sixteen FIFOs, where each FIFO has a read clock (for the input) and a write clock (for the output) – assuming that no read and write requests can coincide. Moreover, every consecutive four FIFOs are grouped together, and each demultiplexer has one of its outputs connected to one of the FIFO groups. The index of the demultiplexer indicates the corresponding FIFO in each group. To demonstrate this, the first output of the first demultiplexer is mapped to the first FIFO, the second to the fifth, the third to the ninth or the fourth to the thirteenth, which is visible in the block diagram attached above. Therefore, the package delivered to the outer is inserted to the correct output queue. Moreover, concerning the read and write requests, to read, an AND operation is performed on the read request and the empty flag coming from the FIFO, inverted, while to write, an AND operation is performed on the write request and the full flag coming from the FIFO, inverted. That is done for each of the sixteen FIFOs in the output queue. However, the first in first out scheduling algorithm requires that the input data waits for its turn, which means waiting for all the packets that arrived first are sent to the next part of the router.

The last part of the router is the output buffer, which holds Round Robin schedulers. This scheduler’s role is to take four data inputs from particular FIFO queues and display one on each clock cycle, in order. This is done using a finite state machine, discussed in detail below. The output of this module is linked to the output of the router.

# Reports

## Register

**HDL Synthesis Report**

Macro Statistics

# Registers : 2

8-bit register : 2

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 16 out of 126800 0%

Number of Slice LUTs: 1 out of 63400 0%

Number used as Logic: 1 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 17

Number with an unused Flip Flop: 1 out of 17 5%

Number with an unused LUT: 16 out of 17 94%

Number of fully used LUT-FF pairs: 0 out of 17 0%

Number of unique control sets: 2

**IO Utilization:**

Number of IOs: 19

Number of bonded IOBs: 19 out of 210 9%

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 1 out of 32 3%

## Demultiplexer

**HDL Synthesis Report**

Macro Statistics

# Latches : 32

1-bit latch : 32

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice LUTs: 4 out of 63400 0%

Number used as Logic: 4 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 4

Number with an unused Flip Flop: 4 out of 4 100%

Number with an unused LUT: 0 out of 4 0%

Number of fully used LUT-FF pairs: 0 out of 4 0%

Number of unique control sets: 4

**IO Utilization:**

Number of IOs: 43

Number of bonded IOBs: 43 out of 210 20%

IOB Flip Flops/Latches: 32

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 1 out of 32 3%

## RAM

**HDL Synthesis Report**

Macro Statistics

# Latches : 72

1-bit latch : 72

# Multiplexers : 8

1-bit 8-to-1 multiplexer : 8

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 64 out of 126800 0%

Number of Slice LUTs: 25 out of 63400 0%

Number used as Logic: 25 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 80

Number with an unused Flip Flop: 16 out of 80 20%

Number with an unused LUT: 55 out of 80 68%

Number of fully used LUT-FF pairs: 9 out of 80 11%

Number of unique control sets: 9

**IO Utilization:**

Number of IOs: 28

Number of bonded IOBs: 26 out of 210 12%

IOB Flip Flops/Latches: 8

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 2 out of 32 6%

## Gray Counter

**HDL Synthesis Report**

Macro Statistics

# Adders/Subtractors : 1

4-bit adder : 1

# Registers : 1

4-bit register : 1

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 4 out of 126800 0%

Number of Slice LUTs: 4 out of 63400 0%

Number used as Logic: 4 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 8

Number with an unused Flip Flop: 4 out of 8 50%

Number with an unused LUT: 4 out of 8 50%

Number of fully used LUT-FF pairs: 0 out of 8 0%

Number of unique control sets: 1

**IO Utilization:**

Number of IOs: 7

Number of bonded IOBs: 7 out of 210 3%

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 1 out of 32 3%

## Grey to binary

**HDL Synthesis Report**

Macro Statistics

# Xors : 3

1-bit xor2 : 3

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice LUTs: 3 out of 63400 0%

Number used as Logic: 3 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 3

Number with an unused Flip Flop: 3 out of 3 100%

Number with an unused LUT: 0 out of 3 0%

Number of fully used LUT-FF pairs: 0 out of 3 0%

Number of unique control sets: 0

**IO Utilization:**

Number of IOs: 8

Number of bonded IOBs: 8 out of 210 3%

## Round Robin

**HDL Synthesis Report**

Macro Statistics

# Multiplexers : 1

8-bit 4-to-1 multiplexer : 1

# FSMs : 1

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 2 out of 126800 0%

Number of Slice LUTs: 10 out of 63400 0%

Number used as Logic: 10 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 10

Number with an unused Flip Flop: 8 out of 10 80%

Number with an unused LUT: 0 out of 10 0%

Number of fully used LUT-FF pairs: 2 out of 10 20%

Number of unique control sets: 1

**IO Utilization:**

Number of IOs: 41

Number of bonded IOBs: 41 out of 210 19%

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 1 out of 32 3%

## FIFO

**HDL Synthesis Report**

Macro Statistics

# Adders/Subtractors : 1

32-bit adder : 1

# Latches : 106

1-bit latch : 106

# Multiplexers : 11

1-bit 2-to-1 multiplexer : 2

1-bit 8-to-1 multiplexer : 8

32-bit 2-to-1 multiplexer : 1

# FSMs : 2

# Xors : 6

1-bit xor2 : 6

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 104 out of 126800 0%

Number of Slice LUTs: 117 out of 63400 0%

Number used as Logic: 117 out of 63400 0%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 173

Number with an unused Flip Flop: 69 out of 173 39%

Number with an unused LUT: 56 out of 173 32%

Number of fully used LUT-FF pairs: 48 out of 173 27%

Number of unique control sets: 14

**IO Utilization:**

Number of IOs: 23

Number of bonded IOBs: 23 out of 210 10%

IOB Flip Flops/Latches: 10

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 4 out of 32 12%

# Timing Analysis

## Register

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

clock | BUFGP | 16 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: 0.648ns (Maximum Frequency: 1542.496MHz)

Minimum input arrival time before clock: 0.837ns

Maximum output required time after clock: 0.640ns

Maximum combinational path delay: No path found

## Demux

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

en | BUFGP | 32 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: No path found

Minimum input arrival time before clock: 0.898ns

Maximum output required time after clock: 0.751ns

Maximum combinational path delay: No path found

## RAM

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

WEA\_CLKA\_AND\_1\_o(WEA\_CLKA\_AND\_1\_o1:O)| BUFG(\*)(RAMArr<1>\_6) | 64 |

REA | BUFGP | 8 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: No path found

Minimum input arrival time before clock: 1.139ns

Maximum output required time after clock: 0.751ns

Maximum combinational path delay: No path found

## Gray Counter

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

CLK | BUFGP | 4 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: 1.059ns (Maximum Frequency: 943.842MHz)

Minimum input arrival time before clock: 0.643ns

Maximum output required time after clock: 0.659ns

Maximum combinational path delay: No path found

## Gray to Binary

**Timing Report**

**Clock Information:**

No clock signals found in this design

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: No path found

Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: 0.934ns

## Round Robin

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

clk | BUFGP | 2 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: 1.082ns (Maximum Frequency: 924.300MHz)

Minimum input arrival time before clock: No path found

Maximum output required time after clock: 1.472ns

Maximum combinational path delay: 0.934ns

## FIFO

**Timing Report**

**Clock Information:**

Clock Signal | Clock buffer(FF name) | Load |

Rd(FIFO\_ctrlr/Mmux\_rd\_valid11:O) | NONE(\*)(ramemory/temp\_1) | 8 |

ramemory/WEA\_CLKA\_AND\_7\_o(ramemory/WEA\_CLKA\_AND\_7\_o1:O)| BUFG(\*)(ramemory/RAMArr<0>\_7)| 64 |

wclk | IBUF+BUFG | 4 |

rclk | IBUF+BUFG | 4 |

wreq | IBUF+BUFG | 32 |

FIFO\_ctrlr/empty\_G(FIFO\_ctrlr/empty\_G:O) | FIFO\_ctrlr/empty) | 1 |

FIFO\_ctrlr/full\_G(FIFO\_ctrlr/full\_G:O) | (FIFO\_ctrlr/full) | 1 |

**Asynchronous Control Signals Information:**

No asynchronous control signals found in this design

**Timing Summary:**

Speed Grade: -3

Minimum period: 2.679ns (Maximum Frequency: 373.218MHz)

Minimum input arrival time before clock: 0.808ns

Maximum output required time after clock: 0.751ns

Maximum combinational path delay: No path found

**HDL Synthesis Report**

Macro Statistics

# Adders/Subtractors : 16

32-bit adder : 16

# Registers : 8

8-bit register : 8

# Latches : 1824

1-bit latch : 1824

# Multiplexers : 180

1-bit 2-to-1 multiplexer : 32

1-bit 8-to-1 multiplexer : 128

32-bit 2-to-1 multiplexer : 16

8-bit 4-to-1 multiplexer : 4

# FSMs : 36

# Xors : 96

1-bit xor2 : 96

**Device utilization summary:**

Selected Device : 7a100tcsg324-3

**Slice Logic Utilization:**

Number of Slice Registers: 2018 out of 126800 1%

Number of Slice LUTs: 1939 out of 63400 3%

Number used as Logic: 1939 out of 63400 3%

**Slice Logic Distribution:**

Number of LUT Flip Flop pairs used: 2998

Number with an unused Flip Flop: 980 out of 2998 32%

Number with an unused LUT: 1059 out of 2998 35%

Number of fully used LUT-FF pairs: 959 out of 2998 31%

Number of unique control sets: 216

**IO Utilization:**

Number of IOs: 73

Number of bonded IOBs: 73 out of 210 34%

**Specific Feature Utilization:**

Number of BUFG/BUFGCTRLs: 32 out of 32 100%

**Timing Summary:**

Speed Grade: -3

Minimum period: 2.679ns (Maximum Frequency: 373.218MHz)

Minimum input arrival time before clock: 1.061ns

Maximum output required time after clock: 1.538ns

Maximum combinational path delay: No path found

## Schedular Design and FSM implementation

This section focuses on our implementation and design of the schedular. We designed a round robin algorithm for the schedular to use which was implemented in a Moore finite state machine (FSM) design.

An FSM is a designing methodology used to model problems in various fields including mathematics, AI, linguistics, and games. The FSMs are composed of a finite set of states, where in each state the machine behaves in a specified way. FSMs are divided into two kinds: Mealy, and Moore. In Mealy machines, the machine determines their output depending on its current state and the value being inputted into the machine. On the other hand, Moore machines determine their output based only on their current state.

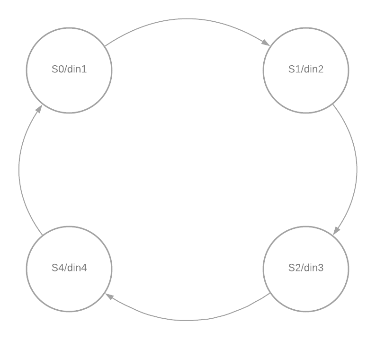


Figure 4: The State Diagram

In our implementation, we used a Moore machine. This is due to the lack of need of input to the machine. In round robin scheduling, the set of data to be scheduled is given a certain quantum (the clock period); therefore, making no input needed. Generally, most FSMs have a reset to set the machine to a known state as a start, but this is not needed as the states are all connected into a cycle and each state leads to one and only one state. This is shown in Figure 4.

We developed the FSM by using three processes in our VHDL code. This does not affect the logic or output of the schedular; it only affects the code’s readability and ease of access. The code’s readability varies with the number of processes as if we use one process, it will be responsible for all the FSM functionalities. On the other hand, if we use two processes, the responsibilities are divided on both processes and so on. The following table showcases each process, and their sensitivity lists for using one, two, or three processes while coding. The sensitivity list for each process coding style for Moore Machines is shown in Table 1.

Table 1: Each Process coding style with their sensitivity list for Moore Machines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number of Processes | One | Two | | Three | | |
| Process Number | P1 | P1 | P2 | P1 | P2 | P3 |
| Sensitivity List (for Moore Machines) | Clock | Clock | Current State | Clock | Current State | Current State |
| Current State | Inputs | Inputs |
| Process Responsibilities | Update Current State | Update Current State |  | Update Current State |  |  |
| Determine Next State |  | Determine Next State |  | Determine Next State |  |
| Determine Outputs |  | Determine Outputs |  |  | Determine Outputs |

The following schematic view diagram shows the synthesized circuit from a top level. As we can see, the input entering a four by one multiplexer where the selector is the output of a counter block with a maximum output of “11” where the counter increments every clock cycle.

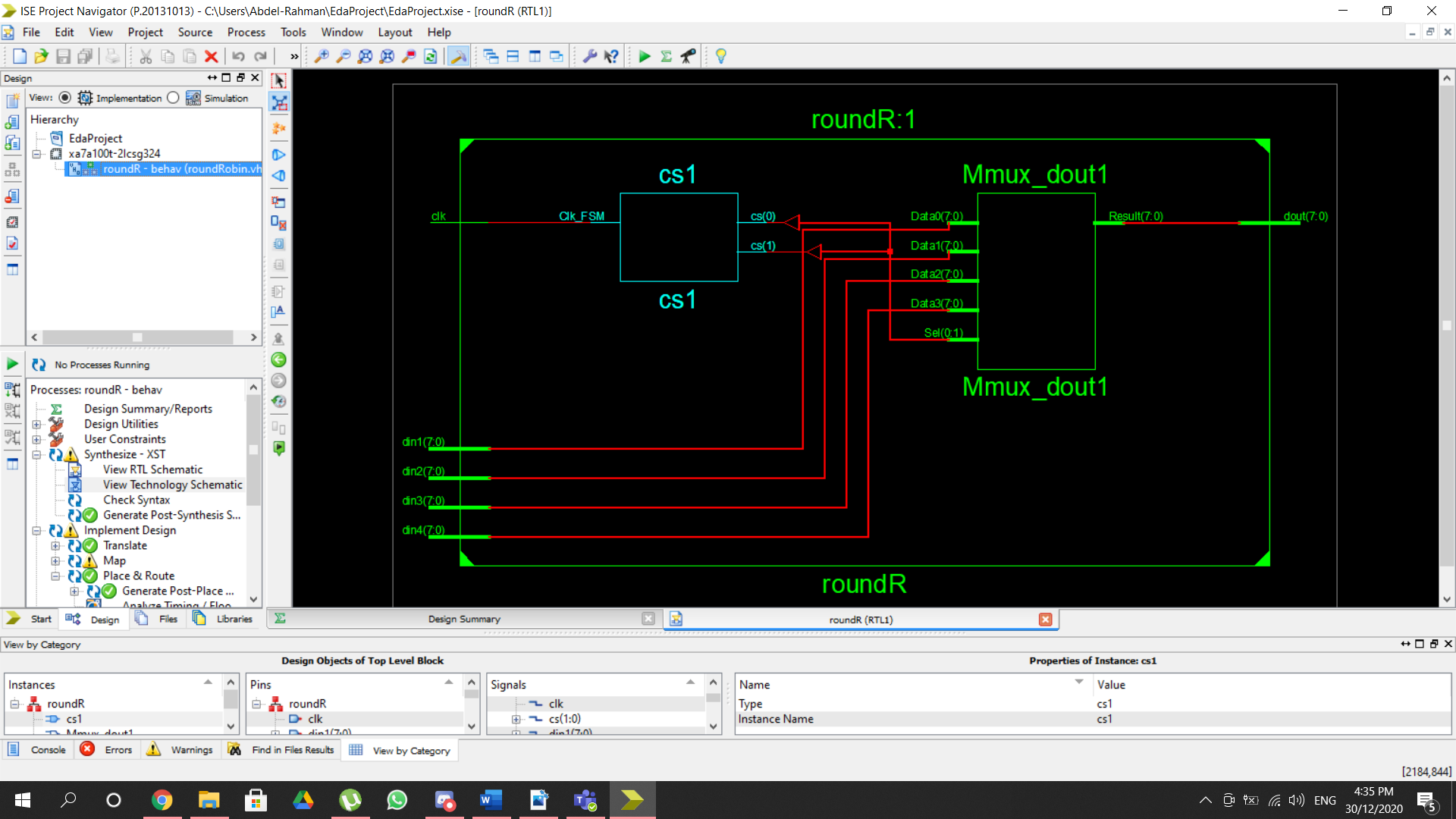


Figure 5: The Schematic View For the Round Robin Schedular

# Test and Simulation Results

## M-ROU-01

Table 2: Test Strategy for the 8-Bit register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tested Feature** | **Clock\_en** | **Reset** | **Data\_in** | **Clock** | **Delay** | **Data\_out** |
| Reset | 0 | 1 | 1111 1111 | 0 | 20ns | 0000 0000 |
| Asynchronous Reset is not affected by the clock | 0 | 1 | 1111 1111 | 1 | 20ns | 0000 0000 |
| Output Hold if the clock has a value of “0” | 1 | 0 | 1111 1111 | 0 | 20ns | 0000 0000 |
| Load when enable has a value of “1” | 1 | 0 | 1111 1111 | 1 | 20ns | 1111 1111 |
| Output Hold check when clock has a value of “0” | 0 | 0 | 1111 1111 | 0 | 20ns | 1111 1111 |
| Output Hold check when enable has a value of “0” | 0 | 0 | 1111 1111 | 1 | 20ns | 1111 1111 |

## M-ROU-02

Table 3: Test Strategy for the 8-Bit Demux

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tested Feature** | **D\_in** | **Sel** | **En** | **Delay** | **D\_out1** | **D\_out2** | **D\_out3** | **D\_out4** |
| First value | 0000 0001 | 00 | 1 | 20ns | 0000 0001 | - | - | - |
| Second value | 0000 0010 | 01 | 1 | 20ns | 0000 0001 | 0000 0010 | - | - |
| Third value | 0000 0011 | 10 | 1 | 20ns | 0000 0001 | 0000 0010 | 0000 0011 | - |
| Fourth value | 0000 0100 | 11 | 1 | 20ns | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 |
| Hold | 0000 0100 | 11 | 0 | 20ns | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 |

## 

## M-ROU-03

Table 4: Test Strategy for the Block RAM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tested Feature** | **Clka** | **Clkb** | **Wea** | **Rea** | **Addra** | **Addrb** | **D\_in** | **Delay** | **D\_out** |
| Default output | 0 | 0 | 0 | 0 | - | - | - | 20ns | 0000 0000 |
| A stores input data in address 0100 | 1 | 0 | 1 | 0 | 0100 | 0000 | 0000 0001 | 20ns | 0000 0000 |
| Output holds | 0 | 0 | 0 | 0 | 0100 | 0000 | 0000 0001 | 20ns | 0000 0000 |
| A stores input data in address 1000 | 1 | 0 | 1 | 0 | 1000 | 0000 | 0000 0011 | 20ns | 0000 0000 |
| Output holds | 0 | 0 | 0 | 0 | 1000 | 0000 | 0000 0011 | 20ns | 0000 0000 |
| B stores input data in address 0001 | 0 | 1 | 1 | 0 | 0000 | 0001 | 0000 0100 | 20ns | 0000 0000 |
| Output holds | 0 | 0 | 0 | 0 | 0000 | 0001 | 0000 0100 | 20ns | 0000 0000 |
| B stores input data in address 0010 | 0 | 1 | 1 | 0 | 0000 | 0010 | 0000 0110 | 20ns | 0000 0000 |
| Output holds | 0 | 0 | 0 | 0 | 0000 | 0010 | 0000 0110 | 20ns | 0000 0000 |
| Read from A at the address 0100 | 1 | 0 | 0 | 1 | 0100 | 0000 | 0000 0000 | 20ns | 0000 0001 |
| Output holds | 0 | 0 | 0 | 0 | 0100 | 0000 | 0000 0000 | 20ns | 0000 0001 |
| Read from B at the address 0001 | 0 | 1 | 0 | 1 | 0000 | 0001 | 0000 0000 | 20ns | 0000 0100 |
| Output holds | 0 | 0 | 0 | 0 | 0000 | 0001 | 0000 0000 | 20ns | 0000 0100 |
| Read from A at the address 1000 | 1 | 0 | 0 | 1 | 1000 | 0000 | 0000 0000 | 20ns | 0000 0011 |
| Output holds | 0 | 0 | 0 | 0 | 1000 | 0000 | 0000 0000 | 20ns | 0000 0011 |
| Read from B at the address 0010 | 0 | 1 | 0 | 1 | 0000 | 0010 | 0000 0000 | 20ns | 0000 0110 |

## 

## M-ROU-04

Table 5: Test Strategy for the Grey Counter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tested Feature** | **En** | **Reset** | **Clk** | **Delay** | **Count\_out** |
| Reset | 0 | 1 | 1 | 10ns | 0000 |
| Counter Hold when clock has a value of “0” | 1 | 0 | 0 | 10ns | 0000 |
| Counter Increases by “1” when enable has a value of “1” | 1 | 0 | 1 | 10ns | 0001 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0001 |
| Counter increases its value to “2” | 1 | 0 | 1 | 10ns | 0011 |
| Counter is Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0011 |
| Counter increases its value to “3” | 1 | 0 | 1 | 10ns | 0010 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0010 |
| Counter Increases its value to “4” | 1 | 0 | 1 | 10ns | 0110 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0110 |
| Counter Increases its value to “5” | 1 | 0 | 1 | 10ns | 0111 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0111 |
| Counter Increases its value to “6” | 1 | 0 | 1 | 10ns | 0101 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0101 |
| Counter Increases its value to “7” | 1 | 0 | 1 | 10ns | 0100 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 0100 |
| Counter Increases its value to “8” | 1 | 0 | 1 | 10ns | 1100 |
| Counter Hold if clock is “0” | 1 | 0 | 0 | 10ns | 1100 |

## M-ROU-05

Table 6: Test Strategy for the Grey to Binary Converter

|  |  |  |  |
| --- | --- | --- | --- |
| Tested Feature | Grey\_in | Delay | Bin\_out |
| Counter converts the value “0” from grey code to binary correctly | 0000 | 10ns | 0000 |
| Counter converts the value “1” from grey code to binary correctly | 0001 | 10ns | 0001 |
| Counter converts the value “2” from grey code to binary correctly | 0011 | 10ns | 0010 |
| Counter converts the value “3” from grey code to binary correctly | 0010 | 10ns | 0011 |
| Counter converts the value “4” from grey code to binary correctly | 0110 | 10ns | 0100 |
| Counter converts the value “5” from grey code to binary correctly | 0111 | 10ns | 0101 |
| Counter converts the value “6” from grey code to binary correctly | 0101 | 10ns | 0110 |
| Counter converts the value “7” from grey code to binary correctly | 0100 | 10ns | 0111 |
| Counter converts the value “8” from grey code to binary correctly | 1100 | 10ns | 1000 |

## 

## M-ROU-06

Table 7: Test Strategy for the FIFO Controller

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tested Feature | Rclk | Wclk | R\_req | W\_req | Rst | Delay | W\_ptr | R\_ptr | Wr\_valid | Rd\_valid | Full | Empty |
| No change if all inputs are 0 | 0 | 0 | 0 | 0 | 0 | 20 ns | - | - |  |  | - | - |
| Reset all inputs | 0 | 0 | 0 | 0 | 1 | 20 ns | - | - | 0 | 0 | 0 | 1 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | - | - | 0 | 0 | 0 | 1 |
| Write valid at first location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0000 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0000 | - | 0 | 0 | 0 | 0 |
| Write valid at second location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0001 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0001 | - | 0 | 0 | 0 | 0 |
| Write valid at third location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0010 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0010 | - | 0 | 0 | 0 | 0 |
| Write valid at fourth location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0011 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0011 | - | 0 | 0 | 0 | 0 |
| Write valid at fifth location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0100 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0100 | - | 0 | 0 | 0 | 0 |
| Write valid at sixth location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0101 | - | 1 | 0 | 0 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0101 | - | 0 | 0 | 0 | 0 |
| Write valid at seventh location | 0 | 1 | 0 | 1 | 0 | 20 ns | 0110 | - | 1 | 0 | 1 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0110 | - | 0 | 0 | 1 | 0 |
| Cannot write if full | 0 | 1 | 0 | 1 | 0 | 20 ns | 0111 | - | 0 | 0 | 1 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 0111 | - | 0 | 0 | 1 | 0 |
| Read valid from first location | 1 | 0 | 1 | 1 | 0 | 20 ns | 1000 | 0001 | 0 | 1 | 1 | 0 |
| No change if clock is low | 0 | 0 | 0 | 0 | 0 | 20 ns | 1000 | 0001 | 0 | 0 | 1 | 0 |
| Read valid from second location | 1 | 0 | 1 | 1 | 0 | 20 ns | 1001 | 0010 | 0 | 1 | 1 | 0 |

## 

## M-ROU-07

Table 8: Test Strategy for the FIFO

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tested Feature | Datain | Reset | Rclk | Wclk | Rreq | Wreq | Delay | Empty | Full | Dataout |
| Reset | 0000 0001 | 1 | 0 | 0 | 0 | 0 | 20 ns | 1 | 0 | 0000 0000 |
| Store data | 0100 0010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0100 0010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0000 0110 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0000 0110 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0110 0010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0110 0010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0100 1010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0100 1010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 1100 0010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 1100 0010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0100 1110 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0100 1110 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0111 0010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0111 0010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| Add data to queue till it’s full | 0110 1010 | 0 | 0 | 1 | 0 | 1 | 20 ns | 0 | 0 | 0000 0000 |
| No change | 0110 1010 | 0 | 0 | 0 | 0 | 1 | 20 ns | 0 | 1 | 0000 0000 |
| Reads first value inserted | 0110 1010 | 0 | 1 | 0 | 1 | 0 | 20 ns | 0 | 1 | 0100 0010 |

## M-ROU-08

Table 9: Test Strategy for the Round Robin Scheduler

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tested Feature | Din1 | Din2 | Din3 | Din4 | Clk | Delay | Dout |
| First input displayed | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 1 | 10ns | 0000 0001 |
| Hold if clock is “0” | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 0 | 10ns | 0000 0001 |
| Second input displayed | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 1 | 10ns | 0000 0010 |
| Hold if clock is “0” | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 0 | 10ns | 0000 0010 |
| Third input displayed | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 1 | 10ns | 0000 0011 |
| Hold if clock is “0” | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 0 | 10ns | 0000 0011 |
| Fourth input displayed | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 1 | 10ns | 0000 0100 |
| Hold if clock is “0” | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 0 | 10ns | 0000 0100 |

## M-ROU-10

Table 10: Test Strategy for the Router

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tested Feature | Datai1 | Datai2 | Datai3 | Datai4 | Wr1 | Wr2 | Wr3 | Wr4 | Wclock | Rreq | Wreq | Rclock | Rst | Delay | Datao1 | Datao2 | Datao3 | Datao4 |
| No output | - | - | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Reset | - | - | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 20ns | - | - | - | - |
| No change | - | - | - | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Data is stored in memory | 1000 0000 | 1100 0001 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| No change | 1000 0000 | 1100 0001 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| New data is stored in memory | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| Write Clock Cycles | | | | | | | | | | | | | | | | | | |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Input remains | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Input remains | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Input remains | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | - | - | - | - |
| Input remains | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 20ns | - | - | - | - |
| Output | | | | | | | | | | | | | | | | | | |
| Output from reset is displayed correctly | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | 0000 0000 | - | - | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20ns | 0000 0000 | - | - | - |
| Datao4 is displayed correctly | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | 0000 0000 | - | - | 1111 0011 |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20ns | 0000 0000 | - | - | 1111 0011 |
| Datao2 and datao3 are displayed correctly | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | 1000 0000 | 1000 0001 | - | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20ns | 1000 0000 | 1000 0001 | - | - |
| Datao2 and datao4 are updated | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | 0000 0000 | 1100 0001 | - | 1100 0011 |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20ns | 0000 0000 | 1100 0001 | - | 1100 0011 |
| Datao3 is displayed correctly | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20ns | 0000 0000 | - | 1110 0010 | - |
| No change | 1000 0001 | 1100 0011 | 1110 0010 | 1111 0011 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 20ns | 0000 0000 | - | 1110 0010 | - |

# Conclusion

In conclusion we find that our router is working as intended while achieving its functions according to the implemented test bench. The router takes the input and with every function of its submodule it successfully and releases the output in its desired location according to the routing address (last 2 bits of the input).

