Department of Electronic and Telecommunication Engineering

University of Moratuwa

EN3030-Circuits and Systems Design

DPUT Processor

Report on Verilog based design and implementation of an image down-sampling processor on FPGA

A.G.D.K. Anhettigama 160026P

S.P. Dissanayake 160134U

M. N. J. U. Indrajith 160221J

W. W. A. T. E. Weerasinghe 160680M

16th July, 2019

Contents

[1.0 Introduction 3](#_Toc14174036)

[2.0 Instruction Set Architecture of DPUT processor 3](#_Toc14174037)

[2.1 Instruction types 4](#_Toc14174038)

[2.2 Registers & Flags 5](#_Toc14174039)

[2.3 Instruction Set 5](#_Toc14174040)

[2.4 Datapath 7](#_Toc14174041)

[3.0 Modules 8](#_Toc14174042)

[3.1 PROGRAM\_COUNTER 8](#_Toc14174043)

[3.2 AC\_ALU 8](#_Toc14174044)

[3.3 UART 9](#_Toc14174045)

[3.3.1 Baudrate Generator 11](#_Toc14174046)

[3.3.2 Transmitter 11](#_Toc14174047)

[3.3.3 Receiver 12](#_Toc14174048)

[3.4 INSTRUCTION\_DECODER 13](#_Toc14174049)

[3.5 LOOP\_REGISTER 15](#_Toc14174050)

[3.6 MEM\_REGISTERS 15](#_Toc14174051)

[3.7 REG\_BANK 16](#_Toc14174052)

[3.8 IRAM and DRAM 17](#_Toc14174053)

[3.9 PROCESSOR\_TOP\_MODULE 17](#_Toc14174054)

[3.10 TEST\_BENCH 17](#_Toc14174055)

[4.0 Design Summary 18](#_Toc14174056)

[4.1 Resource utilization 18](#_Toc14174057)

[4.2 Timing Summary 20](#_Toc14174058)

[4.3 Technical Aspects 20](#_Toc14174059)

[5.0 Simulation 21](#_Toc14174060)

[6.0 Supplementary Tools 22](#_Toc14174061)

[6.1 Compiler 22](#_Toc14174062)

[6.2 Simulator 23](#_Toc14174063)

[7.0 Down-sampling algorithm 24](#_Toc14174064)

[7.1 Filtering 24](#_Toc14174065)

[7.2 Down-sampling 27](#_Toc14174066)

[8.0 Results and Verification 29](#_Toc14174067)

[8.1 Transmitting and receiving down-sampled image 29](#_Toc14174068)

[8.2 Comparison between images down-sampled by the computer and FPGA 31](#_Toc14174069)

[9.0 Reference 32](#_Toc14174070)

[10.0 Appendix 32](#_Toc14174071)

[10.1 Verilog codes 32](#_Toc14174072)

[10.1.1 PROGRAM\_COUNTER 32](#_Toc14174073)

[10.1.2 AC\_ALU 33](#_Toc14174074)

[10.1.3 Baudrate Generator 35](#_Toc14174075)

[10.1.4 Transmitter 36](#_Toc14174076)

[10.1.5 Receiver 37](#_Toc14174077)

[10.1.6 UART 39](#_Toc14174078)

[10.1.7 INSTRUCTION\_DECODER 40](#_Toc14174079)

[10.1.8 LOOP\_REGISTER 43](#_Toc14174080)

[10.1.9 MEM\_REGISTERS 44](#_Toc14174081)

[10.1.10 REG\_BANK 45](#_Toc14174082)

[10.1.11 PROCESSOR\_TOP\_MODULE 46](#_Toc14174083)

[10.1.12 TEST\_BENCH 50](#_Toc14174084)

[10.2 Python codes 51](#_Toc14174085)

[10.2.1 Compiler 51](#_Toc14174086)

[10.2.2 Simulator 59](#_Toc14174087)

[10.3 Assembly code for down-sampling 64](#_Toc14174088)

[10.4 Schematics 68](#_Toc14174089)

# 1.0 Introduction

When a processor for a custom ISA is designed, we should have a tradeoff in hardware complexity, user/ programmer friendliness, performance, size and cost. Specially when your processor supposes to do a specific task, your instruction set and register pool should be able to perform that task efficiently using available resource/for a given cost.

For us our specific task was to down sample an image by a factor of two. Available resources were **Xilinx Spartan – 6 FPGA** with UART interface to communicate between Processor and Computer. We were able to achieve our task by optimized processor for looping operations, a memory which can be used to store two 256x256 images at one time and UART transceiver which can be used to send/ receive data from memory. Verilog HDL was used to implement modules and Xilinx ISE Design Suite as our programming environment.

DPUT Processor is an 8 – bit processor with 64kb (4096x16 bit) instruction memory and 1024kb (131072x8 bit) data memory. Even though this is an 8-bit processor (which has 8-bit dram) all the data registers inside the processor are 16-bit registers. This enables programmer to multiply 2 8-bit values without overflow. Excluding special purpose registers DPUT has 16 general purpose registers which reduces frequent memory access which improves the performance of the processor.

Since lot of looping operations are involved in image processing algorithms, we have introduced a new instruction with a special register and a flag which can reduce no of clock cycles w.r.t normal ISA. Also, instruction set with only 15 instructions have reduced the programming complexity. Since instructions are categorized into 5 instruction types with a meaningful way, instruction set is simple and easily understandable by the programmer.

We have also implemented a python compiler which reads a .txt file into a binary machine code which can be imported to instruction memory. In addition to compiler we also implemented a python-based simulator which reads an instruction txt file and displays the values of the registers when an instruction is getting executed. This can be used to debug your algorithm without repeatedly synthesizing your modules. Also, OUT port can be connected to any inside bus and it can also be used to debug your modules/algorithms.

# 2.0 Instruction Set Architecture of DPUT processor

The design objective of our ISA is to optimize the ISA for image processing applications and Serial Communication through UART interface. Since the image is a 2D-array of data and requires loops to go through pixels and execute operations, we have introduced an instruction called JUMPDEC which can be used to execute loop operations very fast.

DPUT ISA belongs to the RISC category of instruction sets. ISA contains only 15 simple instructions of equal length and all instructions requires only one clock cycle to execute. As in RISC, each instruction performs a specific task and there are separate instructions to access data from memory (Load/Store Architecture).

**Key features of ISA**

* No of instructions **= 15**
* No of instruction types **= 5**
* Avg. clock cycles per instruction (CPI) **= 1**
* Memory used per instruction **= 2 Bytes**

## 2.1 Instruction types

We have divided instructions into 5 main types based on their operation.

* **I-type** – When this type of instruction is executed processor is mainly in idle state (i.e. no internal operation is done.) UART module control instructions also fall into this category because in our processor UART module is operated as a separate module.
* **A-type** – These instructions are used to perform ALU operations
* **S-type** – These instructions perform shift register operations. Here AC register acts as a shift register
* **T-type** – These instructions are used to move data in one register to another.
* **M-type** – Deal with Memory. Contains two sub categories
* **M1-type** – deal with data memory. LOAD/STORE instructions.
* **M2-type** – deal with instruction memory. These instructions are used to perform branching operations

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction Type** |  | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| **type – I** |  | **OPCODE** | | | | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **type – A** |  | **OPCODE** | | | | **R** | | | | | **CONST** | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **type - S** |  | **OPCODE** | | | | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **X** | **N** | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **type - T** |  | **OPCODE** | | | | **X** | **S** | | | | | **X** | **D** | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **type - M1** |  | **OPCODE** | | | | **M** | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **type - M2** |  | **OPCODE** | | | | **INST** | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Instruction Types

* **OPCODE (4 bits)**  – Specifies the operation to be performed.
* **R, D, S (5 bits)** – Specifies Register address
* **CONST (7 bits)** – unsigned integer (0-127)
* **N (4 bits)**  – unsigned integer (0-15) specifies amount of bit shift to be performed
* **M (12 bits)** – Data memory offset
* **INST (12 bits)** – Jump instruction address (0-4095)
* **X (1 bit) -** Unused bit

## 2.2 Registers & Flags

There are 19 registers (including user inaccessible registers) in DPUT Processor. Most of them are 16-bit registers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Register** | **Name** | **Size (bits)** | **Address** | **Description** |
| PC | PROGRAM COUNTER | 12 | - | Contains the location of the next instruction to be executed |
| IR | INSTRUCTION REGISTER | 16 | - | Holds the current instruction being executed. |
| MBR | MEMORY BASE REGISTER | 16 | 00001 | Contains Base ([15:1] bits) of the dram address |
| MDR | MEMORY DATA REGISTER | 8 | 00010 | Stores the data being transferred in and from dram |
| UARTTX | UART TX REGISTER | 8 | 00011 | Contains data to be sent through UART interface |
| UARTRX | UART RX REGISTER | 8 | 00100 | Contains data came through UART interface |
| LR | LOOP REGISTER | 16 | 00110 | Keep the current cycle of the loop |
| ZR | ZERO REGISTER | 16 | 00000 | Contains constant zero value |
| AC | ACCUMULATOR | 16 | 00101 | Stores results of ALU |
| R0 – R15 | GP REGISTERS | 16 | 10000 –  11111 | General purpose registers - can be used to store values |
| Z | ZERO FLAG | 1 | - | Set to one if AC is zero |
| LRZ | LOOP OVERFLOW FLAG | 1 | - | Set to one if LR is zero |
| TXBUSY | TX BUSY FLAG | 1 | - | Indicate UART module is transmitting data |
| RXREADY | RX READY FLAG | 1 | - | Indicate UART module is being receiving data |

Registers and Flags

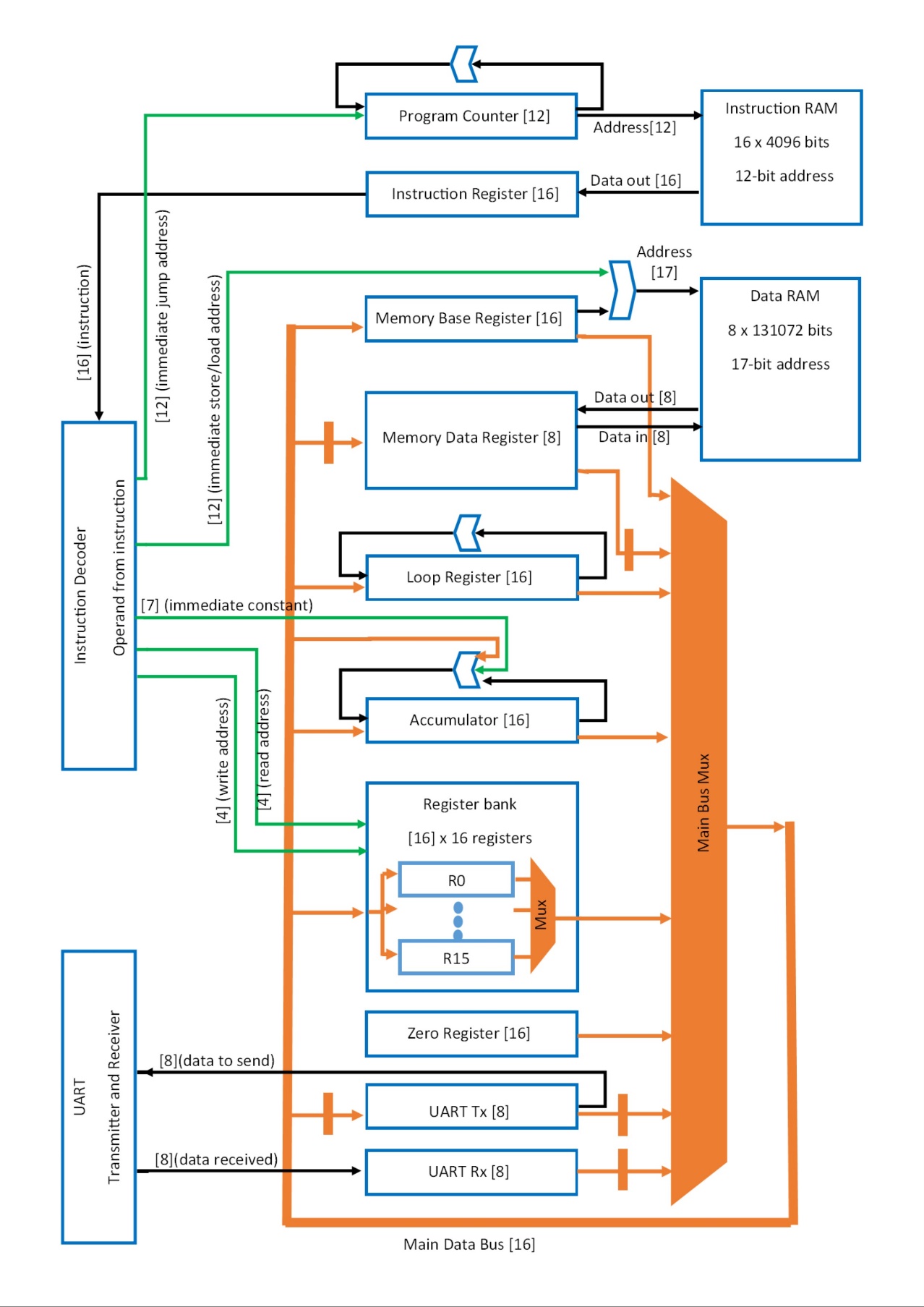
## 2.3 Instruction Set

Typically, all instructions are executed within one clock cycle. But for synchronization purposes we advise to add NOP instruction after UARTTX/UARTRX is used. Since the block RAM modules of the FPGA can be accessed within one clock cycle, LOAD/STORE instructions are also executed within one clock cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction** | **Type** | **Opcode** | **Description** |
| NOP | type - I | 0000 | No Operation |
| UARTSEND | type - I | 1101 | Wait for UART output to complete (followed by a NOP) |
| UARTREAD | type - I | 1110 | Wait for UART input to complete (followed by a NOP) |
| ADD | type - A | 0001 | AC ← AC + ( [ R ]+ [ CONST ] ) |
| SUB | type - A | 0010 | AC ← AC - ( [ R ]+ [ CONST ] ) |
| MUL | type - A | 0010 | AC ← AC \* ( [ R ]+ [ CONST ] ) |
| DIV | type - A | 0100 | AC ← AC / ( [ R ]+ [ CONST ] ) |
| SHR | type - S | 0101 | AC Shift right N bits |
| SHL | type - S | 0110 | AC Shift left N bits |
| LOAD | type - M1 | 0111 | [ M + 2\*MBR ] ← MDR |
| STORE | type - M1 | 1000 | MDR ← [ M + 2\*MBR ] |
| JUMP | type - M2 | 1001 | Jump to [INST] in IRAM |
| JUMPZ | type - M2 | 1010 | Jump to [INST] in IRAM IF Z FLAG =1 |
| JUMPDEC | type - M2 | 1011 | Decrement LR by ONE. Jump to [INST] in IRAM if LRZ = 0 |
| MOVE | type - T | 1100 | [ D ] ← [ S ] |

Instruction Set of DPUT ISA

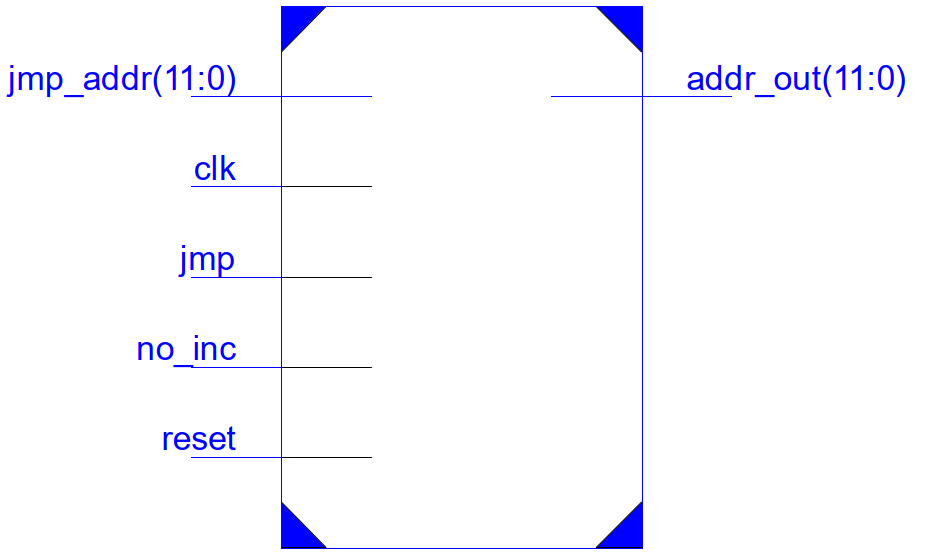
## 2.4 Datapath



Datapath

# 3.0 Modules

## 3.1 PROGRAM\_COUNTER



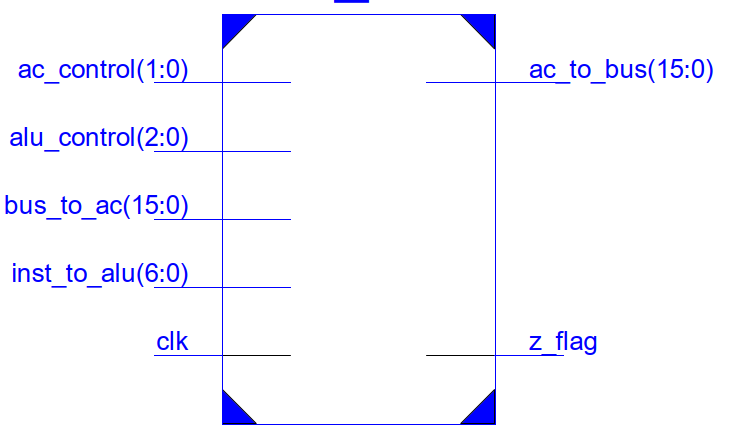
Program Counter is a 12-bit register which is sensitive to positive edge of the clock pulse.

|  |  |
| --- | --- |
| **Control Signals** | **Description** |
| Jmp | if ( jmp ) : addr\_out = jmp\_addr |
| no\_inc | If ( no\_inc ) : addr\_out will not increment |
| Reset | If ( reset ) : addr\_out set to zero |

In default configuration **jmp, no\_inc** & reset signals are set to zero. Therefore, at every positive edge of the clock pulse **addr\_out** will be incremented by 1. i.e it is pointed to next instruction to be fetched.

When type – M2 instructions use **jmp** signal to perform branching. If you want to set the processor into idle state **no\_inc** signal should be set. **reset** pin is used to restart the program.

## 3.2 AC\_ALU



This module includes Accumulator (AC) and Arithmetic and Logic Unit (ALU). ALU can perform 4 basic arithmetic operations: Addition, Subtraction, Division and Multiplication. AC can be used as a shift register which can shift 0-15 positions left/right (But it also done through ALU). **Z\_flag** set whenever AC is zero.