During the design there were a lot of problems and challenges that needed time and effort to fix such as the clock synchronization between the FIFO Controller and the RAM and how will we tackle the write and request signals. For each problem faced, we analyzed and took a decision for a workaround or a solution for the problem to help implement the router which was thankfully successful.

For the FSM we decided to use a 3-process Moore machine since the output doesn’t depend on the input change, it only focuses on the clock and the state. The test strategies used were fairly simple as we decided to test each functionality in each module according to the test tables using a fully implemented testbench to ensure the correctness of the cycle. The timing results were acceptable since the minimum clock time needed for the successful router implementation was 2.679ns which is acceptable and easy to achieve.

Development for this project could be done by enhancing the controller unit, adding more modules that further compacts the inputs and outputs or making the counter binary directly to lower the clock time needed for the full path.

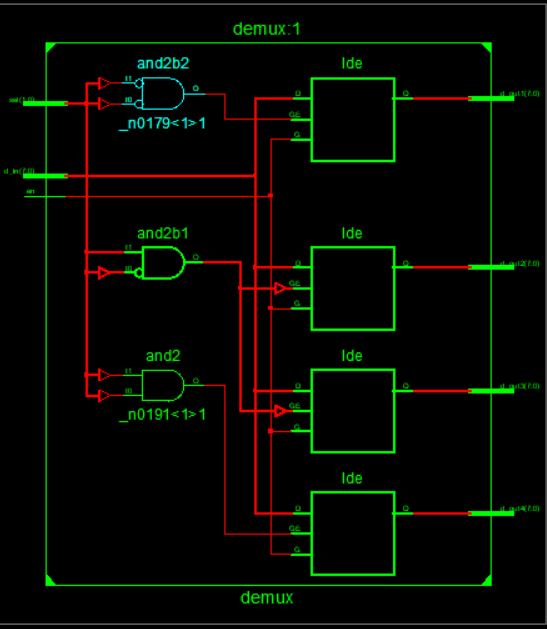
# Task Distribution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student/Tasks | Module development | Module Test Bench | Module Synthesis | Report Headers |
| Abdel-Rahman Megahed | M-ROU-02 (DeMux)  M-ROU-08 (Round Robin) | M-ROU-02 (DeMux)  M-ROU-08 (Round Robin)  M-ROU-09 (Router) | M-ROU-01 (Register)  M-ROU-02 (DeMux)  M-ROU-07 (FIFO)  M-ROU-08 (Round Robin) | Schedular Design and FSM implementation |
| Maryam Nouh | M-ROU-01 (Register)  M-ROU-07 (FIFO) | M-ROU-01 (Register)  M-ROU-07 (FIFO)  M-ROU-09 (Router) | M-ROU-09 (Router) | Design Flow  Test and simulation results |
| Seif Elewa | M-ROU-03 (RAM) M-ROU-06 (FIFO controller)  M-ROU-9 (Router) | M-ROU-03 (RAM) M-ROU-06 (FIFO controller) | M-ROU-03 (RAM)  M-ROU-04 (Gray Counter)  M-ROU-05 (Gray to Binary) M-ROU-06 (FIFO controller) | Introduction  Conclusion |
| Youssef Mansi | M-ROU-04 (Gray Counter)  M-ROU-05 (Gray to Binary)  M-ROU-09 (Router) | M-ROU-04 (Gray Counter)  M-ROU-05 (Gray to Binary) |  | Data Flow |

# RTL Schematic

## Register

## Demultiplexer



## RAM

## Grey Generator

## Grey To Binary

## FIFO

![A picture containing text, clock, display

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDsRXhpZgAATU0AKgAAAAgABAE7AAIAAAALAAAISodpAAQAAAABAAAIVpydAAEAAAAWAAAQzuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHNlaWYgZWxld2EAAAAFkAMAAgAAABQAABCkkAQAAgAAABQAABC4kpEAAgAAAAMzNwAAkpIAAgAAAAMzNwAA6hwABwAACAwAAAiYAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMTowMTowOCAxMzo1NTowNQAyMDIxOjAxOjA4IDEzOjU1OjA1AAAAcwBlAGkAZgAgAGUAbABlAHcAYQAAAP/hCx1odHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIxLTAxLTA4VDEzOjU1OjA1LjM3MTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5zZWlmIGVsZXdhPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIArgCYwMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APDvEuq6jD4m1COK/ukRZ2CqszAAZ7DNZf8AbWqf9BK8/wC/7f41Z8U/8jXqX/Xw386yaALv9tap/wBBK8/7/t/jR/bWqf8AQSvP+/7f41SooAu/21qn/QSvP+/7f40f21qn/QSvP+/7f41SooAu/wBtap/0Erz/AL/t/jR/bWqf9BK8/wC/7f41SooAu/21qn/QSvP+/wC3+NH9tap/0Erz/v8At/jVKigC7/bWqf8AQSvP+/7f40f21qn/AEErz/v+3+NUqKALv9tap/0Erz/v+3+NH9tap/0Erz/v+3+NUqKALv8AbWqf9BK8/wC/7f40f21qn/QSvP8Av+3+NUqKALv9tap/0Erz/v8At/jR/bWqf9BK8/7/ALf41SooAu/21qn/AEErz/v+3+NH9tap/wBBK8/7/t/jVKigC7/bWqf9BK8/7/t/jR/bWqf9BK8/7/t/jVKigC7/AG1qn/QSvP8Av+3+NH9tap/0Erz/AL/t/jVKigC7/bWqf9BK8/7/ALf40f21qn/QSvP+/wC3+NUqKALv9tap/wBBK8/7/t/jR/bWqf8AQSvP+/7f41SooAu/21qn/QSvP+/7f40f21qn/QSvP+/7f41SooAu/wBtap/0Erz/AL/t/jR/bWqf9BK8/wC/7f41SooAu/21qn/QSvP+/wC3+NH9tap/0Erz/v8At/jVKigC7/bWqf8AQSvP+/7f40f21qn/AEErz/v+3+NUqKALv9tap/0Erz/v+3+NH9tap/0Erz/v+3+NUqKALv8AbWqf9BK8/wC/7f40f21qn/QSvP8Av+3+NUqKALv9tap/0Erz/v8At/jR/bWqf9BK8/7/ALf41SooAu/21qn/AEErz/v+3+NH9tap/wBBK8/7/t/jVKigC7/bWqf9BK8/7/t/jR/bWqf9BK8/7/t/jVKigC7/AG1qn/QSvP8Av+3+NH9tap/0Erz/AL/t/jVKigC7/bWqf9BK8/7/ALf40f21qn/QSvP+/wC3+NUqKALv9tap/wBBK8/7/t/jR/bWqf8AQSvP+/7f41SooAu/21qn/QSvP+/7f40f21qn/QSvP+/7f41SooAu/wBtap/0Erz/AL/t/jR/bWqf9BK8/wC/7f41SooAu/21qn/QSvP+/wC3+NH9tap/0Erz/v8At/jVKigC7/bWqf8AQSvP+/7f40f21qn/AEErz/v+3+NUqKALv9tap/0Erz/v+3+NH9tap/0Erz/v+3+NUqKALv8AbWqf9BK8/wC/7f40f21qn/QSvP8Av+3+NUqKALv9tap/0Erz/v8At/jR/bWqf9BK8/7/ALf41SooAu/21qn/AEErz/v+3+NH9tap/wBBK8/7/t/jVKigC7/bWqf9BK8/7/t/jR/bWqf9BK8/7/t/jVKigC7/AG1qn/QSvP8Av+3+NH9tap/0Erz/AL/t/jVKigC7/bWqf9BK8/7/ALf40f21qn/QSvP+/wC3+NUqKALv9tap/wBBK8/7/t/jVjT9Y1NtTtVbUbsgzICDO3PzD3rKqzpv/IVtP+uyf+hCgB+s/wDIdv8A/r5k/wDQjRRrP/Idv/8Ar5k/9CNFAFrxT/yNepf9fDfzrJrW8U/8jXqX/Xw386yaACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBUUu6qvVjgV0et+EYtB1GOxv8AW7ITn/WqqSHyhgnn5eemOM9axtLu5rLU4J7ZgsiuMEqG/QjFen6tO9/8cILa8EcsMQcIjRrjmNjzxz0710QpxlFd2/yVzxsdiq1GslH4VGcn3fLbun3/AK68LpvhX7dod1q8+qWtnaW83k7pVcmRsAjAUE9+9YDABiAcgHgjvXpS6jey/DnULVJURW1Y2y/uU+VDjjp79etTavqS6D400/w/Z2dudNjVFlh8hWabd1JYjP5EUvZxbSXW33tJnNSzKuqk4yjzO8rK9rKKXlu7/wDDHl1bXhnwrqPinUBbafHtUAl53U7Ex6n+lW/F8J0DxBquiWDBbEz7vLKKSOcgbiMjH1rb+F2uaifElrpX2phY4ZvJCgAnHUkDJqKSjKWp14rF1vqEsTh0trq/a19rb+W3mcAw2sVPY4pKdL/rn/3jTazPWWwUUUUDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACrOm/wDIVtP+uyf+hCq1WdN/5Ctp/wBdk/8AQhQA/Wf+Q7f/APXzJ/6EaKNZ/wCQ7f8A/XzJ/wChGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAUUUUAFFFFABRXtGhfBPSNV8P2GoS6neo91bpMyqEwCyg4HHvXMeMPCvg7wjqCWL6lqV9dYzJHAY/wB16ZJHX2rqlhakI88tF6ng0M/wWIrvD0bykr6KL6Hn1Fb2fCP9zW/++4f8KM+Ef7mt/wDfcP8AhWPJ5o9P6y/5Jfd/wTBorez4R/ua3/33D/hRnwj/AHNb/wC+4f8ACjk80H1l/wAkvu/4Jg0VvZ8I/wBzW/8AvuH/AAoz4R/ua3/33D/hRyeaD6y/5Jfd/wAEwaK3s+Ef7mt/99w/4UZ8I/3Nb/77h/wo5PNB9Zf8kvu/4Jg0VvZ8I/3Nb/77h/woz4R/ua3/AN9w/wCFHJ5oPrL/AJJfd/wTBorez4R/ua3/AN9w/wCFGfCP9zW/++4f8KOTzQfWX/JL7v8AgmDRW9nwj/c1v/vuH/CjPhH+5rf/AH3D/hRyeaD6y/5Jfd/wTBorez4R/ua3/wB9w/4UZ8I/3Nb/AO+4f8KOTzQfWX/JL7v+CYNFb2fCP9zW/wDvuH/CjPhH+5rf/fcP+FHJ5oPrL/kl93/BMGit7PhH+5rf/fcP+FGfCP8Ac1v/AL7h/wAKOTzQfWX/ACS+7/gmDRW9nwj/AHNb/wC+4f8ACjPhH+5rf/fcP+FHJ5oPrL/kl93/AATBorez4R/ua3/33D/hRnwj/c1v/vuH/Cjk80H1l/yS+7/gmHG/lyq+M7TnFdNL41kl8cjxIbJQ4z+48w45Ur1x7+lVc+Ef7mt/99w/4UZ8I/3Nb/77h/wqlzK1pLQwqulWd50pPRrbo91v5DJPErv4dutJ+zKBcXhujLv5U8fLjHt1rW/4TyGa6s9Qv9EjuNUtECpdeeVDY6Fkxzj61mZ8I/3Nb/77h/woz4R/ua3/AN9w/wCFCutpL/hlZfgYTo4ee9KXXut9Hs+vbYp3uqrqTXlzqMDTX9zL5guBKVCZOSNuOfzqTwzrzeG9cj1KOBbho1I8tm25z71Yz4R/ua3/AN9w/wCFGfCP9zW/++4f8KSjZ3TRvJ05U3SdOXK9LeW1t9PkYTNuYt6nNJW9nwj/AHNb/wC+4f8ACjPhH+5rf/fcP+FLk80b/Wf7kvu/4Jg0VvZ8I/3Nb/77h/woz4R/ua3/AN9w/wCFHJ5oPrL/AJJfd/wTBorez4R/ua3/AN9w/wCFGfCP9zW/++4f8KOTzQfWX/JL7v8AgmDRW9nwj/c1v/vuH/CjPhH+5rf/AH3D/hRyeaD6y/5Jfd/wTBorez4R/ua3/wB9w/4UZ8I/3Nb/AO+4f8KOTzQfWX/JL7v+CYNFb2fCP9zW/wDvuH/CjPhH+5rf/fcP+FHJ5oPrL/kl93/BMGit7PhH+5rf/fcP+FGfCP8Ac1v/AL7h/wAKOTzQfWX/ACS+7/gmDRW9nwj/AHNb/wC+4f8ACjPhH+5rf/fcP+FHJ5oPrL/kl93/AATBorez4R/ua3/33D/hRnwj/c1v/vuH/Cjk80H1l/yS+7/gmDRW9nwj/c1v/vuH/CjPhH+5rf8A33D/AIUcnmg+sv8Akl93/BMGit7PhH+5rf8A33D/AIUZ8I/3Nb/77h/wo5PNB9Zf8kvu/wCCYNFb2fCP9zW/++4f8KM+Ef7mt/8AfcP+FHJ5oPrL/kl93/BMGit7PhH+5rf/AH3D/hTo/wDhD2kVXGtIpIBbdEdo9elHJ5oPrX9yX3f8E5+ivbtP+CnhvVdPhvdP1y9nt5l3I67CD+nX2rjPiX4BsvBH9nfYbu4uftfmbvOC/Lt29MD3raeEq04c72PMwvEGAxWIWGpt87vo01tucJRRRXKe8FFFFABRRRQAVZ03/kK2n/XZP/QhVarOm/8AIVtP+uyf+hCgB+s/8h2//wCvmT/0I0Uaz/yHb/8A6+ZP/QjRQBa8U/8AI16l/wBfDfzrJrW8U/8AI16l/wBfDfzrJoAKKKKACiiigAooooA9u1z4kJ4X8A6PpelEPqsunQ5Y9IFMY59zjp+ftXik88tzcPPcSNJLIxZ3Y5JNb3jP/kKWP/YLtP8A0Stc9XTiKspys9keLk+AoYWh7Smvenq311/RBRRRXMe0FFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAdn8PviDdeDtQEU+6fS5m/fQ55T/aX39u9dV8b9RtdV0/w7e2EyzW8yzMjqev3K8irodQ/wCSe6J/1+3f8oq64VpexlSe3/BR8/iMsoLMqOOgrSu0/P3Zfic9RRRXIfQBRRRQAUUUUAFWdN/5Ctp/12T/ANCFVqs6b/yFbT/rsn/oQoAfrP8AyHb/AP6+ZP8A0I0Uaz/yHb//AK+ZP/QjRQBa8U/8jXqX/Xw386ya1vFP/I16l/18N/OsmgAooooAKKKKACiiigDofGf/ACFLH/sF2n/ola56uh8Z/wDIUsf+wXaf+iVrnq0q/GzjwP8Au0PRBRRRWZ2BRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAd94m1G60fRvDq6YIoRNp6vJi3RtzYHJJU1V8GX9xrPjSzTVPLnVVbCmFFH5ACrXijxDfafo3h2LSdReEf2evmLC/8AFgdfeqHgjVJrvx3bXWqXRkfaQZJW/rXz8ab+pTk4q9pa9d35fqfLwpS/s+pNwV7T1+1u/L9S54MijPi/Wd0UbeXaTMgZAwUgjBwa51/FOtMhV549pGD/AKLEP/Za6PwXdJbeMNam85Y/9Em2OWA5yMYrmLnxPrd3bvBc6ncSxOMMjPkEV0UqbniJXinpHf0e2jOqjSdTFTvBSVob9NHto/0N2aWPwx4T064s4YX1DVFaV7iRAxjXjAAPHr2pUQeLPCN9dTwxLqWm7WWSFQhlTgHcBxxnsB0olii8U+ENOhsZo11DTEaNrd2CmROMMCePwojb/hE/CF/bXMsf9pajtVIomDmNOCdxHHbtmstLafxeb52v+XL8vmY6cun8bn+dub/0nl+XzOMrT8N6YuseI7OwkO1JpMMfYAk/yrMrU8N6mmj+JLK/lGUhkyw9iCD/ADr16/P7KXJvZ29T3cRz+xn7P4rO3rbQ19S8VS6drDWuj2ttb2VnKY1haFX8zDcliQTz7EVXbT7XxT4oaLQIPsVuY/Mk3nKpj7zfTnpVjUvClxqOsNc6PcW9zZXkpkScyhBHluQwPIx9DWxo1zoGmeJp9N0yb9zPaGFrqRyVaX244FeU6lOnT5qCvO2v/wBt5+uu54kqtKlS58Mm58rv+rl572vruc5ceF0bS7q/0jUU1COzcLOBEUIB/iGScitL/hAEF9b2D61Ct5dQiWKLyTyCMgE54Pap9Ntz4U8O60dVlRZblPs8ESSBvM5HzDH071ZvriE/EzQJBMhjVLfcwYYH1NTLEV3JqE9FezstbJPtbfsRPFYlylGnO8UpNOy1tFPtbdvYq6Xo9jD4I1z7bOEuoZURyYCxiPPAOec+tYr+G4bfwva6zd6isYuw4hhEJYllJGCc8Zx1ro0QXmh+LYbeSNpGu0dFMgG4DdkjPWsrVLpI/BHhsqyu8TyMybuR+8Y8+lOlUq87tLeSvttyX7dyqNat7R2k7ymr7bcl+2moz/hC44ri0s73VorfULtQyW3lFtuem5s8ZrV0LSjY6Ld21/bp9oh1e3jbcoP8QBAPpU+raf8A254r0/WrO5i+wSIhklEoUw7euQefyq22s2ep/wBp3MEqCN9YtymWxuUMPmx6Vz1MRWqQSbvs35Pm2/4fsctXFV6tOKbvezenwvm20/XXQ57UPDv9reLdbKzR2VnZHfLIVyEGOMAfQ1n3XhcJZ2d7YX6XlpcziAyCMoY3PQFc/X8q7C31OKfUfEulxfZHmmkEsH2jDJKQOV9P/wBdZct3qtjY2tre/wBk2MFxdIxtYFAkO1h83y5AH41rTxFdNRTta2ndcu+1/wBNNTajisSnGCdrKOjtquXV7X/TSzOd8SaFB4fvmshqAurmNsSIsRUKMZBznnrWNW943kSXxlqDxurqZOGU5B4FYNevhnKVGMpu7aTPdwcpyw8JVHdtJv5+gUUUV0HUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV0Oof8k90T/r9u/5RVz1dDqH/ACT3RP8Ar9u/5RVpDaXp+qOPEfxKX+L/ANtkc9RRRWZ2BRRRQAUUUUAFWdN/5Ctp/wBdk/8AQhVarOm/8hW0/wCuyf8AoQoAfrP/ACHb/wD6+ZP/AEI0Uaz/AMh2/wD+vmT/ANCNFAFrxT/yNepf9fDfzrJrW8U/8jXqX/Xw386yaACiiigAooooAKKKKAOh8Z/8hSx/7Bdp/wCiVrnq6Hxn/wAhSx/7Bdp/6JWuerSr8bOPA/7tD0QUUUVmdgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVbs9J1HUI2ksLG5uUU7WaGJmAPpwKqV7J8GSF8N6gWIAF1kk9vkFefmGLlhMO6sVfY8vNcdLAYV14q7TWnqeX/APCM67/0Br//AMBn/wAKF8Na4GBOi3xAPI+zPz+lel+O/iVBBanTvDk6yzSDElyh4jHoPf8AlXk/267/AOfqb/v4azwdfF4in7ScVHtvf9LGWAxGOxVL2lSChfZO9/0sdOtjcBQD4DkYgcki55/WqeoaNqV2EFp4TubEr1McUzbv++s1ifbrv/n6m/7+Gj7dd/8AP1N/38NbxoVYu6f/AKV/8kdEcNWjLmUl98n+ci7/AMIzrv8A0Br/AP8AAZ/8KP8AhGdd/wCgNf8A/gM/+FUvt13/AM/U3/fw0fbrv/n6m/7+GtbV+6+5/wCZvbE94/c/8y7/AMIzrv8A0Br/AP8AAZ/8KP8AhGdd/wCgNf8A/gM/+FUvt13/AM/U3/fw0fbrv/n6m/7+Gi1fuvuf+YWxPeP3P/Mu/wDCM67/ANAa/wD/AAGf/Cj/AIRnXf8AoDX/AP4DP/hVL7dd/wDP1N/38NPh1S+t50mhvJ1kjYMreYeCKLV+jX3P/MGsTbRx+5/5lr/hGdd/6A1//wCAz/4VQurS4spzBeQSQSgAlJUKsPwNe6eBPHcHiW1W1vGWLUo1+Ze0o/vL/UV5v8VP+R/uv+uUX/oAry8JmFeri5YatDlaV/yPGwOaYmtjZYTEU1FpN/ivw1OOooor3D6MKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuh1D/knuif9ft3/KKuerodQ/5J7on/AF+3f8oq0htL0/VHHiP4lL/F/wC2yOeooorM7AooooAKKKKACrOm/wDIVtP+uyf+hCq1WdN/5Ctp/wBdk/8AQhQA/Wf+Q7f/APXzJ/6EaKNZ/wCQ7f8A/XzJ/wChGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAqI0jhI1Z2Y4CqMk1cg0bUrmZ4obGcvHneDGRswCeSenAPWk0i+Gmaxa3rIZBBKHKA4ziu50Q69JpcVwdC1i6aRZWWeDhJxIOC2VO7HarjFy2POxuKqYdXilbzdtdfT+mcFdWNzZNtuYWThTnqOVDDn6EGq9ega9pfiLW9KtbX/hFL+GW2CKsixH5gECncMcnIPPpiud/wCEF8U/9AC//wC/Bp+znfZiw+YUp071ZxT/AMS/zH+M/wDkKWP/AGC7T/0Stc9Xb+LvCuuz6nZmHSbtwmnWqNtiJwyxKCPqDWD/AMIf4i/6At7/AN+TWNavRjUknJX9UTgsVh1hoJzWy6oxqK2W8IeIVUs2jXgAGSfJPFU/7G1H/nym/wC+DUKtSltJfeeth4vFJugue29tbfcUqKu/2NqP/PlN/wB8Gj+xtR/58pv++DT9pDujp+pYr/n1L7mUqKu/2NqP/PlN/wB8GprXQL+4ukikt5IVY8yMhwtDq00ruSKhl+LnJRjSld+TMyitjUPDV7ZyqkKtdBlyWjQ4HtVT+xtR/wCfKb/vg0o1qcldSRdXLcbRm6c6Urryv+K0KVFXf7G1H/nym/74NH9jaj/z5Tf98Gn7SHdGX1LFf8+pfcylRV3+xtR/58pv++DR/Y2o/wDPlN/3waPaQ7oPqWK/59S+5lKirv8AY2o/8+U3/fBo/sbUf+fKb/vg0e0h3QfUsV/z6l9zKVFXf7G1H/nym/74NH9jaj/z5Tf98Gj2kO6D6liv+fUvuZSp8MEtxJsgieV8Z2opJ/SrX9jaj/z5Tf8AfBrQ0aPU9IupZ0sLgu8RjUhcbSSDn9KmdVKLcWm/UzqYPGqDcKUm/R/5GedH1EQxSmzm2yttjGw5Y4zwOvQUT6Rf29nDdTWzLDMGKNwc7euR1GPeu2Gr3N7HLaxaJqjPPIZAY3G+M7t2E+XgdvpU076zNoQ0x/C+ounluhlZfnyehBAx9fWvO+vVU1zJLXutvvPArYrGYaahiKag77N2dtdVd9zzaultdUu7D4dzwWkzRJd35jm28Fl8tTjPpVT/AIQ/xF/0Bb3/AL8mtU+Ftd/4Q1Lf+ybvzhfs+zyjnb5YGfpmuivWw8+VOSeq6o0xOIws1BOcWuZdUchRWz/wh/iL/oC3v/fk0f8ACH+Iv+gLe/8Afk10fWaH86+9HX9bw/8Az8j96Mair2oaLqWlKjalYz2okJCGVCu7HXFUa1jKM1eLujeE4zXNB3QUUUVRQUUUqjcwHqcUAJRXpq/Ba7ZQf7Zh5Gf9Qf8AGl/4Urd/9BmH/vwf8a8n+2cB/wA/Pwf+R4f9v5Z/z9/B/wCR5ta3U1ldR3NrI0U0TbkdTgg1u+N7uW/16K7uSGlnsreRyBjJMak8V1n/AApW7/6DMP8A34P+NX9X+E1zqdzDKuqxRiO2igwYSclEC5698VzyzXAOtGpzrRNXs/LyOWedZY8RCoqi0TV7PrbyPJoreafd5EUkmwZbYpOB6nFSyaddxWKXksDpA7bVdhjcfb8q9UsfhTfWNheQRazGGudg3rCRgAnI6981oeIPh2+paYYotR8oCYzsGj3KvB+VQOg5olnmG9ooxkrX31/yB8RYR1lCEk03vZ+Xl3/I8ag0+8udhgtpXWRwisEO0knAGenWrUfh7VZLh4RZSqyZ3FxtUY/2jxXe28TaDBYw2Ra/NujZ3MIoySxOSpBORng5qO5mvL7zopYpI7YohVPPBZmXORnHfcPyrR5hUcvdSt/wex9K8vz6UnyYSXL3affto9tThZNA1OK+W0a1JlYBhtYMpGM53A4xj3pl1o19ZLK1zBtWEgOwYEc9MEHke4ruHM73sM7WcpEUDQY+0oNyspU5AXGcHjAqpf2Nxd6XcWtvB5KeXDHCrzBiQrMxJP8AwKtI46d1zW6X+/1NIZXnrcebCy6X0ffXr2OEora/4RTU/wC7F/38FH/CKan/AHYv+/grv+s0f5ken/YuZf8APiX3MxaK2v8AhFNT/uxf9/BR/wAIpqf92L/v4KPrNH+ZB/YuZf8APiX3MxaK2v8AhFNT/uxf9/BR/wAIpqf92L/v4KPrNH+ZB/YuZf8APiX3MxaK2v8AhFNT/uxf9/BR/wAIpqf92L/v4KPrNH+ZB/YuZf8APiX3MxaK2v8AhFNT/uxf9/BR/wAIpqf92L/v4KPrNH+ZB/YuZf8APiX3MxaK2v8AhFNT/uxf9/BVTUNGu9MjR7pUCucDa2aca9KTtGSuZVcrx1GDqVaUlFbtplCiiitjzgooooAK6HUP+Se6J/1+3f8AKKuerodQ/wCSe6J/1+3f8oq0htL0/VHHiP4lL/F/7bI56iiiszsCiiigAooooAKs6b/yFbT/AK7J/wChCq1WdN/5Ctp/12T/ANCFAD9Z/wCQ7f8A/XzJ/wChGijWf+Q7f/8AXzJ/6EaKALXin/ka9S/6+G/nWTWt4p/5GvUv+vhv51k0AFFFFABTxNKBgSOAOwY0yigVkx/ny/8APV/++jR58v8Az1f/AL6NMooDlXY6TxnLINUscOw/4llp3/6YrXPedL/z0f8A76Nb3jP/AJClj/2C7T/0Stc9V1UudnHgUvq0PRD/ADpP+ejf99Gm72/vH86Sis7I7VpsLvb+8fzo3t/eP50lFMfMxd7f3j+dPiuJoJVkilZHXowPSo6KVkxqcou6eppa3czS6gvmSs2Io8ZPTKgn9TWdvb+8fzq7rH/IQ/7ZR/8AoAqjWdJJU4+h25hOUsXVbf2n+Yu9v7x/Oje394/nSUVqcPMxd7f3j+dG9v7x/OkooDmYu9v7x/Oje394/nSUUBzMXe394/nRvb+8fzpKKA5mLvb+8fzo3t/eP50lFAczHCRx0dh+NL50v/PR/wDvo0yilZCeu4/zpf8Ano//AH0a2Wlk/wCEFQ72z/aTc5/6ZLWHWy3/ACIqf9hJv/RS1hWS931OXEJXh/iX6mT50v8Az0f/AL6NHnS/89H/AO+jTKK3sjpshWdn+8xb6mkoopjCiiigAoBwcjrRRQBuDxr4lAwNbvMf9dTR/wAJt4l/6Dd5/wB/TWHRXP8AVaH8i+5HL9Tw3/PuP3I3P+E28S/9Bu8/7+mtjxJ4t1+1vrVbfV7qNWsbd2CyEZYxqSfxNcXW34r/AOQjZ/8AYPtf/RS1hLDUPbRXItn0Xkc08JhvbwXs47PovIX/AITbxL/0G7z/AL+mj/hNfEv/AEG7z/v6ax4rea4OIInkOQMIpPJ6CpGsLtLb7Q9rMsH/AD1MZ2/nWrw+GTs4R+5G7wuETs4R+5El/q+oapKsmo3ctw6LtVpGyQPSqm9v7x/OpvsV19k+1fZpfs+cebsO3069Kv8A/CL6z9qWA2MoLAHeV+QZ6ZboK056VNWukjb21KkuXmSX3GVvb+8fzo3t/eP51p3Ph3UrZoQYRL5xwhgcSDOcYJXOD7UXPhzUbUMZI0ZViMpZJFYYHXkHqO46ij21J295AsVSdrTWvmZm9v7x/Oje394/nSUVsb8zF3t/eP50b2/vH86SigOZi72/vH86N7f3j+dJRQHMxd7f3j+dG9v7x/OkooDmYu9v7x/Oje394/nSUUBzMXe394/nW1fknwlpuef3j/zNYlbd/wD8ilpv/XR/5muet8UPX9Gevl7bo4n/AAf+3wMSiiiug8cKKKKACuh1D/knuif9ft3/ACirnq6HUP8Aknuif9ft3/KKtIbS9P1Rx4j+JS/xf+2yOeooorM7AooooAKKKKACrOm/8hW0/wCuyf8AoQqtVnTf+Qraf9dk/wDQhQA/Wf8AkO3/AP18yf8AoRoo1n/kO3//AF8yf+hGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQB0Fh4L1PUbCK8hktkilXcPMkwQMkDPHcggfSpU8Ba0yRsywx7uWV3wYh/ebjgdvxqrpvi7VtJCi0kjCrEIQrxBhtDFhwfdjXUWHxEhg09Ip7y/SRgDMkVlAUZvxOSPrVxSbPCxNTNKcm6ai1fTe9vuOX1vwzcaFp9pcXcqGS5LDyh1UAkZ9wcZ/GsSu01fxDoOuRRx6hc6s4jYshW1gUjPbhuntWTs8I/899b/wC/MP8A8VRyPujpw2JrKn+/i+bXZO3kP8Z/8hSx/wCwXaf+iVrnq7fxcvh3+07Pz5dTB/s6127Ioz8vlLjOW6461g7fDH/PbVv+/UX/AMVWVaqlUkrP7gwWISw0Pdey6GNRWyy+Gdp2y6tnHGYo/wD4qqeNL/v3n/fC/wCNQqqfR/cerh17dNr3bd9ClRV3Gl/37z/vhf8AGjGl/wB+8/74X/Gnz+TOr6q/54/eilRV3Gl/37z/AL4X/GjGl/37z/vhf8aOfyYfVX/PH70LrH/IQ/7ZR/8AoAqjWzqg077b+9a63eVH91VxjYMd6p40v+/ef98L/jUU5+4tHsdeNwzeKqPmj8T6ruUqKu40v+/ef98L/jRjS/795/3wv+NXz+TOT6q/54/eilRV3Gl/37z/AL4X/GjGl/37z/vhf8aOfyYfVX/PH70UqKu40v8Av3n/AHwv+NGNL/v3n/fC/wCNHP5MPqr/AJ4/eilRV3Gl/wB+8/74X/GjGl/37z/vhf8AGjn8mH1V/wA8fvRSq9Y6TPqFtcTwtEqW+3cZGxknOAPU8UmNL/v3n/fC/wCNWIbyygtXgjluxHI6SMPLTqucd/c1E5y5fcWvoZ1MLUcfcnG/qvmXB4J1YsAwhUbcuWfiM/3W44PtUlv4G1F3xcyQwKEZnOcmPHTcO2e1X4vHbm6jkuLif5Ody2sRbPryetWZfHVpP/rbvUGJzuJs4Pmz6/NzXmSq5gtLL8f8j53EPNKMuR8r03jd/ocERgkVst/yIqf9hJv/AEUtKV8ME587Vv8Av1F/8VWqV8Pf8Ian73U/I+3tz5ce7d5Y/wBrGMV2Vay933Xv2OmviF7nuvddDkKK2dvhj/ntq3/fqL/4qjb4Y/57at/36i/+Krf2y/lf3HT9YX8svuMair2oDSgqf2U94zZO/wC0oqjHbG0mqNaxlzK5vCXMr2t6hRRRVFBSqNzAepxSUA4ORQB6kvwVdlB/tteRn/j2/wDsqX/hSj/9Bxf/AAG/+yrgx4s8QAYGtX//AIEN/jR/wlniH/oN3/8A4EN/jXg/Vs1/5/r7l/kfM/U87/6CV/4Cv8jvP+FKP/0HF/8AAb/7Kr+rfCV9TuYZRq6x+XbxQY+z5zsQLn73fFeaf8JZ4h/6Dd//AOBDf41seJfEut29/arBq15GrWNu5CzsMsY1JPXqTWMsNmftIp1lfXovLyMJYTOfaxTxCvZ/ZXl5Hd6B8M59Bmmkj1aOUyL8u62+64+633u2al8SfDyXWIEVdVW2t4C7CJLfI29h97qBmvJ/+Es8Q/8AQbv/APwIb/Gj/hLPEP8A0G7/AP8AAhv8aX9m5g6vtXWXN6f8ASynNPbqvKvFy/wr/I7C8sP7N0iLTrFzckW7xGSZvKT5i3OzByRng5p8l7eXTRROiW8JjdJitzuJJ28gYHTb+tefX2pXupyrJqN3NdOo2q0zliB6c1Wrvjl7avOWuvTv81+R9jHDZS4p1MPJy1d/adX/ANuJfgegv5jR2aC0ULaSCSPN6OoOcfc6e360ksbyQzQwQR20bwz9bneWkkAGegwOK8/orT6jbaX5/wCZr7DJ1th5f+DH/wDIm1/wit//AM9Lb/v7R/wit/8A89Lb/v7WLRXXyVv5l93/AATu+sZd/wA+Jf8Agf8A9qbX/CK3/wDz0tv+/tH/AAit/wD89Lb/AL+1i0Uclb+Zfd/wQ+sZd/z4l/4H/wDam1/wit//AM9Lb/v7R/wit/8A89Lb/v7WLRRyVv5l93/BD6xl3/PiX/gf/wBqbX/CK3//AD0tv+/tH/CK3/8Az0tv+/tYtFHJW/mX3f8ABD6xl3/PiX/gf/2ptf8ACK3/APz0tv8Av7WfqGnT6ZOsNzt3Mu4bTkY/yKq1t+KP+Pu0/wCvVP61KdSNRRk73v0NZU8HWwlSrSpuLg47yvvfyXYxK27/AP5FLTf+uj/zNYlbd/8A8ilpv/XR/wCZp1vih6/ozPLv4OJ/wf8At8DEoooroPICiiigArodQ/5J7on/AF+3f8oq56uh1D/knuif9ft3/KKtIbS9P1Rx4j+JS/xf+2yOeooorM7AooooAKKKKACrOm/8hW0/67J/6EKrVZ03/kK2n/XZP/QhQA/Wf+Q7f/8AXzJ/6EaKNZ/5Dt//ANfMn/oRooAteKf+Rr1L/r4b+dZNa3in/ka9S/6+G/nWTQAUUUUAFFFFABRRRQB0PjP/AJClj/2C7T/0Stc9XQ+M/wDkKWP/AGC7T/0Stc9WlX42ceB/3aHogooorM7AooooAKKKKAL2sf8AIQ/7ZR/+gCqNXtY/5CH/AGyj/wDQBVGs6X8Neh2Y7/eqn+J/mFFFFaHGFFFFABRRRQAUUUUAFFFFABRRRQAVst/yIqf9hJv/AEUtY1bLf8iKn/YSb/0UtYVvs+pzYjeH+JfqY1FFFbnSFFFFABRRRQAUUUUAFFa1vocE9tHK2t6bCXUExyNJuX2OEIzUv/CPW/8A0MOk/wDfcn/xFYPEU07fo/8AI5niqSdnf7n/AJGJW34r/wCQjZ/9g+1/9FLR/wAI9b/9DDpP/fcn/wARWv4k0OCa+tS2t6bFixt1w7SZOI1GeE6HrXPLEU/bRfk+j8vI5Z4ql7eD12fR+XkcpZ2N1qE/k2MEk8uCdka5OBVmPQdWlthcRadcNCRkOIzgjGa6Tw8LbQmuN+q6PdJOFBUzSrjGcc+WfWtCfWPPnkk/tfSVV9nyefLgbc/9M+9ZVMZVU2qcdO+phWx9dVHGnC8e7T/r/hvM4+28N6tdPEsdjKPOUtGzqQHAGTg/SqVxY3Vozi5gkj2NsbcuMH0ru5/FVtFfzSWr20iTIscm+4cAgKAdo2fLz3rH8Q38WtW1jEt5aq0CEysWb53br/D6Ac06WJxEprnhZP8Ar/gHo4ejmU5x9pTSi/70dOvf5f1Y5Wirv9nR/wDQRtP++m/+Jo/s6P8A6CNp/wB9N/8AE13