Main bus is connected to module through **bus\_to\_ac** port and constants from instruction **(CONST/N)** is connected via **inst\_to\_alu** port. AC’s value is always given out through **ac\_to\_bus** port.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **control Signal** | | **Selection** | **Operation** | **Description** |
| alu\_control (2:0) | | 000 | Addition | Data in Main bus and Constant from instruction is added to AC and stored back in AC |
| 001 | Subtraction | Data in Main bus and Constant from instruction is added together and substracted from AC and stored back in AC |
| 010 | Multiplication | Data in Main bus and Constant from instruction is added together and multiplied by AC and stored back in AC |
| 011 | Division | Data in AC is divided by the summation of Main bus value and Constant |
| 100 | Shift Left | Data in AC is shifted to N bits left (N is given as Constant) |
| 101 | Shift Right | Data in AC is shifted to N bits right ((N is given as Constant)) |
| ac\_control (1:0) | ac\_control[0] | 0 | AC = ALU output | write ALU output to AC |
| 1 | AC = Main bus input | write data in main bus to AC |
| ac\_control[1] | 0 | AC write disabled | don't write input data to AC register |
| 1 | AC write enabled | write input data to AC register |

## 3.3 UART

UART stands for *Universal Asynchronous Receiver and Transmitter.* This ‘Asynchronous’ term conveys that this protocol does not have a clock signal to synchronize output bits of the transmitter to the input bits of receiver. UART adds start bit and a stop bit at the beginning and at the end of each data packet to distinguish the data accurately. The UART module transmits data which is received by a data bus. In this implementation, the data which comes in a parallel form gets added the start and stop bits and sent to the Tx pin bit by bit serially. At the other end; the receiver collects the serially incoming bits and converts the data into parallel form and then removes the start and stop bit.