+0ier9TreX3x/zKVFXf7Oj/6CNp/303/xNH9nR/8AQRtP++m/+Jo9pEPqdby++P8AmUqKu/2dH/0EbT/vpv8A4mj+zo/+gjaf99N/8TR7SIfU63l98f8AMpUVd/s6P/oI2n/fTf8AxNH9nR/9BG0/76b/AOJo9pEPqdby++P+ZSoq7/Z0f/QRtP8Avpv/AImj+zo/+gjaf99N/wDE0e0iH1Ot5ffH/MpVt+KP+Pu0/wCvVP61R/s6P/oI2n/fTf8AxNa/iG0Sa5ti13bxYt1GHLc9eeBWE6kfawfqephsLVWArrTVw6ru/M5utu//AORS03/ro/8AM1R/s6P/AKCNp/303/xNa95Zo3hmwjN3bqFd8OS21uT04oq1IuUPX9GGX4WqqWIvbWHdfzR8zm6Ku/2dH/0EbT/vpv8A4mj+zo/+gjaf99N/8TW/tInl/U63l98f8ylRWpb6PDOWDaxp8OO8jPz+Smp/+Eet/wDoYdJ/77k/+IqHXpp2f5P/ACPPrVI0JunPddtfxV0YldDqH/JPdE/6/bv+UVQ/8I9b/wDQw6T/AN9yf/EV0F54ft5PA2kRHxBpSBLq5YSM8m1siPgfJ1GOfqK6KFSNTmUe3p1Xc87EYqlz0t/i7P8All5HDRRSTzJFCjPI7BVVRkknoK1f+ET8Qc/8Se84OD+5PBra0HTrbQ9Yhvx4h0OfywwMbSygMGUqefL966C91uO5uA0Os6LBEIUiEQupiBhmOc+V33fpWqpvqYYnMK0aijQheNt2nv26HBWnhrWLw2/k2E2y5bbFIykKxwTgH6A1Vu9MvLGSVLu2kiaJtr7lxg9v513OpeOINL1RDYxw3xNskcskVw3ldOQoKjB965/xF4nTWtJsrVISkkTNJMxPU5IUZ74TaM+1ZNNLzNMPicbUnFzppQf3rf8A4Y5yiiikewFWdN/5Ctp/12T/ANCFVqs6b/yFbT/rsn/oQoAfrP8AyHb/AP6+ZP8A0I0Uaz/yHb//AK+ZP/QjRQBa8U/8jXqX/Xw386ya1vFP/I16l/18N/OsmgAooooA1PD6WDahI+rRrLBHA8gjaQpvYDgZBFduNC8HXdvOYpLW3CSmMsbz5wFIwygtg5/GvNK2rbxFHbWscJ0LSJyihfMlhcs3uSHHNUkno3Y8nHYWtUkp0pyT7J2X52NDxbpGn2On2lxp8MMYkd0zDceaHUE4Y8nBIxkVytdAfFaFQp8PaKVHQeRJgf8Aj9J/wlEX/QuaJ/4Dv/8AF1XLG+5eHeJpU+ScXJ97r/NjvGf/ACFLH/sF2n/ola56u38Xa6kWp2YOjaXJu061bLxMcZiU4HzdB0FYX/CRR/8AQC0j/vy//wAXWNadRVJJR/FCwVSr9Wh7nRdUYtFbLeIY2Uj+w9JGR1EL8f8Aj9U/7RX/AJ8LT/vg/wCNQpze8fxR62HSqJ+1fJ+N/uKVFXf7RX/nwtP++D/jR/aK/wDPhaf98H/GnzS7HT7Gj/z8/BlKirv9or/z4Wn/AHwf8aP7RX/nwtP++D/jRzS7B7Gj/wA/PwYusf8AIQ/7ZR/+gCqNbOqXype4Nnat+6jOWQ5+4Peqf9or/wA+Fp/3wf8AGopylyLTodeNpUXiql6n2n0fcpUVd/tFf+fC0/74P+NH9or/AM+Fp/3wf8avml2OT2NH/n5+DKVFXf7RX/nwtP8Avg/40f2iv/Phaf8AfB/xo5pdg9jR/wCfn4MpUVd/tFf+fC0/74P+NH9or/z4Wn/fB/xo5pdg9jR/5+fgylRV3+0V/wCfC0/74P8AjR/aK/8APhaf98H/ABo5pdg9jR/5+fgylT4QrToH4UsAfpmrX9or/wA+Fp/3wf8AGj+0V/58LT/vg/40c0uwnRo2/i/gzqLS18Lz6lPBLDFFELkwxubk4wFJ3E56EjGferp8N6A9w0dt9mnaTcXVLvPkgAY2fNlu/XNcbDqqRShzptjJj+F42IP/AI9VtfEio2V0PSVPqIX/APi68yeGr3vCUvv/AOCfPYrA1oT/AHFWUlbv1+bv/XqYrDDEe9bDf8iKn/YSb/0UtL/wkUf/AEAtJ/78v/8AF1qnXU/4Q1Jf7G0zH29l8vym2/6sHON3WumrOp7vude6Na9Sr7nudV1RyFFbX/CRR/8AQC0j/vy//wAXR/wkUf8A0AtI/wC/L/8Axdb+0qfyfijp9rV/59/ijFoq9qGpLfqgXT7O02EnNsjKW+uSao1rFtq7VjeDbV5KwUUUVRQUUUUAFFFFABW34r/5CNn/ANg+1/8ARS1iVt+K/wDkI2f/AGD7X/0Utc8/40PR/ocs/wDeIekv0MSitfQdAbXVuxHcpDJAgKIw/wBYxzhfbpUtn4XnubGO6lmWJJZNiYXOfkZs/wDjuPxpyxFKLak9UOeKowbjKWq3+eph0UUVudIUUUUAFFFFABRRRQAUUUUAFFFFABW34o/4+7T/AK9U/rWJW34o/wCPu0/69U/rXPP+LD5nr4X/AJF+I9YfmzErbv8A/kUtN/66P/M1iVt3/wDyKWm/9dH/AJmit8UPX9GGXfwcT/g/9vgYlFFFdB5AUUUUAFdDqH/JPdE/6/bv+UVc9XQ6h/yT3RP+v27/AJRVpDaXp+qOPEfxKX+L/wBtkc9RWjoOlLrWsw2L3K2qyBiZXGQoCk9PwrY0zwJd3+tXNi9wkSW5YGYLuzgAjjjqCKzKrYyhQbVSVrK/yOWopWG1iPQ4pKDqCiiigAqzpv8AyFbT/rsn/oQqtVnTf+Qraf8AXZP/AEIUAP1n/kO3/wD18yf+hGijWf8AkO3/AP18yf8AoRooAteKf+Rr1L/r4b+dZNa3in/ka9S/6+G/nWTQAUUUUAFFTW1nc3jbbWCSY7guEXPJ6Crtj4c1bUAzW1lK0aLveQqQqjBIJPbIBp2ZlOtTgm5ySt5mZRVu90y70/BuYiFIXDjlcsoYDPrgiqlIuMozXNF3R0PjP/kKWP8A2C7T/wBErXPV0PjP/kKWP/YLtP8A0Stc9WlX42cuB/3aHogooorM7AooooAKKKKAL2sf8hD/ALZR/wDoAqjV7WP+Qh/2yj/9AFUazpfw16HZjv8Aeqn+J/mFFFFaHGFFFFABRRRQAUUUUAFFFTQWlxdf8e8Ly8hflXPJzgfoaTaSuxNpK7IaK0IdA1a4SN4NOuZFk+4VjJ3cZ4/CnXGgahbWEN3LB+6mDkAHJUL97I7YrP21O9uZfeZe3pX5eZX9TNrZb/kRU/7CTf8Aopaxq2W/5EVP+wk3/opamt9n1IxG8P8AEv1MaiiitzpCiiigAooooAKKKKANi20bT57aOWXxDYwOygtE8cpZD6HCEflUv9g6Z/0M+n/9+p//AIisKisHSm38b/D/ACOZ0ajd/aP7o/5G7/YOmf8AQz6f/wB+p/8A4itfxJo2ny31qZPENjCRY26gPHKcgRqA3CdD1ri62/Ff/IRs/wDsH2v/AKKWueVKfto++9n28vI5Z0ant4fvHs+kfLyL+nW9vpTs9l4q01CxBOYJj06fwVclvBLFHEPFGkRxxNuVY7SVQDtK9o/RjXE0U5YNSlzSld+kf8ipYBTlzSm2/SP/AMiaDadaq5A1a2YA4BCSc/8AjtJ9gtf+gpb/APfD/wDxNUKK6uSX8z/D/I9xV6SVnSX3y/8Aki/9gtf+gpb/APfD/wDxNH2C1/6Clv8A98P/APE1Qoo5JfzP8P8AIf1ij/z5j98v/ki/9gtf+gpb/wDfD/8AxNH2C1/6Clv/AN8P/wDE1Qoo5JfzP8P8g+sUf+fMfvl/8kX/ALBa/wDQUt/++H/+Jo+wWv8A0FLf/vh//iaoUUckv5n+H+QfWKP/AD5j98v/AJIv/YLX/oKW/wD3w/8A8TR9gtf+gpb/APfD/wDxNUKKOSX8z/D/ACD6xR/58x++X/yRf+wWv/QUt/8Avh//AImj7Ba/9BS3/wC+H/8AiaoUUckv5n+H+QfWKP8Az5j98v8A5Iv/AGC1/wCgpb/98P8A/E1reIbSCW5tjJfQwkW6gBlY5HPPArmq2/FH/H3af9eqf1rCcZe1j73ft/kerhq9J4Cu/ZLeHWXd/wB4p/YLX/oKW/8A3w//AMTWteWkDeGbCM30KqrviQq2G5PTjNc1W3f/APIpab/10f8AmaKsZc0Pe6+XZ+QYCvSdLEWpJe53l/NH+8U/sFr/ANBS3/74f/4mj7Ba/wDQUt/++H/+JqhRW/JL+Z/h/keV9Yo/8+Y/fL/5I1bbSbGYsJdds4MdC8cpz+SmrH9g6Z/0M+n/APfqf/4isKiodObek3+H+R59eEqlRyhJxXZWt+Kb/E3f7B0z/oZ9P/79T/8AxFdBeaFpr+B9IjfxLp8aLdXLLK0U2HJEeQMJnIwPzrgq6HUP+Se6J/1+3f8AKKuihGUOZylzadbd12sebiKNXnpfvX8XaP8ALLyLNlpGn6fdLcW3jDS1kUEAmCc9Rg/8s/Q1uQ6xJbtuh8Y6Gjn7zixl3NwByfL54Arzmir54/yr8f8AMqrlyrO9Sbfqof8AyJb1O1hs71ora/hv0wD50KsqknthgD+lVKKKh7npRTjFJu4UUUUigqzpv/IVtP8Arsn/AKEKrVZ03/kK2n/XZP8A0IUAP1n/AJDt/wD9fMn/AKEaKNZ/5Dt//wBfMn/oRooAteKf+Rr1L/r4b+dZNa3in/ka9S/6+G/nWTQAUUUUAbPh3xJceHJp5LaJJDKgA3fwsPut+Ga7JNS1vVUivo9AQp5cgtvLvkjVUcdCvfHvivNKKtS6M83E5fSrT9okubu7+nSSO81uw8Ra3pttaz6LFG1qFWKRbqPO0KBg888gn8awf+EI8Qf8+K/+BEX/AMVWDRTvDs/v/wCAVRw1ajHkpyil/hf/AMmdv4u8MavPqdmYrUEJp1qjfvkGCIlBHX1rB/4RHW/+fNf+/wDH/wDFVY8Z/wDIUsf+wXaf+iVrnqxrKt7SVpK3o/8AMzwUcR9WhaS2X2X/APJGyfCetKpJtFwBk/v4/wD4qqf9kXv/ADxH/fxf8apUVCVXq193/BPWw9op+397tb3fz5r/AIF3+yL3/niP+/i/41d0vw3NezOly3kKq5DDDZOfrWLVrTZZItRg8qRk3SKDtbGRkcVNRVeR2kvu/wCCeng54FYiHtKUmr7c3+UV+ZavdBura8kihXzUU4DllGePTNQf2Re/88R/38X/ABqC9Znvp2dix8xuSc96hpxVTlV2vu/4JnXlg/ay5ackrv7S7/4TZ1TTLuS93JECPKjH317IB61T/si9/wCeI/7+L/jS6x/yEP8AtlH/AOgCqNKmp8i1W3b/AIJrjZYb61UvCXxP7S7/AOEu/wBkXv8AzxH/AH8X/Gj+yL3/AJ4j/v4v+NUqKu0+6+7/AIJyc+F/kl/4Ev8A5Au/2Re/88R/38X/ABo/si9/54j/AL+L/jVKii0+6+7/AIIc+F/kl/4Ev/kC7/ZF7/zxH/fxf8aP7Ivf+eI/7+L/AI1SootPuvu/4Ic+F/kl/wCBL/5Au/2Re/8APEf9/F/xo/si9/54j/v4v+NUqKLT7r7v+CHPhf5Jf+BL/wCQLv8AZF7/AM8R/wB/F/xrX0G61XQXmMFpHKJQMBpF+Vh0PX0J/OuboqKlJ1I8s7Nen/BM6scFVg4Tpyaf95f/ACJ3T63qupTNA+lKYZFKLDHdKm0Bw64PbGAKsy3GuzaMNMfREaDY6EteIWOehznt+teeUVyPAR05baa7Pf8A8CPIr5dh3Jewjypa2d3r30cfy+Zs/wDCI63/AM+a/wDf+P8A+KrWPhjV/wDhDUg+yjzPt7Pt85Onlgdc1yFbLf8AIip/2Em/9FLWlaNb3feW/Z//ACRNeOIvC8luvsv/AOSD/hEdb/581/7/AMf/AMVR/wAIjrf/AD5r/wB/4/8A4qsait+Wv/Mvuf8A8kdPLiP5l/4C/wD5Iu6ho19pao19CIxISFxIrZx9CapUUVrHmS956/16m8FJL33d/d+rCiiiqKCiiigAorVt9N0uW3R5tdihkZctGbaRip9MgYNSf2Vo/wD0McP/AICS/wCFYOvBO1n/AOAv/I53iYJ2s/8AwGX+RjVt+K/+QjZ/9g+1/wDRS03+ytH/AOhjh/8AASX/AArY8SabpUl9amXXYoiLG3ABtpDkCNcHgd+tc8q8PbRdns/svy8jlniYe3g7PZ/Zl5eRzdhpcmoQXMySxRJbKGcysRnPQDAPPFa48Dan5as8lvGxGCjM2VbGdhwuM8H296ijtNMht5YI/EsIjmKlx9jl528j+H3rdHie4EYT/hL7baPXTm5+vyc1lWrYi/7rbzjL/L1MMRiMTzfuNvOMuy/uvrf8Dnrrw1NY6PNd3kqRzxMqtbDllzjG704IPGai07w3faparcWuwxEsGYk/Jtxknj3/AJ1d1HWJdRhkt7rWIJI3YFmWzKl8dMkLntUWmapLo6Omn60sSyMGYeQxyRnHUe9aKeI9m/5r9na33HqRweYOi3zQ52/5o2t99/61LKeCrkNGjzCR5nKxeSMh/wB2XGM4PUY5FZ+raF/ZVhaztcpM85YMsfRCO2fWtKXxPfTSwSPrkW+Bi0ZFpjBKlc/d54JqlqN//aqRpearCVjJKhLUpyep4AzUU5YrnTqPTro/Py9CKOBzL2kXVnC3W04+f/AMOirv2Sy/6CSf9+X/AMKPsll/0Ek/78v/AIV3+0Xn9zPT+qVO8f8AwOP+ZSoq79ksv+gkn/fl/wDCj7JZf9BJP+/L/wCFHtF5/cw+qVO8f/A4/wCZSoq79ksv+gkn/fl/8KPsll/0Ek/78v8A4Ue0Xn9zD6pU7x/8Dj/mUqKu/ZLL/oJJ/wB+X/wqC7tzaXckBYOUONwHWmppuxFTD1Kced2tto0/yb7ENbfij/j7tP8Ar1T+tYlbfij/AI+7T/r1T+tZT/iw+Z34X/kX4j1h+bMStu//AORS03/ro/8AM1iVt3//ACKWm/8AXR/5mit8UPX9GGXfwcT/AIP/AG+BiUUUV0HkBRRRQAV0Oof8k90T/r9u/wCUVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/wDQhVarOm/8hW0/67J/6EKAH6z/AMh2/wD+vmT/ANCNFGs/8h2//wCvmT/0I0UAWvFP/I16l/18N/OsmtbxT/yNepf9fDfzrJoAKKKKACiuj0/wRqmp2MV3beX5MoUhiegLEfpjJ+oq7L8ONRSNvLvLWWQNsWNC2WYHDDJGBj3I9qbTW5wSzHCQlyyqK5x9Fa+teHpNGhila6guUkZkJiz8rKcFTkDoQayKR106sKseeDujofGf/IUsf+wXaf8Aola56uh8Z/8AIUsf+wXaf+iVrnq0q/GznwP+7Q9EFFFFZnYFT2H/ACEbb/rqv8xUFT2H/IRtv+uq/wAxUz+Fm2H/AI0PVfmNu/8Aj8m/66N/Ooqlu/8Aj8m/66N/OoqI7IVb+JL1Ze1j/kIf9so//QBVGr2sf8hD/tlH/wCgCqNTS/hr0N8d/vVT/E/zCiiitDjCiiigAooooAKKKKACiirmmaZPq1y0FrgyKhfB7gED+tTKSiuaWxMpRhHmk7Ip0V1S/D/UckS3FvE3mMiqxOX25yRgdMAn6VE/gi8jZg91bqGJ8gkt+/wATjjjr3xXMsbh3tI41mGFbspo5qtlv+RFT/sJN/6KWsYjBIrZb/kRU/7CTf8Aopa0rfZ9Ua4jeH+JfqY1FFFbnSFFFFABRRRQAUUUUAFFFFABW34r/wCQjZ/9g+1/9FLWJW34r/5CNn/2D7X/ANFLXPP+ND0f6HLP/eIekv0MSitPRdCutdmaOzKAoyhtxxjcTz9Bita38ETC5so9Qu44Gu22rEATIOCT2xxj1oqYqjTbjKWqFVxlClJxnLVdOvc5aium1PwVdWMNzPDPHLFCSUHIaRR95hxgYOeDzxxWZpGg3WtiT7EULRuqlCecHPP0GP1ojiaMoOalohwxlCdN1FLRGZRXUHwXLFCZZrkPH8gzGjDDGRUKkMB6k+nFVtT8Mf2Xpk9xNcqZ4ZhG0AHKZ6bj0zjB4z1qY4ujJ2TIjjsPOSjGWrdjAooorqO0KKKKACiiigAq7rH/ACFp/wDeH8hVKrusf8haf/eH8hWb+Nej/Q64f7rP/FH8pFKtvxR/x92n/Xqn9axK2/FH/H3af9eqf1rOf8WHzOzC/wDIvxHrD82Ylbd//wAilpv/AF0f+ZrErbv/APkUtN/66P8AzNFb4oev6MMu/g4n/B/7fAxKKKK6DyAooooAK6HUP+Se6J/1+3f8oq56uh1D/knuif8AX7d/yirSG0vT9UceI/iUv8X/ALbI56iiiszsCiiigAooooAKs6b/AMhW0/67J/6EKrVZ03/kK2n/AF2T/wBCFAD9Z/5Dt/8A9fMn/oRoo1n/AJDt/wD9fMn/AKEaKALXin/ka9S/6+G/nWTWt4p/5GvUv+vhv51k0AFFFFAGjbeINVs7L7JbXskcGGGwY6MAG/kK6Sw8a21vYmO4m1gyynfOYpYArOeSRmMkc+9cVRVKTRx1sDh6y96K7nZ6n4m8PavZQW99aas/ksWDrPCrHPriPFZfneEP+fPWv/AqL/43WBRVe0fZfcRTwFKlHlg2l6s7jxdJ4dGp2fn22pk/2da7dk8YwvlLjPydcdawfN8L/wDPrq3/AIER/wDxFT+M/wDkKWP/AGC7T/0Stc9WNainUk7vfuzHBYdPDQfM9l1ZstL4Z2nba6rnHGbiP/4iqe7S/wDnld/9/F/+JqlRWapJdX956+Hl9XTS1v8Aza/mXd2l/wDPK7/7+L/8TU9m2m/brfZFdBvNXGZFxnP0rLqew/5CNt/11X+YolBcr1f3nfQxUnVj7sd19ldy3ctpn2qXdFdZ3nOJF9fpUW7S/wDnld/9/F/+Jqvd/wDH5N/10b+dRU401Zav7xVcVL2kvdju/so2dUbTvtv72O5LeVH92RQMbBjtVPdpf/PK7/7+L/8AE0usf8hD/tlH/wCgCqNRTguRavbubY3FSWKqLlj8T+yu5d3aX/zyu/8Av4v/AMTRu0v/AJ5Xf/fxf/iapUVp7Nd395yfWpfyx/8AAUXd2l/88rv/AL+L/wDE0btL/wCeV3/38X/4mqVFHs13f3h9al/LH/wFF3dpf/PK7/7+L/8AE0btL/55Xf8A38X/AOJqlRR7Nd394fWpfyx/8BRd3aX/AM8rv/v4v/xNG7S/+eV3/wB/F/8AiapUUezXd/eH1qX8sf8AwFF3dpf/ADyu/wDv4v8A8TU9pfWdhdLcWf22KVejLIv/AMTWXRSdJNWYniXJWcI2/wAKOkg8TKt1BJPPqTCHG3bLHkYXaMZQ9uK1IvG9itx5sy6pccEbZJIcc/SMY/CuHornngaE90eXicHh8TLmlBLS2it+RtmXwwST9l1b/wACI/8A4itUyeHf+ENQ/ZtT8j7e3Hnx7t3ljvsxjFcfWy3/ACIqf9hJv/RS0qtFLl1e/dnPXw6XJ7z3XVi+b4X/AOfXVv8AwIj/APiKPN8L/wDPrq3/AIER/wDxFYtFb+wX8z+9nT9Wj/NL72XtQfSmVP7KhvI2yd/2mVWyO2MKKo0UVrGPKrG0I8kbX+8KKKKosKKKKANa3k8PC3jF1bak0235zHPGFJ9gUJxUnm+F/wDn11b/AMCI/wD4isWisHRTd7v72czw6bvzP72bXm+F/wDn11b/AMCI/wD4itjxJJ4eF9a/abbUmb7Db7fLnjA2+WuOqdcda42tvxX/AMhGz/7B9r/6KWueVFe2jq9n1fkcs8OvbwXM9n1fkT2mp6DYNI1nHrERkQoxW5i5B7fcqzJ4k0uW+jvJDrLXETM6OZ4sqW6/8s65SitHhKbd3e/qaywNGT5ndv1OmufF0tzBLbyXF80EgIKExcj6hKoWOqW2mStJYtewuy7WKyJyP++ayKKqOFpRi4paM9KEqcIckaULf4Vr66HSz+Lp7mLy57m+dNwbBMfUHI/g9RUF74hXUbcw3sl5KjNuYZjBJ9SQuTWDRSjhKMdVEIypRacaUFb+5H/Iu7tL/wCeV3/38X/4mjdpf/PK7/7+L/8AE1Sorb2a7v7zf61L+WP/AICi7u0v/nld/wDfxf8A4mjdpf8Azyu/+/i//E1Soo9mu7+8PrUv5Y/+Ao07SHTLu7igVLpTIwUEyLx+lZhGGI96u6P/AMhm0/66iqbfeP1pRVpten6l1pKeHhPlSd5LRW6R/wAxKu6x/wAhaf8A3h/IVSq7rH/IWn/3h/IU38a9H+hEP91n/ij+UilW34o/4+7T/r1T+tYlbfij/j7tP+vVP61nP+LD5nZhf+RfiPWH5sxK27//AJFLTf8Aro/8zWJW3f8A/Ipab/10f+ZorfFD1/Rhl38HE/4P/b4GJRRRXQeQFFFFABXQ6h/yT3RP+v27/lFXPV0Oof8AJPdE/wCv27/lFWkNpen6o48R/Epf4v8A22Rz1FFFZnYFFFFABRRRQAVZ03/kK2n/AF2T/wBCFVqs6b/yFbT/AK7J/wChCgB+s/8AIdv/APr5k/8AQjRRrP8AyHb/AP6+ZP8A0I0UAWvFP/I16l/18N/OsmtbxT/yNepf9fDfzrJoAKKKKACiiigAooooA6Hxn/yFLH/sF2n/AKJWuerofGf/ACFLH/sF2n/ola56tKvxs48D/u0PRBRRRWZ2BU9h/wAhG2/66r/MVBU9h/yEbb/rqv8AMVM/hZth/wCND1X5jbv/AI/Jv+ujfzqKpbv/AI/Jv+ujfzqKiOyFW/iS9WXtY/5CH/bKP/0AVRq9rH/IQ/7ZR/8AoAqjU0v4a9DfHf71U/xP8wooorQ4wooooAKKKKACiiigAooooAKKKKACvUfhvoFl4j8E6nZahHlWuso4+9G2wcg15dXaeCvH6eEdNuLVtPa686XzNwl2Y4Ax0PpXmZpTr1cM1Q+K6a+88fOaOIrYRxw3x3TWttmZ/izwTqHhSRGnIntZOFnjBwD6H0P865uvUL/4uWWp2MlpfeHjLDKMMrXI/wDia4z7Z4X/AOgPqP8A4Hr/APG6nB18X7O2Jpvm7q2v4k4DEY/2VsZSfMuqcdfxMKit37Z4X/6A+o/+B6//ABuj7Z4X/wCgPqP/AIHr/wDG67fbS/59v8P8z0PrE/8An1L/AMl/+SMKit37Z4X/AOgPqP8A4Hr/APG6Ptnhf/oD6j/4Hr/8bo9tL/n2/wAP8w+sT/59S/8AJf8A5IwqK3ftnhf/AKA+o/8Agev/AMbo+2eF/wDoD6j/AOB6/wDxuj20v+fb/D/MPrE/+fUv/Jf/AJIwqK3ftnhf/oD6j/4Hr/8AG6fDqHhaGdJDol9IEYNse+Uq3sf3fSl7af8Az7f4f5ieInb+FL/yX/5I3vh/8P5NckTU9WQpp6nKIeDOf/iffvVD4oRpF47uI4kVEWGIKqjAA2DgCuni+M1vDEscWglEUYVVuQAB/wB81wXivXx4l8Qy6mtubcSKq+WX3YwMdcCvJwkcdUxrrYiPLGzSV1pquzPDwMcyq5i6+Khyw5Wkrp21XZ7+ZjUUUV759OFFFFABRRRQAUUUUAFFFFAF3R/+Qzaf9dRVNvvH61c0f/kM2n/XUVTb7x+tZr+I/Rfqdkv90h/il+URKu6x/wAhaf8A3h/IVSq7rH/IWn/3h/IUP416P9BQ/wB1n/ij+UilW34o/wCPu0/69U/rWJW34o/4+7T/AK9U/rWc/wCLD5nZhf8AkX4j1h+bMStu/wD+RS03/ro/8zWJW3f/APIpab/10f8AmaK3xQ9f0YZd/BxP+D/2+BiUUUV0HkBRRRQAV0Oof8k90T/r9u/5RVz1dDqH/JPdE/6/bv8AlFWkNpen6o48R/Epf4v/AG2Rz1FFFZnYFFFFABRRRQAVZ03/AJCtp/12T/0IVWqzpv8AyFbT/rsn/oQoAfrP/Idv/wDr5k/9CNFGs/8AIdv/APr5k/8AQjRQBa8U/wDI16l/18N/OsmtbxT/AMjXqX/Xw386yaACiiigAooooAKKKKAOh8Z/8hSx/wCwXaf+iVrnq6Hxn/yFLH/sF2n/AKJWuerSr8bOPA/7tD0QUUUVmdgVPYf8hG2/66r/ADFQVPYf8hG2/wCuq/zFTP4WbYf+ND1X5jbv/j8m/wCujfzqKpbv/j8m/wCujfzqKiOyFW/iS9WXtY/5CH/bKP8A9AFUavax/wAhD/tlH/6AKo1NL+GvQ3x3+9VP8T/MKKKK0OMKKKKACiiigAooooAKKKKACiiigArrvAttbS2+uz3NrBcva2LSxCeMOFYZ5wa5Gu1+Hkz29p4imiIDppzMpIB5APY8GuHMG1hpW8vzR5uaNrCSt5fmjCl8STSxMh0/S1DDGVsYwR9DitXxRY2tt4U0Ga3t4opZocyOiAFz7nvWLd+ItSvrZoLmaJo26hbeNT+YUGuh8Xf8ib4c/wCuFYzi6dWkrWu3s/J+SMKkXTrUUko3k9nf7L8kWPF93FoNxp0Gn6ZpoSWxjlcyWaOSxzk5I9qzNAt4/FGtu+oQW0NvZ27TyR28QiDgY4+X3IrZ8da7f6bd6XDZyxqh06JiGgR+cnuwJrI8Ja3G3iC6/teVE+32rW/m7VRVJIIJAAHauWiqiwftIrW299d/Tt5nFQjVWX+1jH3rPVNuW/p28wsfEFjf6xHaXWiWCWE7rFhIVWSNTxu3gZz3rH8R6T/YmvXNiCWSNv3bH+JT0NbNh4K1W11yJ7+DyrCCRZJLtztiMYOchuh4rN8W6tHrPiW5uoDmDOyI4xlR0rroSh7e1B3jbXqr9Pnud2HlT+spYd3hy663V7q3ztcxa6ZfBciJaC/1Wxsbi7UNFBOW3YPTJAIGfc1zcTiOZHYbgrAketd74r0DUfEeqWN9o0LXVrcWyKJV5SIjggntitMVWlCcY83KnfX06amuMryp1IQ5+SLvd6dNlrp/wxV0vw7YnwXrE95dWgu4ZEUMQzeT14yARzjqM1j6z4XbRLCCe71C1MtxEk0Vugcuyt3zjHH1rZ0LT7iTwL4ktLdRcSxzRE+Sd4IG7JBHUVW8fnN5pXPTTof/AEEVyUatT604c+jb7fyxfyOGhWq/XHTU7pyd9tlGLXonc0Nf8IR3urXs1nLaadaWVvC8u5SANwPOFB9K4WeNYriSOOVZlViBIgIDj1Gea9S8QH/iX+JcH/l2tf5PXlVa5bUqVKb5ntZfgn+ptlFWpUpPnldKyX/gKfr1CiiivVPbCiiigAooooAKKKKACiiigC7o/wDyGbT/AK6iqbfeP1q5o/8AyGbT/rqKpt94/Ws1/Efov1OyX+6Q/wAUvyiJV3WP+QtP/vD+QqlV3WP+QtP/ALw/kKH8a9H+gof7rP8AxR/KRSrb8Uf8fdp/16p/WsStvxR/x92n/Xqn9azn/Fh8zswv/IvxHrD82Ylbd/8A8ilpv/XR/wCZrErbv/8AkUtN/wCuj/zNFb4oev6MMu/g4n/B/wC3wMSiiiug8gKKKKACuh1D/knuif8AX7d/yirnq6HUP+Se6J/1+3f8oq0htL0/VHHiP4lL/F/7bI56iiiszsCiiigAooooAKs6b/yFbT/rsn/oQqtVnTf+Qraf9dk/9CFAD9Z/5Dt//wBfMn/oRoo1n/kO3/8A18yf+hGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAUUUUAFFFFAHQ+M/8AkKWP/YLtP/RK1z1dD4z/AOQpY/8AYLtP/RK1z1aVfjZx4H/doeiCiiiszsCp7D/kI23/AF1X+YqCp7D/AJCNt/11X+YqZ/CzbD/xoeq/Mbd/8fk3/XRv51FUt3/x+Tf9dG/nUVEdkKt/El6svax/yEP+2Uf/AKAKo1e1j/kIf9so/wD0AVRqaX8Nehvjv96qf4n+YUUUVocYUUUUAFFFFABRRRQAUUUUAFFFFABShmXO0kZ4OD1pKKAClLMQASSB0BPSkooAVmZvvEntyaSiigB5mkK4MjEem41bsLnToEYahp0l2xPylbkx4/8AHTmqNFTKKkrP/L8iJQUlyv8ABtfkbD6hoRRgmhzKxHB+3k4P/fFZXnOMhHZV7AMeKZRUxpxjtf5tv8yYUow2v823+bYqu6ghWYZ64PWgsW+8SfqaSitDWw4yOc5Zjnrz1ptFFABRRRQAUUUUAFFFFABRRRQAUUUUAXdH/wCQzaf9dRVNvvH61c0f/kM2n/XUVTb7x+tZr+I/Rfqdkv8AdIf4pflESrusf8haf/eH8hVKrusf8haf/eH8hQ/jXo/0FD/dZ/4o/lIpVt+KP+Pu0/69U/rWJW34o/4+7T/r1T+tZz/iw+Z2YX/kX4j1h+bMStu//wCRS03/AK6P/M1iVt3/APyKWm/9dH/maK3xQ9f0YZd/BxP+D/2+BiUUUV0HkBRRRQAV0Oof8k90T/r9u/5RVz1dDqH/ACT3RP8Ar9u/5RVpDaXp+qOPEfxKX+L/ANtkc9RRRWZ2BRRRQAUUUUAFWdN/5Ctp/wBdk/8AQhVarOm/8hW0/wCuyf8AoQoAfrP/ACHb/wD6+ZP/AEI0Uaz/AMh2/wD+vmT/ANCNFAFrxT/yNepf9fDfzrJrW8U/8jXqX/Xw386yaACiiigAooooAKKKKAOh8Z/8hSx/7Bdp/wCiVrnq6Hxn/wAhSx/7Bdp/6JWuerSr8bOPA/7tD0QUUUVmdgVPYf8AIRtv+uq/zFQVPYf8hG2/66r/ADFTP4WbYf8AjQ9V+Y27/wCPyb/ro386iqW7/wCPyb/ro386iojshVv4kvVl7WP+Qh/2yj/9AFUavax/yEP+2Uf/AKAKo1NL+GvQ3x3+9VP8T/MKKKK0OMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigC7o//IZtP+uoqm33j9auaP8A8hm0/wCuoqm33j9azX8R+i/U7Jf7pD/FL8oiVd1j/kLT/wC8P5CqVXdY/wCQtP8A7w/kKH8a9H+gof7rP/FH8pFKtvxR/wAfdp/16p/WsStvxR/x92n/AF6p/Ws5/wAWHzOzC/8AIvxHrD82Ylbd/wD8ilpv/XR/5msStu//AORS03/ro/8AM0Vvih6/owy7+Dif8H/t8DEoooroPICiiigArodQ/wCSe6J/1+3f8oq56uh1D/knuif9ft3/ACirSG0vT9UceI/iUv8AF/7bI56iiiszsCiiigAooooAKs6b/wAhW0/67J/6EKrVZ03/AJCtp/12T/0IUAP1n/kO3/8A18yf+hGijWf+Q7f/APXzJ/6EaKALXin/AJGvUv8Ar4b+dZNa3in/AJGvUv8Ar4b+dZNABRRRQAUUUUAFFFFAHQ+M/wDkKWP/AGC7T/0Stc9XQ+M/+QpY/wDYLtP/AEStc9WlX42ceB/3aHogooorM7Aqew/5CNt/11X+YqCp7D/kI23/AF1X+YqZ/CzbD/xoeq/Mbd/8fk3/AF0b+dRVLd/8fk3/AF0b+dRUR2Qq38SXqy9rH/IQ/wC2Uf8A6AKo1e1j/kIf9so//QBVGppfw16G+O/3qp/if5hRRRWhxhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAXdH/AOQzaf8AXUVTb7x+tXNH/wCQzaf9dRVNvvH61mv4j9F+p2S/3SH+KX5REq7rH/IWn/3h/IVSq7rH/IWn/wB4fyFD+Nej/QUP91n/AIo/lIpVt+KP+Pu0/wCvVP61iVt+KP8Aj7tP+vVP61nP+LD5nZhf+RfiPWH5sxK27/8A5FLTf+uj/wAzWJW3f/8AIpab/wBdH/maK3xQ9f0YZd/BxP8Ag/8Ab4GJRRRXQeQFFFFABXQ6h/yT3RP+v27/AJRVz1dDqH/JPdE/6/bv+UVaQ2l6fqjjxH8Sl/i/9tkc9RRRWZ2BRRRQAUUUUAFWdN/5Ctp/12T/ANCFVqs6b/yFbT/rsn/oQoAfrP8AyHb/AP6+ZP8A0I0Uaz/yHb//AK+ZP/QjRQBa8U/8jXqX/Xw386ya1vFP/I16l/18N/OsmgAooooAKKKKACiiigDofGf/ACFLH/sF2n/ola56uh8Z/wDIUsf+wXaf+iVrnq0q/GzjwP8Au0PRBRRRWZ2BU9h/yEbb/rqv8xUFT2H/ACEbb/rqv8xUz+Fm2H/jQ9V+Y27/AOPyb/ro386iqW7/AOPyb/ro386iojshVv4kvVl7WP8AkIf9so//AEAVRq9rH/IQ/wC2Uf8A6AKo1NL+GvQ3x3+9VP8AE/zOjt7qLTvDFtcfY7ed5JWUmVAfWoP+EkT/AKBVl/37pt1/yJ1l/wBfDf1rFrmp0YT5nJdX+Z7OMzHFYX2VOjKy5IdF1iu6Nz/hJE/6BVl/37o/4SRP+gVZf9+6w6K1+rUu35nF/bWP/n/CP+Ruf8JIn/QKsv8Av3R/wkif9Aqy/wC/dYdFH1al2/MP7ax/8/4R/wAjc/4SRP8AoFWX/fuj/hJE/wCgVZf9+6w6KPq1Lt+Yf21j/wCf8I/5G5/wkif9Aqy/790f8JIn/QKsv+/dYdFH1al2/MP7ax/8/wCEf8jc/wCEkT/oFWX/AH7o/wCEkT/oFWX/AH7rDoo+rUu35h/bWP8A5/wj/kbn/CSJ/wBAqy/790f8JIn/AECrL/v3WHRR9WpdvzD+2sf/AD/hH/I3P+EkT/oFWX/fuj/hJE/6BVl/37rDoo+rUu35h/bWP/n/AAj/AJG5/wAJIn/QKsv+/dH/AAkif9Aqy/791h0UfVqXb8w/trH/AM/4R/yNz/hJE/6BVl/37o/4SRP+gVZf9+6w6KPq1Lt+Yf21j/5/wj/kbn/CSJ/0CrL/AL90f8JIn/QKsv8Av3WHRR9WpdvzD+2sf/P+Ef8AI3P+EkT/AKBVl/37o/4SRP8AoFWX/fusOij6tS7fmH9tY/8An/CP+Ruf8JIn/QKsv+/dH/CSJ/0CrL/v3WHRR9WpdvzD+2sf/P8AhH/I3P8AhJE/6BVl/wB+6P8AhJE/6BVl/wB+6w6KPq1Lt+Yf21j/AOf8I/5G5/wkif8AQKsv+/dH/CSJ/wBAqy/791h0UfVqXb8w/trH/wA/4R/yNz/hJE/6BVl/37o/4SRP+gVZf9+6w6KPq1Lt+Yf21j/5/wAI/wCRuf8ACSJ/0CrL/v3R/wAJIn/QKsv+/dYdFH1al2/MP7ax/wDP+Ef8jc/4SRP+gVZf9+6P+EkT/oFWX/fusOij6tS7fmH9tY/+f8I/5HS6fr6z6jBENNtI97gblTkfSq58SKGP/Eqsuv8AzzrP0f8A5DNp/wBdRVNvvH61Cw9Lnat0XfzOqWcY76tGXPrzS6R7R8jb/wCEkT/oFWX/AH7qzqGvrBfyxnTbSTafvMnJ4rmqu6x/yFp/94fyFDw9LnSt0ffyFHOMd9XlLn15o9I9peXkX/8AhJE/6BVl/wB+60tb1lbOe3U2FtNvgV8yJnGc8D2rka2/FH/H3af9eqf1qJ4emqsVbudWHzbGywVebnqnC2i6t+Qv/CSJ/wBAqy/791V1PWW1K3ih+zxQJExIEYwOazaK3jQpxfMkeRVzbG1qbpzno99EvyQUUUVueWFFFFABXQ6h/wAk90T/AK/bv+UVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/9CFVqs6b/wAhW0/67J/6EKAH6z/yHb//AK+ZP/QjRRrP/Idv/wDr5k/9CNFAFrxT/wAjXqX/AF8N/OsmtbxT/wAjXqX/AF8N/OsmgAooooAKKKKACiiigDofGf8AyFLH/sF2n/ola56uh8Z/8hSx/wCwXaf+iVrnq0q/GzjwP+7Q9EFFFFZnYFT2H/IRtv8Arqv8xUFT2H/IRtv+uq/zFTP4WbYf+ND1X5jbv/j8m/66N/Ooqlu/+Pyb/ro386iojshVv4kvVl7WP+Qh/wBso/8A0AVRq9rH/IQ/7ZR/+gCqNTS/hr0N8d/vVT/E/wAzauv+ROsv+vhv61i1tXX/ACJ1l/18N/WsWoobS9X+Z1Zp/Epf9e4f+ko2dHs7GXTby7v45JBblcBGweaX7R4d/wCfK7/77H+NO0v/AJFfVv8AgH86w6zjDnnO7ej7+SOutiFhcNh+SnB80W23FN35pLd+SNr7R4d/58rv/vsf40faPDv/AD5Xf/fY/wAaxaK09gv5n97OT+1J/wDPqn/4BH/I2vtHh3/nyu/++x/jR9o8O/8APld/99j/ABrFoo9gv5n97D+1J/8APqn/AOAR/wAja+0eHf8Anyu/++x/jR9o8O/8+V3/AN9j/GsWij2C/mf3sP7Un/z6p/8AgEf8ja+0eHf+fK7/AO+x/jR9o8O/8+V3/wB9j/GsWij2C/mf3sP7Un/z6p/+AR/yNr7R4d/58rv/AL7H+NH2jw7/AM+V3/32P8axaKPYL+Z/ew/tSf8Az6p/+AR/yNr7R4d/58rv/vsf40faPDv/AD5Xf/fY/wAaxaKPYL+Z/ew/tSf/AD6p/wDgEf8AI2vtHh3/AJ8rv/vsf40faPDv/Pld/wDfY/xrFoo9gv5n97D+1J/8+qf/AIBH/I2vtHh3/nyu/wDvsf40faPDv/Pld/8AfY/xrFoo9gv5n97D+1J/8+qf/gEf8ja+0eHf+fK7/wC+x/jR9o8O/wDPld/99j/GsWij2C/mf3sP7Un/AM+qf/gEf8ja+0eHf+fK7/77H+NH2jw7/wA+V3/32P8AGsWij2C/mf3sP7Un/wA+qf8A4BH/ACNr7R4d/wCfK7/77H+NH2jw7/z5Xf8A32P8axaKPYL+Z/ew/tSf/Pqn/wCAR/yNr7R4d/58rv8A77H+NH2jw7/z5Xf/AH2P8axaKPYL+Z/ew/tSf/Pqn/4BH/I2vtHh3/nyu/8Avsf40faPDv8Az5Xf/fY/xrFoo9gv5n97D+1J/wDPqn/4BH/I2vtHh3/nyu/++x/jR9o8O/8APld/99j/ABrFoo9gv5n97D+1J/8APqn/AOAR/wAja+0eHf8Anyu/++x/jR9o8O/8+V3/AN9j/GsWij2C/mf3sP7Un/z6p/8AgEf8ja+0eHf+fK7/AO+x/jR9o8O/8+V3/wB9j/GsWij2C/mf3sP7Un/z6p/+AR/yOj06fQm1KAW9pdLKXGws/AP51Xa48Pbjmyu+v98f41R0f/kM2n/XUVTb7x+tZqiudq72XX1OqWZT+rRl7OHxS+wu0TZ+0eHf+fK7/wC+x/jVnUZ9CXUJRcWl00mfmKvwePrXOVd1j/kLT/7w/kKHRXOveez6+go5lP6tN+zh8UfsLtIu/aPDv/Pld/8AfY/xqvreow6ldRSWyOiRxCMB8Z4zWbRWsaMYy5rt+rOKtmVWrSdHljFO1+WKW22wUUUVseaFFFFABRRRQAV0Oof8k90T/r9u/wCUVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/wDQhVarOm/8hW0/67J/6EKAH6z/AMh2/wD+vmT/ANCNFGs/8h2//wCvmT/0I0UAWvFP/I16l/18N/OsmtbxT/yNepf9fDfzrJoAKKKKACiiigAooooA6Hxn/wAhSx/7Bdp/6JWuerofGf8AyFLH/sF2n/ola56tKvxs48D/ALtD0QUUUVmdgVPYf8hG2/66r/MVBU9h/wAhG2/66r/MVM/hZth/40PVfmNu/wDj8m/66N/Ooqlu/wDj8m/66N/OoqI7IVb+JL1Ze1j/AJCH/bKP/wBAFUavax/yEP8AtlH/AOgCqNTS/hr0N8d/vVT/ABP8zauv+ROsv+vhv61i1tXX/InWX/Xw39axaihtL1f5nVmn8Sl/17h/6Sjc0v8A5FfVv+AfzrDrc0v/AJFfVv8AgH86w6VL45+v6IrMP93wv+B/+lzLNjbR3MknnyNHHHGZCVXceMdsj1qbytK/5+7r/vwP/iqbp33bz/r2b+YqlV2cpPU51OnSowfs027737+TRf8AK0r/AJ+7r/vwP/iqPK0r/n7uv+/A/wDiqoUU+R/zP8P8iPrMP+fUf/Jv/ki/5Wlf8/d1/wB+B/8AFUeVpX/P3df9+B/8VVCijkf8z/D/ACD6zD/n1H/yb/5Iv+VpX/P3df8Afgf/ABVHlaV/z93X/fgf/FVQoo5H/M/w/wAg+sw/59R/8m/+SL/laV/z93X/AH4H/wAVR5Wlf8/d1/34H/xVUKKOR/zP8P8AIPrMP+fUf/Jv/ki/5Wlf8/d1/wB+B/8AFUeVpX/P3df9+B/8VVCijkf8z/D/ACD6zD/n1H/yb/5Iv+VpX/P3df8Afgf/ABVHlaV/z93X/fgf/FVQoo5H/M/w/wAg+sw/59R/8m/+SL/laV/z93X/AH4H/wAVR5Wlf8/d1/34H/xVUKKOR/zP8P8AIPrMP+fUf/Jv/ki/5Wlf8/d1/wB+B/8AFUeVpX/P3df9+B/8VVCijkf8z/D/ACD6zD/n1H/yb/5Iv+VpX/P3df8Afgf/ABVHlaV/z93X/fgf/FVQoo5H/M/w/wAg+sw/59R/8m/+SL/laV/z93X/AH4H/wAVR5Wlf8/d1/34H/xVUKKOR/zP8P8AIPrMP+fUf/Jv/ki/5Wlf8/d1/wB+B/8AFUeVpX/P3df9+B/8VVCijkf8z/D/ACD6zD/n1H/yb/5Iv+VpX/P3df8Afgf/ABVHlaV/z93X/fgf/FVQoo5H/M/w/wAg+sw/59R/8m/+SL/laV/z93X/AH4H/wAVR5Wlf8/d1/34H/xVUKKOR/zP8P8AIPrMP+fUf/Jv/ki/5Wlf8/d1/wB+B/8AFUeVpX/P3df9+B/8VVCijkf8z/D/ACD6zD/n1H/yb/5Iv+VpX/P3df8Afgf/ABVHlaV/z93X/fgf/FVQoo5H/M/w/wAg+sw/59R/8m/+SNrS4tNGq2xiubhn8wbQ0IAJ+u6qrRaVuP8Apd11/wCeA/8AiqZo/wDyGbT/AK6iqbfeP1rNQfO/eey7efkdUsTD6rB+yj8Uv5u0f7xe8rSv+fu6/wC/A/8Aiqh1CdLnUJZot2xj8u4YPSq1FaqFndu5xVMRzw9nGCirp6X6X7t92FFFFWcoUUUUAFFFFABRRRQAV0Oof8k90T/r9u/5RVz1dDqH/JPdE/6/bv8AlFWkNpen6o48R/Epf4v/AG2Rz1FFFZnYFFFFABRRRQAVZ03/AJCtp/12T/0IVWqzpv8AyFbT/rsn/oQoAfrP/Idv/wDr5k/9CNFGs/8AIdv/APr5k/8AQjRQBa8U/wDI16l/18N/OsmtbxT/AMjXqX/Xw386yaACiiigAooooAKKKKAOh8Z/8hSx/wCwXaf+iVrnq6Hxn/yFLH/sF2n/AKJWuerSr8bOPA/7tD0QUUUVmdgVPYf8hG2/66r/ADFQVPYf8hG2/wCuq/zFTP4WbYf+ND1X5jbv/j8m/wCujfzqKpbv/j8m/wCujfzqKiOyFW/iS9WXtY/5CH/bKP8A9AFUavax/wAhD/tlH/6AKo1NL+GvQ3x3+9VP8T/M2rr/AJE6y/6+G/rWLW1df8idZf8AXw39axaihtL1f5nVmn8Sl/17h/6Sjc0v/kV9W/4B/OsOtzS/+RX1b/gH86w6VL45+v6IrMP93wv+B/8Apcy7p33bz/r2b+YqlV3Tvu3n/Xs38xVKtI/Eziq/wKfz/Mkt4HurmOCLG+RtoycDNWf7N/6fLT/v7/8AWo0f/kM2n/XUVTb7x+tD5nKyZUI0oUFUnG7ba3tsl/mXP7N/6fbT/v7/APWo/s3/AKfbT/v7/wDWqlRRyy7k+1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZd/s3/p9tP8Av7/9aj+zf+n20/7+/wD1qpUUcsu4e1of8+/xZq6faJa6jBPLe2uyNwxxJz/KstuWP1pKKcYtO7YqteM6apwjZJt733t/kFFFFWcoUUUUAFFFFABRRRQAUUUUAFdDqH/JPdE/6/bv+UVc9XQ6h/yT3RP+v27/AJRVpDaXp+qOPEfxKX+L/wBtkc9RRRWZ2BRRRQAUUUUAFWdN/wCQraf9dk/9CFVqs6b/AMhW0/67J/6EKAH6z/yHb/8A6+ZP/QjRRrP/ACHb/wD6+ZP/AEI0UAWvFP8AyNepf9fDfzrJrW8U/wDI16l/18N/OsmgAooooAKKKKACiiigDofGf/IUsf8AsF2n/ola56uh8Z/8hSx/7Bdp/wCiVrnq0q/GzjwP+7Q9EFFFFZnYFT2H/IRtv+uq/wAxUFT2H/IRtv8Arqv8xUz+Fm2H/jQ9V+Y27/4/Jv8Aro386iqW7/4/Jv8Aro386iojshVv4kvVl7WP+Qh/2yj/APQBVGr2sf8AIQ/7ZR/+gCqNTS/hr0N8d/vVT/E/zNq6/wCROsv+vhv61i1tXX/InWX/AF8N/WsWoobS9X+Z1Zp/Epf9e4f+ko3NL/5FfVv+AfzrDrc0v/kV9W/4B/OsOlS+Ofr+iKzD/d8L/gf/AKXMu6d928/69m/mKpVd077t5/17N/MVSrSPxM4qv8Cn8/zLuj/8hm0/66iqbfeP1q5o/wDyGbT/AK6iqbfeP1oX8R+i/Ucv90h/il+URKKKK0OMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuh1D/knuif9ft3/KKuerodQ/5J7on/AF+3f8oq0htL0/VHHiP4lL/F/wC2yOeooorM7AooooAKKKKACrOm/wDIVtP+uyf+hCq1WdN/5Ctp/wBdk/8AQhQA/Wf+Q7f/APXzJ/6EaKNZ/wCQ7f8A/XzJ/wChGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAUUUUAFFFFAHQ+M/+QpY/wDYLtP/AEStc9XQ+M/+QpY/9gu0/wDRK1z1aVfjZx4H/doeiCiiiszsCp7D/kI23/XVf5ioKnsP+Qjbf9dV/mKmfws2w/8AGh6r8xt3/wAfk3/XRv51FUt3/wAfk3/XRv51FRHZCrfxJerL2sf8hD/tlH/6AKo1e1j/AJCH/bKP/wBAFUaml/DXob47/eqn+J/mbV1/yJ1l/wBfDf1rFrauv+ROsv8Ar4b+tYtRQ2l6v8zqzT+JS/69w/8ASUbml/8AIr6t/wAA/nWHW5pf/Ir6t/wD+dYdKl8c/X9EVmH+74X/AAP/ANLmXdO+7ef9ezfzFUqu6d928/69m/mKpVpH4mcVX+BT+f5l3R/+Qzaf9dRVNvvH61c0f/kM2n/XUVTb7x+tC/iP0X6jl/ukP8UvyiJRRRWhxhRRRQAUUUUAFFFFABRRRQAUUVr+FLWC98V6fbXcSywyS4dG6MMGoqTVODm+iuZ1aipU5VH0Tf3GRRXc6atrqHi5tJn8P2SWrPJH5kcbq6gA4bJYjjGelZPhvRbe+8WPDcfNYWjtJMW6GNT0J9xXL9cilJyVrK/3nGsfBKTmmuVKXTZ37ddDnKK6bxnp1lb3FlqGkwCCxv4BJHEDnYR1GfxrS8YeHrKHRdMutItDHJ5cUdwE6MzIrBvqS2KFjab5NPiv8rd/yFHMKb9ndNc9/k1un89Dh6K7Hxro1hpGj6QllAEn/eR3EneRgEOT/wB9Vx1bYevGvTVSO2v4Ox0YXERxNJVYbO/4OwUUUVudIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFdDqH/ACT3RP8Ar9u/5RVz1dDqH/JPdE/6/bv+UVaQ2l6fqjjxH8Sl/i/9tkc9RRRWZ2BRRRQAUUUUAFWdN/5Ctp/12T/0IVWqzpv/ACFbT/rsn/oQoAfrP/Idv/8Ar5k/9CNFGs/8h2//AOvmT/0I0UAWvFP/ACNepf8AXw386ya1vFP/ACNepf8AXw386yaACiiigAooooAKKKKAOh8Z/wDIUsf+wXaf+iVrnq6Hxn/yFLH/ALBdp/6JWuerSr8bOPA/7tD0QUUUVmdgVPYf8hG2/wCuq/zFQVPYf8hG2/66r/MVM/hZth/40PVfmNu/+Pyb/ro386iqW7/4/Jv+ujfzqKiOyFW/iS9WXtY/5CH/AGyj/wDQBVGr2sf8hD/tlH/6AKo1NL+GvQ3x3+9VP8T/ADNq6/5E6y/6+G/rWLW1df8AInWX/Xw39axaihtL1f5nVmn8Sl/17h/6Sjc0v/kV9W/4B/OsOtzS/wDkV9W/4B/OsOlS+Ofr+iKzD/d8L/gf/pcy7p33bz/r2b+YqlV3Tvu3n/Xs38xVKtI/Eziq/wACn8/zLuj/APIZtP8ArqKpt94/Wrmj/wDIZtP+uoqm33j9aF/Efov1HL/dIf4pflESiiitDjCiiigAooooAKKKKACiiigArX8K3sGn+KrC7u5PKhil3O+CdowfSsiioqQVSDg+qsZ1aaq05U3s01953Vh43muNS1DTtY1Kd9KvWdRPvbdCOdpXvjoCPSqtpqWneGdB1GPTby3v726mEXzQlkMI9QwHXJ/KuPorj+oUlotE7XWlnY4P7MoJ2jpF2ulazt30+866TX7LV/A9zY6gLa0uraUS2iW8OwSZyGGFGB1rctfFOinUBHe3W6zS1t5Fwp/10SqcdPVcV5rRUzy+lJNXaWr9L22+4mplVCaau0rt6dL22+6/zZ0viTWodW0TTgJ/MuUmneZcHKhtmP5VzVFFdlGlGjDkjtr+Lud1ChGhD2cNtfxdwooorU3CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArodQ/5J7on/AF+3f8oq56uh1D/knuif9ft3/KKtIbS9P1Rx4j+JS/xf+2yOeooorM7AooooAKKKKACrOm/8hW0/67J/6EKrVZ03/kK2n/XZP/QhQA/Wf+Q7f/8AXzJ/6EaKNZ/5Dt//ANfMn/oRooAteKf+Rr1L/r4b+dZNa3in/ka9S/6+G/nWTQAUUUUAFFFFABRRRQB0PjP/AJClj/2C7T/0Stc9XQ+M/wDkKWP/AGC7T/0Stc9WlX42ceB/3aHogooorM7Aqew/5CNt/wBdV/mKgqS3lEN1FKRkI4Ygd8GplrFo1oyUasW+6Fu/+Pyb/ro386iq/K+mSyu5+1gsxOPl703/AIlf/T3/AOO1CnZbM6amHUptqcd+4usf8hD/ALZR/wDoAqjWzqn9n/bf3v2rd5Uf3duMbBiqf/Er/wCnv/x2ppz9xaPY6Mbhr4qo+ePxPr5l26/5E6y/6+G/rWLXR3H2H/hFbTd9o8rz22427s89ayf+JX/09/8AjtRRno9Hu/zOnM8PedP34/BDr/dRf0v/AJFfVv8AgH86w66XTvsX/CO6n5f2jy/k37tu7r2rH/4lf/T3/wCO0qU/fno9/wBEVj8PfD4Zc8fgfX+/INO+7ef9ezfzFUq2LD+ztt1s+1f8e7bs7emRVT/iV/8AT3/47WkZ+89GcdXDfuafvx69fMNH/wCQzaf9dRVNvvH61r6V/Z39rW3lfat/mDG7bjNVW/szcf8Aj76/7NCn770ey/UcsN/ssFzx+KXXyiUaKu/8Sv8A6e//AB2j/iV/9Pf/AI7V8/kzk+rf34/eUqAMkAd6u/8AEr/6e/8Ax2j/AIln/T5/47Rz+TD6t/fj95txeFrGTUP7NbUpVv4wTKot8xggcgNuyefYVBpXhVtTsbq6WcqlrMyygJk7FGSQM8n0H61PD4ue3CeVLMHQAeYYIi7DGMM2Mt+NSR+LIHkBnkuoQkglX7JBDHlvU4AzXmt4xJ28v+D06/geJVwuYUoOSqRe2zu/Oy5Vv66FH/hHFfw7JqkE8jFZG2RPFtzGuNzE54+8OKwa7B/E+mvJua41Xb5XleV5cPl7f9zGPxxWR/xTH/UW/wDIVb0atVX9qn5af8MTh61dX9tFvtp+HQ9P+D8Ub+ErkyRqx+2Nyyg/wrWf8QfH0VpI2leHvLEynE9yij5P9lff3rN0/wAQ6Zonw/eCwfU4or27kQyr5YkXCJnHYZBHPWuT/wCKYPX+1/8AyFXjUcDGpjJ4mrFtX0Vvz/yPnqGWxrY+pi68W1fRW/F/5Cf8Jfr/AP0FJ/zFH/CX6/8A9BSf8xR/xTH/AFFv/IVU9Q/sz93/AGV9r77/ALTt/DG38a9uNOhJ29l+CPo40sPJ29jb/t1Fz/hL9f8A+gpP+Yo/4S/X/wDoKT/mKxqK2+rUP5F9yN/qmH/59x+5Gz/wl+v/APQUn/MUf8Jfr/8A0FJ/zFY1FH1ah/IvuQfVMP8A8+4/cjZ/4S/X/wDoKT/mKP8AhL9f/wCgpP8AmKyrdUe5iWT7hcBsnHGa9n/4Q74df8/Fp/4Mv/sq8/GVsLhHFTpXv2imeXj8RgsC4qpRve+0U9jzrSfHutafqUVxc3DXsKn54JsYYflwfeva9C13RfEdt5umNE7BQXiKAMmexFcz/wAId8Ov+fi0/wDBl/8AZVd0nRfA+iagl7pl/bQzpxkajkEehG7BH1r5zMK2DxUb04SjJeWj9dT5PNK+AxkOalTnCa2tFWfrr+J5h8SlVPiBqIQBR+74Ax/AtZGjaLLrE5VZEhhVgryueATnA+pwfyrt/iVF4bXxKk873Tz3EKvIbSRGX0GcnjgD8Oa57TNa0TSfMFm2pgSEFhJFC4yM4OCDzya+hwtebwUPZxd7JbfI+qweIqSy6n7GL5uVLVdtH/wDFOlyvrMun2x3tG7ruPAwucsfYAE1ck8KagkkCgwulwxEMiOSsmACSOOnOPwq3DqWhw6mb6KXVEuWLEu6RFctkHI9OTxWiviW3XAXVyAv3F+xJiP/AHRt+X8K2qVsQmuRdOqe/wAkexQw+OxetFwikteeSi2/K5BN4Kt7K8WC+1F/3svlxfZ4Q5PB3EgsMAVQi0HT3vL0nU3+xWjBTKsOZCS20HbnGM+/StSfX7G6Mck+qO1xET5cwtgCoKkMMAYOc9T6VBp+q6bpkIjtNSwvcPZK+7nIzlecGsYzxXL7zd/T/wC17G8MlzPk96vTvp/y8hbfX8PxuZGsaGdJt4pGmEjPLJEQBjBUKf8A2YVk11N9d6PqFrHFc6lMzJK8pk8k5Yttz2/2aofY/D//AEEp/wDv0f8ACuyjWkofvE7+jO+hlOJVNKrVp3/xw/zMWitr7H4f/wCglP8A9+j/AIUfY/D/AP0Ep/8Av0f8K29uuz+5m/8AZVT/AJ+U/wDwZH/MxaK2vsfh/wD6CU//AH6P+FH2Pw//ANBKf/v0f8KPbrs/uYf2VU/5+U//AAZH/MxaK2vsfh//AKCU/wD36P8AhR9j8P8A/QSn/wC/R/wo9uuz+5h/ZVT/AJ+U/wDwZH/MxaK2vsfh/wD6CU//AH6P+FH2Pw//ANBKf/v0f8KPbrs/uYf2VU/5+U//AAZH/MxaK2vsfh//AKCU/wD36P8AhR9j8P8A/QSn/wC/R/wo9uuz+5h/ZVT/AJ+U/wDwZH/MxaK9a0jwt4LnhB1H7FENilWGp8se+Rv4rR/4Q74df8/Fp/4Mv/sq8qedUIS5XCX3L/M+OxudUMHXdCcXJrrHlkvk1JpnildDqH/JPdE/6/bv+UVelf8ACHfDr/n4tP8AwZf/AGVdBL4P+HUmi2lvPcWgs45JGhJ1HALsF34bdz0XjtXr5XiY5h7RU042S+LTd/PseRieJMNzU37Oejv8Pk138z58sLGbUbxLa2ALsCcscBQBkk+wAJ/Curt/h1cXWqQWsOo2xSRQxkO7n128c8Y64r1Ww8KfDTTLxLq0vLFZVBALanuGCMEEF8Hg1vwXXg+2YGDU9MQA5CreIFHGOmcdq9ZYGWl5L7zgxnFNSUv9lpySt1j1/E+dT4Tu3jvZbaeCVLVmGMsGkCnBK8Y/Wqul6Bd6tA8tu0SAMUjEjEGVwM7F45OPXFej/EzWrTQWtrXwheWjW14sz3SROkwDMR9dvfpivN9N8QX+lW7QWbxhGfeN8SuVbGNykj5T7iuKcHTk4S1svxPpcFicZi8L7aKSvtfe2zvp5Grp3ge9vLq7huJorf7KGDsST8wxx+tTT+EtMtLC9e61WdLizCq6i2BjMhwNgbdnjPp0BqovjvXVORPBkklj9mjy+f73HPTvVC+166vWO5Y0VpluHQICGlAwWOfXnjpzUaXZap5jKfvzSWm347rqaa+Fbb+y4pn1Am5eD7VsSLKeUGKkhs/e+U8Y/Gs6XTzpXiwWLOJDb3axlgOuGFSy+LNUmsTZu8AhK7SEt0U7c52ggZA5PHvUBv31PxQl7Kqo890rsq9ASwoetjqw8cVGo/bO61/S3ReZBrP/ACHb/wD6+ZP/AEI0Uaz/AMh2/wD+vmT/ANCNFI7y14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAUUUUAFFFFAHQ+M/+QpY/wDYLtP/AEStc9XQ+M/+QpY/9gu0/wDRK1z1aVfjZx4H/doeiCiiiszsCiiigAooooAvax/yEP8AtlH/AOgCqNXtY/5CH/bKP/0AVRrOl/DXodmO/wB6qf4n+ZtXX/InWX/Xw39axa2rr/kTrL/r4b+tYtRQ2l6v8zqzT+JS/wCvcP8A0lG5pf8AyK+rf8A/nWHW5pf/ACK+rf8AAP51h0qXxz9f0RWYf7vhf8D/APS5l3Tvu3n/AF7N/MVSq7p33bz/AK9m/mKpVpH4mcVX+BT+f5l3R/8AkM2n/XUVTb7x+tXNH/5DNp/11FU2+8frQv4j9F+o5f7pD/FL8oiUUUVocYUUVa0u1jvtVtrWeUQxyyBWkP8ACD3qZSUU5PoTKShFyfQq0V1kOjWl3qkMDaPc2UZmMZLyk7/kYjrznIHTIqTSPClvc2F1JfJJDPFcMsMUhKGcAZCDPqeM1ySxtKKvL9OvozglmNGCvK/Tt19GcfRXdyeFtO/4Ro3ItWiu2tWlCtI+VcHpyNuPqa4StaGIhXvy9NDbDYqniebkvo7am3P/AMiHZf8AYQm/9Fx1iVtz/wDIh2X/AGEJv/RcdYlGH+F+r/MeG+GX+KX5sKKKK6DqCiiigAooooAKKKKACiiigDb8Xf8AIfH/AF623/olKxK2/F3/ACHx/wBett/6JSsSufDfwIei/I5cH/u1P0X5BRRRXQdQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXQ6h/yT3RP+v27/lFXPV0Oof8k90T/r9u/wCUVaQ2l6fqjjxH8Sl/i/8AbZHPUUUVmdgUUUUAFFFFABVnTf8AkK2n/XZP/QhVarOm/wDIVtP+uyf+hCgB+s/8h2//AOvmT/0I0Uaz/wAh2/8A+vmT/wBCNFAFrxT/AMjXqX/Xw386ya1vFP8AyNepf9fDfzrJoAKKKKACiiigAooooA6Hxn/yFLH/ALBdp/6JWuerofGf/IUsf+wXaf8Aola56tKvxs48D/u0PRBRRRWZ2BRRRQAUUUUAaM91p91Isk0VyH2Kp2OuOAB6e1R7tL/553f/AH2v+FUqKzVNJWTZ2yxk5ycpRi2/JHR3Bsf+EVtNyXHlee20Bl3Z568Vk7tL/wCed3/32v8AhV26/wCROsv+vhv61i1jRho9Xu/zPSzPENTp+7H4IdF/KjpdONl/wjup+WtwI/k3hmXcee3FY+7S/wDnnd/99r/hV/S/+RX1b/gH86w6VKHvz1e/6IrH4hrD4Z8sdYPov55GxYHTtt1sS6/4923ZdemR7VU3aX/zzu/++1/wo077t5/17N/MVSrSMPeerOOriX7Gn7sevRdzY0o6d/a1t5SXQfzBjc64z+VVWbTNx/d3fX++v+FJo/8AyGbT/rqKpt94/WhQ996vZfqOWJf1WD5Y/FLou0S5u0v/AJ53f/fa/wCFG7S/+ed3/wB9r/hVKir5PNnJ9af8kfuRd3aX/wA87v8A77X/AAo3aZ/zzvP+/i/4VSoo5F3YfWn/ACR+5Gq2pQOyM8+psyHKk3AJX6ccVLHqWnytm/m1eUKQU23K5B9eQaxaKh0IMwqyjUg4ckV5pJP5M6J9U0WSIxPLrzRtwUN2hB/DbVbf4Y/54at/3/i/+IrGoqVh4rZv7zgWFhHZv7zsZn8Pf8IXaZh1PyPt0u0CaPdu2JnJ24xjFY+/wx/zw1b/AL/xf/EU6f8A5EOy/wCwhN/6LjrErGhRXK/ee76+ZhhsOnGXvP4pdX3Nnf4Y/wCeGrf9/wCL/wCIqnqDaYfL/sqO7Trv+0urZ9MbQPeqVFdUaSi73f3nXGiou/M/vCiiitTcKKKKACiiigAooooA2/F3/IfH/Xrbf+iUrErb8Xf8h8f9ett/6JSsSufDfwIei/I5cH/u1P0X5BRRRXQdQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXQ6h/wAk90T/AK/bv+UVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/9CFVqs6b/wAhW0/67J/6EKAH6z/yHb//AK+ZP/QjRRrP/Idv/wDr5k/9CNFAFrxT/wAjXqX/AF8N/OsmtbxT/wAjXqX/AF8N/OsmgAooooAKKKKACiiigDofGf8AyFLH/sF2n/ola56uh8Z/8hSx/wCwXaf+iVrnq0q/GzjwP+7Q9EFFFFZnYFFFFABRRRQAUUUUAbV1/wAidZf9fDf1rFrauv8AkTrL/r4b+tYtYUNper/M9bNP4lL/AK9w/wDSUbml/wDIr6t/wD+dYdbml/8AIr6t/wAA/nWHSpfHP1/RFZh/u+F/wP8A9LmXdO+7ef8AXs38xVKrunfdvP8Ar2b+YqlWkfiZxVf4FP5/mXdH/wCQzaf9dRVNvvH61NZXAtL6G4K7xG4YqDjNWDNpROfsl1/3/H/xNJ3U72v/AEzSEadTDqDmotNvW/VR7J9ihRV/zdK/59Lr/v8Aj/4mjzdK/wCfS6/7/j/4mnzv+V/h/mT9Wh/z9j/5N/8AIlCir/m6V/z6XX/f8f8AxNHm6V/z6XX/AH/H/wATRzv+V/h/mH1aH/P2P/k3/wAibvhvRdJ1HSHu75Wzbt5cgDkb2c4T8uelbZ8KaLAt3F9kM0tpIkTOfOcMSiknEeT1NcSt1pyKVS3vFUnJAuQOf++atWmqaVFI7XEGptu/55XwQk+/yHNeZWoV5Scoylbt/T9f6R4mMyyquarDE37RXNfdbXSXf5ehpnw1bv4bkl8jy9QbfOiCUlhEMADYTnnJ7dq5Guh/tLw3v3/2fq27GN39oLnHp/q6j+1+Fv8AoFan/wCByf8AxuuijKrC/PFu/p/8kY4edenfnjJ3flp5fENn/wCRDsv+whN/6LjrErtJrnw5/wAIbaM2m6gYPt0oVBeLuDbEyc+X0xjjFZH2vwt/0CtT/wDA5P8A43SoVpWfuPd9u/qGGrzUZfu5bvt39TCord+1+Fv+gVqf/gcn/wAbqjqUulS+X/ZNpdW+M+Z584k3emMKMd66o1ZSdnBr7v8AM7IVpSlZwa9bf5lCiiitjoCiiigAooooAKKKKANvxd/yHx/1623/AKJSsStvxd/yHx/1623/AKJSsSufDfwIei/I5cH/ALtT9F+QUUUV0HUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV0Oof8k90T/r9u/wCUVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/wDQhVarOm/8hW0/67J/6EKAH6z/AMh2/wD+vmT/ANCNFGs/8h2//wCvmT/0I0UAWvFP/I16l/18N/OsmtbxT/yNepf9fDfzrJoAKKKKACiiigAooooA6Hxn/wAhSx/7Bdp/6JWuerofGf8AyFLH/sF2n/ola56tKvxs48D/ALtD0QUUUVmdgUUUUAFFFFABRRRQBtXX/InWX/Xw39axa2rr/kTrL/r4b+tYtYUNper/ADPWzT+JS/69w/8ASUbml/8AIr6t/wAA/nWHW5pf/Ir6t/wD+dYdKl8c/X9EVmH+74X/AAP/ANLmXdO+7ef9ezfzFUqu6d928/69m/mKpVpH4mcVX+BT+f5hRRRWhyBRRRQAUUUUAFFdB4Qs7G7v7o6pEkkMNs0g37iAdwGcKQT1rsE8O6PNZPPbaNHJyuyRIp5EkBUHKqrZHXByeoNefXx8KE+SUWeXiczp4ap7OUW/y/M8vorsW8PWz+HpFNtHHqj77lEVzvEYwFUITnnJ/KuOrpo141r8vQ66GJhXvy9Gbc//ACIdl/2EJv8A0XHWJW3P/wAiHZf9hCb/ANFx1iUsP8L9X+YsN8Mv8UvzYUUUV0HUFFFFABRRRQAUUUUAFFFFAG34u/5D4/69bb/0SlYlbfi7/kPj/r1tv/RKViVz4b+BD0X5HLg/92p+i/IKKKK6DqCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK6HUP8Aknuif9ft3/KKuerodQ/5J7on/X7d/wAoq0htL0/VHHiP4lL/ABf+2yOeooorM7AooooAKKKKACrOm/8AIVtP+uyf+hCq1WdN/wCQraf9dk/9CFAD9Z/5Dt//ANfMn/oRoo1n/kO3/wD18yf+hGigC14p/wCRr1L/AK+G/nWTWt4p/wCRr1L/AK+G/nWTQAUUUUAFFFFABRRRQB0PjP8A5Clj/wBgu0/9ErXPV0PjP/kKWP8A2C7T/wBErXPVpV+NnHgf92h6IKKKKzOwKKKKACiiigAooooA2rr/AJE6y/6+G/rWLW1df8idZf8AXw39axawobS9X+Z62afxKX/XuH/pKNzS/wDkV9W/4B/OsOtzS/8AkV9W/wCAfzrDpUvjn6/oisw/3fC/4H/6XMmtbp7SRmRUfepRlcZBBqf+0v8ApytP+/X/ANeqVFauEW7s8+GJrU48sXoXf7S/6crT/v1/9ej+0v8ApytP+/X/ANeqVFL2cexf1yv/ADfgi7/aX/Tlaf8Afr/69H9pf9OVp/36/wDr1Soo9nHsH1yv/N+CLv8AaX/Tlaf9+v8A69H9pf8ATlaf9+v/AK9UqKPZx7B9cr/zfgi8NTK5xZ2gz1xH/wDXqxbeIJbUnbY2UgIxiSIkD6c1k0UnRg1Zoyq1qlaDp1HdP0Nv/hJ33Z/sjSs4xn7Of8ab/wAJH/1BtJ/8Bf8A69Y1FT9Xpdjz/qlHt+Z2M2u48F2k39laYd19Kuw2/wAowicgZ68/pWP/AMJH/wBQbSf/AAF/+vTp/wDkQ7L/ALCE3/ouOsSsKGHpuL06v8znw2GpOMtPtS79zZ/4SP8A6g2k/wDgL/8AXqnqGpf2h5f+hWlrsz/x7RbN2fXmqVFdMaNOLukdccPTg+aK1CiiitTcKKKKACiiigAooooA2/F3/IfH/Xrbf+iUrErb8Xf8h8f9ett/6JSsSufDfwIei/I5cH/u1P0X5BRRRXQdQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXQ6h/wAk90T/AK/bv+UVc9XQ6h/yT3RP+v27/lFWkNpen6o48R/Epf4v/bZHPUUUVmdgUUUUAFFFFABVnTf+Qraf9dk/9CFVqs6b/wAhW0/67J/6EKAH6z/yHb//AK+ZP/QjRRrP/Idv/wDr5k/9CNFAFrxT/wAjXqX/AF8N/OsmtbxT/wAjXqX/AF8N/OsmgAooooAKKKKACiiigDofGf8AyFLH/sF2n/ola56uh8Z/8hSx/wCwXaf+iVrnq0q/GzjwP+7Q9EFFFFZnYFFFFABRRRQAUUUUAbV1/wAidZf9fDf1rFrauv8AkTrL/r4b+tYtYUNper/M9bNP4lL/AK9w/wDSUbmlAnwxqwHJ+T+dYvlv/db8quafrF3piutoygSEFty56Vc/4SzVP78f/fsVFqsJycUmm+/l6HRKpgMRh6MatSUZQi1pFNfE3vzLv2Mfy3/ut+VHlv8A3W/Ktj/hLNU/vx/9+xR/wlmqf34/+/Yquav/ACr7/wDgGPsMr/5/T/8AAF/8mY/lv/db8qPLf+635Vsf8JZqn9+P/v2KP+Es1T+/H/37FHNX/lX3/wDAD2GV/wDP6f8A4Av/AJMg8PxwDXLeXUH8q3hbzX3LneF5249TjFdelzZvHe3WnX9nBc3aRsDMAm1gW3fKQcDBWuX/AOEs1T+/H/37FH/CWap/fj/79iuOvh61aXM1+PnfscGIyzKq8uZ4ia/7hq29/wCc6+4v7Jre1Fle2McaM329GUZnPO8qMZ55xg9xUOva1pl1YXlppe1f3KhPMK+WV5yFAUEMBjua5b/hLNU/vx/9+xR/wlmqf34/+/YrCOAmpJtbefnfscsciyeMlJ4ibt/cXe+vv6/5HUo2jSaa9vbPFc3thbiK2/d7BK2PnYdd3rUNlPp76RZLqc9lIYZ42IdhwnG4AAAg4z1JzXOf8JZqn9+P/v2KP+Eq1BuJktZ07pNAGU/gaf1Kra3z31/Iitk+XRpSdLETlLdXglr2vzaJ+nV7m3eTW09pcrJcWt1cizmLSQ4wAGTYPqMt+dcVWz/wksv/AEC9J/8AAFP8KP8AhJZf+gXpP/gCn+FddCnVoppRv8/+AcGHp1qCaUb/AD/4Bst4U12bwTZRRaVcvJ9sll2hOdjImG+hwayv+EI8Tf8AQEvP+/dd4vxUurbwvaXo0uAs07W+wSEABEU5HH+10qn/AMLqvP8AoDwf9/j/AIV5dOvmivy0o7vr5+p41LE5ylLkoRau+vn6nH/8IR4m/wCgJef9+6q6h4b1nSrX7RqWm3FtDuC75EwMntXdf8LqvP8AoDwf9/j/AIVi+KviNceKdGGnzafFbqJVk3rIWPGeOnvXVRrZnKolVpJR66/8E7MPiM4lViq1GKjfV36fecZRRRXsn0AUUUUAFFFFABRRRQBt+Lv+Q+P+vW2/9EpWJW34u/5D4/69bb/0SlYlc+G/gQ9F+Ry4P/dqfovyCiiiug6gooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACuh1D/AJJ7on/X7d/yirnq6HUP+Se6J/1+3f8AKKtIbS9P1Rx4j+JS/wAX/tsjnqKKKzOwKKKKACiiigAqzpv/ACFbT/rsn/oQqtVnTf8AkK2n/XZP/QhQA/Wf+Q7f/wDXzJ/6EaKNZ/5Dt/8A9fMn/oRooAteKf8Aka9S/wCvhv51k1reKf8Aka9S/wCvhv51k0AFFFFABRRRQAUUUUAdD4z/AOQpY/8AYLtP/RK1z1dD4z/5Clj/ANgu0/8ARK1z1aVfjZx4H/doeiCiiiszsCiiigAooooAKKKKANq6/wCROsv+vhv61i1tzoz+D7IIrMfPboM+tZH2eb/nlJ/3ya56LVper/M9jM4Sc6TS/wCXcP8A0lEdFSfZ5v8AnlJ/3yaPs83/ADyk/wC+TW90eV7OfZkdFSfZ5v8AnlJ/3yaPs83/ADyk/wC+TRdB7OfZkdFSfZ5v+eUn/fJo+zzf88pP++TRdB7OfZkdFa3h+GKPXLebUd8cEDea37snft52498Yrr1vLZ0vbrTtRitLq8SM7pkKEMC27IAOOCv5Vx18U6UuVRv/AMP6P1OHEV6tGXKqbfnZ239H6nnVFel3GpWzwWos7+GGKNm+3RmMg3J53EDHO7nHI61ZHiTTn1LyrcmGBPLIMpJjK87gF28MBjua5XmFS2lJ/f8A8A4njsVa6w7/AB/y/I8ror0nU7uzvJlaFkuLuOCYQKGLjJI6sVGDtBwMH61Us7uy/sizXVbm1k8ieNwJAxwnG4AbeDjPc5rRY6Tipcj9P6RosbW5Od0WvLXz8v6+RwNFdre3ltPaXKPeW95cfY5i8sfTG5Ni5IGSPm/OuKrroVnVTbVjqw9d1k242t/Xkbc//Ih2X/YQm/8ARcdYldq3g/X5/BNlDDpczyfbJJtox9xkTB698Gsv/hAfFH/QGuP0/wAa56OKw6TTmt31Xc5MPjcLGMk6kfil1Xc56iuh/wCEB8Uf9Aa4/T/GqmpeFtb0e0+06np0tvDuC73xjJ7dfauiOKoSfLGab9UdccZhpyUY1ItvzRk0UUV0HUFFFFABRRRQAUUUUAbfi7/kPj/r1tv/AESlYlbfi7/kPj/r1tv/AESlYlc+G/gQ9F+Ry4P/AHan6L8goooroOoKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArodQ/5J7on/X7d/wAoq56uh1D/AJJ7on/X7d/yirSG0vT9UceI/iUv8X/tsjnqKKKzOwKKKKACiiigAqzpv/IVtP8Arsn/AKEKrVZ03/kK2n/XZP8A0IUAP1n/AJDt/wD9fMn/AKEaKNZ/5Dt//wBfMn/oRooAteKf+Rr1L/r4b+dZNa3in/ka9S/6+G/nWSQR1BFABRQQQcEY+tGDjOOPWgAopVRn+4pb6DNL5b7tuxt3pjmgV0NopSpHUEfUUlAzofGf/IUsf+wXaf8Aola56uh8Z/8AIUsf+wXaf+iVrnq0q/GzjwP+7Q9EFFFFZnYFFFFABRRRQAUUUUAaNnr1/YWwgtpVWNSSAUB61P8A8JVq3/Pdf+/a/wCFY9FYuhSbu4r7j0aea4+nBQhWkktkpM2P+Eq1b/nuv/ftf8KP+Eq1b/nuv/ftf8Kx6KX1ej/KvuL/ALYzH/n/AD/8Cf8AmbH/AAlWrf8APdf+/a/4Uf8ACVat/wA91/79r/hWPRR9Xo/yr7g/tjMf+f8AP/wJ/wCZsf8ACVat/wA91/79r/hR/wAJVq3/AD3X/v2v+FY9FH1ej/KvuD+2Mx/5/wA//An/AJmx/wAJVq3/AD3X/v2v+FH/AAlWrf8APdf+/a/4Vj0UfV6P8q+4P7YzH/n/AD/8Cf8AmbH/AAlWrf8APdf+/a/4Uf8ACVat/wA91/79r/hWPS7Tuxg59MUfV6P8q+4P7YzH/n/P/wACf+Zr/wDCVat/z3X/AL9r/hTk8V6orZdoJR/dlt0dT+BGKxQCTgDJpxjcLkowHrij6vR/lX3GVbNMbWpulVrScZKzTk7NPobf/CX6h/z66Z/4Lof/AImj/hL9Q/59dM/8F0P/AMTWGUZeqkd+RSVP1Wh/IvuPF+p4b+RfcerD4qX9v4WtL0adal2ne3K5IGERDnj/AHqpf8Lo1L/oFWn/AH21cjP/AMiHZf8AYQm/9Fx1iV51LKsFJNumt337nk0Mky+ak5Ulu+/f1PSf+F0al/0CrT/vtqxvFHxFvPFGjjT7mxggQSrJvjYk5GeOfrXH0V008rwdKanCFmvU66WTYCjNVKdNJrbf/MKKKK9E9YKKKKACiiigAooooA2/F3/IfH/Xrbf+iUrErb8Xf8h8f9ett/6JSsSufDfwIei/I5cH/u1P0X5BRRRXQdQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXQ6h/yT3RP+v27/lFXPV0Oof8AJPdE/wCv27/lFWkNpen6o48R/Epf4v8A22Rz1FFFZnYFFFFABRRRQAVZ03/kK2n/AF2T/wBCFVqs6b/yFbT/AK7J/wChCgB+s/8AIdv/APr5k/8AQjRRrP8AyHb/AP6+ZP8A0I0UAbF99j/4WJdf2mQtr9pfeWBIBwcZx2ziuqkvvCl4oF9faeXEPlEJatsHy9UO3K8+grh/FP8AyNepf9fDfzrJpvVW/rax52KwKxM1Nzat2t/kenaleeEb8XMkl9ZFpYiqr9lbcuANuG25B61yegXumwaeI9SMb7bhpfJkUsrYhYLkY/vYFc7RS63IpZbGlSdLnk0/TT00PStIurO6ubCLw/KlpcXl4i3YtSYmAOAFXpxkHpW5p3g/xlfalqGp39ulnd3SFYJWcF4TvU53DPG0EceteTaNqkuia1aanbokktrKsqq+dpIPfFei/wDC+te/6BenflJ/8VXVR9i03UbTPCzHL8whO2CjGSa1ct99vTb5Kw/xN4A8SXti8VnpDyOt7uBDoNy+WoZuvQvuP41y/wDwq3xn/wBAKX/v7H/8VXS/8L617/oF6d+Un/xVH/C+te/6BenflJ/8VWvJg/5n/XyM8PLiKhT5I0off/8AbFDxb4C8TXOo2bwaVI6pp9tExEicMsSqw69iMVg/8K78V/8AQHk/7+J/8VXdeJ/i3q1lfWscdhZMJLOCclg/BkjDEfe6ZNYv/C5tZ/6B1j+T/wDxVfP46pmMcTNYeEXDo3v+ZrhK2efV4ctKFrd3/mc+fh34qAydHkA/66J/8VWVf6DqemSrHe2jozLuAUh+PqpNdo3xk1h0Ktp1iVYYIw//AMVXN6l4xvr2ZHtoobFVXBSEEhj6/MTWWHqZi5WrQil5f8O/yPpcpnXqSkszXJ25En995q3yT+Ri/Y7n/n3l/wC+DR9juf8An3l/74NXv+Ek1b/n7P8A3wv+FH/CSat/z9n/AL4X/Cu+9fsvvf8AkfQ8mV/z1P8AwGP/AMmUfsdz/wA+8v8A3waPsdz/AM+8v/fBq9/wkmrf8/Z/74X/AAo/4STVv+fs/wDfC/4UXr9l97/yDkyv+ep/4DH/AOTKP2O5/wCfeX/vg0fY7n/n3l/74NXv+Ek1b/n7P/fC/wCFH/CSat/z9n/vhf8ACi9fsvvf+QcmV/z1P/AY/wDyZR+x3P8Az7y/98Gj7Hc/8+8v/fBq9/wkmrf8/Z/74X/Cj/hJNW/5+z/3wv8AhRev2X3v/IOTK/56n/gMf/kyj9juf+feX/vg0fY7n/n3l/74NXv+Ek1b/n7P/fC/4Uf8JJq3/P2f++F/wovX7L73/kHJlf8APU/8Bj/8mUfsdz/z7y/98Gj7Hc/8+8v/AHwavf8ACSat/wA/Z/74X/Cj/hJNW/5+z/3wv+FF6/Zfe/8AIOTK/wCep/4DH/5Mo/Y7n/n3l/74NH2O5/595f8Avg1e/wCEk1b/AJ+z/wB8L/hR/wAJJq3/AD9n/vhf8KL1+y+9/wCQcmV/z1P/AAGP/wAmUfsdz/z7y/8AfBo+x3P/AD7y/wDfBq9/wkmrf8/Z/wC+F/wo/wCEk1b/AJ+z/wB8L/hRev2X3v8AyDkyv+ep/wCAx/8Aky34atoY72STUIhG6Kpha4hZo924ZyAD/DnHvirDOh8fJe7ZPsguEYyLG2AoAyemazP+Ek1b/n7P/fC/4Uf8JJq3/P2f++F/wrmlRqynKTtqrb/8A5ZYPKZTlN1amqt8MdPT3jq9+mSy3rWFlFHcxiRLMrCUDKf4icdcZ61HpmpXMtjpVjczSzQxuJrmKeKR/uvgIowf4QD+Ncx/wkmrf8/Z/wC+F/wpr+ItWddpvZF90wp/Mc1h9Sm1Z2++/wCnz9Thq5VlfspKnUqOXTmjHezW/M9Nb6LfU7C+hubi1uYz519P9jm3TeRIMjcuxcsoJI+b864n+x9S/wCgdd/9+G/wpf7a1X/oJ3n/AIEN/jR/bWq/9BO8/wDAhv8AGuihQq0U0mjzsPh6+HTjFpnWP4I8RTeDLO3j0uQzC8kmKb1BCMiAHr7Gsv8A4V34r/6A8n/fxP8A4qvXPh94iuvEnhtri7jjSSCTyBsz8wCrycnrzXJ678TvEGg61cafc6dZFoWIDlJBvXsw+boa8KjjswlWnQpwjzJtvfr8z5vD5lmssRUw1OEOaLbd79X6nH/8K78V/wDQHk/7+J/8VVPVfCWuaJZ/atU097eDcE3s6nk9Bwfaus/4XNrP/QOsfyf/AOKrI8TfETUfFGkjT7y0tYYxIJN0QbORn1J9a9OjUzR1EqsIqPW3/Dnr4etnLqxVanBRvrZ62+85GiiivZPoAooooAKKKKACiivRvh78PW1RotX1lCtmDuhhPHne5/2f51y4rFUsLSdSo9PzOPG42jgqLrVnp+b7I5Txd/yHx/1623/olKxK6n4kqqePtQVFCqojAAGAB5a1y1GDlzYanLul+QYCXPhKUu8V+QUUUV1HYFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV0Oof8k90T/r9u/5RVz1dDqH/JPdE/6/bv8AlFWkNpen6o48R/Epf4v/AG2Rz1FFFZnYFFFFABRRRQAVZ03/AJCtp/12T/0IVWqzpv8AyFbT/rsn/oQoAfrP/Idv/wDr5k/9CNFGs/8AIdv/APr5k/8AQjRQBa8U/wDI16l/18N/OsmtbxT/AMjXqX/Xw386yaACiiigAooooAKKKKAOh8Z/8hSx/wCwXaf+iVrnq6Hxn/yFLH/sF2n/AKJWuerSr8bOPA/7tD0QUUUVmdgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHo3w+8d6T4Y0Gaz1JbkyvcGQeVGGGCoHcj0qz4s8XeDfFdkEuEvobmP/VXC26kr6gjdyK8wqxbafeXoJs7Se4CnB8qItg/gK8meWYf27xN2pb3ueHUybCrEvF8zjK973sav2Pwv/wBBjUf/AAAX/wCOUfY/C/8A0GNR/wDABf8A45VD+w9W2F/7Lvdozk/Z3wP0pkGk6jcxh7bT7qVDnDRwswP5CurlW/tX/wCS/wCR28qtf2z/APJf8jS+x+F/+gxqP/gAv/xyj7H4X/6DGo/+AC//ABysyXStQt4GmnsLmKJTgyPCwUdupFFtpl/eRGSzsrmeNTgvFEzAH0yBT5dL+1dv+3f8h8mnN7Z2/wC3f8jT+x+F/wDoMaj/AOAC/wDxyj7H4X/6DGo/+AC//HKy7fTb68UtaWVxOoOCYomYA+nAp76NqccbvJp12qRjLs0DAL9eOKOVJ2dV/wDkv+QOCTs6zv8A9u/5Gj9j8L/9BjUf/ABf/jlH2Pwv/wBBjUf/AAAX/wCOVlWun3l9u+xWk9xs+95MZfb9cCnQ6XqFw7rBY3MrRtscJCxKt6HA4NDhbeq//Jf8gdO106z0/wAP+R0GkjwbY6lFcX13qF7FGc+QbNVDHtk7zx7V6Kvxd8NIoVIr5VAwAIV4/wDHq8YlsLyC4W3mtJ45n+7G8ZDN9B1qGSKSGVo5kaORThlYYIPuK48RllDFtSqycu2q/RHBisnw2OkpVpylbbVfoja8Y6xba94ru9RsRIIJtu0SLhuFA6fhWHRRXp0qcaVNU47JW+49ijSjRpxpQ2ikl8gooorQ1CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK6HUP+Se6J/1+3f8oq56uh1D/knuif8AX7d/yirSG0vT9UceI/iUv8X/ALbI56iiiszsCiiigAooooAKs6b/AMhW0/67J/6EKrVZ03/kK2n/AF2T/wBCFAD9Z/5Dt/8A9fMn/oRoo1n/AJDt/wD9fMn/AKEaKALXin/ka9S/6+G/nWTWt4p/5GvUv+vhv51k0AFFFFABRRRQAUUUUAdD4z/5Clj/ANgu0/8ARK1z1dD4z/5Clj/2C7T/ANErXPVpV+NnHgf92h6IKKKKzOwKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArs/DMs0fgDXhbOUkkmgQMDgjOe/auMqeO9uYbSW1inkSCYgyRhsK5HTIrnxFF1ocq7p/c7nLiqDr01Bd0/uaZ6n4diez8WWlpqF9fajfxoobbhIYkKA4Iwd/XrxWZfXl5pvhHS44JHgc6k+7HB4I/xriz4k1preOA6pdGKIqY0804Ur93H0xUFxq2oXSKlzeTSqj+YodycN6/XivMjl1T2inNrp+F/K3Xb8zx4ZTU9qp1Gntf5c3SyS32/M9Gu7qW88QeL7e4cvDHaErGfughgAcetQ6hdWllY6D9gtNSlhaAGFrG4VEaTJ3Agoct06+1ef/2tqHnXE32ybzLldsz7zmQdcH1qSx17VtMgMOnajc20TNuKRSFQT6/pT/s2SSs1pbTVL4bdBrKZxSs1pbTVL4eV7Hf6nqcsfhHWbm0s5NKkku0DwlvmU8knIx3FWEv7m78bWdtPKzwyaR+8Q9HJhzkj6mvMW1K9a2kt2upTDI+94yxwzepHrT11fUVuluVvZxOkflrIHO4LjG3Ppjil/ZnutK3X8UkvusL+x2oySau+a2/VJLz0sdRftdW/gnTToZlRGupRcmHr5ny7QSOfXH410/iWdrXwFcurGLUZFtvtpX5WEm1Tzjoa8xsNa1LSzIdOvp7YykF/Kcrux6/nTJNTvZYp45LqVkuJPMmUucSN6n1NXLL5SnFtqyd/N63s/T/I0nlc51IttWjLm83dp2fp0+Wx6JoskGq6Pp3iK8DSSaKki3Ds2Wbb8ycf8CNebXNxJdXUk8zl5JGLMx6kmtn+2rWz8KyaZp32jzrpw1y8gCqAOirgnI69cVg10YSg6cpye19PTf8ANv5WOrA4Z0p1JvZuy8ldv82/lYKKKK9A9QKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK6HUP+Se6J/wBft3/KKuerodQ/5J7on/X7d/yirSG0vT9UceI/iUv8X/tsjnqKKKzOwKKKKACiiigAqzpv/IVtP+uyf+hCq1WdN/5Ctp/12T/0IUAP1n/kO3//AF8yf+hGijWf+Q7f/wDXzJ/6EaKALXin/ka9S/6+G/nWTWt4p/5GvUv+vhv51k0AFFFFABRRRQAUUUUAdD4z/wCQpY/9gu0/9ErXPV2fjrQNQgtNH1lod1jc6dbIsi87WWJRhvTpXGVrWTU3c4MuqQqYWDg72VvmtwooorI7wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArodQ/5J7on/AF+3f8oqxLOzuNQvI7WyhaaeVgqRoMlia7fx34YuPCfhHw7YXkivO8lxNIFHCFhH8ue/TrW0IycJS6W/VHm4qvTjiKFFv3nJtLyUZHBUUUViekFFFFABRRRQAVZ03/kK2n/XZP8A0IVWqzpv/IVtP+uyf+hCgB+s/wDIdv8A/r5k/wDQjRRrP/Idv/8Ar5k/9CNFAFrxT/yNepf9fDfzrJrW8U/8jXqX/Xw386yaACiiigAooooAKKKKAPqfQ9Ng1j4Zadp92itFcabEh3KGxmMYOD3B5H0r548YeD9Q8H6w1peoWhfmC4A+WRf8fUVLa/EPxXZWkVra61PHBCgSNAq4VQMAdKJ/iH4pugBc6s8wXoJIo2x+a16FevRrQSs00fHZXlOZZdiKk1KMoTbbV2vR7HNUVvf8Jv4g/wCf5f8AwHi/+Jo/4TfxB/z/AC/+A8X/AMTXHaHd/d/wT6bnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRW9/wm/iD/AJ/l/wDAeL/4mj/hN/EH/P8AL/4Dxf8AxNFod393/BDnxX8kf/An/wDImDRXaPfePI7eCeSGZYrgqIXNpHhy3THy980Q3/ju4t5p4YpnigYrI4tI8KR1B+WnyRXf7v8AgnP9cna/uf8Agf8A9qcXRW9/wm/iD/n+X/wHi/8AiaP+E38Qf8/y/wDgPF/8TStDu/u/4J0c+K/kj/4E/wD5EwaK3v8AhN/EH/P8v/gPF/8AE0f8Jv4g/wCf5f8AwHi/+JotDu/u/wCCHPiv5I/+BP8A+RMGit7/AITfxB/z/L/4Dxf/ABNH/Cb+IP8An+X/AMB4v/iaLQ7v7v8Aghz4r+SP/gT/APkTBore/wCE38Qf8/y/+A8X/wATR/wm/iD/AJ/l/wDAeL/4mi0O7+7/AIIc+K/kj/4E/wD5EwaK3v8AhN/EH/P8v/gPF/8AE0f8Jv4g/wCf5f8AwHi/+JotDu/u/wCCHPiv5I/+BP8A+RMGit7/AITfxB/z/L/4Dxf/ABNH/Cb+IP8An+X/AMB4v/iaLQ7v7v8Aghz4r+SP/gT/APkTBore/wCE38Qf8/y/+A8X/wATR/wm/iD/AJ/l/wDAeL/4mi0O7+7/AIIc+K/kj/4E/wD5EwaK3v8AhN/EH/P8v/gPF/8AE0f8Jv4g/wCf5f8AwHi/+JotDu/u/wCCHPiv5I/+BP8A+RMGit7/AITfxB/z/L/4Dxf/ABNH/Cb+IP8An+X/AMB4v/iaLQ7v7v8Aghz4r+SP/gT/APkTBore/wCE38Qf8/y/+A8X/wATR/wm/iD/AJ/l/wDAeL/4mi0O7+7/AIIc+K/kj/4E/wD5EwaK3v8AhN/EH/P8v/gPF/8AE0f8Jv4g/wCf5f8AwHi/+JotDu/u/wCCHPiv5I/+BP8A+RMGit7/AITfxB/z/L/4Dxf/ABNH/Cb+IP8An+X/AMB4v/iaLQ7v7v8Aghz4r+SP/gT/APkTBore/wCE38Qf8/y/+A8X/wATR/wm/iD/AJ/l/wDAeL/4mi0O7+7/AIIc+K/kj/4E/wD5EwaVVZ3CopZmOAAMkmt3/hN/EH/P8v8A4Dxf/E0o8ceIVYFb9QQcgi3i4/8AHaLU+7+7/ghz4r+SP/gT/wDkT2H4TfD+fw7btrGroq3tzGBHCVBaFfXPYkdvSsX9oD/mBf8Abb/2SuD/AOFm+Mf+g9cf98p/hWVrXibWfEXk/wBtX8l35GfL3gDbnGeg9hXdPE0vYexgmfLYXJMw/tZZhipxdr6K+1mklp0MqiiivNPtAooooAKKKKACrOm/8hW0/wCuyf8AoQqtVnTf+Qraf9dk/wDQhQA/Wf8AkO3/AP18yf8AoRoo1n/kO3//AF8yf+hGigC14p/5GvUv+vhv51k1reKf+Rr1L/r4b+dZNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVJbvGlzG00fmxhvmTdt3D0z2qOgcMKqLtJMTV1Y9R8XRWGseM/DGnSWjxxzRWyOwnJ/dsF+UemM9aXxOun2Hw2NhaWLhF1KWKMm4Y4YBfmPHPXpWLqHifTLjxt4f1KKVzbWMdss7GM5BQLuwO/Q03xL4n07UfDRtLKV2n/tKS5AMZA2ELg5/DpXRUmuSVt3J/dzxt+F/zPj6WErp4aDjLljZvfR3e5JeaJoHhyLSLXV7Oa7vL5ElnkE5jWBWx0A6kZrn/Fugr4d8QzWUUnmw4EkTnqVYZGa6G91nQPEsekXerX8lnd2MaRTxeQziZVxyCOhOK5/xdry+IvEMt7DGYoAqxxKeoVRgZrKrbp3f3dP0/E9LAfWvbR9pzbS5r3te+nLfTa+2lt9TEooorI98KKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACrOm/8AIVtP+uyf+hCq1WdN/wCQraf9dk/9CFAD9Z/5Dt//ANfMn/oRoo1n/kO3/wD18yf+hGigC14p/wCRr1L/AK+G/nWTWt4p/wCRr1L/AK+G/nWTQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVZ03/kK2n/AF2T/wBCFVqs6b/yFbT/AK7J/wChCgB+s/8AIdv/APr5k/8AQjRRrP8AyHb/AP6+ZP8A0I0UAauqjRNS1a5vRq0sYnkL7DZk7c9s7qqfYdF/6DUv/gEf/iqKKAD7Dov/AEGpf/AI/wDxVH2HRf8AoNS/+AR/+KoooAPsOi/9BqX/AMAj/wDFUfYdF/6DUv8A4BH/AOKoooAPsOi/9BqX/wAAj/8AFUfYdF/6DUv/AIBH/wCKoooAPsOi/wDQal/8Aj/8VR9h0X/oNS/+AR/+KoooAPsOi/8AQal/8Aj/APFUfYdF/wCg1L/4BH/4qiigA+w6L/0Gpf8AwCP/AMVR9h0X/oNS/wDgEf8A4qiigA+w6L/0Gpf/AACP/wAVR9h0X/oNS/8AgEf/AIqiigA+w6L/ANBqX/wCP/xVH2HRf+g1L/4BH/4qiigA+w6L/wBBqX/wCP8A8VR9h0X/AKDUv/gEf/iqKKAD7Dov/Qal/wDAI/8AxVH2HRf+g1L/AOAR/wDiqKKAD7Dov/Qal/8AAI//ABVH2HRf+g1L/wCAR/8AiqKKAD7Dov8A0Gpf/AI//FUfYdF/6DUv/gEf/iqKKAD7Dov/AEGpf/AI/wDxVH2HRf8AoNS/+AR/+KoooAPsOi/9BqX/AMAj/wDFUfYdF/6DUv8A4BH/AOKoooAPsOi/9BqX/wAAj/8AFUfYdF/6DUv/AIBH/wCKoooAPsOi/wDQal/8Aj/8VR9h0X/oNS/+AR/+KoooAPsOi/8AQal/8Aj/APFUfYdF/wCg1L/4BH/4qiigA+w6L/0Gpf8AwCP/AMVR9h0X/oNS/wDgEf8A4qiigA+w6L/0Gpf/AACP/wAVR9h0X/oNS/8AgEf/AIqiigA+w6L/ANBqX/wCP/xVH2HRf+g1L/4BH/4qiigA+w6L/wBBqX/wCP8A8VR9h0X/AKDUv/gEf/iqKKAD7Dov/Qal/wDAI/8AxVH2HRf+g1L/AOAR/wDiqKKAD7Dov/Qal/8AAI//ABVH2HRf+g1L/wCAR/8AiqKKAD7Dov8A0Gpf/AI//FUfYdF/6DUv/gEf/iqKKAD7Dov/AEGpf/AI/wDxVH2HRf8AoNS/+AR/+KoooAPsOi/9BqX/AMAj/wDFUfYdF/6DUv8A4BH/AOKoooAPsOi/9BqX/wAAj/8AFUfYdF/6DUv/AIBH/wCKoooAPsOi/wDQal/8Aj/8VR9h0X/oNS/+AR/+KoooAPsOi/8AQal/8Aj/APFUfYdF/wCg1L/4BH/4qiigA+w6L/0Gpf8AwCP/AMVR9h0X/oNS/wDgEf8A4qiigA+w6L/0Gpf/AACP/wAVR9h0X/oNS/8AgEf/AIqiigA+w6L/ANBqX/wCP/xVH2HRf+g1L/4BH/4qiigA+w6L/wBBqX/wCP8A8VR9h0X/AKDUv/gEf/iqKKAD7Dov/Qal/wDAI/8AxVH2HRf+g1L/AOAR/wDiqKKAD7Dov/Qal/8AAI//ABVH2HRf+g1L/wCAR/8AiqKKAD7Dov8A0Gpf/AI//FUfYdF/6DUv/gEf/iqKKAD7Dov/AEGpf/AI/wDxVSW1totvdwzHWJWEbq+PsZ5wc/3qKKAM3UZ0udUup4s7JZndcjsWJFFFFAH/2Q==)