![A screenshot of a cell phone

Description generated with very high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RDiRXhpZgAATU0AKgAAAAgABAE7AAIAAAAIAAAISodpAAQAAAABAAAIUpydAAEAAAAQAAAQyuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFRoaWxpbmEAAAWQAwACAAAAFAAAEKCQBAACAAAAFAAAELSSkQACAAAAAzA3AACSkgACAAAAAzA3AADqHAAHAAAIDAAACJQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE5OjA1OjA3IDEzOjI0OjM2ADIwMTk6MDU6MDcgMTM6MjQ6MzYAAABUAGgAaQBsAGkAbgBhAAAA/+ELGmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTktMDUtMDdUMTM6MjQ6MzYuMDcwPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlRoaWxpbmE8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgCWAMEAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKakiSAmNgwBwcGvJP2ifE2reHPAsX9jXbWxuphFKyjkqQTwe3TtW78Ep5bn4U6TNcSPLK8eWd2JJPuTRD31J9gn7vL5noFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHz/+0T448S+FNY0qLw7rFxp6TKxkWIj5unqK9u8OXEt34Z064uXMksttG7uerEqCTXzl+1T/AMh7Rf8Acf8A9lr6I8Kf8ijpX/XpF/6CKKWtBt/zMK2laKXb9Ec98YdZ1DQfhjqeoaPdSWl3EE2TR9VywFc5+z54o1rxT4OuLrxBqM1/OsxUPLjIGfatb48f8ke1f6J/6GK5P9l3/kQrr/r4P8zRQ1dW/ZfoFXSNO3d/kbnxe+MifD0Rafplut3qs67lV8lUHuBzXlcVx8fvEsY1fS5L6C0u/wB5FFHLGiqPQBuR+Nex+L/hv4FufEaeKvFly8E4ddrXF2EiyBwMHjt0qleftDfD3S7ySya+uXMB2bobYuhx6MOCKiFre8/e/QqV7+6tP1PJNO+LfxH+G/iQWXj5ZryKVkLi6wSi99jLxn86+o9K1KDV9Ktr+0bdFcRh19sjpXyT8bvi5o3xCWCy0ewkMVq26O8kO0nPUbccdPWvor4PStN8KtGd+vlkcf7xraF5Um5LVNGc7RqJJ7o8l+JXxS8QeEfjJ9nTVbhdKiQsbRcbWPOB0z1xWVHqnxy8YyNrehSX1nplw5MSJIiqi/RuayvjNape/Hy3t5PuSSxK30319Z2FrFZafBa267YoYwiD0AFZ0V+5jN93/X5F1Xaq4rsj47+LVn8S4LOJvGtxcS6WXUReZIhXft9Bz610/wAGbP4pGbR5rWa6HhfP3RLHs249Otdl+1L/AMiLY/8AX2P/AEFq7D4Gf8ki0f8A65VWHdoyf8rXz9Sa6vyrvf5eh6HRRRSGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAfMX7VSMNb0VyDtKPz/3zX0J4RkWTwdpTRsGX7JHyP90Vwvxy+Gt34/8ADsL6QQdQsSWijPHmA4yPrxXi/hv4l/Ez4cWkmgSaLJerA2FW4gd/K9lK8YqaUkoSpve7f3lVYuUozXa39fce0/tC6lDZfCe9gkYB7llRATySGBrA/Zd/5EK6/wCvg/zNcDrnh3x/8S9Bu/E/i4TWthDH51nZRr94njCryR+Nei/szWd1ZeBrlLy2mt3M5IWWMqevvWlGLi6nNu0n+KX6EVZKSp8u13+R5n8QLvUfil8bo/DH2t4bFJhCiqeF45OO5zXruk/s4+A7LTYoNRs5dQuFHz3DzMhc/QHFee/GL4aeJNA8bN418HRyOikTO0H3oWHHTvn2zVOP4/8AxLa1FqvhpDIy7BKbSXdk8Z64zWVJpUVH7XX1Lqp+0cvs9C1+0DoPhnwb4S03QvD9lBbyPK0n96TGR1Y84r2P4M/8km0b/rm3/oRr5k8X+APiLqWlS+LPFkNxNIzIqI3zuyt0wF6AV9J/A+4kl+FWmxXEEkEsIZXSRCpHzHsa2ppqnPm3um/u/wCGM6lueDjtZr9Tw34tf8nEWf8A12i/9Dr6yj/1a/Svlf4q6Xfz/tAWc8FjcyQiaLMiQsVHz+oGK+qE/wBWv0qKX+7x9ZFVf479EeHftS/8iLY/9fY/9Bauw+Bn/JItH/65VR+PnhDUfFvgARaPD51zazCbZ3YAEYH515J8Lfih4z8N/wBm+GZdBZ9PEojMsltJuQE469KMPrz0+raCvtCfRXPqyigHIooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACkdxGjOxwqjJPtS1Def8eM//XNv5UAcE/x3+HUbsjeIowynBHkydf8Avmk/4Xz8OP8AoYo/+/En/wATXxNef8f0/wD10b+dQ0Afb/8Awvn4cf8AQxR/9+JP/iaVfjx8OnYKniFGYnAAgkJJ/wC+a+HwCzAKMknAAr6X+B3wMEKweKPGVtlyA9nZSL930dwe/oO2AaAPWP8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDi/8AhbXhH/n8uv8AwAn/APiKP+FteEf+fy6/8AJ//iK7SigDz2T47fDuKRo5fECo6nDK1vICD6Ebab/wvn4cf9DFH/34k/8Aia4j44/A9Nbjn8TeErcJqCgvdWqDAnHUso/vdfUnNfLMkbxSNHKpR0JVlYYII6g0Afbv/C+fhx/0MUf/AH4k/wDiaP8AhfPw4/6GKP8A78Sf/E18QUUAfoL4V8a6B41tbi48NX63sVu4SVgjLtYjIHIFb1eAfsm/8ir4g/6/Y/8A0Cvf6ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqG8/48Z/+ubfyqaobz/jxn/65t/KgD86Lz/j+n/66N/OoQCTgDJ9KnulL6hMqgsxlYAAck5r6Q+BnwOEIt/FXjC3zJw9nZSD7vcO3v0IH50AHwN+Bv2fyPFHjG2/e8PZ2Ug4A7O4/kORg19GAAAADAHQUAAAADAHQCigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArwX43fAwa8sviPwfbAakObizjGBP7qP73Tjvya96ooA/ObU9LvdG1KbT9VtpLW7gbbLDIMMp9DVWvtT4v8AwgsviFpZvLFUttbt0Jimx/rgP4G/XHua+NtV0q90TU59P1S3e2uoGKSRuOQR/P60AfTP7Jv/ACKviD/r9j/9Ar3+vAP2Tf8AkVfEH/X7H/6BXv8AQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFIWAIBIBPQZpaACiiigAooooAKKKAQelABRRRQAUUUUAFFFFABRRQTjrQAUUUUAFFGQelFABRQSAMk4HvRQAUUgIJIBBI600yxqcNIoPoWoAfRSKwYZUgj1BpenWgAopqyI5wrqx9jSgg9CD9KAFooooAKhvP+PGf/rm38qmqG8/48Z/+ubfyoA+efgh8Fre4dPGHiZI7hJXMlhbcMvU/O39B2xmvoyuP+E3/JKdA/69j/6E1dhQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXmHxg+D9l8Q9Ma8sFS312Bf3U3QTAfwN/IHtXp9FAHhf7MWl3ui6T4n0/U7d7e6gv0V43Ugg7P5V7pXE+Df+SgeOv+v22/8ARArtqACiiigAooooAKKKKACignHWigAooooAKKKKACiigEHpQAUUUUAFFFIGBJAIJHXBoAWiiigAooooAKKQkDqQPrSk460AFFAIIyDkUzzoh1kT/voUAPooBBGQcikZlQZZgo9zQAtFIrKwypBHqDQCGGQQR7UALRRRQB4/8UtbbQ/ib4XuRHNcfLKFt4TzKx24X0/Ous0f4ii71+HR9d0S80K7uV3Wy3To4m9gVJAP1rnPiDdWlp8YPCD3zKqHzVUscDcdmKsfEox3vjXwjaWREl8l6JsK3Kx4YE/TJFFL4IrvKS/H+mFT4pPtFM7DQ/FtvrmuavpsNtLE+lzGKR3Iw5wDkfnSeHPF9r4ifUxFbyW4064eCQyEHcVJBIx24rj/AABcxQ/ETxvHM4R0uTIwY4wuF5/Sl+ElxbXn/CUTI6SW8mpzHdn5Su80o3a/7cv89Aen/gVvwZfb4n3Ny1zNoPhPUdX0+3cob2CWNVYjg4DEHggjp2rrtB1u38Q6PDqFokiJKM7JUKsp9CDXnkPhjXfD1lqV58OfFVk+mM7TLaXUfnqj5+ZVfcAozntXY+AvE0vizwhaapcQiKaRcSKDxkentVRs0/K3r/kKWj8tf69Tb1G5mtNPmntrV7uWNCywRsAzn0BPFeafBnxJq+sW2pDVtNu0X7TIxu5pEKjDEbMA5yBx+Fepv9w/SvLvhQHl8F64kB/efbJwMHod7VEXyub3tH9UU1eMV/eX5M1rv4nMdSu4NB8OX+tW9kSLi7t5EVEI6j5iCcDnjNdP4c8RWHijRo9R0uTfE/yspGCjDqpHqK4n4O3EMHw/ntrqRVuLWSQXSM3zIcZO78KT4MjzLPXbmAH7JNqc7QtnIYbzyK0S1cd9E7/d/n+BLfXza/P/ACPTKwND8W2+ua5q+mQ20sT6XMYZHcjDnAOR+db5rzDwFdQw/EbxxHK4RkujKwbjC4Xn9KhP3mvJv8hv4b+aOw8NeLLfxNNqMdvbyQGwuXt3MhB3FSRkY7cVhah8THimvDonhu/1mzsm2z3dvJGqKe4wxBOPasv4TXsMq+KbuBhLEdSndSpzuG89KwtPk1vxF4R1TUY9XsdB8PLLKJLW1t9k2AxBJlDcEkZ6d6U21G66RT+9L+tC0lzW/vNfmdhqXxZ0mx0LR9Ut7W4vI9WkMcKREAq4GcHP5Vz/AMSvGfiKPwXZTR+GtR02Se+jSQNPESF3fdOD/FXFeHPLl8J+AgG8xRqUnJOf4K9Q+M5C+DbQk4A1O2yT/v1c4pWf95L8v8zODvp5P82v0Ov8O3lxf6BaT3ljLYStGMwSsrMvHqOOetXrqV4LSWWGFp3RSVjUgFj6DNNsDnTrYj/nkv8AIVMfun6UVd5WCGyPJfhb4q1zU9Y18ahot8QLqXEkksZEW1jiLAPXtnpXSr8T9NPhW71eS0nSe1mMDWBI80vkgKO3IGazPhOw/tDxSMjI1WbI9PnNcVq1xaH4pDxNbWzt4ftb1I7tlbEbTbcCXGOcYK/jQtXCPeMflovz29bDenPLs389/wDh/kztfGGr3Ovav4d8OQJJaNqD/aLyJmG5I1G4Ake611niW81fSdDC+F9K+33eRHGhYBYx/ebJGQPbmuMnZf8AhoOzkfBSbTFELbfdz1+leoUrXprzbf4tW+5Bf3vRL8r3/E8l+DraxFq3i5fEd4Lq9S6RpGUnauY87Rn06fhXNaFd+A9U13Xb7xkGu7m4v2MI8udwiYGOU4HOa7L4fOkfjDxw8qlkW7jLADOR5Q7V0vha58NXvhm5n8MWsdrYuX82NYfK+bHOV9cUr6Kf91foN/FOP94sWjWWh+F4m8JaW17akgx29tIAcHqcuawPifrd9baLpmnWDSWlxq9yluzqRuiBUk/jxio/g/8AaP7C1PexNoNSnFsCOi7u3t0qD4vQmF/DuqspMFjqKPM+cBF2tyfzqpJOUU9m4/jbQUbpStulL71cy/FegWnw3j0nXPDCPbP9oSG8UOSJ1YhfmyfVs8Vs2GoDw18VW0kO32HW4jcQqTkJIOWx9Swqt8YZ47/w3pGnWxWWe+vIjEityQrKxP5A1X8bxbfil4Ktrf8A1sQk3Y67Rsog25pPrJr5cqf4PUU7KOn8t/mnp960PVqKKKQwqG8/48Z/+ubfyqaobz/jxn/65t/KgDlfhN/ySnQP+vY/+hNXYVx/wm/5JToH/Xsf/QmrsKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOJ8G/8lA8df8AX7bf+iBXbVxPg3/koHjr/r9tv/RArtqACiig0AYGh+LbfXNc1fTIbaWJ9LmMMjuRhzgHI/Ojw14st/E02ox29vJAbC5e3cyEHcVJGRjtxXH+ArqGH4jeOI5XCMl0ZWDcYXC8/pSfCa9hlXxTdwMJYjqU7qVOdw3npSi9Lv8Akv8APQclrZfzW/BmpqHxMeKa8OieG7/WbOybbPd28kaop7jDEE49qTUvizpNjoWj6pb2txeR6tIY4UiIBVwM4Ofyrj9Pk1vxF4R1TUY9XsdB8PLLKJLW1t9k2AxBJlDcEkZ6d65rw55cvhPwEA3mKNSk5Jz/AAVVNOUlB/3fxFNpe8v734J/5ana/Erxn4ij8F2U0fhrUdNknvo0kDTxEhd33Tg/xV6V4dvLi/0C0nvLGWwlaMZglZWZePUcc9a5D4zkL4NtCTgDU7bJP+/Xd2Bzp1sR/wA8l/kKIfBL/F+iFLePp+pOeBXA2fxPk1XWbvT9F8M6hftZXLQXMkcsarHg43fMRkHB6V3p+6fpXm/wnVRqHilsAE6rNk/8DNTHWTXZX/FL9SnpG/ml+f8AkaafE/TT4VutXktJ457WUwNYEjzS+SAo7cgZrrrC5e8sIbiWB7d5EDGJyCyexxXhOrXFofikPE1tbO3h+1vUju2VsRtNtwJcY5xgr+Ne+RuskauhyrAEEelVHWCl3/D/AIff0sTLSbj/AF/S29SDUbma00+ae2tXu5Y0LLBGwDOfQE8V5p8GfEmr6xbakNW027RftMjG7mkQqMMRswDnIHH4V6m/3D9K8u+FAeXwXriQH959snAweh3tURfK5ve0f1RTV4xX95fkzWu/icx1K7g0Hw5f61b2RIuLu3kRUQjqPmIJwOeM10/hzxFYeKNGj1HS5N8T/KykYKMOqkeorifg7cQwfD+e2upFW4tZJBdIzfMhxk7vwpPgyPMs9duYAfsk2pztC2chhvPIrRLVx30Tv93+f4Et9fNr8/8AI9Lb7p+leHeFfF17pHxB8W2lhot9rc5vWYxW8ir5a4XnLkD8BXuLfdP0ryn4YXdmvxE8aW5dVuvthcgsMlML2qIfxP8At1/mipfw/mv1O18KeMbPxVazmKGS0u7VylxaTY3xH3xweh6UeGfGFt4n0q7vre2lgS1leJlkIJYqSMjH0rkfBYW7+LXiq+sMNZlIo2kVsq0g3ZH4Gq/ws1C1tvBfiJriZYxb3k/mljjZlmxmk5e45f3L/O6Fa0uX+9b8GdPafEnS7jwSfEs8Mttb7tixOQWZs4AGPUkCqcHxKvUuLX+2fB+qaZaXTBUupHjkUE9MqhJ/SuY8EaXo/iD4Jpaa1draW8suYpzKEKPuBUgn3xVvUNR8a/D1NJk1HWrLW9JknjtWX7P5c3zEBW3FjuP4Vq0lOz8vxS/G4nqny+f4PQ1NXvX8TfFTT9Ehcmx0+H7ZcqDjc+RtB/Bqr/Fa88Xy6Teaf4ctVsbFbcvPqUjA8DPyqAcgnHcd6Z4OfPxo8TmYYkmjikTK4+XYldh47/5EXVv+vZ/5VhPSipeTfz1/L9DWnrWt6L8mZng6/Nn8H9NvpnJMWmrIzMfRAa8z8EL8ObrSw/ihDdapcXMjPI8c5UZc4+YfL0xXo/hK4sbX4L6bNq0Jmsk0xDPGI9+5NgyNvfirqvoFx8N55NHhig0h7disaoEUDPp25rWs+SdSfb/g/mY0VzU4R7/1+BpSXH9i6fYQaFpMl7akhFFvIoEKf3iWPIriNUiXxv8AFefQNU3SaRplusj2wbCyuwOCcc8Y/Wtz4Sm9Pw10s6gzFvLOzeMHbk4zWLpCrpXx61ZLkeX/AGhaRtAzN9/aDnH5iqcbVuV67/fbT9fmCl+6bXkvlfX+uweGQfDHxOvPCNuznSbm1NzbxM2RDgqCB3wSSaueC786L421fwhI7NDF/pNnuOdsZONo/EE1Tt1XVf2gJLi3HmRafp5jldWyFcspAP4VHN+9/aNi+z9I9NXzsf7z1NNtuF+ql+F2vyKno526OL+btf8ANnqVFFFIDnvEfgXQfFd1Bca3atNLbqyxMsrIU3YyRg9eOtM8PeAtB8M3b3WmwTNcOMebczvMyj0Bckj8K6Sihe7sD13OW1r4deHdf1j+0763nS6KhWe2uXh8wA5+YKRu/GtDw/4T0fwxbzwaNa+RHcOZJFLFgSevWtmihaKyB66s4y9+FHhe+vZbl4byIzNueO3vpYoyf9xWA/SrV7rkfhPytM03wtq95bxoNrafbK0a+2dw5rqaKNUrLYN3dnGf8LCuf+hJ8Uf+AS//ABdXfDfg7RdIvZtX0yzurOe9G6WGWZsAnk5TO0GumoprTUN9DktW+GfhrWdUfULq3uIp5P8AWfZrqSFZP95VIB/GrN9qMPg6ytrLS/DmpXtuFwq6bbq4TH97kc10lFTsrLYN3dnGf8LCuf8AoSfFH/gEv/xdNm8E+H/GM8Wu6jpeoafeSIA8Zme3cgHOHVGwfxzXa0U9A1Mbw/4U0jwvBPDotr5EdxIZJF3FgSevWsdvhX4WOqPffZbgNI+94RdSCFj7x52n8q7GijrcOljkrP4ZeF7CSBrWykT7PcG4iXz22o5GCQM46DpW14g8Pad4n0l9N1iEzWzkMVDFSCOhBHIrToo3XL0DZ3OKj1xPCEY0XTfCviO9trYYSaKLzlYdeHZ8mpIvH9xLMkZ8GeJkDMBuezUAe5+fpXY0UXb1eoadDmLTwFoNvq91q9tb3MF1fKwnC3Dqp3dflBwDz1qzF4L0OHwzLoCWf/Evm3b0LElsnJO7rnNb1FGlrdA1vfqcD4s8Jz2Z0HU/DcMss+jS7REGLvJG3ykZPJwCTXeRP5kSuUZNwztYYI9jTqKd3a3nf7xWV7+VjL0zw7pukX1/eWMJSbUHElwS5O4gYHB6cVhar8N/C9zeT6pdrdW3WWUQXkkUfHJJRSB29K7GggEEEZB6g0u3kPv5nnV344+GV3o66WfEtnDaqQQtvcNEcj3XBqxe/ET4bajpLabe+I9PmtXjEbI8hOR9cdfeu0/s6x/587f/AL9L/hR/Z1j/AM+dv/36X/Ch63uC02PLtD1T4Q+Hr/7bYeIbeSZRhDc3kkwT/dD52/hWfpXj7wfqvxJvPEeq6/Z28Fon2axjlbBPUO/TocDFew/2dY/8+dv/AN+l/wAKP7Osf+fO3/79L/hTTd0+36isrNdzl/8AhbXgP/oadP8A++z/AIUf8La8B/8AQ06f/wB9n/Cuo/s6x/587f8A79L/AIUf2dY/8+dv/wB+l/wpDOX/AOFteA/+hp0//vs/4VyfxF+PGhaB4cWfwvd2Ot3UsvlPAJSNqFTluB9K9U/s6x/587f/AL9L/hWR4l8H+HvEOjvb6zpFrdRRgyIrJjDAHB4xQB80eF/2ldU8MeGLHRofD1pOlnHsWV7hgW5J6Y961v8AhrPV/wDoV7H/AMCX/wAK9B+HHwn8Dav8OtGv9S8N2lxdTwbpJX3ZY7iM9a6f/hSvw6/6FSy/N/8A4qgDxf8A4az1f/oV7H/wJf8Awo/4az1f/oV7H/wJf/CvaP8AhSvw6/6FSy/N/wD4qj/hSvw6/wChUsvzf/4qgDxf/hrPV/8AoV7H/wACX/wo/wCGs9X/AOhXsf8AwJf/AAr2j/hSvw6/6FSy/N//AIqj/hSvw6/6FSy/N/8A4qgDxf8A4az1f/oV7H/wJf8Awo/4az1f/oV7H/wJf/CvaP8AhSvw6/6FSy/N/wD4qj/hSvw6/wChUsvzf/4qgCl8HfiddfE7SNSvLzTobBrOdYlWKQuGBXOea9GrG8OeEdB8I280HhvTIdPincPKsWcMwGAeSa2aACiiigAooooAKKKKACiiigAooooAKKKKACsHxx4ik8J+B9V12CBbiSxgMqxOxAc5HBI+tb1VNU0uy1rS59O1S3W5s7hNksL5w6+hxQB80/8ADWer/wDQr2P/AIEv/hR/w1nq/wD0K9j/AOBL/wCFe0f8KV+HX/QqWX5v/wDFUf8AClfh1/0Kll+b/wDxVAHi/wDw1nq//Qr2P/gS/wDhR/w1nq//AEK9j/4Ev/hXtH/Clfh1/wBCpZfm/wD8VR/wpX4df9CpZfm//wAVQB4v/wANZ6v/ANCvY/8AgS/+FH/DWer/APQr2P8A4Ev/AIV7R/wpX4df9CpZfm//AMVR/wAKV+HX/QqWX5v/APFUAeL/APDWer/9CvY/+BL/AOFH/DWer/8AQr2P/gS/+Fe0f8KV+HX/AEKll+b/APxVH/Clfh1/0Kll+b//ABVAHn3wk+L+h6je+JtZ8UXtnotxqF3E6W7SEjCxBcg49q9J/wCFteA/+hp0/wD77P8AhWN8OfDejaN4t8aadpmm28FpBeW4iiCbgmYQTjOe5r0H+zrH/nzt/wDv0v8AhQBy/wDwtrwH/wBDTp//AH2f8KP+FteA/wDoadP/AO+z/hXUf2dY/wDPnb/9+l/wo/s6x/587f8A79L/AIUAeX61qvwh1/WP7TvvENsl0VCs9teSQ+YAc/MFxu/Gr/h/xf8ACvwvBPDouv6fBHcSGSRfOZgSevWvQf7Osf8Anzt/+/S/4Uf2dY/8+dv/AN+l/wAKForIHrqzyZpvg0dUe+/t+ANI+94RfSCFj7x52n8qks734N2EkDWut2afZ7g3ES/an2o5GCQOnQdK9V/s6x/587f/AL9L/hR/Z1j/AM+dv/36X/ChaWt0B67mNNH4b+JHhhkSePU9Mkkx5kDkDcvoRyCKzI9cTwhGNF03wr4jvba2GEmii85WHXh2fJrsooo4U2QxrGv91FAFPpbbB6nHRePriWZIz4M8TIGYLuazUAZ7n562NN8MaXp9terZW8kA1F2luB5h3Fm5Pfjr2rZop6WDUwYvBehw+GZdASz/AOJfNu3oWJLZOSd3XOap3muDwmkGlWPhzXNRghjASW0hEqgehZmyTXVUUahocZ/wsK5/6EnxR/4BL/8AF1d8N+DtF0i9m1fTLO6s570bpYZZmwCeTlM7Qa6aimtNQ30OS1b4Z+GtZ1R9Qure4ink/wBZ9mupIVk/3lUgH8as32ow+DrK2stL8Oale24XCrpturhMf3uRzXSUVOystg3d2cX/AMLBuT/zJPij/wAAl/8Ai6pwfDbwt4iEurz6TqWm3V65lmV7h4JckYw21unHSvQKKem4amboegad4c05bLSbcQwrycnczH1LHkn3NYd78MPDF/rE2pTWkyTTtulSK5dIpDjHKA7T+IrrqKHq7sForIwbTwVoNl4aOgwWQ/s8/wDLJmLY9wTyDWbpnwu8M6VqEV5BFeTSQ8xrdXssyKfXa5IzXYUU7u9+obqxw+vaHe2HxD0vxJpNtJNGyG0vI4lydhOd+PbaBXX39jb6pp8tneIXgnQq65xkGrNFTZcvK/6uO75uZblPTtJs9K0eDTLOEC0gjESRsd3ygYwc9a5ofC3wyNR+2LFeK2/f5IvZRFn/AK57tuPbFdjRT683UWy5ehmah4fsNRjtY5lljS1ffEsErRAH3CkZHtVbxH4O0fxUsQ1eGQvCcxywStE6/wDAlINblFD1DYx9C8NaX4V094NHt2AOWZncySOfdm5P4msDwVoN43iDVvE+s27291fPsghlXDRQjkA++c129FO7vfysKytYKKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqG8/48Z/8Arm38qmqG8/48Z/8Arm38qAOV+E3/ACSnQP8Ar2P/AKE1dhXH/Cb/AJJToH/Xsf8A0Jq7CgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDifBv/ACUDx1/1+23/AKIFdtXE+Df+SgeOv+v22/8ARArtqACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoqCa+tLe4it57mKOabPlxu4DPjrgd6SPULOa8ktIrqF7mIZkhVwWT6jqKALFFFFABRVTUNVsNJtxPql7b2cRO0PPIEUn0yaq2PijQdUuRb6brNhdzHpHDco7H8AaN9g2NWis/Udf0jR3RNV1Szsmf7ouJ1Qt9MmrsM0VxCssEiyRuMq6HII9jQA+iiigAooooAKhvP8Ajxn/AOubfyqaobz/AI8Z/wDrm38qAOV+E3/JKdA/69j/AOhNXYVx/wAJv+SU6B/17H/0Jq7CgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDifBv8AyUDx1/1+23/ogV21cT4N/wCSgeOv+v22/wDRArtqACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAGu6xxs7nCqMk15zB498U+IZLu68GaDY3ml2zlBNd3LxvMRydgCkH867/Uomn0y5ij+88bAflXm/wi1rT9J8Eyadqk8Njd6fI6zxzNtbjnOD1/CktZSv0V/x/Qb0ivN/oV9d1C51Tx34Hur6xmsLh1m8y3mGCjfJn8Kq2WuWXh34weMdS1OURQQwIxJ7/InAq74k1ey1z4ieDL3TJvOt3M4V9pXONnqBXK6t4MTxb8YPFgE8kdzbwI9vg/KHCJgkd+9EW1ZrT4/0Cye/aP8A6UeteDNf1rxHZyX2qaVHp1pIc2o3kyOvqykcfma6auQ+HHioeJfDarcgRajZnyLuDGCjj2/EV19XJJOy2/rUiLbWu55R8XNQtv8AhKPC+m3drNfJ9p+0NaQRCRpgMjAU8HrXReHdZ8OyakYIfDbaDqCxl40u7GOCSQY6rt/GtDWfFNlpHirT9N1GxZRdKTHfvt8uM5+6SeQa5bxgF174jeGU0WZZmgeR55oGDKiFGABI461Edkl1b/r8C5dZPol/XzuQ+AtNs/GGr+IdX163ivm+2SWsSzAOsaIxHyg8DjuK5bT/AIiXnw9m1PRorP8AtGKHVVt4I2kJZEdS/Hrz2rq/h1fWvhjVfEej6xPFZypeyXKea20OjsSCCevFeb31u93ry66Y8W+oeII/IYgjcqxspPPuKcNZwitmop/fH/g/iKfwzb3Tdvuf6W/A9F1H4neKfDupWNv4j8NWqLqXyWhs52kbfjgOCBgdM9a19L8ca7b+MbbQPGOlWVhJexs9tLaXDSqxGMg7gMHkVkfE8Z8YeCs/8/bf+gU/xx/yWXwl/uzf+yU6erjfrJr7l+Yp6KVukU/6+49SooopDCobz/jxn/65t/KpqhvP+PGf/rm38qAOV+E3/JKdA/69j/6E1dhXH/Cb/klOgf8AXsf/AEJq7CgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDifBv/JQPHX/X7bf+iBXbVxPg3/koHjr/AK/bb/0QK7agAooooAKKKKACiiigAopGdUXc7BR6k4pQcjI5FABRRTVkRmIVlJXqAelADqKKKACiimrIj52MrY4ODnFADqKKKACiio47iGZ3SKWN2Q4YKwJU+/pQBJRRRQAUUUUAFFMaaJZViaRBIwyqFhk/hTmZUUs7BVAySTgCgBaKj8+LyPO81PKxu8zcNuPXNVl1vSmYKup2ZY8AC4Xn9aALtFHXpUVxdW9pHvup4oEzjdI4UfmaAJaKhS9tZLc3EdzC0I6yLICo/HpUqsGUMpBB6EHrQAtFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFYOp+BvC+tX/23VtBsLy6/57TQhm/Ot6ijzAoHQ9LaS0kNhAXsl22zbBmIccL6dBT4dI0+21KbUILOGO8nAEs6oAz49T+FXKKAOa1S1uNDu5L/AMK+F7W/vbxv9KkFwluze5JB3VR/4SHx1/0IsP8A4OI//ia7OigDDSw/4SjQxD4w0G3j3HLWcsi3Cj3yBip9J8PaF4Yt5f7G06002EgvJ5KBF47mtWkdFkRkdQysMMpGQR6U+ugepxWs3Hwz8QzpNrl14dv5EGFe4nicgfiaszax8P7iC2hn1PQpIrRg0CNcxkREDAK88Vr/APCK+Hv+gDpn/gHH/hR/wivh7/oA6Z/4Bx/4UlpsG5l3niDwJqE9vNe6vok8ts26F5LqMmM4xkc8Utx4h8C3d/Be3Wr6JLdW+fJme6jLJnrg546Vp/8ACK+Hv+gDpn/gHH/hR/wivh7/AKAOmf8AgHH/AIUAQf8ACceFf+hk0r/wMj/xo/4Tjwr/ANDJpX/gZH/jU/8Awivh7/oA6Z/4Bx/4Uf8ACK+Hv+gDpn/gHH/hQBB/wnHhX/oZNK/8DI/8aiu/G3hZrKcDxHpRJjYAC8j9PrVz/hFfD3/QB0z/AMA4/wDCorvwr4fFlORoWmAiNsEWcfp9KAOS+F/i7w5Z/DHQ7e717TYJo7ch45LpFZTuPUE11f8AwnHhX/oZNK/8DI/8a5n4W+HNDufhfoc1zo2nzSvbks8lqjM3zHqSK63/AIRXw9/0AdM/8A4/8KAIP+E48K/9DJpX/gZH/jR/wnHhX/oZNK/8DI/8an/4RXw9/wBAHTP/AADj/wAKP+EV8Pf9AHTP/AOP/CgCD/hOPCv/AEMmlf8AgZH/AI0f8Jx4V/6GTSv/AAMj/wAan/4RXw9/0AdM/wDAOP8Awo/4RXw9/wBAHTP/AADj/wAKAIP+E48K/wDQyaV/4GR/40f8Jx4V/wChk0r/AMDI/wDGp/8AhFfD3/QB0z/wDj/wo/4RXw9/0AdM/wDAOP8AwoAv2V9a6japdafcxXVu/wB2WFw6t9CKnqK1tLeyt1gsoIreFfuxxIEUfQDipaACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4nwb/yUDx1/wBftt/6IFdtXE+Df+SgeOv+v22/9ECu2oAKKKKAOc0Dxamu6/rOmLaNCdKnMJkL58zgHOMcdaXwt4tTxNNqca2rW/8AZ909uSX3b9pIz046VyHga+t7T4j+OIrmQRstyZju4+TA5/Q034R6hDcw+J760YTRNqUzoV/iG80otWu/5L/O6/4I5LWy/mt+DNa5+ImpT/brjw34bbVNPsWKS3LXawkkddqkEtjnpVfUPi7Z2/h/RNTsdOlvBq8phSIPtKOB0PHrXLWkl94p8F6tq2reIk0rR4pZRJp1pbpHhVYjmRcMCSM/jXNeGWik8J+AljIdV1KUcnP8FVTTlNQf938fwFNpLmX978Ezt/ih4g8Vp4KspJNBFhJLfxLKqagrYG7hcgchv0r0rw/PeXGg2kmo2Ys5zGN0IlEmOOPmFcd8aGCeDLRmOANTtiT6fPXd6ewbTbYqcgxLgj6CiHwS/wAX6IUt4+n6j7l5Y7WR7eLzpVUlIy23cfTPavKfhVrXiK91jxD9u0nKi8lzI14rGNgxxGBjkds160fun6V5t8J5U/tHxSm4bl1SYkZ6fOamHxy/wv8ANDl8C9V+TLo+KVsPDV5fS6fImoWtwbb+zvMy7yZO0A47gZ6V21jPNc2MM1zB9nldQzRbt2w+me9eD6rcxTfEr/hNraz3aFY3iwXDbiFZgMGXb0OCNv4175DKk0KSRkFHUEEelVHWmpPd/h/w+/zCWk3H+v6WxFqEtzBYTSWVuLmdVJSIyBN59Nx6V5X8JfEmqLousXviCz8qzhuJXkvHug5BDkbduM8dM+1etv8AcP0rwjR7WfUvgv4rtdP+e4+1TfIp5/1hNZ8zi5tfy/qi1FS5V/eX5M7IfE/Uzpja2PCk/wDYKtxeG4HmFM43eVt3YrvNM1K21fTLe/sJPMt7hBJG3qCM1wq6/pJ+CLXX2mJ4BYmI8/xkEAfXJFafwptZ7P4aaRHdI0bmBWCt1AIGK2cUpTj/AC2177/5X+ZipNxjLvf9DsG+6fpXhvg/xBqun/EbxfZ6FoTatcNel3BnEKouF/iIIJ46V7k33T9K8l+GWqWMXxG8aWckkaXQvDIcnkrhf8Kzh/F/7df5o0l/D+a/U6XT/iRb3nh3Wb2awkt7/RkdrqxZ+QVB6NjkHB5xV258apb/AA6k8VmyZkSDzvs+/k84xnHvXm8q/wBq6p8RNU0wCSyawMAdDkPIquGH51b1XW9PT9mxi9wi+dbeRGCeWcN0/Q1MpP2bl1tH8b3KjFe0Uel2vut/wTtdX+ICafpmlPaae93qOrf8etmJNu44ycvjAwM0um+MNZOvRaZ4h8MT6aZ1Jimhm+0x8dmZVwv41h6homg+IPCfhu01PVf7L1NVD6fPG+HDgZOBkZ4HQ0yLWfE/hz4h2GganrUeuWeoQuUf7OkUkJUDkheo59e1bTXLNx83b5XsvUxTbpqXkm/8/QseDpj4r+IGta7MRJbae4tLMHnbwCxH/Agah+LUXjG/0a8tdFeHTtKSBnuLwuGkkxn5AoII6dfenfBb5NG1i3cETRajJ5gIweWYj9K6vx3/AMiLq3/Xs/8AKsaq5aat0Sf3q/5s6KWtZ+tvudvyRweo6iun/s12rSSCP7Rp0cAY+rJiqvhCP4ZPDpttJ4WMF55aD7ZeaS0SM+OvmMMda6DQ9Y0vRfg3oFzrtkbuzNvAjL5KyBSVGGIPYetS/FeW3uPhjMLPHmTKn2RY+u44IxjpxW1R8tST/vL+vxOekuenGL7P+vwOye5v01aG3g09XsGTLXXngbD6bMZPavPL62h8bfF660bWo1utJ0u3RxaOcxu7A8svfG2vRdHWaPR7SO7fdOIl3k9ScV57p23RfjzqQu9sSapaIYGY/fKA7gP++hUpL2qi9fi++2n/AACuZuk5LTb7rq5TsdLtdJ+IepeC4olGh6hYtPHaZ+SNgVGFHYZJNdD8LtblvNLvdGvHaS50e4Nszt1cAAg/rWVZ7dZ+PVzeWu2W302xMMkingOSrAflTfhmPO+InjW6i/1DXpVfTOFpU22kn1i390tPw/AdTRu3Rr8Vr+J6jRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVDef8AHjP/ANc2/lU1Q3n/AB4z/wDXNv5UAcr8Jv8AklOgf9ex/wDQmrsK4/4Tf8kp0D/r2P8A6E1dhQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFZHivxDD4U8Kahrt1DJPDYxGV44yAzDPQZ+ta9ZviHQrPxN4evNG1PzPsl7H5cvlttbHse1AHiv/AA1h4e/6F3U/+/kf+NH/AA1h4e/6F3U/+/kf+Nbv/DMnw/8A7up/+Bf/ANjR/wAMyfD/APu6n/4F/wD2NAGF/wANYeHv+hd1P/v5H/jR/wANYeHv+hd1P/v5H/jW7/wzJ8P/AO7qf/gX/wDY0f8ADMnw/wD7up/+Bf8A9jQBhf8ADWHh7/oXdT/7+R/40f8ADWHh7/oXdT/7+R/41u/8MyfD/wDu6n/4F/8A2NH/AAzJ8P8A+7qf/gX/APY0AYX/AA1h4e/6F3U/+/kf+NH/AA1h4e/6F3U/+/kf+Nbv/DMnw/8A7up/+Bf/ANjR/wAMyfD/APu6n/4F/wD2NAGZ8Nfi94b1DWvFOsXrXGnpqF3C8ULwPKwCxBTkoCOor0D/AIWz4O/6CU3/AIBT/wDxFYfwj8Maf4P17xjomjiUWdtewbPNfc3MIJyfqa9OoA43/hbPg7/oJTf+AU//AMRR/wALZ8Hf9BKb/wAAp/8A4iuyooA8t1fX/hTr2pLf6tH9oulAUSm0uVOAc84UZq7o/jf4caBHLHo8htEmcvIsdlP8zHqeVr0WihaaIHrueTNqfwibVDqJtx9qZt5f7Jc4J9cYx+lS22t/Cazkje2hEbRTGdCLO4+VyMFh8vpXqlFC0tboD13MFG8P/EDw4d0Y1DTZXKlZY3jyy+xwaxH1nUPCjf2R4f8AAGp3Vhb8RS29xEEOeeN75/Ou5opDOHh8c+I5JkR/h3rMaswBdri3wo9fv1s2/gzw9HqU+qJpMcN7doRO4ZssG6g84rfop6C1MmPwxosOgyaNHp0Q06QHfb87WycnvnrXOy+JdX0GQ6XpHw/1W4srX5IZYbiEIy+o3Pn867iijW9w0scVaeNvEVxeRQzfD7WLeN2CtK9xAVQepw+a2tM8IaDpOoT6jp2lx211dA+c6s3zZ5ORnFbdFAHMf8K38I/2mdQ/sSD7SX8wvufG7125x+lM1rxRrOkagbTTfBepapAqjbcW00KofYBmBrqqKPLoHmcKvjrxIzAH4ca0ATgk3Nvx/wCP1en+HPhPVme81Dw/ELi5bzZtzsGLEYOSrV1lFGgalHTdG07R9OWw020jt7VRgRqMj9etYX/CsPBm+Zv7Bt8z58wF3w2Tk8ZwOa6uih6u7BaKyMnUfC2i6rpMemahp8U1nGAEiJI2/Qg5qDRPBXh3w5cPPoulxW0rjDOGZjj/AIETW7RTu73DpY4rS9AvvD/xGv7qygaXS9WUSSlSAIZFAUce4BNdfeWdvqFnJa3kQlglUq6N0YelTUUre6o9h395yKMWjadDoy6THaRiwSPylg6qFxjH5VlaZ8P/AAvo9+LzTtIihuB0fe7Y/AkiujoovrfqK2luhRk0awl1aLU5LcG8iXYku4/KPTGcd6r674X0XxLCkeuWEd2sZym4kFfxBBrWoot0AwDo1t4W8MXUPhXSRvVCY7eJuXY+7H37mq/gLwy/hnw6sV22+/uGM11J/ec//WArp6Kabu33Dol/X9f5hRRRSAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACijp1qOKeGcEwSpIFOCUYHB9OKAJKKKKACiiigAopnnxed5Pmp5uM7Nw3Y9cU+gAooooAKhvP+PGf/rm38qmqG8/48Z/+ubfyoA5X4Tf8kp0D/r2P/oTV2Fcf8Jv+SU6B/wBex/8AQmrsKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOJ8G/8lA8df9ftt/6IFdtXE+Df+SgeOv8Ar9tv/RArtqACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK5zxz4vi8E+Hxqs9o90hnSExo2D8xxnoa6OvPfjOAfBtoCMj+07b/0Opd7pd2vxZUbX1N7TvFF3/wj9zrHiPShpFrCglQ/aVlLoRnJwBj6Vzg+KGp/2Umtv4TnGhuwxdi5BfYTjd5W3dirPxQsLq/+D97DYA+YLVThepGB271g2GmX+ofDuO4n+IyQ6RJb7X/4lsIVFPBGapt3m19m39MlLSKfW/8AS/M6q98f+dqFvYeFNNOtXc0AuCvniFUjIyCWII5HapvCXjuPxHJqFrfWLaZf6c225t3kDheM5DYAPFed+E47y61yTwz4Y8TQ2ljY2URXUFtI5JbpWU45PQDHr3qn8PLa2bxR43sdGvpLximFmdy7O+0bjyT3zx2oejdtfdbXy2BapX7pP5ncS/Em/wBTs7678N+GpdS0u23JJdm5EROByVQjLDHpXLfC/wAUjSvhvr3iGGykuY11CeZoc7GCbiSeR2rovhfq2n23wpFrcyxwzadE8d6j4BRhknP4EVm/BqGz13wXr0Ue17O8vrhQVHBRmP8AShpxc1HX3f1X59AupKLlp736P8jtdW8b22naDpmowwfaW1J0SGFXAJZlzjPtiunRtyKxGMjOPSvBfAnna74w07QLpJCnhuSWWQuON3mHYP8Avlq97qmla62eq9OhOqfK91o/UhvLuGwspru6cRwwoXdj2A61wMXxK1i902bV9K8IT3ekRsQLgXQWRwDglYtu410HxCtp7z4e61BaqXle0kCqvU/KeKzvh9rmlxfDKyu3uYoLe1i2zMzYEZBwc/jUR15m+lvxv/kW9orvf9P8zEhu1vvjpa3UaSIsukIwWRCrD5n4IPStTRviLqHiHxBeabo/hxpo7C7a3urh7tUCANjcAR83Q8CsuO8t9Q+O1td2Uyz28ukK0ciHIYbn5FT/AAnAF/4oOBk6rNk/8DNVT1jFPopP/wAnt+pMur84/wDpJ6SOnNFFFIYVDef8eM//AFzb+VTVDef8eM//AFzb+VAHK/Cb/klOgf8AXsf/AEJq7CuP+E3/ACSnQP8Ar2P/AKE1dhQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBxPg3/koHjr/r9tv/AEQK7auJ8G/8lA8df9ftt/6IFdtQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVLVdH0/W7VbbVbVLmFJFlVHzgMpyDx6VdooAZ5Ufk+VsBj27dpHGPSuWk+F/g2W7Ny+hQmUuJCfMfG7rnG7H6V1lFHW/UOljntW8BeGNcuEn1TSIZpY12qwZkIHp8pFTWHg7w/pmqDUbDTIYLsR+X5qE52+nXB6CtuihabBuc1qXw88KavqDX2o6NDNcscs+913fUAgVatdH0/wjo9z/AMI5o/GWl+y2xwZXPPG44ya26KWystA3d2cb4F0O4trnVNd1KwNjeapKHNu5BaIKNuCQSOcA03WPHGt6VdXSx+BNXvLa3yftMM0O11AySAWz+ldpRT7W2Wgd2+p4va/tHabc3UUE/hHXYI5GCtLJCCqA9yBzirkt98Hrq8N7Pbjz5DvY/ZroDP0AxXrlFHn1DyPPF8c/DhNWTU0n23iRCFZRZTghBn5cbcdzTtN8ffDzSHuG026Nu1zIZZitnP8AO5OSTla9Boo2A43/AIWx4N/6Ckn/AIBzf/EUf8LY8G/9BST/AMA5v/iK7KigDjf+FseDf+gpJ/4Bzf8AxFRXXxX8HPZzKuqSEmNgP9Dm9P8Acrt6hvP+PGf/AK5t/KgDyv4bfErwrpnw30azvdRkjnhg2uotZWwdx7hcV1H/AAtjwb/0FJP/AADm/wDiKd8Jv+SU6B/17H/0Jq7CgDjf+FseDf8AoKSf+Ac3/wARR/wtjwb/ANBST/wDm/8AiK7KigDjf+FseDf+gpJ/4Bzf/EUf8LY8G/8AQUk/8A5v/iK7KigDjf8AhbHg3/oKSf8AgHN/8RR/wtjwb/0FJP8AwDm/+IrsqKAKek6tZ65psd/pkpltpc7HKMmcHHRgDVyiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOJ8G/8AJQPHX/X7bf8AogV21cT4N/5KB46/6/bb/wBECu2oAKKKKACiiigAooooAKKKguL60tHjS7uoYGlbbGskgUufQZ6mgCeiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACobz/jxn/65t/KpqhvP+PGf/rm38qAOV+E3/JKdA/69j/6E1dhXH/Cb/klOgf9ex/9CauwoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4nwb/yUDx1/wBftt/6IFdtXE+Df+SgeOv+v22/9ECu2oAKKKD0oA86t/H3iPV/GeqaFoOh2Uq6Zc+XNPc3LIDHxyMKctz0q/4k8b6ha+I4fDnhfTYdR1do/NlSeUxxRJ6lgCf0rM+Hg/4uH41Pf7cf5LXNarotzF8b9Qa48S3vh9L62jNvcWwTEuAcqS6kDGR6daUdfZrur/O17Dlpzvt/Vzr7H4lS2T6jZ+M9OXTNQsYfP8u3cyRyx8DKscE8kDpWfe/EbxVo9pa6vrPh2yh0S4kVfMjumadAxwCyFQByfWuU8T2lj4fur3WLbxjf+IdesrUqkc8aOiKWBySiAenGazfiBp1pD4IstQ1XxZdeILyaWEwROU2QHeCxARRjjI+ai+il5peW/wDl8lqFteXvf12X9eZ6/wCKfHKaKlhbaVbfb9T1P/j0tycKRxyxHIAyOcGvMfG+u+Jbjxr4VsPFOjW1jN9uWWF7SdpY3AVhgkqMHJrb1SdNL8ceD/EN0xbS2s2gMw5SN2Chc/Wo/ihr+l33jjwfZWV1FcTi+WQ+WdwC4buPrVqNqkf8Vvuf+WpDd6cv8F/vX9I7Pw744OpWOtnUrZLS50eR0kiVychVBzz65rV8Ia9N4l8OQ6pNbpbickxqjFsp2P4ivJviL9p0Hx5Na6fDIyeJYIoTt+7vD5c/9817Ro2mx6Rotpp8IAS2iWMY9hilHWHM/JfNb/p945aS5V6/J7fr9xdooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooJA6nFJuHqPzoAWiiigAooooAKhvP+PGf/rm38qmqG8/48Z/+ubfyoA5X4Tf8kp0D/r2P/oTV2Fcf8Jv+SU6B/wBex/8AQmrsKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoJABJOAOpNFBAIIIyD1BoAxT4y8MAkHxFpQI6g3sf+NH/CZ+GP8AoYtK/wDA2P8AxqQ+E/DpJJ0DSyT1Js4/8KT/AIRLw5/0ANL/APAKP/CgBn/CZ+GP+hi0r/wNj/xo/wCEz8Mf9DFpX/gbH/jT/wDhEvDn/QA0v/wCj/wo/wCES8Of9ADS/wDwCj/woAZ/wmfhj/oYtK/8DY/8aP8AhM/DH/QxaV/4Gx/40/8A4RLw5/0ANL/8Ao/8KP8AhEvDn/QA0v8A8Ao/8KAGf8Jn4Y/6GLSv/A2P/Gj/AITPwx/0MWlf+Bsf+NP/AOES8Of9ADS//AKP/Cj/AIRLw5/0ANL/APAKP/CgDivCPirw/D468ayza5p0cc15bmJ2u0AcCAAkHPPNdl/wmfhj/oYtK/8AA2P/ABrkPCPhvQ5vHfjWKXRdPeOG8txGjWqEIDACQBjjmuy/4RLw5/0ANL/8Ao/8KAGf8Jn4Y/6GLSv/AANj/wAaP+Ez8Mf9DFpX/gbH/jT/APhEvDn/AEANL/8AAKP/AAo/4RLw5/0ANL/8Ao/8KAKFpr3gmxuri5s9X0SGe6bfPIl3EDI3qeeag1rUfAHiO2W313UdBv4VOVS4uYmAP4mtb/hEvDn/AEANL/8AAKP/AAo/4RLw5/0ANL/8Ao/8KPIDD0uX4b6LZXFnpN14es7a5/10UNxEqyfUZ5qpaWHwpsVnWz/4RmFbhdswSaIbx1wea6f/AIRLw5/0ANL/APAKP/Cj/hEvDn/QA0v/AMAo/wDCgBtpp/hzVvDgsLCGwvdHx5YihKvDgcY44qrZfD7wlpro9h4d0+3aOQSoY4ANrgYBHvW5Z2Npp1uINPtYbWEEkRwRhFye+BxU9HW/UOljjLjRNT17x1aXuqadHa6fpLM1s/nLIZ2Klc4HK9e9dnRRQtEkG7uFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB5R8XbaK/8AFfhKxutz2090VljDsocbl4OCKl8d/Dzwxongy/1PSLY6Ze2sfmQ3CXEhKsD6MxFV/jBZ/wBoeLfCFsbie28y6I823fY6/MvIPasf4leE7zw7Np+p3usa1rXhqKUf2hY3V0ZOP7x46fhUL4LLS8nr9xf2u9lt82d1Y+NZ9N+GWla/qljcXTSxp5/lYBTLY3HPatKXxxZHX9N0rT7eW/kv4vO8yEjbCmAQzZ9Qf0pdT1PQ4fh7LdkRtpTWuEVcbWBGAB+PFeefBtf7H8Q3djq8MsV3fQJNYtM+f9HAJWMem0ECtvirSXz/AOB+vomY/DSi/l+Wv6fNHtFFFFQWFQ3n/HjP/wBc2/lU1Q3n/HjP/wBc2/lQByvwm/5JToH/AF7H/wBCauwrj/hN/wAkp0D/AK9j/wChNXYUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcT4N/5KB46/6/bb/wBECu2rifBv/JQPHX/X7bf+iBXbUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFMmmjt4JJp3CRxqXdm6KAMk0+kIDKQwBBGCD3oA841fxn8Ldc1GwvtR8T6c8+nv5luy3W3acg8gdegq9qXxJ+G+rabPYX/AIo0uW3nXZIhnHIrr/7J07/oH2v/AH5X/Cj+ydO/6B9r/wB+V/wpWTVug7tO55gmqfCCPw/baIPE9obC2lEscTagx5BBAPPIyOnStHUPGPws1LU7C/uPE2nLc6ef3DxXWzaOODjqOBxXff2Tp3/QPtf+/K/4Uf2Tp3/QPtf+/K/4VV3e5NlaxzX/AAtrwD/0Nul/9/xR/wALa8A/9Dbpf/f8V0v9k6d/0D7X/vyv+FH9k6d/0D7X/vyv+FIZzX/C2vAP/Q26X/3/ABUN18WfAT2cyr4s0wkxsABOOeK6v+ydO/6B9r/35X/CobvSdOFjPiwtf9W3/LFfT6UAeb/DX4l+C9L+G+jWWoeJdPt7mGDbJFJMAynceCK6n/hbXgH/AKG3S/8Av+Kq/CrTbGX4WaE8tlbu7W5yzRKSfmb2rr/7J07/AKB9r/35X/CgDmv+FteAf+ht0v8A7/ij/hbXgH/obdL/AO/4rpf7J07/AKB9r/35X/Cj+ydO/wCgfa/9+V/woA5r/hbXgH/obdL/AO/4o/4W14B/6G3S/wDv+K6X+ydO/wCgfa/9+V/wo/snTv8AoH2v/flf8KAOa/4W14B/6G3S/wDv+KP+FteAf+ht0v8A7/iul/snTv8AoH2v/flf8KP7J07/AKB9r/35X/CgBNK1ew1zTY9Q0i7ivLSXOyaJsq2Dg4P1q5TIoo4IxHDGsaDoqKAB+Ap9ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcT4N/wCSgeOv+v22/wDRArtq4nwb/wAlA8df9ftt/wCiBXbUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABUN5/x4z/9c2/lU1Q3n/HjP/1zb+VAHK/Cb/klOgf9ex/9Cauwrj/hN/ySnQP+vY/+hNXYUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcT4N/5KB46/wCv22/9ECu2rifBv/JQPHX/AF+23/ogV21ABRRRQAUUZHrRketABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRTJJooQDNIke44G5gMmn0AFFFFABRRRQAUUUUAFFFFABUN5/wAeM/8A1zb+VTVDef8AHjP/ANc2/lQByvwm/wCSU6B/17H/ANCauwrj/hN/ySnQP+vY/wDoTV2FABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHE+Df8AkoHjr/r9tv8A0QK7auJ8G/8AJQPHX/X7bf8AogV21ABRRRQBwnjfwH4budI1bVptO3XxheTzhPIDuC9cBsfpWN8N/BHh6TwPpOty6W9xqKwrMJPtEm5nAB6Fsda7vxf/AMidqv8A16yf+gmsj4V/8kz0b/r3T/0EUoac9tLW/Uc/s38/0Ik+Jul/8Indaw9rcRyWsphaxJHnFwSAo7ZIGfpXW2F017p8NzJBJbtKgYxSY3J7HHFeE6rPZf8AC1B4itreQ+H7a+SO8KtiNp9uBLjHOACv4175G6vGrIcqwyCPSqjrBS7/ANfjv6WJl7s3Ht/X4beo6iiikMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAPIvi7feJIPFXhmLS7O1ktmum8vfdMnmvt5VgB0A5zzzXW6n4w1HQdY0e31vToIbTUAI5biOYt5UxwAnQZBJ6+1YfxXlSDX/B8szBI11B9zNwB8lWPitcw6tpFr4Ys0S4vtWkAiYHPlKOsmR6ZBqYtqkravmfz2/r5FNJz10937tWbnhjxXc+JtX1IW9ki6TayeVDd7zumYdeMYwDkde1dRXAfCW9WHw9L4euYUt7/AEiQxTRrn5skkNzzyMH8a7+tJJKyWq79/Mzi27308uxylz4wls/iRbeGri1jS3urdpYrnzDuZlAyu3Hv61meHfiaut+MdY0WayW3SwQvFPvJ85RgH6cnFZvxmmbQbfSvFcKKX06cRscHO1yM/wAq87103HhDQdI1y3icXOqwy27bepMspcH8hWcW7el16t6x/wAjSS1S72fyXxf5npUXxYf/AIRNdVn0xRcXlybfTreOQt9o6ck446npnpVm18d+ItL1iytfHOg2um298wjhuLO5aZRIeArZVcZJrifGnhWbR9E8EeVe3Gm2tlIVmu4FDPCWU4bBBHUgdK1tY8L6Zcz6bH4h+J2qagjXCSW9uyQtvcEEfcTIrRJKXzt936vczbuvVX/P8Fsewg5GRRTIEEdvGiksqoACe/FPpPca2Cobz/jxn/65t/KpqhvP+PGf/rm38qQzlfhN/wAkp0D/AK9j/wChNXYVx/wm/wCSU6B/17H/ANCauwoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACkZgilmOFAySe1LR160Ace3xZ8AoxVvFulhgcEG4HFJ/wALb8Af9DdpX/gQK6X+yNNPXT7X/vwv+FH9kab/ANA+1/78L/hQBzX/AAtvwB/0N2lf+BAo/wCFt+AP+hu0r/wIFdL/AGRpv/QPtf8Avwv+FH9kab/0D7X/AL8L/hQBzX/C2/AH/Q3aV/4ECj/hbfgD/obtK/8AAgV0v9kab/0D7X/vwv8AhR/ZGm/9A+1/78L/AIUAc1/wtvwB/wBDdpX/AIECj/hbfgD/AKG7Sv8AwIFdL/ZGm/8AQPtf+/C/4Uf2Rpv/AED7X/vwv+FAHlvhX4leDLTxt4wubnxLp0UF3dwPbyNMAJQIQCR64PFdb/wtvwB/0N2lf+BAqh4P0ywfx944V7K3ZUvLcKDEpC/uB04rtP7I03/oH2v/AH4X/CgDmv8AhbfgD/obtK/8CBR/wtvwB/0N2lf+BArpf7I03/oH2v8A34X/AAo/sjTf+gfa/wDfhf8ACgDk774nfDrUbGa0uvFmltDMhRwLkDIIweaqaf48+GWl+H00ay8V6dHZpF5SgXnzBcY+91z712/9kab/ANA+1/78L/hR/ZGm/wDQPtf+/C/4UrbruB51F4k+EcPhebw+viPSzp824yK13lmJOSd3XOe9a+nfEz4daXp0Fla+LtOMMC7E8y73tj3J5Ndd/ZGm/wDQPtf+/C/4Uf2Rpv8A0D7X/vwv+FVdgM0bXNL8RacL/Q76G/tGYoJoH3KSOozV+o4YIbaPy7eJIkznaihR+QqSkAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZ2s+H9J8RWq22uafb38KtuWOdNwB9abb+G9GtL2C8ttNt4ri3i8mKVUAZExjaD6YrToo2Dc57VrR9KvH1fQPDUOpanP8AJM4nSByvuzDnoOKzn8T+Mo42eTwIqooyzNrUIAHr0rsqR0WSNkkUMjDDKwyCPSgDi4PE/h7xPoj2/i46LbI7fNaS6pBcKw7HKnFW7y98BahbW1ve6hoU8NoQ0CPdRERkDAI544rS/wCEQ8Nf9C9pX/gFH/8AE0f8Ih4a/wChe0r/AMAo/wD4mjqBXu/EPhC/sntL3WtHnt5F2vE95EVYemN1Yuj6f8LvD+oC+0Q+G7G6VSomguIlYA9RnNdF/wAIh4a/6F7Sv/AKP/4mj/hEPDX/AEL2lf8AgFH/APE0LR3QbqzD/hL/AA1/0MOlf+Bsf/xVH/CX+Gv+hh0r/wADY/8A4qj/AIRDw1/0L2lf+AUf/wATR/wiHhr/AKF7Sv8AwCj/APiaAD/hL/DX/Qw6V/4Gx/8AxVQ3fi7w2bKcDxDpRJjbAF7H6f71Tf8ACIeGv+he0r/wCj/+JqG78I+GxZTkeHtKBEbYIso/T/doA5j4XeJ9AtfhhocF1rmmwypbkNHJdxqynceoJ4rrf+Ev8Nf9DDpX/gbH/wDFVyXwu8MaBdfDDQ57rQ9Nmle3JaSS0jZmO49SRzXW/wDCIeGv+he0r/wCj/8AiaAD/hL/AA1/0MOlf+Bsf/xVH/CX+Gv+hh0r/wADY/8A4qj/AIRDw1/0L2lf+AUf/wATR/wiHhr/AKF7Sv8AwCj/APiaAD/hL/DX/Qw6V/4Gx/8AxVH/AAl/hr/oYdK/8DY//iqP+EQ8Nf8AQvaV/wCAUf8A8TR/wiHhr/oXtK/8Ao//AImgA/4S/wANf9DDpX/gbH/8VR/wl/hr/oYdK/8AA2P/AOKo/wCEQ8Nf9C9pX/gFH/8AE0f8Ih4a/wChe0r/AMAo/wD4mgDStLy2v7ZbixuIrmB/uywuHVvoRxU1Q2lnbWFstvY28VtAn3YoUCKv0A4qagAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOJ8G/8AJQPHX/X7bf8AogV21cT4N/5KB46/6/bb/wBECu2oAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACobz/jxn/65t/KpqhvP+PGf/rm38qAOV+E3/JKdA/69j/6E1dhXH/Cb/klOgf9ex/9CauwoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