## Round Robin

## A picture containing text, device Description automatically generatedRouter

# APPENDIX A – VHDL Codes

## Register (M-ROU-01):

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

use IEEE.numeric\_std.all;

ENTITY reg IS

PORT (data\_in : IN STD\_LOGIC\_VECTOR (7 DOWNTO 0);

clock, reset, clock\_en: IN STD\_LOGIC;

data\_out : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0));

END ENTITY;

ARCHITECTURE behave OF reg IS

SIGNAL dout: STD\_LOGIC\_VECTOR (7 DOWNTO 0);

BEGIN

p1: PROCESS(reset,clock) IS BEGIN

IF reset = '1' THEN dout <= (OTHERS => '0');

ELSIF rising\_edge(clock) THEN

IF clock\_en = '1' THEN dout <= data\_in;

ELSIF clock\_en = '0' THEN dout <= dout;

END IF;

data\_out <= dout;

END IF;

END PROCESS;

END ARCHITECTURE;

## Demultiplexer (M-ROU-02):

LIBRARY IEEE;

USE ieee.std\_logic\_1164.ALL;

ENTITY demux IS

PORT(d\_in: in std\_logic\_vector (7 downto 0);

sel: in std\_logic\_vector (1 downto 0);

en: in std\_logic;

d\_out1,d\_out2,d\_out3,d\_out4: out std\_logic\_vector (7 downto 0));

END ENTITY demux;

ARCHITECTURE behav OF demux IS

Signal temp,temp1: std\_logic\_vector (7 downto 0);

BEGIN

PROCESS (sel, en, d\_in) IS

VARIABLE temp,temp1: std\_logic\_vector (7 downto 0);

BEGIN

if en='1' THEN

temp:= d\_in;

CASE sel IS

WHEN "00" => d\_out1<= temp;

WHEN "01" => d\_out2<= temp;

WHEN "10" => d\_out3<= temp;

WHEN "11" => d\_out4<= temp;

WHEN OTHERS=>

END CASE;

ELSIF en='0' THEN

temp1:=temp;

END IF;

END PROCESS;

END ARCHITECTURE behav;

## Block RAM (M-ROU-03):

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

USE ieee.numeric\_std.ALL;

Entity RAM IS

port( CLKA,CLKB,WEA,REA : in std\_logic;

ADDRA,ADDRB : in std\_logic\_vector(3 downto 0);

d\_in : in std\_logic\_vector(7 downto 0);

d\_out: out std\_logic\_vector(7 downto 0));

end Entity RAM;

Architecture behave of RAM IS

type RAM\_Arr is array (0 to 7) of std\_logic\_vector(7 downto 0);

signal RAMArr :RAM\_Arr ;

signal temp : STD\_LOGIC\_VECTOR(7 downto 0);

BEGIN

write\_process: process(CLKA,WEA)

begin

--Rising clock edge causes a problem in the synchronization between The RAM and the FIFO controller in the FIFO Module

--We decided to solve this problem using a level triggered RAM

if(WEA='1' AND CLKA='1') then

RAMArr(to\_integer('0' & unsigned( ADDRA(2 downto 0))))<= d\_in;

end if;

end process write\_process;

read\_process: process(CLKB,REA)

begin

--Rising clock edge causes a problem in the synchronization between The RAM and the FIFO controller in the FIFO Module

--We decided to solve this problem using a level triggered RAM

if(REA='1' and CLKB='1' ) then

temp<=RAMArr(to\_integer('0' & unsigned(ADDRB(2 downto 0))));

else temp<=temp;

end if;

end process read\_process;

d\_out<=temp;

end architecture behave;

## Grey Counter (M-ROU-04):

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE ieee.numeric\_std.ALL;

entity greyGenerator is

port ( CLK , RESET , EN : in std\_logic;

COUNT\_OUT : out std\_logic\_vector(3 downto 0));

end greyGenerator;

architecture BEHAVE of greyGenerator is

signal Currstate, Nextstate, hold, next\_hold: std\_logic\_vector (3 DOWNTO 0);

begin

P1 : process(clk) is begin

if (Clk = '1' AND Clk'EVENT) THEN

if (Reset = '1') THEN

Currstate <= (OTHERS =>'0');

elsif (En = '1') THEN

Currstate <= Nextstate;

end if;

end if;

end process;

hold <= Currstate XOR ('0' & hold(3 DOWNTO 1));

next\_hold <= std\_logic\_vector(unsigned(hold) + 1);

Nextstate <= next\_hold XOR ('0' & next\_hold(3 DOWNTO 1));

COUNT\_OUT <= Currstate;

end architecture BEHAVE;

## Grey to Binary (M-ROU-05):

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

entity greyBinary is

port( gray\_in : in std\_logic\_vector(3 downto 0);

bin\_out : out std\_logic\_vector(3 downto 0)

);

end greyBinary;

architecture behav of greyBinary is

begin

bin\_out(3) <= gray\_in(3);

bin\_out(2) <= gray\_in(3) xor gray\_in(2);

bin\_out(1) <= gray\_in(3) xor gray\_in(2) xor gray\_in(1);

bin\_out(0) <= gray\_in(3) xor gray\_in(2) xor gray\_in(1) xor gray\_in(0);

end behav;

## FIFO Controller (M-ROU-06):

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE ieee.numeric\_std.ALL;

Entity FIFO\_Controller IS

Port( reset,rdclk,wrclk,rreq,wreq : in std\_logic;

write\_valid,read\_valid,empty,full: out std\_logic;

wr\_ptr,rd\_ptr : out std\_logic\_vector(3 downto 0));

end Entity FIFO\_Controller;

Architecture behave of FIFO\_Controller IS

component greyBinary is

port( gray\_in : in std\_logic\_vector(3 downto 0);

bin\_out : out std\_logic\_vector(3 downto 0)

);

end component;

FOR ALL:greyBinary USE Entity work.greyBinary(behav);

component greyGenerator is

port ( CLK , RESET , EN : in std\_logic;

COUNT\_OUT : out std\_logic\_vector(3 downto 0));

end component;

FOR ALL:greyGenerator USE Entity work.greyGenerator(BEHAVE);

signal wr\_valid,rd\_valid: std\_logic;

signal grey\_read,grey\_write: std\_logic\_vector(3 downto 0);

signal w\_count : integer:=0;

signal r\_count: integer:=0;

Begin

read\_counter:greyGenerator port map(rdclk,reset,rreq,grey\_read);

write\_counter:greyGenerator port map(wrclk,reset,wreq,grey\_write);

read\_bin:greyBinary port map(grey\_read,rd\_ptr);

write\_bin:greyBinary port map(grey\_write,wr\_ptr);

p1:process(reset,wrclk,wreq)

begin

if(reset='1') then

wr\_valid<='0';

w\_count<=0;

elsif(wreq='1' AND wrclk='1') then

if(w\_count=8) then

w\_count<=0;

else

w\_count<=w\_count+1;

end if;

wr\_valid<='1';

else wr\_valid<='0';

end if;

end process p1;

p2:process(reset,rdclk,rreq)

begin

if(reset='1') then

rd\_valid<='0';

r\_count<=0;

elsif(rreq='1' AND rdclk='1') then

if(r\_count=7) then

r\_count<=0;

else r\_count<=r\_count+1;

end if;

rd\_valid<='1';

else rd\_valid<='0';

end if;

end process p2;

p3:process(w\_count,r\_count,reset)

begin

if(reset='1') then

full<='0';

empty<='1';

elsif(w\_count=8) then

full<='1';

empty<='0';

elsif(w\_count/=0) then

empty<='0';

end if;

end process p3;

read\_valid<=rd\_valid;

write\_valid<=wr\_valid;

end architecture behave;

## FIFO (M-ROU-07):

library IEEE;

USE IEEE.std\_logic\_1164.ALL;

ENTITY fifo IS

PORT ( reset, rclk, wclk, rreq, wreq: IN std\_logic;

data\_in : IN std\_logic\_vector (7 DOWNTO 0);

data\_out : OUT std\_logic\_vector (7 DOWNTO 0);

empty, full: OUT std\_logic);

END ENTITY;

ARCHITECTURE struct OF fifo IS

SIGNAL Rd, Wr: std\_logic;

SIGNAL ADDA,ADDB: std\_logic\_vector (3 DOWNTO 0);

COMPONENT RAM IS

PORT (CLKA,CLKB,WEA,REA : in std\_logic;

ADDRA,ADDRB : in std\_logic\_vector(3 downto 0);

d\_in : in std\_logic\_vector(7 downto 0);

d\_out: out std\_logic\_vector(7 downto 0));

END COMPONENT;

FOR ALL:RAM use ENTITY work.RAM(behave);

Component FIFO\_Controller IS

Port( reset,rdclk,wrclk,rreq,wreq : in std\_logic;

write\_valid,read\_valid,empty,full: out std\_logic;

wr\_ptr,rd\_ptr : out std\_logic\_vector(3 downto 0));

end Component;

FOR ALL:FIFO\_Controller use ENTITY work.FIFO\_Controller(behave);

BEGIN

FIFO\_ctrlr : FIFO\_Controller

PORT MAP (reset, rclk, wclk, rreq, wreq, Wr, Rd, empty, full, ADDA, ADDB);

ramemory : RAM

PORT MAP (CLKA=>wclk,CLKB=> rclk, WEA=>Wr, REA=>Rd, ADDRA=>ADDA, ADDRB=>ADDB, d\_in=>data\_in, d\_out=>data\_out);

END ARCHITECTURE;

## Round Robin Scheduler (M-ROU-08):

LIBRARY IEEE;

USE ieee.std\_logic\_1164.ALL;

ENTITY roundR IS

PORT( din1,din2,din3,din4: IN std\_logic\_vector (7 downto 0);

clk: IN std\_logic;

dout: OUT std\_logic\_vector (7 downto 0));

END ENTITY roundR;

ARCHITECTURE behav OF roundR IS

TYPE state IS (s0,s1,s2,s3);

SIGNAL cs:state :=s0 ;

Signal ns: state;

BEGIN

PROCESS (clk) IS

BEGIN

IF rising\_edge(clk) THEN

cs<=ns;

END IF;

END PROCESS;

PROCESS (cs,din1,din2,din3,din4) IS

BEGIN

CASE cs IS

WHEN s0=>

ns<= s1;

WHEN s1=>

ns<=s2;

WHEN s2=>

ns<=s3;

WHEN s3=>

ns<=s0;

END CASE;

END PROCESS;

Process (cs) IS

BEGIN

CASE cs IS

WHEN s0=>

dout<= din1;

WHEN s1=>

dout<=din2;

WHEN s2=>

dout<=din3;

WHEN s3=>

dout<=din4;

END CASE;

END PROCESS;

END ARCHITECTURE behav;

## Router (M-ROU-09):

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

use IEEE.numeric\_std.all;

ENTITY router IS

PORT (datai1,datai2,datai3,datai4: IN std\_logic\_vector (7 DOWNTO 0);

datao1,datao2,datao3,datao4: OUT std\_logic\_vector (7 DOWNTO 0);

wr1,wr2,wr3,wr4,wclock,rclock,rst: IN std\_logic;

wreq , rdreq : IN STD\_LOGIC);

END ENTITY router;

ARCHITECTURE behav OF router IS

Component reg IS

PORT (data\_in : IN STD\_LOGIC\_vector (7 DOWNTO 0);

clock, reset, clock\_en: IN STD\_LOGIC;

data\_out : OUT STD\_LOGIC\_vector (7 DOWNTO 0));

END Component;

for all: reg use entity work.reg(behave);

COMPONENT demux IS

PORT( d\_in: IN std\_logic\_vector(7 DOWNTO 0);

sel: IN std\_logic\_vector(1 DOWNTO 0);

en: IN std\_logic;

d\_out1, d\_out2, d\_out3, d\_out4: OUT std\_logic\_vector(7 DOWNTO 0));

END COMPONENT demux;

for all: demux use entity work.demux(behav);

Component fifo IS

PORT( reset, rclk, wclk, rreq, wreq: IN std\_logic;

data\_in : IN std\_logic\_vector (7 DOWNTO 0);

data\_out : OUT std\_logic\_vector (7 DOWNTO 0);

empty, full: OUT std\_logic

);

END Component;

for all: fifo use entity work.fifo(struct);

COMPONENT roundr IS

PORT (clk: in std\_logic;

din1,din2,din3,din4: in std\_logic\_vector (7 downto 0);

dout: out std\_logic\_vector (7 downto 0));

END COMPONENT roundr;

for all: roundr use entity work.roundr(behav);

signal rego1,rego2,rego3,rego4: std\_logic\_vector(7 downto 0);

signal d11,d12,d13,d14,d21,d22,d23,d24,d31,d32,d33,d34,d41,d42,d43,d44: std\_logic\_vector(7 downto 0);

signal f11,f12,f13,f14,f21,f22,f23,f24,f31,f32,f33,f34,f41,f42,f43,f44: std\_logic\_vector(7 downto 0);

signal fe11,fe12,fe13,fe14,fe21,fe22,fe23,fe24,fe31,fe32,fe33,fe34,fe41,fe42,fe43,fe44: std\_logic :='1';

signal ff11,ff12,ff13,ff14,ff21,ff22,ff23,ff24,ff31,ff32,ff33,ff34,ff41,ff42,ff43,ff44: std\_logic :='0';

signal rd11\_val,rd12\_val,rd13\_val,rd14\_val,rd21\_val,rd22\_val,rd23\_val,rd24\_val,rd31\_val,rd32\_val,rd33\_val,rd34\_val,rd41\_val,rd42\_val,rd43\_val,rd44\_val: STD\_LOGIC;

signal wr11\_val,wr12\_val,wr13\_val,wr14\_val,wr21\_val,wr22\_val,wr23\_val,wr24\_val,wr31\_val,wr32\_val,wr33\_val,wr34\_val,wr41\_val,wr42\_val,wr43\_val,wr44\_val: STD\_LOGIC;

begin

---------------------------Registers Decleration---------------------------

reg1: reg port map( data\_in=>datai1, clock=>wclock, reset=>rst ,clock\_en=>wr1, data\_out=>rego1);

reg2: reg port map( data\_in=>datai2, clock=>wclock, reset=>rst ,clock\_en=>wr2, data\_out=>rego2);

reg3: reg port map( data\_in=>datai3, clock=>wclock, reset=>rst ,clock\_en=>wr3, data\_out=>rego3);

reg4: reg port map( data\_in=>datai4, clock=>wclock, reset=>rst ,clock\_en=>wr4, data\_out=>rego4);

---------------------------Multiplexers Decleration---------------------------

demux1: demux port map( rego1 ,rego1( 1 downto 0), wr1, d11, d12, d13, d14);

demux2: demux port map( rego2 ,rego2( 1 downto 0) , wr2, d21, d22, d23, d24);

demux3: demux port map( rego3 , rego3(1 downto 0), wr3, d31, d32, d33, d34);

demux4: demux port map( rego4 , rego4(1 downto 0), wr4, d41, d42, d43, d44);

-------------------------------FIFO Decleration------------------------------

fifo1: fifo port map ( rst , rclock , wclock ,rd11\_val , wr11\_val , d11, f11, fe11, ff11 );

fifo2: fifo port map ( rst , rclock , wclock ,rd12\_val , wr12\_val , d12, f12, fe12, ff12 );

fifo3: fifo port map ( rst , rclock , wclock ,rd13\_val , wr13\_val , d13, f13, fe13, ff13 );

fifo4: fifo port map ( rst , rclock , wclock ,rd14\_val , wr14\_val , d14, f14, fe14, ff14 );

fifo5: fifo port map ( rst , rclock , wclock ,rd21\_val , wr21\_val , d21, f21, fe21, ff21 );

fifo6: fifo port map ( rst , rclock , wclock ,rd22\_val , wr22\_val , d22, f22, fe22, ff22 );

fifo7: fifo port map ( rst , rclock , wclock ,rd23\_val , wr23\_val , d23, f23, fe23, ff23 );

fifo8: fifo port map ( rst , rclock , wclock ,rd24\_val , wr24\_val , d24, f24, fe24, ff24 );

fifo9: fifo port map ( rst , rclock , wclock ,rd31\_val , wr31\_val , d31, f31, fe31, ff31 );

fifo10: fifo port map( rst , rclock , wclock ,rd32\_val , wr32\_val , d32, f32, fe32, ff32);

fifo11: fifo port map( rst , rclock , wclock ,rd33\_val , wr33\_val , d33, f33, fe33, ff33);

fifo12: fifo port map( rst , rclock , wclock ,rd34\_val , wr34\_val , d34, f34, fe34, ff34);

fifo13: fifo port map( rst , rclock , wclock ,rd41\_val , wr41\_val , d41, f41, fe41, ff41);

fifo14: fifo port map( rst , rclock , wclock ,rd42\_val , wr42\_val , d42, f42, fe42, ff42);

fifo15: fifo port map( rst , rclock , wclock ,rd43\_val , wr43\_val , d43, f43, fe43, ff43);

fifo16: fifo port map( rst , rclock , wclock ,rd44\_val , wr44\_val , d44, f44, fe44, ff44);

---------------------------Valid Gates---------------------------

rd11\_val<=rdreq AND NOT(fe11); wr11\_val<=wreq AND NOT(ff11);

rd12\_val<=rdreq AND NOT(fe12); wr12\_val<=wreq AND NOT(ff12);

rd13\_val<=rdreq AND NOT(fe13); wr13\_val<=wreq AND NOT(ff13);

rd14\_val<=rdreq AND NOT(fe14); wr14\_val<=wreq AND NOT(ff14);

rd21\_val<=rdreq AND NOT(fe21); wr21\_val<=wreq AND NOT(ff21);

rd22\_val<=rdreq AND NOT(fe22); wr22\_val<=wreq AND NOT(ff22);

rd23\_val<=rdreq AND NOT(fe23); wr23\_val<=wreq AND NOT(ff23);

rd24\_val<=rdreq AND NOT(fe24); wr24\_val<=wreq AND NOT(ff24);

rd31\_val<=rdreq AND NOT(fe31); wr31\_val<=wreq AND NOT(ff31);

rd32\_val<=rdreq AND NOT(fe32); wr32\_val<=wreq AND NOT(ff32);

rd33\_val<=rdreq AND NOT(fe33); wr33\_val<=wreq AND NOT(ff33);

rd34\_val<=rdreq AND NOT(fe34); wr34\_val<=wreq AND NOT(ff34);

rd41\_val<=rdreq AND NOT(fe41); wr41\_val<=wreq AND NOT(ff41);

rd42\_val<=rdreq AND NOT(fe42); wr42\_val<=wreq AND NOT(ff42);

rd43\_val<=rdreq AND NOT(fe43); wr43\_val<=wreq AND NOT(ff43);

rd44\_val<=rdreq AND NOT(fe44); wr44\_val<=wreq AND NOT(ff44);

---------------------------Round Robin Decleration---------------------------

round1: roundr port map(rclock, f11,f21,f31,f41 , datao1);

round2: roundr port map(rclock, f12,f22,f32,f42 , datao2);

round3: roundr port map(rclock, f13,f23,f33,f43 , datao3);

round4: roundr port map(rclock, f14,f24,f34,f44 , datao4);

end architecture behav;

# APPENDIX B – VHDL Testbenches

## Register Testbench (M-ROU-01):

ENTITY regtb IS

END ENTITY;

ARCHITECTURE test of regtb IS

COMPONENT reg IS

PORT (data\_in : IN bit\_vector (7 DOWNTO 0);

clock, reset, clock\_en: IN bit;

data\_out : OUT bit\_vector (7 DOWNTO 0));

END COMPONENT;

SIGNAL clk, rst, en: bit;

SIGNAL din, dout: bit\_vector (7 DOWNTO 0);

BEGIN

dut: reg

PORT MAP (din, clk, rst, en, dout);

p1: PROCESS IS BEGIN

en <= '0';

rst <= '1';

din <= "11111111";

clk <= '0';

WAIT FOR 20 ns;

ASSERT dout <= "00000000" REPORT "reset error, case1" SEVERITY error;

clk <= '1';

WAIT FOR 20 ns;

ASSERT dout <= "00000000" REPORT "reset error, case2" SEVERITY error;

en <= '1';

rst <= '0';

clk <= '0';

WAIT FOR 20 ns;

ASSERT dout <= "00000000" REPORT "remain error, case3" SEVERITY error;

clk <= '1';

WAIT FOR 20 ns;

ASSERT dout <= "11111111" REPORT "load error, case4" SEVERITY error;

en <= '0';

clk <= '0';

WAIT FOR 20 ns;

ASSERT dout <= "11111111" REPORT "remain error, case5" SEVERITY error;

clk <= '1';

WAIT FOR 20 ns;

ASSERT dout <= "11111111" REPORT "load error, case6" SEVERITY error;

WAIT;

END PROCESS p1;

END ARCHITECTURE test;

## Demultiplexer Testbench (M-ROU-02):

LIBRARY IEEE;

USE ieee.std\_logic\_1164.ALL;

ENTITY bench IS

END ENTITY bench;

ARCHITECTURE test OF bench IS

SIGNAL i,clock,rst: std\_logic;

SIGNAL o: std\_logic\_vector (1 downto 0);

COMPONENT demux IS

PORT(d\_in: in std\_logic\_vector (7 downto 0);

sel: in std\_logic\_vector (1 downto 0);

en: in std\_logic;

d\_out1,d\_out2,d\_out3,d\_out4: out std\_logic\_vector (7 downto 0));

END COMPONENT demux;

FOR ALL: demux USE ENTITY work.demux (behav);

SIGNAL s: std\_logic\_vector (1 downto 0);

SIGNAL e: std\_logic;

SIGNAL inp,o1,o2,o3,o4: std\_logic\_vector (7 downto 0);

BEGIN

demx: demux PORT MAP(inp,s,e,o1,o2,o3,o4);

PROCESS IS

BEGIN

inp<="00000001"; s<="00"; e<='1';

ASSERT o1="00000001"

REPORT "FIRST VALUE ERROR"

SEVERITY ERROR;

WAIT FOR 20 ns;

inp<="00000010"; s<="01"; e<='1';

ASSERT o2="00000010"

REPORT "SECOND VALUE ERROR"

SEVERITY ERROR;

WAIT FOR 20 ns;

inp<="00000011"; s<="10"; e<='1';

ASSERT o3="00000011"

REPORT "THIRD VALUE ERROR"

SEVERITY ERROR;

WAIT FOR 20 ns;

inp<="00000100"; s<="11"; e<='1';

ASSERT o4="00000100"

REPORT "FOURTH VALUE ERROR"

SEVERITY ERROR;

WAIT FOR 20 ns;

inp<="00000000"; s<="11"; e<='0';

ASSERT o4="00000100"

REPORT "ENABLE ERROR"

SEVERITY ERROR;

WAIT;

END PROCESS;

END ARCHITECTURE test;

## Block RAM Testbench (M-ROU-03):

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE ieee.numeric\_std.ALL;

Entity RAM\_test IS

End Entity RAM\_test;

Architecture tb of RAM\_test IS

Component RAM IS

port( CLKA,CLKB,WEA,REA : in std\_logic;

ADDRA,ADDRB : in std\_logic\_vector(3 downto 0);

d\_in : in std\_logic\_vector(7 downto 0);

d\_out: out std\_logic\_vector(7 downto 0));

end Component RAM;

for all:RAM use entity work.RAM(behave);

signal CA,CB,WE,RE: std\_logic;

signal ADDA,ADDB: std\_logic\_vector(3 downto 0);

signal inp,oup : std\_logic\_vector(7 downto 0);

Begin

DUT:RAM port map(CA,CB,WE,RE,ADDA,ADDB,inp,oup);

p1:process

begin

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CA<='1';WE<='1';ADDA<="0100";ADDB<="0000";inp<="00000001";

WAIT FOR 20 ns;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CA<='1';WE<='1';ADDA<="1000";ADDB<="0000";inp<="00000011";

WAIT FOR 20 ns;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CB<='1';WE<='1';ADDA<="0000";ADDB<="0001";inp<="00000100";

WAIT FOR 20 ns;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CB<='1';WE<='1';ADDA<="0000";ADDB<="0010";inp<="00000110";

WAIT FOR 20 ns;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CA<='1';RE<='1';ADDA<="0100";ADDB<="0000";inp<="00000000";

WAIT FOR 20 ns;

ASSERT oup="00000001" REPORT"Read A1 Error" severity error;

wait for 20 ns;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CB<='1';RE<='1';ADDA<="0000";ADDB<="0001";inp<="00000000";

WAIT FOR 20 ns;

ASSERT oup="00000100" REPORT"Read B1 Error" severity error;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CA<='1';RE<='1';ADDA<="1000";ADDB<="0000";inp<="00000000";

WAIT FOR 20 ns;

ASSERT oup="00000011" REPORT"Read A2 Error" severity error;

CA<='0';CB<='0';WE<='0';RE<='0';

WAIT FOR 20 ns;

CB<='1';RE<='1';ADDA<="0000";ADDB<="0010";inp<="00000000";

WAIT FOR 20 ns;

ASSERT oup="00000110" REPORT"Read B2 Error" severity error;

Wait;

end process p1;

end architecture tb;

## Grey Counter testbench (M-ROU-04):

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY greyTest IS

END greyTest;

ARCHITECTURE tb OF greyTest IS

COMPONENT GreyGenerator IS

PORT (Clk, Reset, En: IN std\_logic;

count\_out : OUT std\_logic\_vector (3 DOWNTO 0));

END COMPONENT;

SIGNAL Clk\_s, Rst, En\_s: std\_logic;

SIGNAL countout: std\_logic\_vector(3 DOWNTO 0);

BEGIN

CompToTest: greyGenerator PORT MAP (Clk\_s, Rst, En\_s, countout);

Clk\_proc: PROCESS is

BEGIN

Clk\_s <= '1';

WAIT FOR 10 ns;

Clk\_s <= '0';

WAIT FOR 10 ns;

END PROCESS clk\_proc;

output : PROCESS

BEGIN

Rst <= '1';

WAIT UNTIL Clk\_s='1' AND Clk\_s'EVENT;

WAIT FOR 5 NS;

Rst <= '0';

En\_s <= '1';

FOR index IN 0 To 3 LOOP

WAIT UNTIL Clk\_s='1' AND Clk\_s'EVENT;

END LOOP;

WAIT FOR 5 NS;

WAIT;

END PROCESS output;

END ARCHITECTURE TB;

## Grey to Binary Testbench (M-ROU-05):

library ieee;

use ieee.std\_logic\_1164.all;

entity gbTestbench is

end gbTestbench;

architecture behavior of gbTestbench is

component greyBinary is

port( gray\_in : in std\_logic\_vector(3 downto 0);

bin\_out : out std\_logic\_vector(3 downto 0)

);

end component;

signal bin\_out,gray\_in : std\_logic\_vector(3 downto 0) := (others => '0');

begin

dut: greyBinary port map (

gray\_in => gray\_in,

bin\_out => bin\_out

);

sim : process

begin

gray\_in <= "0000"; wait for 10 ns;

gray\_in <= "0001"; wait for 10 ns;

gray\_in <= "0011"; wait for 10 ns;

gray\_in <= "0010"; wait for 10 ns;

gray\_in <= "0110"; wait for 10 ns;

gray\_in <= "0111"; wait for 10 ns;

gray\_in <= "0101"; wait for 10 ns;

gray\_in <= "0100"; wait for 10 ns;

gray\_in <= "1100"; wait for 10 ns;

gray\_in <= "1101"; wait for 10 ns;

gray\_in <= "1111"; wait for 10 ns;

gray\_in <= "1110"; wait for 10 ns;

gray\_in <= "1010"; wait for 10 ns;

gray\_in <= "1011"; wait for 10 ns;

gray\_in <= "1001"; wait for 10 ns;

gray\_in <= "1000"; wait for 10 ns;

wait;

end process;

end ARCHITECTURE;

## FIFO Controller Testbench (M-ROU-06):

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE ieee.numeric\_std.ALL;

Entity FIFO\_Controller\_test IS

End Entity FIFO\_Controller\_test;

Architecture tb of FIFO\_Controller\_test IS

Component FIFO\_Controller IS

Port( reset,rdclk,wrclk,rreq,wreq : in std\_logic;

write\_valid,read\_valid,empty,full: out std\_logic;

wr\_ptr,rd\_ptr : out std\_logic\_vector(3 downto 0));

end component;

FOR ALL : FIFO\_Controller use ENTITY work.FIFO\_Controller(behave);

signal rst,rclk,wclk,r\_req,w\_req,wr\_valid,rd\_valid,em,fl: std\_logic;

signal w\_ptr,r\_ptr : std\_logic\_vector(3 downto 0);

Begin

DUT:FIFO\_Controller port map(rst,rclk,wclk,r\_req,w\_req,wr\_valid,rd\_valid,em,fl,w\_ptr,r\_ptr);

p1:process

Begin

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='1';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

ASSERT wr\_valid ='0' REPORT"Write valid reset error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid reset error" SEVERITY error;

ASSERT fl ='0' REPORT"full reset error" SEVERITY error;

ASSERT em ='1' REPORT"empty reset error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0001" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0010" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0011" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0100" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0101" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='0' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0110" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='1' REPORT"full error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="0111" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='0';wclk<='1';r\_req<='0';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='1' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='0' REPORT"read valid error" SEVERITY error;

ASSERT fl ='1' REPORT"full 7 error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT w\_ptr ="1000" REPORT"Write pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='1';wclk<='0';r\_req<='1';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='0' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='1' REPORT"read valid error" SEVERITY error;

ASSERT fl ='1' REPORT"full 7 error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT r\_ptr ="0001" REPORT"read pointer1 error" SEVERITY error;

rst<='0';rclk<='0';wclk<='0';r\_req<='0';w\_req<='0';

wait for 20 ns;

rst<='0';rclk<='1';wclk<='0';r\_req<='1';w\_req<='1';

wait for 20 ns;

ASSERT wr\_valid ='0' REPORT"Write valid error" SEVERITY error;

ASSERT rd\_valid ='1' REPORT"read valid error" SEVERITY error;

ASSERT fl ='1' REPORT"full 7 error" SEVERITY error;

ASSERT em ='0' REPORT"empty error" SEVERITY error;

ASSERT r\_ptr ="0010" REPORT"read pointer2 error" SEVERITY error;

wait;

end process p1;

end architecture tb;

## FIFO Testbench (M-ROU-07):

library IEEE;

USE IEEE.std\_logic\_1164.ALL;

ENTITY fifotb IS

END ENTITY;

ARCHITECTURE test of fifotb IS

COMPONENT fifo IS

PORT ( reset, rclk, wclk, rreq, wreq: IN std\_logic;

data\_in : IN std\_logic\_vector (7 DOWNTO 0);

data\_out : OUT std\_logic\_vector (7 DOWNTO 0);

empty, full: OUT std\_logic);

END COMPONENT;

FOR ALL: FIFO use entity work.fifo(struct);

SIGNAL din : std\_logic\_vector (7 DOWNTO 0);

SIGNAL rst, rclk, wclk, rreq, wreq: std\_logic;

SIGNAL e, f: std\_logic;

SIGNAL dout : std\_logic\_vector (7 DOWNTO 0);

BEGIN

dut: fifo

PORT MAP (rst, rclk, wclk, rreq, wreq,din,dout,e, f);

p1: PROCESS IS BEGIN

din <= "00000001";

rst <= '1';

rclk <= '0';

wclk <= '0';

rreq <= '0';

wreq <= '0';

WAIT FOR 20 ns;

--ASSERT dout <= "00000000" REPORT "reset error, dout" SEVERITY error;

ASSERT e <= '1' REPORT "reset error, empty flag" SEVERITY error;

ASSERT f <= '0' REPORT "reset error, full flag" SEVERITY error;

rst <='0';

wreq <='1'; wclk <='1'; din <= "01000010";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "00000110";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "01100010";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "01001010";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "11000010";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "01001110";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "01110010";

WAIT FOR 20 ns;

wclk <= '0';

WAIT FOR 20 ns;

wreq <='1'; wclk <='1'; din <= "01101010";

WAIT FOR 20 ns;

rreq<='0';

rst <= '0';

rclk <= '1';

wclk <= '0';

wreq <= '0';

WAIT FOR 20 ns;

ASSERT dout <= "00000000" REPORT "remain error, dout case1" SEVERITY error;

ASSERT e <= '0' REPORT "remain error, empty flag case1" SEVERITY error;

ASSERT f <= '1' REPORT "remain error, full flag case1" SEVERITY error;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

ASSERT dout <= "01000010" REPORT "read error, dout case1" SEVERITY error;

ASSERT e <= '0' REPORT "read error, empty flag case1" SEVERITY error;

ASSERT f <= '1' REPORT "read error, full flag case1" SEVERITY error;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

rclk<='0';

wait for 20 ns;

rclk<='1';

rreq <= '1';

WAIT FOR 20 ns;

WAIT;

END PROCESS p1;

END ARCHITECTURE test;

## Round Robin Scheduler Testbench(M-ROU-08):

LIBRARY IEEE;

USE ieee.std\_logic\_1164.ALL;

ENTITY rbench IS

END ENTITY rbench;

ARCHITECTURE test OF rbench IS

COMPONENT roundR IS

PORT( din1,din2,din3,din4: IN std\_logic\_vector (7 downto 0);

clk: IN std\_logic;

dout: OUT std\_logic\_vector (7 downto 0));

END COMPONENT roundR;

FOR ALL: roundR USE ENTITY work.roundR (behav);

SIGNAL d1,d2,d3,d4,d0 : std\_logic\_vector (7 downto 0);

SIGNAL clock: std\_logic;

BEGIN

DUT: roundR PORT MAP(d1, d2, d3, d4, clock, d0);

PROCESS IS

BEGIN

d1<="00000001"; d2<="00000010"; d3<="00000011"; d4<="00000100"; clock<='0';

WAIT FOR 10 ns;

clock<='1';

ASSERT d0 = "00000001"

REPORT "ERROR in FIRST INPUT"

SEVERITY error;

WAIT FOR 10 ns;

clock<='0';

WAIT FOR 10 ns;

clock<='1';

ASSERT d0 = "00000010"

REPORT "ERROR in SECOND INPUT"

SEVERITY error;

WAIT FOR 10 ns;

clock<='0';

WAIT FOR 10 ns;

clock<='1';

ASSERT d0 = "00000011"

REPORT "ERROR in THIRD INPUT"

SEVERITY error;

WAIT FOR 10 ns;

clock<='0';

WAIT FOR 10 ns;

clock<='1';

ASSERT d0 = "00000100"

REPORT "ERROR in FOURTH INPUT"

SEVERITY error;

WAIT FOR 10 ns;

END PROCESS;

END ARCHITECTURE test;

## Router Testbench (M-ROU-10):

LIBRARY IEEE;

LIBRARY IEEE;

USE ieee.std\_logic\_1164.ALL;

ENTITY tb IS

END ENTITY tb;

ARCHITECTURE testr of tb IS

COMPONENT router IS

PORT (datai1,datai2,datai3,datai4: IN std\_logic\_vector (7 DOWNTO 0);

datao1,datao2,datao3,datao4: OUT std\_logic\_vector (7 DOWNTO 0);

wr1,wr2,wr3,wr4,wclock,rclock,rst: IN std\_logic;

wreq , rdreq : IN STD\_LOGIC);

END COMPONENT router;

FOR ALL: router USE ENTITY work.router (behav);

SIGNAL datai1,datai2,datai3,datai4,datao1,datao2,datao3,datao4: std\_logic\_vector (7 downto 0);

SIGNAL wr1,wr2,wr3,wr4, wclock, rclock, rst,wereq,rreq: std\_logic;

BEGIN

DUT: router

PORT MAP(datai1,datai2,datai3,datai4,datao1,datao2,datao3,datao4,wr1,wr2,wr3,wr4, wclock, rclock,rst,wereq,rreq);

PROCESS IS

BEGIN

---------- RESET-------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --20 ns

rst<='1';

wait for 20 ns; --40 ns

assert datao1<="00000000" Report "reset 1 error" Severity error;

assert datao2<="00000000" Report "reset 2 error " Severity error;

assert datao3<="00000000" Report "reset 3 error " Severity error;

assert datao4<="00000000" Report "reset 4 error " Severity error;

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --60 ns

wereq<='1'; wclock<='1'; wr1<='1';datai1<="10000000";datai2<="11000001";datai3<="11100010";datai4<="11110011";

wr2<='1';wr3<='1';wr4<='1';

wait for 20 ns; --80 ns

assert datao1<="00000000" Report "write 1 error" Severity error;

assert datao2<="00000000" Report "write 2 error " Severity error;

assert datao3<="00000000" Report "write 3 error " Severity error;

assert datao4<="00000000" Report "write 4 error " Severity error;

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --100 ns

wereq<='1'; wclock<='1'; wr1<='1';datai1<="10000001";datai2<="11000011";

wr2<='1';wr3<='1';wr4<='1';

wait for 20 ns; --120 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --140 ns

wereq<='1'; wclock<='1'; wr1<='1';

wr2<='1';wr3<='1';wr4<='1';

wait for 20 ns; --160 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --180 ns

wereq<='1'; wclock<='1'; wr1<='1';

wr2<='1';wr3<='1';wr4<='1';

wait for 20 ns; --200 ns

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --220 ns

wereq<='1'; wclock<='1'; wr1<='1';

wr2<='1';wr3<='1';wr4<='1';

wait for 20 ns; --240 ns

-----------------------------

rst<='0';

wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --260 ns

rreq<='1'; rclock<='1';

wait for 20 ns; --280 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --300 ns

rreq<='1'; rclock<='1';

wait for 20 ns; -- 320 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --340 ns

rreq<='1'; rclock<='1';

wait for 20 ns; --360 ns

assert datao4<="11110011" Report "read 1 error " Severity error;

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --380 ns

rreq<='1'; rclock<='1';

assert datao1<="10000000" Report "read 2 error" Severity error;

assert datao2<="10000001" Report "read 3 error " Severity error;

wait for 20 ns; -- 400 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --420 ns

rreq<='1'; rclock<='1';

assert datao2<="11000001" Report "read 4 error " Severity error;

assert datao4<="11000011" Report "read 5 error " Severity error;

wait for 20 ns; --440 ns

-----------------------------

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --460 ns

rreq<='1'; rclock<='1';

assert datao3<="11100010" Report "read 6 error " Severity error;

wait for 20 ns; --480 ns

rst<='0'; wereq<='0';rreq<='0';wclock<='0';rclock<='0';wr1<='0';wr2<='0';wr3<='0';wr4<='0';

wait for 20 ns; --500 ns

WAIT;

END PROCESS;

END ARCHITECTURE testr;

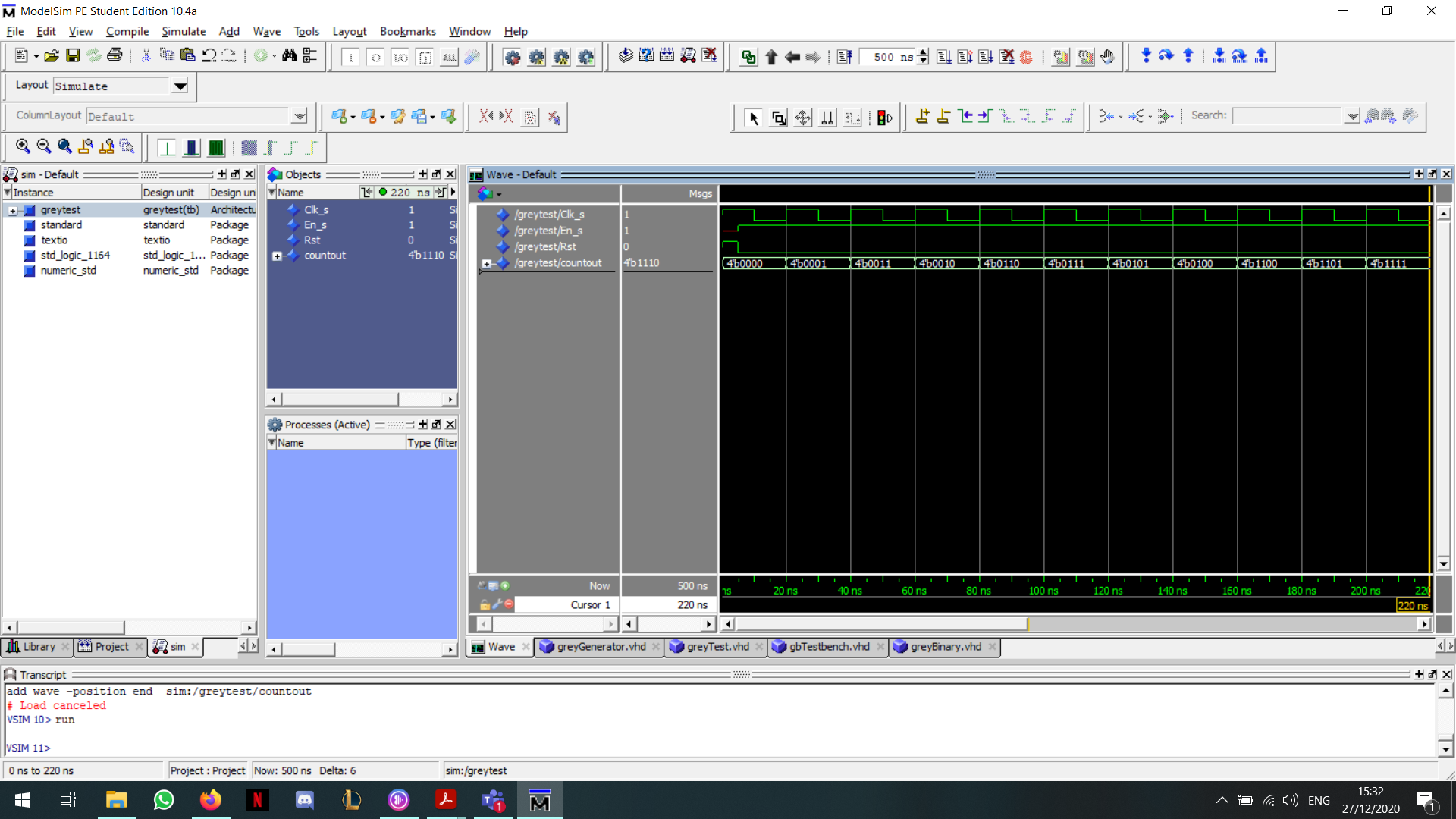
# APPENDIX C – Waveform Screenshots

## Register Waveform (M-ROU-01):

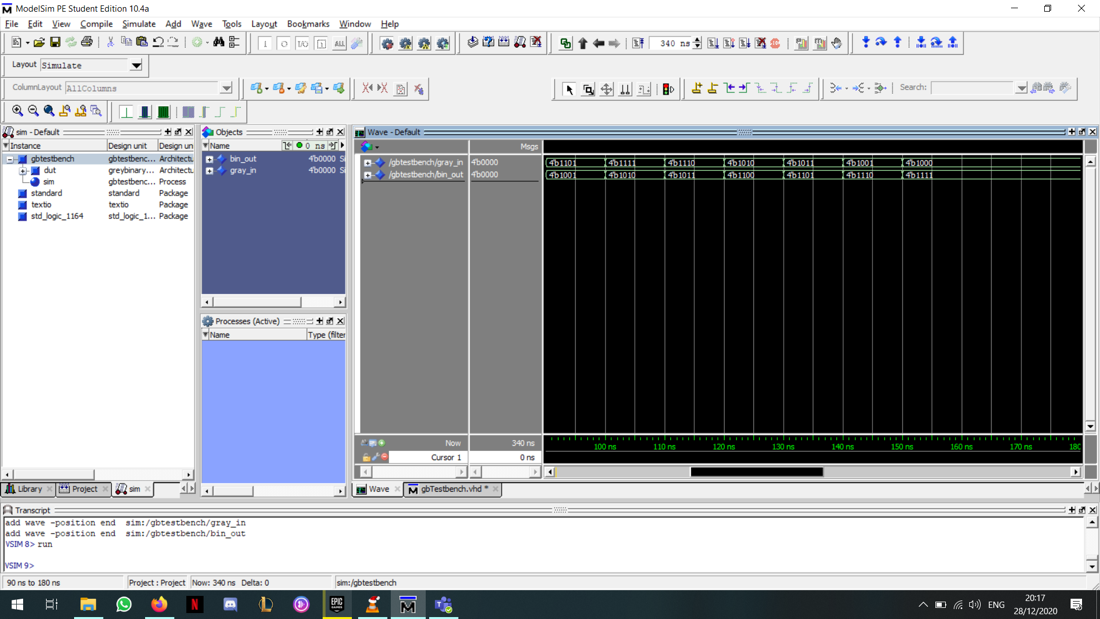
## Demultiplexer Waveform (M-ROU-02):

## Block RAM Waveform (M-ROU-03):

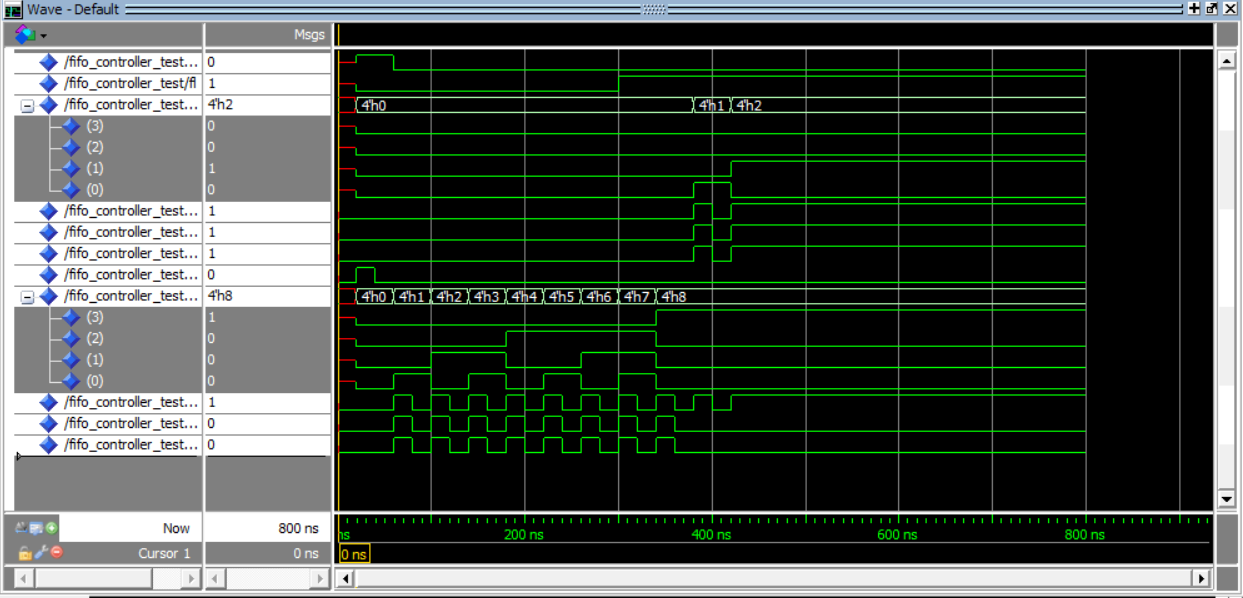
## Grey Counter Waveform (M-ROU-04):



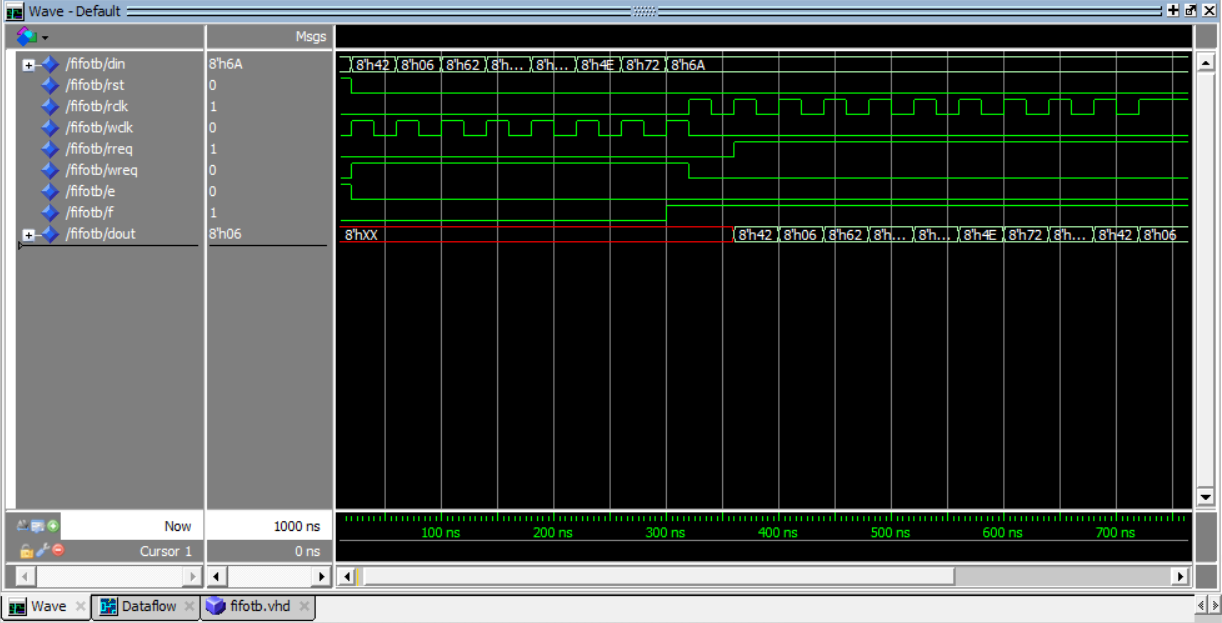
## Grey to binary Waveform (M-ROU-05):



## FIFO Controller Waveform (M-ROU-06):



## FIFO Waveform (M-ROU-07):



## Round Robin Scheduler Waveform (M-ROU-08):

## Router Waveform (M-ROU-10):

# References

1. Badrouchi, S., Zitouni, A. D., Torki, K., & Tourki, R. (2005). Asynchronous NoC router design.
2. Pallavi V., Gadgay B., Pujari V., (2016), Implementation of Enhanced NOC Router.
3. George, S. (2012). Design and FPGA Implementation of a Router for NoC.
4. Chatrath, A. K., Gupta, A., & Pandey, S. (2016). Design and implementation of high speed reconfigurable NoC router. *2016 International Conference on Inventive Computation Technologies (ICICT)*.
5. Avani, P., & Agrawal, S. (2018). Efficient Dynamic Virtual Channel Architecture for NoC Systems. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.