4nwb/yUDx1/wBftt/6IFdtXE+Df+SgeOv+v22/9ECu2oAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACobz/jxn/wCubfyqaobz/jxn/wCubfyoA5X4Tf8AJKdA/wCvY/8AoTV2Fcf8Jv8AklOgf9ex/wDQmrsKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAOJ8G/8AJQPHX/X7bf8AogV21cT4N/5KB46/6/bb/wBECu2oAKKKKACiq0mpWMMhSW9t43XqrSqCPwzSJqmnyOFS+tmYnAAmUk/rQBaooooAKKOnWmRTRTKTDIkgBwSjA0APooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqjrWotpGi3V+lu1ybeMv5SttL+2avVHcwrcWssTgMrqVIPeplflfLuONrq5w+q/FSx0zwBY+KPsMk8d4QFt0kG5cnB5x261fvfH1vbzaBFa2T3Ta0A0YWQDy14yx45xuryDSLQ6trcngeWM+XoctwzAMTgNEQpP41sfCqV/EXjaETBinhm1NmMjgsRtJ/wDHa0jaTv0ev/bttvv0+ZMrxT7rT530/DU7qf4i3T+M77w1o3h2bULyy2tI32pYl2kA5ywxxkcV3CFmjUuuxiOVznFeceEwP+Fz+LTgZ/dDP/bNa9JqY/w4vq1ccvja7f5Ihu7uGxs5bq6cRwwoXdj2A6muAHxSvptNl1iz8J3VxokTEG9FyoYqDgsItu4jNbnxKs7q++HOswWIJma1fAU8n5TxXE+FNP1m9+HEE0Xj6C202ODZMv8AZcJWLHDAtnsam7959rf0/uKsvdXe5uX3xatl1yx0vRNIn1WfULUXNsY5QisCSCCSPlxg8mtfwr46HiDWL3R9Q0yTStUssGS2eUSjac4IYDB6V5v4A0iw0f4oafa6Xqy6xappSmO7UABvnfpgnFdVpigftBaxgAZsIf8A0Fq0ik3Fd1J/de35Gbbs/K342/zPSqKKKkoKhvP+PGf/AK5t/KpqhvP+PGf/AK5t/KgDlfhN/wAkp0D/AK9j/wChNXYVx/wm/wCSU6B/17H/ANCauwoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4nwb/wAlA8df9ftt/wCiBXbVxPg3/koHjr/r9tv/AEQK7agAooooA4Pxz8P/AAleaNq+q3Xh+xlv2geQ3LRAuW29c+tY3wz8A+FT4F0jWB4bsZtSWFZVmMYDlwAQd3rnvXeeL/8AkTtV/wCvWT/0E1kfCvj4ZaNn/n3T/wBBFKGinbTb9Rz15b+f6FRfijZDwzd6hJp8qXtrObY6dvBkaTJwAcdwM9K7SynlubGGaeA28kihmiLBtntkda8H1a5gk+J3/CZW1pu0GxvFhuW3kKzgY83b0O3BX8a99ikWWFJIzlWAII9KqOsFLv8A1+O/o0KXuzcf6/pbDLv/AI85f9w14f8ACnxXqtjo2q2ujeGptWW3vZmnl+0rCE+djgbh83HpXuF3/wAecv8AuGvK/glqunf8IvrMQljSW2vZjOC3T52OT+FRH4pv+7+pUvhj/i/Rna6H42sdf8KXGt2cbj7Mjma3fh0ZRkqc1FZeN4b34eSeK1snSKO3af7OXBYgDOM4ri/AURuPDvjXUbVf9BvZp3tmU5DjywMj8qq6RrFjb/s0XUk86xqLSW3+Y4/ebcbfrmipJqEpLTSL9L7hBXmo7+819x2t78RbWz8Madqa2MstzqbBLSyV/mkcgnbuxgdDzTIvHWr2+pQ2mveEbuwFwp8mWKcXKlgCcNsHy9Oprn49G0HxB8MPD1jrOp/2bdPtawuFk2OsuDjbgjPGeKluNY8VeDfGWi6Zqmsw65p+pExAG2SGSMhSc/LnI6d62lG1Vw82l/XcyUr01Lyu/wDP0Om0jx5Z6n4X1HWZbdrUadJLHPA7gsDH1598VsaBq/8AbuiW+o/Z2thOCyxs24gZ4OffrXivi9ZNJ8dap4WtVk8rxFLAyADAAJPmn/x4V7np1mmn6ZbWkYAWGJUGPYYqI+9Dn72+/r+JctJ8vr9z2/As0UUUgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMq28MaNZ65daxbWEceoXi7Z5wTlx784pdI8N6RoM11LpFjHavdyGSdkJ+dic5OT71qUULTYHruZsejafp+oXmq2dkPt1yAZnQndLgYA5OOgFctJ8QdejZv+Lc6+VXPzeZb4x6/6yu7pk0MdxBJDOgeORSjqw4YEYIpDOK0X4iS6vqS2l94ZvdMhcHdcXVzblB7Ha5P6UXXgX4e3l1JcXFrZNJI259t6yqx/3Q+P0qx/wqXwD/0KWl/9+BR/wqXwD/0KWl/9+BT03FqaMFh4WttUj1G3NhHdxxCFJFmAwgzgYzjuakjh8ORa5LrEctmt/MgjefzxllHQdcd6yv8AhUvgH/oUtL/78Cj/AIVL4B/6FLS/+/AoCx0v9rad/wBBC1/7/L/jR/a2nf8AQQtf+/y/41zX/CpfAP8A0KWl/wDfgUf8Kl8A/wDQpaX/AN+BQB0v9rad/wBBC1/7/L/jUN3q2nGxnxf2v+rb/lsvp9awP+FS+Af+hS0v/vwKhuvhN4CSzmZfCemAiNiCIBxxQBH8KtTsYvhZoSS3tujrbnKtKoI+Zveuv/tbTv8AoIWv/f5f8a8x+Gvw08F6p8N9GvdQ8NafcXM0G6SWSEFmO48k11P/AAqXwD/0KWl/9+BQB0v9rad/0ELX/v8AL/jR/a2nf9BC1/7/AC/41zX/AAqXwD/0KWl/9+BR/wAKl8A/9Clpf/fgUAdL/a2nf9BC1/7/AC/40f2tp3/QQtf+/wAv+Nc1/wAKl8A/9Clpf/fgUf8ACpfAP/QpaX/34FAHS/2tp3/QQtf+/wAv+NH9rad/0ELX/v8AL/jXNf8ACpfAP/QpaX/34FH/AAqXwD/0KWl/9+BQB1sU0c8YkhkWRD0ZGBB/EU+qelaRYaHpsen6RaRWdpFnZDEuFXJycD61coAKKKKACiiigAooooAKKKKACiiigAooooAKOnWikZQ6lWGQRgg96AKv9racOt/a/wDf5f8AGj+1tO/6CFr/AN/l/wAa5tvhN4CdizeE9MJJySYBzSf8Kl8A/wDQpaX/AN+BQB0v9rad/wBBC1/7/L/jR/a2nf8AQQtf+/y/41zX/CpfAP8A0KWl/wDfgUf8Kl8A/wDQpaX/AN+BQB0v9rad/wBBC1/7/L/jR/a2nf8AQQtf+/y/41zX/CpfAP8A0KWl/wDfgUf8Kl8A/wDQpaX/AN+BQB0v9rad/wBBC1/7/L/jR/a2nf8AQQtf+/y/41zX/CpfAP8A0KWl/wDfgUf8Kl8A/wDQpaX/AN+BQBn+D9SsU8feOGe9t1V7y3KkyqA37gdOa7X+1tO/6CFr/wB/l/xryzwr8NfBl3428YW1z4b0+WC0u4EgjaEERgwgkD0yea67/hUvgH/oUtL/AO/AoA6X+1tO/wCgha/9/l/xo/tbTv8AoIWv/f5f8a5r/hUvgH/oUtL/AO/Ao/4VL4B/6FLS/wDvwKAN+6vNHvbWS2ury0khlUq6GdfmB6jrVe0Tw9Y6OulWk9nFZLH5YiW4GAuMYznNZH/CpfAP/QpaX/34FH/CpfAP/QpaX/34FAF6PS/CcWgyaLH9hGnSZ3weeMNk5POc9ayNT8ZxeFWg0zRPDl3qlpFGAklpdQ7V5+787g1Z/wCFS+Af+hS0v/vwKP8AhUvgH/oUtL/78CjUNDOj+I2s3cJMXw712aNuCyTW5B/ESVbt/hj4OvbWOefw59lkky7xGZ1YMxyd21sE5NdPo2h6Z4e04WGh2MNjaKxcQwrtUE9Tir9GgalezsLWwsks7OBIbdF2rGo4ArnJPhj4PlubieTRYi9xuMo819rbuvy7sfpXV0UPV3YLRWRkXPhTRLzQ00e50+N7BBtSHJG0exByPzqpo/gPw34fvDeaTpaxXG3AdpXkI+m5jiuioou736h0scJb6ZeeIviNHrGo6NNp9vpKPFA9wVJuC3G5dpPHy98da7uiijZJA9W2FFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFQ3n/HjP/wBc2/lU1Q3n/HjP/wBc2/lQByvwm/5JToH/AF7H/wBCauwrj/hN/wAkp0D/AK9j/wChNXYUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcT4N/5KB46/6/bb/wBECu2rifBv/JQPHX/X7bf+iBXbUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABUN5/x4z/APXNv5VNUN5/x4z/APXNv5UAcr8Jv+SU6B/17H/0Jq7CvkX4Q/GmfwVrT6Lr8jzaHNOwVictaknqM/w9OOAOTX1paXcF/ZxXVnKk1vMgeORDlXU8gg0ATUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRXgHxx+OK6THP4Y8I3Ae+YGO8u0P+o7FVP97rk8EYoA9E8C3EN1478dS2s0c0f263G+NgwyIACMj3ru68B/ZQdpPDPiJ5GLu18hZmOSSU6179QAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFQ3n/HjP/1zb+VTVDef8eM//XNv5UAfnRef8f0//XRv51678FfjTP4LvI9F8QStNoczgK7HJtSf4h/s+vXpxXkV5/x/T/8AXRv51DQB+jtpdwX1pFc2cqzQSqGSRDkMDU1fH3wW+NU/gq6j0TxBI82hSthDyTaknqP9nrkepr68tbqC+tI7mzmSaCVQySRsGVh7EUAS0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAU2SRIoy8rqiKMlmOAPxpJZo4IXlnkWKNBuZ3YBVHqSelfKPxv+N8niaabw54WmaLSUJS5mGQbkjsP9n+eM0Adl8cfjiukxz+GPCNwr3rjZdXkZyIR3Vf9r17YNfLju8kjPIzO7HJZjkk0mc9aKAPqT9k3/kVfEH/AF+x/wDoFe/14B+yb/yKviD/AK/Y/wD0Cvf6ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqG8/48Z/+ubfyqakdQ6MjDKsMEeooA/OO8/4/p/8Aro386hr7lf4HfDp3Z28M25ZjknzZOT/31Tf+FGfDj/oWLf8A7+yf/FUAfDlex/BX40z+C7uPRPEEjzaHM2FcnLWpPcf7PTjgDk19Af8ACjPhx/0LFv8A9/ZP/iqP+FGfDj/oWLf/AL+yf/FUAd1aXcF9ZxXVnKk1vMgeORDlXU9CDU1cWnwj8FxRqkekOiKMKq3cwAH/AH3Tv+FT+Df+gXJ/4GTf/F0AdlRXG/8ACp/Bv/QLk/8AAyb/AOLo/wCFT+Df+gXJ/wCBk3/xdAHZUVxv/Cp/Bv8A0C5P/Ayb/wCLo/4VP4N/6Bcn/gZN/wDF0AdlRXG/8Kn8G/8AQLk/8DJv/i6P+FT+Df8AoFyf+Bk3/wAXQB2VFcb/AMKn8G/9AuT/AMDJv/i6P+FT+Df+gXJ/4GTf/F0AdlRXG/8ACp/Bv/QLk/8AAyb/AOLo/wCFT+Df+gXJ/wCBk3/xdAHZUVxv/Cp/Bv8A0C5P/Ayb/wCLo/4VP4N/6Bcn/gZN/wDF0AdlRXG/8Kn8G/8AQLk/8DJv/i6P+FT+Df8AoFyf+Bk3/wAXQB2VFcb/AMKn8G/9AuT/AMDJv/i6P+FT+Df+gXJ/4GTf/F0AdlRXG/8ACp/Bv/QLk/8AAyb/AOLo/wCFT+Df+gXJ/wCBk3/xdAHZUVxv/Cp/Bv8A0C5P/Ayb/wCLo/4VP4N/6Bcn/gZN/wDF0AdlTJpY4IXmmcJHGpZ2Y4CgDJNch/wqfwb/ANAuT/wMm/8Ai6RvhL4MZSraU5BGCDeTc/8Aj9AHgPxv+N8niaabw74VnKaQhxPcocG5I7Aj+H8ec14bX3H/AMKM+HH/AELFv/39k/8AiqP+FGfDj/oWLf8A7+yf/FUAfDlFfcf/AAoz4cf9Cxb/APf2T/4qj/hRnw4/6Fi3/wC/sn/xVAHAfsm/8ir4g/6/Y/8A0Cvf6w/DHg3QPBttPb+GtOSwiuHDyqjMdzAYB5JrcoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAorjvip4j1Hwp8O9Q1fRpEju4AuxnQMBlgOh+tYPwN8ca3468Kz33iGaKWdJSgMcQQYz6CiHvuSXT+v1CXupN9dD0+iimTuUt5HXqqkj8qTdlcaV3YfRXgfw9+LPirxF8aLvw5qdzA+nRNOFRLdVb5Tx8w5r3yqt7ql31JbtJx7BRRXzT8UPjx4o0jx1eaX4PnhW0tBtk326yHcPvcmockmkWotpvsfS1Fcd8LvGZ8ceB7XVJmU3XKzhQAA2T2/CuxrSUXGVmZxkpK6CisfxdqNxpHg/VdQsmVbi2tZJYyy5AYKSOK+YtH+NHxi8RSSroEK6h5Rw/2fTlfb9cCs1K8nFdC2rR5j61or5f8A+E+/aB/6Adz/AOCkf4VWv/iv8btCtxfa5pclvZxsN7zaaqKfbOOKrTqLXofVNFeVfCf42WfxDkk0++tlsNTjUFYw+5ZB3IPHOe1df8Qr/X9N8FXt14SgafVU2+TGsXmE/MM/L34zRO8FdhC03ZHTUV8v/wDCfftA/wDQDuf/AAUj/CsbWPjb8X/D1zHb65GlhNKMok+nKpYe2RQB9c0V8uxfEP4+zwrLDotw8bjKsulLgj8q7f4WeKfitq/i4W/jnTJrbTfKYl3sBEN3b5sVSi27EuSSue10UUVJQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHnXx4/wCSPav9E/8AQxXJ/su/8iFdf9fB/ma6z48f8ke1f6J/6GK5P9l3/kQrr/r4P8zRh/iq+i/QK3wU/V/ke41Fdf8AHnN/1zb+VS1Fdf8AHnN/1zb+VTL4WVH4kfKfwh/5OV1D/fuv519Y18nfCH/k5XUP9+6/nX1jWi/g0/Qzl/HmYPjXXY/Dfg7UdSlcJ5ULbST/ABEYH6mvmz4O+C5fHtt4o1bUstJcxOkbkdZHDZP51137T3ikpYWHhi0J825cSSgHqvYfmK9J+D3hePwt8N9Otwu2aeMTy567mAOKzpx51Um/8K/U0qSceSC9X8tjxv8AZ08RyaB4z1LwjfOwSSRvLDf31O3/ABr6er5J+Lmny/Dr41WmvaaGjjuHFwuOmRgH9c19T6FqsGt6FaalasGiuIg6kHNaRl7SjGfVaP8Ar+tjOUVTquK2eqMv4g/8k713/rxl/wDQTXh/7Kv/AB863/vD+Qr3D4g/8k713/rxl/8AQTXyf8I/ixafDOfUDd6XNf8A2puPKlCbfzFRRklVnft/mXUTdKNu/wDkfadRXVrBe27QXcKTRMMMjrkGvAf+Gs9J/wChWvP/AAKT/CsvX/2pLnU9NNp4U0Ga1v5WCo8ziXr2CqAc+lEttrgt97HFJAnhb9pCO30VtkSXwwFPHIyR+Zr7KQ5jUnqQK+Yfgp8Lda1jxcvjDxXbyQxJIZo1mG13kz3U8gda+oAMDArRJwowhLdENqdWU47MK+W/2n/+R60P/rif/QxX1JXy3+0//wAj1of/AFxP/oYrF/xafr+jN4fDL0PpLw7/AMi5Yf8AXFa0qzfDv/IuWH/XFa0q3qfG/U5qX8OPogooorM0CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4D43W5ufhHrCLnO1TwPRga4f9lm4D+Db+AYzHNk8+pNezeINJj13w/eabMMrcRMn0OOP1r5O8I+JtU+A/j2/sNasZ5LCU4dQOXUZ2svbvSotRqTi/tL8R1E5U4tfZf5o+wqhvGCWFw7cBYmJ/KuEs/jl8Pbmzjml8R21u7qCYpA25PY4FcL8Tf2hdFj0q40jwe51G5uYtguox+7XPBHPOce1TUuk0txws2mcZ8F0F3+0RqVzESUDXJBA6gmvquaVYIXlkICopJJrwr9nDwBeaPYXHiXWI2iuL0YhRx8wXnLH65Fdt8bPFf/CK/De9ljYLcXQ8iLnkE85/SrrP2VKMeqVvmTTXtaspdG/wPmbxjqWt+Pfi3f3/AIbs5tTe3nzbxxR+btVTkcemc13KePPj9HGqR6DcKqjAA0kcD8q3/wBl3w0YtM1LxFcJ89y/koSOwwcj86+g6rk9lFQv/wAOLn9pJzsfGPj+4+K3i3TFl8Y+H7lbaz+fzhp4i2D3YDpzXr/7NHi4ar4Rl0G4cfaNOPyKTz5fAH65r2XVLCLVNKubK4XdHPGUYGvkDwVfS/C746yafcyFLf7QbeXPRgfu/qaVF/vHS/mX4/1YKqbp+0W8fy/q59TfEH/kneu/9eMv/oJrwH9mbRNL1e41n+1dOtb3Yw2/aIVk29OmRXvvj9g/w41xlOQbCUgjv8prxD9lX/j51v8A3h/IUqH8afp/mOq/3Mbfzf5HtmqfDjwlqumzWU2gWESSrgvBbojj6EDIr5g8U+HtS+BfxDiv9OtYb3T3bfbvcwhxjPKgnOCOOetfY1ct8Q/BVp468JXOl3Kr5pUtBIR9xx0/Wok5QftI7r8UXHlkuSWxL4F8Z2HjjwxBqmnuuSAJoweY29DXSV8c/CvxXf8Awr+Jcmi60WisppfJuFfovPDj8B+tfYcUqTQrLEwZHGVI7it5crSnHZmMbxbhLdD6+W/2n/8AketD/wCuJ/8AQxX1JXy3+0//AMj1of8A1xP/AKGKwf8AFp+v6M6IfDL0PpLw7/yLlh/1xWtKs3w7/wAi5Yf9cVrSrep8b9Tmpfw4+iCiiiszQKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArF8R+D9B8WWwh8QaZBehAQhkXJTPcUUUmk9xptbHmsn7MHgV5GbzdUXcSdq3C4HsPlrd8M/AnwP4YnM0WntfyhgyPekSGMjuvAxRRVJtbEtJ7nowAUYUYA7CuP8AHnwy0X4iLbLrs14iW/3Et5QoJ9TkGiipaT3KTa2Nvwz4b0/wnoFvpGkIy21uu1S5yze5PrWtRRVNtu7JSSVkFed+LPgl4W8Y+Ihreom8gvOCTbShASDnJ4PNFFTZXT7FXdrHYT6BbXPhd9Cnlme2ktzbs5b94VxjOcdawPAnwu0L4ePctoT3bG5OX+0SBvywBRRVXak5dWTZcqj0R2dFFFIZ554z+CnhbxzrY1XVftcN0FwWtpQgb3OQeeK7LQNFh8PaJb6ZbTzzxW67Ee4fc+Pc4oooj7q5VsD953e5o1w3jj4SeHvH+qWt/rcl4s1qu1BBKFGM55yDRRSsrp9h3aOzs7WOys4raHOyJQq7jzipqKKptt3ZKSSsgooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD/9k=)

|  |  |
| --- | --- |
| **Control Signals** | **Description** |
| Clk\_50m | Clock signal for the UART module |
| ready\_clr | Receiving process starting logic for Receiver |
| Rx | Receiver pin in FPGA |
| tx\_we | Data is assigned from bus to data\_in of uart\_tx if high |
| wr\_en  Tx\_busy | If *not(Tx\_busy) & wr\_en* : enabling the writing process of uart\_tx |
| Tx | Transmitter pin in FPGA |
| ready | This flag is high before receiving process. When it has started the flag becomes low. |
| bus\_to\_uart\_tx | Carries the data which is sent to uart\_tx |
| uart\_rx\_to\_bus | Serial to parallel converted data of uart\_rx to is added to this bus |
| uart\_tx\_to\_bus |  |

![A picture containing object

Description generated with high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RDiRXhpZgAATU0AKgAAAAgABAE7AAIAAAAIAAAISodpAAQAAAABAAAIUpydAAEAAAAQAAAQyuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFRoaWxpbmEAAAWQAwACAAAAFAAAEKCQBAACAAAAFAAAELSSkQACAAAAAzM2AACSkgACAAAAAzM2AADqHAAHAAAIDAAACJQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE5OjA1OjA3IDIyOjAwOjAzADIwMTk6MDU6MDcgMjI6MDA6MDMAAABUAGgAaQBsAGkAbgBhAAAA/+ELGmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTktMDUtMDdUMjI6MDA6MDMuMzU2PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlRoaWxpbmE8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgAtgLWAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKbIcRsR1ANADqKw/ttx/z1aj7bcf89Wrk+tw7G3sWblFYf224/56tR9tuP8Anq1H1uHYPYs3KKw/ttx/z1aj7bcf89Wo+tw7B7Fm5RWH9tuP+erUfbbj/nq1H1uHYPYs3KKw/ttx/wA9Wo+23H/PVqPrcOwexZuUVh/bbj/nq1H224/56tR9bh2D2LNyisP7bcf89Wo+23H/AD1aj63DsHsWblFVdPleW3LSMWO4jJq1XTGXNFNGTVnYKKKKoQUV594+8ea/oviXTfDPgrw9/bWs3kLXT+c2yGGBSVyWyMHcOpIHQclgKraH8SdYs/EmsaF8RNIh0ufT9PfU0u7Jnkt5bZMb2BPzEjPYZ6jHqAelUV4WfjJ8QU8JDxo/gzT/APhGDIGMgvP33k79u7G7rk46decYrc1P4qeINZ8Xtovwv0ay1uO3sY7y4ubuZoQBIm+MANtPKlfXlhnGDgA9YorzTwv8YLO48Ha3qfjSKLR7/wAP3b2mo2sTFwJASFCHoxJBXAJ5UnODVGLxh8YNQsZdR07wFpsNsxV7W2vL3ZcSRtyCRuAUgY3Btp56UAes0VyvgDx9pnj/AEJruxPk3ts3lX1i7AvbSeh9VODhhwcHuCB1VABRRRQAUVXv5GisZHjO1hjBH1rE/tC6/wCezVSi2B0dFc+l3fSZ8t5Gx12rmnefqP8A02/74/8ArU+UDeorB8/Uf+m3/fH/ANajz9R/6bf98f8A1qOUDeorB8/Uf+m3/fH/ANajz9R/6bf98f8A1qOUDeorB8/Uf+m3/fH/ANajz9R/6bf98f8A1qOUDeorB8/Uf+m3/fH/ANajz9R/6bf98f8A1qOUDeornft92OszVLaXty93ErysVLAEUcjA3aKKKgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiqGp69pGimMazqtjp5lz5f2q4SLfjGcbiM4yPzqh/wnXhH/oadF/8ABjF/8VQBh+O5dam8UaJpui6pc2CTWd5cTfZwuXMb26rncp6ea351j/2R4u/6GjVv/IX/AMRR4t+IHh+y8caHeWevabOi6dfRO8V5G4UtJaEAkHjOw4+h9KZ/wt3Rv+gzY/8AgQn+NAGh4Z/4SGw8dWdpqutXt9a3NlcOYrgJgMjRYIwoPRz+dei15NpnxG0LUviBpktzrmnRww2N2DJJdRqoLNBgZJxk7T+Rru/+E68I/wDQ06L/AODGL/4qgDeorB/4Trwj/wBDTov/AIMYv/iqP+E68I/9DTov/gxi/wDiqAOFhi8V6vcahdQ+I9RhiGpXkMUcXl7USO5kjUDKE8BBU39keLv+ho1b/wAhf/EVmaP8TtG0yG+tRrNjtGq37qftKYZWu5WBHPIIYEH0NaH/AAt3Rv8AoM2P/gQn+NAHXfD+51C48MONXu5by5hvbmAzSgbiElZRnAA6Adq6evO/AXjrw0fD9zJeeItKgkl1G7k2S3saNhpmIOC3Qium/wCE68I/9DTov/gxi/8AiqAN6isWDxn4XuriO3tfEmkTTSuEjjjvomZ2JwAAGySTxitqgAooooAKKKKACiivIviP8T/E1n44g8F/DjR1v9ZWMXF1JOm5EjK5A6jHUZYnrgdTQB67RXjfgT4neMT8Sk8FfEjR4LO9u4pJLSe2QhX2KW9SCpVXOc9QBivZKACiiigApsn+qf8A3TTqbJ/qn/3TSewHPUUUV4Z3hRRRQAUUUUAFFYvis+IhoLnweto2p712i8z5e3PzdO+K8n1nxl8XNC17R9Hv4vD/ANq1iRo7XYjFcrjO47uPvCtI03LZkuVj3KivP9CPxTkN+PEP9hwj7IxtGt1Zv3+RtDDP3cbv0q34P8fxar4dv5vEIj07UtH3LqcDHAjx/GM/wnHFDg+mocx2tFeb6X408R6j4I8S+LJLeC1sYLWebSYXiO9wiEh3OeQSOnFYOgeIPjH4k0G01fTYfDptbtN8fmBlbGSORn2p+zfcXMj2aivP/DPj3VW8VDwv420yLTNUljMltJDJuiuAOoU+tegVEouL1KTua2l/8eh/3z/IVdqlpf8Ax6H/AHz/ACFXa9aj/DRyT+JhRRRWpBz/AIw8caF4G0tb3xBeCHzSy28CKWkuHAztRR17DPQEjJGRXm954c1nUfC3jPx94sV7PUL3w7eW1npgYkWduYmbDf7ZIBPTFdl4y+FWheN9fstZ1S51K2vrGMJby2Nz5RTDFgwOCQwJ6g1c8N+AoPDepPeJ4h8RakXiMRh1PU3uIgCQchT/ABcdfc0AcbdatpH/AAya959lH2JtA8hY/JXiYr5Qbb0/1vzZ69+tZfwEurQeIdcszF/pv9maVKJNg/1f2KIY3devataX9njwBbyLJcSailn9qWUWcl9iAuzABcYzzkL1ycgZzXQ+MfhF4b8b67FrGpvqFtfRwG386yufKLIcjB4PZmHHUE5oA8c8Y3Vjer8TLy00557C18R6ebu3SPbvEQKTk46BnVvm77snk19KWFzb3mm21zZOklvNErxPGwZWUjIII6jHesLwd4M8PeEPDU3h7Q0E1msr/aknkErO7qCQ/blCvGBxj1yePl/Z38GOt5Db3Os2lneSiSWyt77bCSCSo27TkLk4zkigCP4TTW2ofE74l6npVt5enTX9vFHMh3xzSosglZX6HJYNgdN49RXrNUNE0XT/AA7ottpOjWyWtlapsiiQcDuSfUk5JPckmr9ABRVTU9VsdF09r7VrqK0tUdEeaZtqqXYIuT2BZgMnjmrMUsc8KSwuskcihkdDkMDyCD3FAFfUv+QdL9B/MVj2NuJ5vn+6oyR61sal/wAg6X6D+YrO00gLIe+RWkfhA0lIVcKAAOwpd1Rbq8L8eaMfin8SNa0yN5Da+GdLIi2Nj/THG4foAPqDUCPed1G6uN+GPiQ+J/hzpWoTNm5WLyLn2ljO1v1Gfoa8/wDjlrHiy48D64ILVdL0O2dIZJZH3TX26QIQoH3U5zzyaAPct1G6uM1LxCfC/wAITrSqGez0pJEU9C3lgKPzIryT/hGfCV/Fp9945+IF5D4tvoRPG63u1bZnXcABj5BgjjIoA+jt1G6sDwja6rp/heztde1GPU7yNMNdxAgSr2PPU471tbqLAS7qN1RbqN1FgIr23SeJmAAcDIPrWbZf8f0P++K191ZNr/yEY/8Arp/WtI7DOjooorIAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK8S/aolkh+GemNFIyE6vGMq2P+WM1AHttFeQ/EH4k634E8H+FbTw1pK6lqesWe2JnDP5ZjSIk7F5bIc9xjGeelcl+zBc6hceJPGZ1eSRrvdCZg/aTfLu4HA5z0oA+i6K8S8b/ALPo8TeKtU8RzeMptNiuW854vsuVhAUA/N5g44znArz/AOAfhGXV/ilNrWl6rc3GiaHI2ye4iKNdMysoG3cdvUtyTxj8AD6tooooA+bP2tG23XhT/cu/5w185M26vpv9p/WdR0TUvC8+k39zYyvFdoz28rRlhmE4JHbivCf+E/8AFf8A0Murf+Bkn+NAxso/4trpX/YXvv8A0TaVz0nWtTVPEmra2sS6xql5frCWMQuZ2k2bsZxk8Z2jP0FZbnPSgDotTH/FGeHP925/9Gmudk+9XRan/wAiZ4c/3bn/ANGmudl+9QB6D4zAHwt+HZA5NleZ/wDAlq89k+8a9D8Z/wDJLPh3/wBeV5/6UtXnkn3jQI6bxCP+KZ8H/wDYHk/9L7uuZfqa6fxD/wAiz4O/7A8n/pfd1zD9TQB0niwf6Vpn/YJs/wD0Qtc0/wB410viz/j60z/sE2X/AKISuaf7xoA9B+1R6f8AGnRLuVHaO3vtNlZY1yxCiEkAdzxX174d8bab4mv57OxhvIZ4IllZbqDy8qSQCOeeRXyIqo/xs0JZeI2v9MDfTEOa+qdCgtIPiJciybKnSkLcd/NagDtKKKKACiiigArM1/xBpHhfSpNU169hsbVCFMshxknoo7k+w9PatOvDfix4H8e+KfiRY6lpej6Zq2i6XGv2az1C4IhlcjLNJHvGTuI9AQoByMggDvh/Y6n8Rvi/J8TL6Oe20Syje30aOUlTICpQttOflwzk4xkkehr3CvMvBl38WV1+ws/E/h/w5p+gIrJK2nsQ8SiM7Ag8wjG4KOnTNem0AFFFFABTZP8AVP8A7pp1Nk/1T/7ppPYDnqKKK8M7wooooAKKKKACvKfib/yV34b/APX3P/7Sr1auC8beFNV1v4heDNW0+JHtNJuJXu2aQKVDbMYB6/dPStabSlr5/kTLY72vn74qva3/AI+uptDt5prTT7aP/hJ2tpNqzReapVD6sACT7D2Ne0+KX1pfDl0PDEEc2puuyHzHCqhPG459OuKzvBfgu28MeFzp9ztu7q7zJqE7jJuJG+9n27U6clD3hSV9CDxRc2F58Gdan0cxmxfRZzB5f3QnktgCo/hD/wAkj8P/APXsf/Q2rBt/BPiHRdH8V+GNOSO60HULGf8AsstKFa2ldCPKIJ+7k8HtWR4ct/jL4Y8O2ejWGh+H5LazTZG00pLkZJ5IkA7+lVypxaT6iu73sbXxS80eO/h+bLH2r+1cH18vjf8A+O7q9Prz7w14I1u48Wr4q8e3ttd6jBGY7O2tFIhtgepGec16DWc2rJLoVHqzFuviXpXh6+uNMu9N1yeWFhuktNMlmjOVB4ZRg9fzqP8A4XLoX/QG8Tf+CWb/AArtNL/49D/vn+Qq7XqUf4aOWfxM8+/4XLoX/QG8Tf8Aglm/wo/4XLoX/QG8Tf8Aglm/wr0GitSDz7/hcuhf9AbxN/4JZv8ACnRfGTw69zBFcWOuWSTTJD9ovNMkhhjLsFBd2ACjJHJrv6zfEeg2fifw3f6Lqabra+haJ8AEoT0dcggMpwwODggGgDhPiPZS/EHXofh9aXctpai3N9qtxCTuQYIgTGecv8xB4ITqDg1Y0z4jS6f8J9S1nxHHu1nw8j2upQDjfcodobgDCyEqwIGMMccCtX4c+Db7wlo07a/qK6prl9IJL29G795tUIg564UDnAJJJOTzWbr3wxbWPiRaa5HfCHSHaO41LTscXU8P+pfgDkZ5JPIUDBFAHM+Hf7Z+El8J/Ei6nrNr4iiN7fyWVs0/2bUSxaTCJwEYMBkDkpngYFdP/wALl0L/AKA3ib/wSzf4V6DRQB59/wALl0L/AKA3ib/wSzf4Uf8AC5dC/wCgN4m/8Es3+Feg0UAeC/Gb4l6V4g+Ems6ZaaZrkE0/kbZLvTJYYxtnjY5dhgcD88CvCfBHxX8WeAZAuiaiZLLJLWF1mSA8HkLnKnJzlSMkDORxX1z8YtA1HxT8KdW0bRYPtF9dvbrFHkDOLiMkknoAAST6CvNfAn7L2m2Xl3vjy6OozYz/AGfbOUhGQeHcYZsZB+UqMjqwOKAOt+H3xjtPiVptxbjSbuxvYU3SHaXg4I4EmMZ5HBwetdtYHCP9atNpljpGgtZ6VZ29lbRgBIbeMRovPYDiqVocK31rSPwiYa1q0OiaHe6ndECK0haVvfAzivB/AFr8WrLSbnWfDum6HLD4gnOovLesfMffyOAwwO4HvXtPinw7aeLfD0+jajNcQ2txjzDbsFZgDnGSDwcVp28UVraxW8ChIokCIo7ADAFKwrnkXwel1rwv441/wp4qhtrW6v8AGrwRWxzHliVfac9OBx22mt79oBs/BbV/+ulv/wCjkrqNT8K6fqninSvEEslxFf6WHWJoWULIr9VcEHI+mOppfF/hex8aeGbjQtVluIrW4KF3tmVXG1gwwSCOoHai2gXOV+ImnS6t+zvd2tvkuumW8wCjJIj2SEfktQaH4K+H938Kbe7udK0yeKTTg8+oTRq0obZ8x80/MCDnvxivRLa0htdNhsVy8MUSwgSYO5QMc/hXnt38CPBN3rBvTBexRMSXsYborAxJznHUfQED2osFy38Crq+ufhFpbakSSpdIWZiSYw3y/wBR+FeibqqWltb2FnFa2UKQW8KBI441wqqOgAqXdTsIm3Ubqh3UbqAJt1Z9p/yEIv8AroKtbqq2n/IQi/3xTRSOjooorIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcL8V/iT/AMKx8O2mqjSv7U+03Qt/K+0+TtyjNuztbP3cYx3ryD4/fEPQPGPwr8ODS7tDeXl0l6bQMGeFBHIjByOAQzY98H0r6C8UeF9K8YaBcaPrtsJ7WYfRkbsynsR615rof7M/gvRdattRkuNS1EW77xbXckZic9twVATg84zQB6Z4eijbw1o8jxqzx2UWxiuSuYxnB7V4l+zr/wAlD+In/X2v/o2avoFVCKFUBVAwABwBXHeCfhlpPgTWta1PS7u9nm1mQSTrcshVCGZvl2qO7nrntQByn7SPiyXw98Mjp1nN5VzrUv2YkBs+SBmTBHAz8qnPUMwxXQ/Bbw1F4Z+E+jQqkYmvIBeTuhJ3tINw6+ilRjpkGrHxG+GGkfEy1sINau722Wxd3jNoyAsWABzuU+ldfa20NlZw2tpEsMEEaxxRoMBFUYAHsAKAJaKKKAPnv9p59JTUvC51yzvbuHybvYlneJbsGzDyWaKQEYzxgfWvCvtHgj/oA+IP/B7D/wDIle5ftRaVPq2o+FobWW0iZIrtibq7jt1IzCODIwBPPTrXg/8Awhep/wDP9on/AIO7T/45QBN9o8Ef9AHxB/4PYf8A5EpGuvA69dB8Qf8Ag8h/+RKi/wCEL1P/AJ/tE/8AB3af/HKP+EL1P/n+0T/wd2n/AMcoAbrmtaZfWWnWWjWNzZ21kjgC6u1uHcu24ncscYH0x+NYUh5re/4QrUv+f3RP/B1af/HKP+EJ1H/n90T/AMHdp/8AHKBnSeM/+SWfDv8A68rz/wBKWrzyT7xrrvFWlW2leGfDtvDrn267MU73drFeR3EVo5k4CFMgblAJ5PNcg+Q3Jz+FAjqPEP8AyLPg7/sDyf8Apfd1zD9TXT+If+RZ8Hf9geT/ANL7uuYfqaAOk8Wf8fWmf9gmy/8ARCVzT/eNdL4s/wCPrTP+wTZf+iErmn+8aAO/ltpbz4yaPa20ywTT3mnRxysm8RsVhAYrkZAJzjIz6ivrHwj4L1Lw/rV1qOq63BqTz26wIsNi1uEAYtk5lfPX2r5Xs+Pjx4c4z/xMtL4/7816D8Wvjx4w03ULjRNJ0O58NJyn2m9QG4kGCMrjKKPQqW6AhhQB7v4q8deHPBdn9o8R6pDaA/dizukf6IOT+VS+DvFVn418JWXiDTYZ4bW8DmNLgAONrshzgkdVPevgDUNRvdWvpLzU7qa7uZWLPLM5ZmJOTyfcmvof4WfE3xdoHwz0nTNI+GWp63ZwCXy7+CdlSbMrscARN0JK9T0oA+k6K8a/4XL48/6I1rP/AIEv/wDGKP8Ahcvjz/ojWs/+BL//ABigD2Wq2o39vpWmXOoX0ixW1rE00rscBVUZJ/IV5H/wuXx5/wBEa1n/AMCX/wDjFcx43+K/jTxHpsfhV/hrqumS6of3kYuHaa4t0IaVEHlDGRwWIYAEjBzwAe8eGtZfxD4asdXkspbE3kQlW3lOWRT93P1GD+NaleLW/wAXvG9rbRW9t8FtXjhhQJGi3LgKoGAB+49Kk/4XL48/6I1rP/gS/wD8YoA9lrhfHfxY0X4d69pdj4hhuFt9RikcXUK7xEVIGGXrg56jP071yn/C5fHn/RGtZ/8AAl//AIxXjHx28X654t1HR5PEHhC88MvbxSrEl1IXM4JXJGUXpgevWgD680XX9J8RWC3uh6hb39u3SSCQMByRz6cg/lV6T/VP/umvzy0DxNrPhbUFvdA1GexnUgkxPgN7EdCPrX1D8HvjL4k8bt/ZuueGp7jamG1eyQLEvHHmqxAGdrnKnrgBO9J7Ael0UUV4Z3hRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGtpf8Ax6H/AHz/ACFXapaX/wAeh/3z/IVdr2KP8NHHP4mFFFFakBRRRQAUUUUAFFFFABRRRQAUUUUAVdS/5B0v0H8xWLbnCt9a2tS/5B0v0H8xWFEcA1rHYT2LO6jdUW6jdVWIJd1G6ot1G6iwEu6jdUW6jdRYCXdRuqLdRuosBLuo3VFuo3UWAl3VDZ/8f8X++KXdSWf/AB/w/wC+KOhUTpKKKKwKCiiigAooooAKKKKACiiquqCQ6PeCAMZTA+wJ97dtOMY70AJb6tp12t0bW+t5hZyGO5Mcqt5LgZKtg/KQCDg1RuvGPhqxsbS9vNf0y3tb3P2WeW7RUnx12MThuo6V4HoejeI1/Zf8TaaPC2qWWryXsQMP2ecXF8DJCXcqRuOV3A4GMA+9ZHxmSSP4X/DZJ9KbR5FEytYsrKYMeXwQ/wA2e/PrQB9WVm6P4i0XxDFLJoOrWWpRwsFka0uFlCE9ASpOK0q+ff2TP+Rb8Rf9fcX/AKAaAPaNW8Y+GtBvBaa54g0zTrkoHEN1dpGxU5AOGIOOD+VatrdQXtpDdWc0c9vOgkiljYMsikZDAjggg5zXxt4+W/8Aiv8AFbxXe6OTcWukW8hheNhJGYYOMhxhQGO9wc98DNe7fs3+I01r4SW1i8m650iaS2kDS73KFi6Njqq4baP9w49AAdtd/EHwdYXk1pe+KtGt7mBzHLDLfRq0bA4IIJyCD2otPiD4Ov7uK1svFWjXFxMwSOKK+jZnY9AADya+W2t/Atz8dvGi/Eq7mtdOF7cmB4hISZfP6fIpP3d3tWP8RbP4bwXFkPhbf6nd3jOgZWjfy+rdC4D787OAMYPXNAH2NrHi7w54fukttd13TdNnkTzEju7pImZckZAYjjIIz7Vn/wDCzPAv/Q46F/4MYv8A4qsPWvhH4c8fWmj3/iyC8e+t9PitztnaMjA3EMPXLHrXh/wf+Fnhrxn4u8X6frkNw9vpNwsdsIpyhAMki8nvwgoA+mv+Ey8NeZYJ/wAJBpm/UsfYl+1pm5y20eXz82W4478VYh8RaLca5Lo0GrWUmqQrukskuFMyDAOSmcjqO3cV5J4o+Gpj8f8Aw90zwtoly1poUsdzJqU0rGKGCOfzDGTjBcnp36dBk1p6D4Uvj+0jrniWLR5rHTorXyJLud2xeSsqYaMEfdAXBxxkdc0AetUUUUAfNn7Wjbbrwp/uXf8AOGvnTzTX0Z+1mQLrwpkZ+S7/AJw186709B+VAxvmmjzTTt6eg/KkYo3t9BQAnmmjzqAIx2NODIOi/pQA3zqYzbjmpd6eg/KjenoPyoA6TxD/AMiz4O/7A8n/AKX3dcw/U1O908kcaSSOyRLsjUnIRclsD0GWJ+pNV2Oc0AdL4s/4+tM/7BNl/wCiErmn+8a6XxZ/x9aZ/wBgmy/9EJXNP940CPSLL/kvPhz/ALCWl/8AtGvsfWvD+k+I7A2Wu6db39uf+Wc6Bscg8enQdK+MWuxYfGjRbxopJlt73TZTHFjc+0QnAyQMnHGSB719c+GfHFp4n1K4sYdN1CxngiWYi8WMBlJI42O3cd8UAeLeOv2W0ZJbzwDfbWVS39nXr8NgdEk7E46Nxk9QK9W+DOjah4e+Euj6TrVq9pfWrXCTQyYyp+0SHqOCCCCCMgggjg13NFABRRRQAVw/hzTr3Uvih4h8S6raTWv2SNdH05JIyoeAESPKDxuDORjjjaRk5ruKKACiiigArwX9oTwD4j8d+KfDVr4b06S5CQzCa4b5YYcsnLueBxk45JwcA171RQB4b4F/Zk0LRRHd+MLj+2rwAE26ApbocDI9Xwc8nGRj5RXtUFnbafp4tbG3jt4I1ISKJAqr34AqxTZP9U/+6aT2A56iiivDO8KKKKACiiigDzv4keO/EPhjxBoWj+FtLtNRu9W80LHcMV+ZduADuUDqetZY8W/GG2zNfeAbB4IwWdYLxN5HtiU/yNUfjHcanafEzwNPoNml9qKNcG3t5HCrI3ycEkgD86j1P4jfE6w1Cz0vU/C+l6RPqJMdtcT3OY93puDFQfY8muuMfdVkjJvV6noGg+LpfGHgNta8OW6R3210FrdZISZeqNgj8/cU3wZ46t/Evhqe+1BF0+809mj1G2c4+zuvX/gJ6g0fDnwe3grwmLC4mWe8uJ3urqRB8rSNjOPYAAfhXlXxXa0h8fXS6Kt0bSS2jPin7IAVEPmoFz/tYzn2P1xEYxlJxQ22ldnfaX8QNX1Xwl4l8TxadBBpdhBNJpnmht9z5aE725+6SMcYNc5o/j34ua9pFvqmk+C9KuLO5XfFL9qVdwzjo0wI6dxXb+KVsE+DWtLo/l/YRok/keWfl2eS2MVF8If+SR+H/wDr2P8A6G1F4qLlbqGt7XIfCfxEuNV8QSeHfFGiy6FrSx+bHC8gdJl9VYdT+f1ruq8v+KIaDx94AubKNTdnVPKJxz5RwG59MFjXqFZzSsmupUb7M1tL/wCPQ/75/kKu1wHi74oaB8N9Hjk1ozy3NwXNtawRktLjGfm+6oyR1OeeAa+cvHX7QnizxYZLbTJP7D05sgRWrHzWGTjdJ16EdMdK9Sj/AA0cs/iZ9jW95bXbTLa3EUxgfy5RG4by22htpx0O1lOPQj1qavmj9n/4n+E/B3gK/sPFGrtaXk2qSXCobaaXchiiG7KIR1Vu+eK9S/4X38Nv+hjP/gvuf/jdakHotFedf8L7+G3/AEMZ/wDBfc//ABuj/hffw2/6GM/+C+5/+N0Aei0V51/wvv4bf9DGf/Bfc/8Axuj/AIX38Nv+hjP/AIL7n/43QB6LRXnX/C+/ht/0MZ/8F9z/APG6P+F9/Db/AKGM/wDgvuf/AI3QB6LRXnX/AAvv4bf9DGf/AAX3P/xuj/hffw2/6GM/+C+5/wDjdAHoFzdW9lD515PHBFvVN8jBRuZgqjJ7liAPUkVKCGUFTkHkEd68C+Mnxc8EeKPhLrOj6HrRur+48jyovsc6bts8bH5mQAcKTya8e8D/ABw8X+CdkEd3/aenqMfZL0lwowQNrdV6j8gKAPtTUv8AkHS/QfzFc+pxXOeCfjN4f+I9nNZ2UdxZaokfmPZzIWG0MASsgGCOR1weeneuhJxWsNiZbFXW9Qk0zw/qF/CqvJa2skyq3QlVJAPtxXk2h/EP4r+JNFg1bRfB2lXNlcbvKl+0qm7axU8NKD1UjpXpfixv+KL1v/sHz/8Aotq8a+G1v8UJPh7preFr/RYtKPm+Ql0reYP3r7s4U/xbu/TFU9yFsel+BvH9x4l1TUtE1zSm0fW9NCtNbGUOGQgfMpH1HqOV5OeG/Eb4mWHgfR5jC8d3q20GK0HzbcnG58fdX8s9BXOfDy5uNL+IWrab4ws1XxVfQLctfxyBo7mEfKFQYG0AKBjvtPpUnxm0XTdO+Fuv3dlZxRXN5PDJPMB80hMy9T/SjXlDqejw6vGvhyLVr91hj+yrcTHsg27jXnP/AAsnxxqkQ1fw14H+06CNzLNNcqs0yD+JV3Aj6YNW/iHdTW3wAunt1LO2n28ZAH8LFFY/kTWVZ6N8RLjwvp2paX4htNOjt7JGg0xYNyuAgwHf1PXpxmhgj0nwv4jt/FPh211azSSJJ15jkGGRh1U1rbveuW8AeLP+Ey8GWersixzSApMingOvB/xrpN1UthEu73o3e9RbqN1OwEu73qWy/wCP6H/fFVd1WrL/AI/of98VL2LidJRRRXOWFFFFABRRRQAUUUUAFFFFABXz5+1hFMNJ8MXaQu0MNzMHkCnarEIVBPYnacfQ+lfQdVtQ06y1awkstUtIL20lx5kFxGJEfBBGVPBwQD9RQB5FJ+0t4QXwUdQjklbWvJIXTjC3+u2Z5bp5e7jOc47V578HNfl8F/Avxrr3zQO0ixWUhIXfMUKjYTwxUsGxz0r2b/hQ/wANc/8AIrw/+BM3/wAXXQ3/AID8M6n4Vt/DV5pEL6PbFTFaIzIqlc4OVIOeTznkkk0AfNnwY8JfEZfD13rng230VbXUt1vI2qqxMqL1AGMFCSR35BHatH4BXt/4G+MOseDNexby3QaNoydiGaM5Vl3AFgyk7cdQwNfSmh6Hp3hvRbfSdEtha2NsCIoQzMFyxY8sSepJ61kz/Dzwtc+Mh4rm0lG1tXVxd+bJnKqEB27tvCgDpQB4J4K8H6H40/aJ8fWHiSxW8t45LuSNSzLsf7QF3AgjnBNYPiTw/efAH4tWOq2ttb3mjXMjPZmdVlPlgruXnlZFypDDHUYPUV9QaV4I8O6H4kv9f0rTVt9U1Hd9quBI5Mm5gzcEkDLAHgCrHiPwvovi7STpviPT4r+03iQRyZG1h0IIIIPJHB6EjoaALmmajbavpdtqNhKsttdRLLE6kEFSM9RxXg/7Ov8AyUP4if8AX2v/AKNmr2/w/wCHdK8LaPHpWgWgs7KJmZIVdmALHJ5Yk9TVTw/4J8PeFr/UL3QdNWzuNScPdOsjt5pyTnDEgcsemOtAG9RRRQAUUUUAfPP7UE9hBqPhdtTsGvozFdhUE5i2nMPOQDn6V4T/AGj4Z/6FmT/wYv8A/E17r+1Fpy6nqPheJr+0sdsV22+7dlVuYeBtU814P/wikf8A0Muh/wDf6X/43QA/+0fDP/Qsyf8Agxf/AOJo/tHwz/0LMn/gxf8A+Jpn/CKR/wDQy6H/AN/pf/jdH/CKR/8AQy6H/wB/pf8A43QA/wDtHwz/ANCzJ/4MX/8AiaP7R8M/9CzJ/wCDF/8A4mmf8IpH/wBDLof/AH+l/wDjdH/CKR/9DLof/f6X/wCN0AP/ALR8M/8AQsyf+DF//iaP7R8M/wDQsyf+DF//AImmf8IpH/0Muh/9/pf/AI3R/wAIpH/0Muh/9/pf/jdAD/7R8M/9CzJ/4MX/APiaP7R8M/8AQsyf+DF//iaZ/wAIpH/0Muh/9/pf/jdH/CKR/wDQy6H/AN/pf/jdAEGvasmsX8c0NqLSKK3it44RIX2rGgUcnqcCsV/vGuh/4RSP/oZdD/7/AEv/AMboPhOM/wDMy6H/AN/pf/jdAHRKEb42aEJv9Wb/AEwN9MQ5r6p0KO0j+IlyLE5U6Um76+a1fJ6iLV/jHo8OnXqMk2oWFul1GpZQwMSFgDjIBB9M4r648K+Cp/Dur3OoXmsNqMk8CwAG3EYQBi394560AdZRRRQAUUUUAFFFFABRRRQAUUUUAFNk/wBU/wDumnU2T/VP/umk9gOeooorwzvCiiigAooooA898a+G9W1T4peC9VsLNprLTZJjdzB1HlBtuOCcnoema6jxX4as/Fvhy50q/HyyrmOQfeicfdYHsQa2qKvnenkKyPPtGvPHGmfDm5tLzRnutesiba1k86Mrcr0WUndxgdQeePer/gvwHb6J4YubbWUjvdQ1bMmqTOM+czZyv+6MkAfWuyopubewuU8otfC/ibRPDnivwfBZvfaRPp9x/ZFwJFyjOjAQNkg9TwelZXhbUvit4W8L2OiW/gKCeOzj8tZXvYwW5JyQH969soqva90Ll8zznQPCfiPWfGUHinx4beKWxQrYWFq5ZIiRguT3PNejUUVnKTkUlYtHSNP1zQ5rDWLKC9tZWIeKdAynjrz0Pv1FeL+Ov2XrC9Ml54GvPsExy32K5JaInJOFbqvUDnPAr3TS/wDj0P8Avn+Qq7XrUf4aOSfxM8p/Z68Laz4P8C6ppXiKxksrtdXkcI2CHQww4ZSOCOCMjuCOoNerUUVqQFFFFABRRRQAUUUUAFFFFAHB/GvR9Q8QfCHWNK0a0kvL66e2SKGMcsftMZPsAACSTwACTXk/gb9lxnEd548vtgIz9gs25GQeGk/EdPQ819KUUAYNn4b0bwv4dksfD+m29hb8EpCmCxz1Y9WPuSazpTgiul1L/kHS/QfzFcvcHDLW1MmWxm+IYJr3wxqlrbJ5k09nNHGmQNzMhAHPua8m8IXnxP8ACHhSz0O28EQ3EVrv2yyXiBm3Oz9A/wDtYr2XdRurVxuZXOC8LeGvEN946k8Y+M47W2uktRa2lpbtv8pMkklvX5m/OtD4r6NqHiL4b6hpujWxubuV4ikQZVziRSeSQOgNdbuo3UculguZFxoa6v4D/sLUAYxPYrbyYOSh2Afoa4eN/ijpemr4dtNK066ijh8iLVWn2hVxhSV65A9q9P3UbqOULmL4J8NR+D/CNno8biRoVJlkAxvc8k1v7qi3UbqdrBcl3Ubqi3UbqLCJd1XrH/j9h/3xWZurTsf+PyD/AHhUS2NIHS0UUVzGgUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHN+LPh94Y8cNaN4p0z7cbMOID58sezdjd9xhnO0dfSud/4UF8NP8AoWv/ACfuf/jlejUUAeF+Kvgj4Nj8XaPp2jaKYI7ixvJ5lF3M24xyWyqeXOMea3T1+lQf8KC8P/8AQLf/AMCJf/iq7vx9qz6F4z0C/Ecxj/s++hZ44mcAtJasAdoOM7D+RrK/4WSPS6/8BZP/AImgDmdL+CHhOLxtYWOp6Q0trcWdzIYzdTLlkaIA5Dg8Bz+ddr/woL4af9C1/wCT9z/8cqLw14ifxF8RNPZY7jZbWF1ud4HQAs8OBkgddp/KvS6APOf+FBfDT/oWv/J+5/8AjlH/AAoL4af9C1/5P3P/AMcr0aigD54074I+HdRW9mj0xvLXUr2CMC4l+VI7mSNR97sqgfhVz/hQXh//AKBb/wDgRL/8VW7ZeMn0KXUtPljukaPVr9gBbSEENdyuCCByCGB/GrP/AAskel1/4Cyf/E0AZvgn4IeBL/Qpn1fQfPuYr65hLm8nX5UlZVGA4HAArof+FBfDT/oWv/J+5/8AjlbXw3uHu/Cst1IkiG41C7lAkQqSGmYg4PNdZQBwWnfBH4e6VqlrqNh4f8q6s5kngk+23DbHRgynBkIOCAcHiu9oooAKKKKACiiigAooooAKKKKACiiigApsn+qf/dNOpsn+qf8A3TSewHPUUUV4Z3hRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGtpf8Ax6H/AHz/ACFXapaX/wAeh/3z/IVdr2KP8NHHP4mFFFFakBRRRQAUUUUAFFFFABRRRQAUUUUAVdS/5B0v0H8xXL3SEoGA6da6jUv+QdL9B/MVz1bU9EJq6sZm6jdVuSzRjlSVP6VH9h/6af8Ajtb8yMeSRBuo3VP9h/6af+O0fYf+mn/jtHMg5ZEG6jdU/wBh/wCmn/jtH2H/AKaf+O0cyDlkQbqN1T/Yf+mn/jtH2H/pp/47RzIOWRBuo3VP9h/6af8AjtKLEZ5kJHsKOZBySIYwZHCgZ9a17Hi9h/3xVaOJIlwg+p9atWX/AB/Q/wC+Kzk7msY2OkooormKCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACorm7t7OLzbueKCPON8rhRn0yalrx39p7/kkK/8AYSh/9BegD16CeG5hWa2lSaJvuvGwZT9CKjutQsrEqL27gty/3fOlVM/TJrxf9nTUJtI/4SPwFqs0RvdFvnKBN53rna5Utj5cqCOAfmzivJ/jrqkvjP4ia9fWE0cmm+HUisgw3jc27D4zwT5jMM8AhQRnqQD7CN5bC6W2NxEJ3XesW8bmX1A6kcHmpq83a68KL8afD9rdWF6/iltDDW12rfuEt/3mVYb/AL2Q/wDCeo59H6z8a/D2l+KZ9AsNO1nXr22XM/8AZFssyREAkqSXXkAZOMj3yCAAei0V5xa/HLwne+AtS8V26X7W+lyRx3dkYkFzGXcIp2l9uCTnO7se4xTdL+OnhTUkubiSDVNPsLeza8F/e2wjhmUMqbUO4szbmwBjkg4oA9JoryRP2jPC4tLO+u9D8SWem3kvlR389knk5zyciQk4zk4BPsa9VtLu3v7OK7spo7i3mQPHLGwZXU9CCOooAW5urezh827nigjzjfK4UZ+pqWvIf2mv+SPP/wBf8H/s1etw/wCpT/dH8qAGLd2z3T2yXETXEYy8QcF1HqR1HUfnUNxq+m2kxhutQtYZV6pJOqsPwJrxvSpLmL9pr4jyaepa7TRImgAXcS4hg28d+ccV5/8AC7wb4B8X211dfETXZv8AhI57qS2awuroW7I3y7WUZ3MeSMnAzkbflyQD6st7mC7hEtrNHNE3R42DKfxFVG17SEcq+q2SspwQbhAQfzrzj4T/AA48Q/D3xDqkSa3b3vhC5RmsrfzjJMsm5drn5Aoyu4HacE449PKfAfh74X6xdeJpviLe2lvfJrM6wLPqDQExZznAYZ5J5oA+pLXUbK+LCyvLe4K/e8mVXx9cGrFfMPwu0jSbP9oaMfDDUr6/8PQW8rak8nyxqCrhUBODIu/Zg4zxnkAmvp6gAooooAKKKKACiiigAooooAKgF9aNeNaLdQm5UZMIkG8Dr93rU9eMfG3wZqdtqFl8RvBsbf2xpGGvESQjzoF55Udccg8j5fpwAexXF1BaQmW7njgjBwXlcKB+JqGfVNOghje5vraKOdcxs8yqHHqCTz1HT1r50l1HU/2ivHmnWVtHNbeEtNggm1JEmIXzmTc65KglgxMY46IWBGa6D436HpWoePvhnomoxKulyTXEEkYcxgR/uRjIxigD137f4d/6CVj/AOBa/wCNX1sbR1DIu5SMghiQRXjHiP4KfCGz0C5nkvV0cIAfti6iXMZzx8rEg56Yx345qh8I/wDhKfEfwH1/SfDWsql9FqD2um392GQCHKFjkKzDKl8cEgkDjFZ+zh2RXNLue1LcaE1wIFvbQzFtgjFyNxbOMYznOe1Xf7Ptv+ef/jxr5x8ZfATw/wCCvhdd6/f+IL2PXbKMTfalOY5ZyV2xhcbsFuA2cjduPAxXqHhPxB4j/wCFAaVqun2B1fXms444LeQ7BIxkEasx/uqCGJ4yFPI60ezh2Qc0u53Hk6d9q+y74/tG3d5Pm/Pj1xnOKSFdLuJ5IbeaGWWI4kjSbcyHOOQDxzXg3w9tPFdp+0sf+E9vILrV5tDMreQcrEpK4TgAAjBzjIz0Jq7p8Mnw9/aruonRo9M8Wo0kTKNwaVvmbLNjH7wMcLn7y/QHs4dkHNLue4TWtjbwtLPtijUZZ3kwAPck0kFvYXUKzWrJNE33Xjk3KeccEGvM/wBorXpdN+Go0iyEj3mt3KWkUaIGLDOSMHnnAAwOprWstK174f8AwT07SPCWlx6hrdvbpEsPmHYJpGzJIS2PlDMzYOB24o9nDsg5pdztFh017prVJI2uEXc0Ily6j1IzkDkfnU39n23/ADz/APHjXgnwks/EVj+0j4lh8Z3kN7rJ0cPcSwHKfM0DKo4HRSBgDHHFem/FP4hw+AfC7S2vlXOt3ZEWnWLElpnJA3bQCSFBz2B4GQSKPZw7IOaXc6iKHTpppIYXjkliOJESXLJ9RninT21hawtNclIYl+88km1R25JNeG/s6Wmp2Pjnx1Br7h9SDWzXJGfvsZGI5AxjOMY4qH4+SfEDVdC19/Kg0zwfprQqwd/3uoEyKAwAGcBmU4OBxnJIo9nDsg5pdz3pLKzljWSNQ6MAysrkgg9wc1Tku9AilaOW/s0dCVZWulBUjqCM1xGt69eeGf2ZYdV0xhHdw6DaLE/OULpGm4Y7jdke4Fea+GPgNpGv/Bj/AISLUF1V/EN7aSXsQEnJYbzGoXHzBxsPOSc8EZo9nDsg5pdz6VhhSFNsQwpOeuafXln7O/iHUPEHwnh/tWUzS6fdSWaSuSXZFCsu4k8kb9v0Ar1OrSSVkSFFFFMAooooAKKKKACiiigAoorn/Huo3ukfD3XdQ0pyl7bWMskDKu4hwpIOKANObWtLt5min1KzikU4ZHnUEH3BNWLe5gu4RLaTxzxk4DxOGH5ivijwToHgXxfDdyeNvG15peu3Tv5XnQDylbBbzHlOQ4PHBKHIIycg17t8Efh14o8C6jdvN4i03VPDN3EWgS0mdy0m5dkmCuFyucgMeo69aAPY5YkmjMcgyrdRmq/9l2n/ADy/8eNeceJP2gvCPhXxNqWhapa6t9r08gO0UEbJKx2nCHfnOGz8wA4PtlPDn7Qng/xH4mt9DFvqumXVywjibUIEjQufuplXYgnPGRTuwPSP7LtP+eX/AI8aP7LtP+eX/jxrkPH3xe8NfDu9tbLWBeXd7crvW1sY1eRVzgMdzKACRgc5NZvg/wCPPhLxn4kXQrWLUtPvpMiJdQhRBIw/gBV2+brwcdKLsD0H+y7T/nl/48aP7LtP+eX/AI8a898VfHnwr4O8X3vhzV7XVTeWaKzPBAjxyFo1dVU7wckMByAM98c1x3xW+McGq/B9n0bTvEGlTatM8FvPPbrFhYmiMhYhyQrByoIzkhhxii7A9xTTrGRA8ah1PRlckH9ad/Zdp/zy/wDHjXmHwS+IOl6j8M1triK8sx4esYze3d4gWJwQ5LI24lgNhySB1FUpf2ovBUe9k0zxBLCjlBOlrFsY+xMoPI5wQD7UXYHrn9l2n/PL/wAeNH9l2n/PL/x41zM3xR8PJ8M5PHVu1xd6VEql0gRTMrM6psKlgAwLDPP0zxXGyftO+B1s4p4LPXLnchaWOK0TNvzgByXC89flJH48UXYHrH9l2n/PL/x40f2Xaf8APL/x41h+BPiBofxE0RtS8PyShY38ua3uFCywt23AEjkcggkfiCK6ei7Ay7qPR7Hb9tmgtt+dvnT7N2OuMmprS20+ZUubJklTOVkjk3Kce4OK+SSIvjJ8QNZ1bxz4oi8P6TZOYrfz5UIjyzbIUyV7KzFsHO3nk16v8AfA2t+E7jULseJdI1rw7eRbIf7PuHlxKj8HlAF4Z8gHqR1xmi7A9tooopAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeO/tPf8AJIV/7CUP/oL17FWD4x8G6R460IaRr6SvaCVZsRSFDuXOOR9TQB4d8VrmX4bfHe38XWc8dvHq2mzK7spkPmrEVGVxgAkRDj0OcVyUfh06X+ydf61OF8/W9VilyGP+rSQooIIwDuDnjOQR+H0r43+H+hfEGxtrTxHHNJFayGWMQylCGIx29qXWfh/oWu+BrXwlqEczaVaxwxRqkpVwsQATLfQCgDzu7/5Or8J/9iv/AFnrP/Zpn02F/GkTK8WprqStO8jAI0Pz+WAM9QwlJ46Fevb1l/BOjyeNbHxU0c39p2Nn9ihbzDsEXzcFe5+duawvEvwX8HeKdfm1m/tLiC9nTZM9ncGES9QSwHUkHB9aAPL/AB1rfgjVvhP8QofAukvZSWtzZpf3PkhEunNzwV+Yk4IfqB96rfxvs7ey/Z08Jw28Hl20V1ZlkjHQG3kJ/U/rXqh+F3hb/hX8ngyKyaHSJSrSrHIVkkZXDbi/UnKjn04rduNA0288N/2De2q3OnG3W3aGX5gyAADPvwOfUZoA83+I2oeEG/ZuneMI2lTWEaaZEjpvE2B5QHPLK3L4JOA/WsbwVp/xIufhF4RPgDU9IsLdbOT7QuooxZm858FcRtxj6V0kP7PfgWL7MjwahPb20vmx20187RA5BPy9Occ+temxRRwQpFCixxxqFREGAoHAAHYUAeDfF/TvGI/Z+uV8aT2moanFqUcry6ch8tIegJ+VcYJOTjuK9N8K/Ejwz4j8HQa5Hq9jbRrAj3kctwoNo5GCj7sYwQQCQAcZHBFdNqGn2mq6fPY6lbx3NrcIUlhkXKup7GvMbj9m/wCHdxdSzfYLuLzHLeXHdMFXJzgDsBQBxHg/xp9v+L/xI8daJpc+pWNtpwWJItwE/l+Wi/Nt+UssTPgjIGfTNdDpUHwm+NWmNrF/ZWtlqxJmvoPtPk3EZUAEsRjcnI+bGD3wQQPTvDHgrw/4O0d9M0DTYre3lA87I3NMQu3Lk9ePw5PrXG6l+zx8PdS1CS7bTJ7YyEEx21wyIOOw7UAch8EtQGmfFzxF4R8Pa/JrXhe3szcWzsd6xuHjUKregDsMjhtoNQ/B3wb4R8SR+K7nxRpVje3EevTxxvc/eCcHA56ZJr2Pwb4F0HwFpclh4bs/IjlffLI7bpJD23MeTjsO1cbefs6eAL++nu7i2vjLcSNK5F2wG5jk/qaAPNvE2meG/B37QnhCP4d3Qtprm+hi1G2s5QyIrSqrIcMcZUnKkdMHPp9O1w3g34P+D/AurPqWhWMn2xk2LNcSmQxg9duemehPpXc0AFFFFABRRRQAUUUUAFFFFABXkPxw8dXdlaQ+BvC1u15r+voYgsTjdDGeDkZyCwyMnAwGOeK9erlbL4ceHbLxzd+L/s8lxrN0cme4k3iPgD5B0XAAA9qAPD/D1rrH7OvjezTWo/tXhzXoYIZ7mOXKQ3AUbzkhQNrF8Z6oc5yDjqPjjpVh4n+I3w10y/8A39hf3Fyknlvjeh8now/nXrHivwlo/jTQ30nxBai4tmIYYOGRh/ErdjWWnwz8Pq3hpmF1I3hjd/ZxeckoDt4b+8AFUD2FAHifxe+AeleGtBTxH4QhuGtrEqb/AE93eRpI93Lo3JBGec8Y54xz6/4F8S+ELX4SRa74fh/s3QbWB5pYdhLQlcmQEdWOQeR1ruJYknheKVQ0cilWU9weCK4O0+C3hCx8N6xoNtDeJp2sPHJcwfamIBjfcuz+7yBn1AHpQBwEMWtfHrUrS88SpHongW1mNxbWvmqJ79gSFLc5XjOTwMHC5zuHvFnDb21jBBYqiW0UaxxKn3VUDAA9gBXlf/DNPw8/59b/AP8AAxq9I8O6BY+F/D9poukq62dmmyISNuYDJPJ78mgDzAf8nfN/2Lv/ALNVf9pXQZpvCGn+J9NDJfaHdrIJY4yWRCRzn+EBgpz9K9L/AOEN0n/hO/8AhLtkv9rfZPsm7zDs8vOfu+vvV/XNFsfEWh3ekatD51ndxmOVM4yPY9iDzQB892mrD4x/tAeHZUj3aXo+mwXkkYbz41laNZWBIxtId1Q57x/gPpOuO8EfC3wx8Pri6uPDltMk10gSSSaYyHaDnAz0Gf5V2NAHjXhv/k73xd/2BIv5W1c5r/hT4oxfHK98Zad4Uttait2aPTftV5CsaR42qwXzVOcZ69znrXtNp4L0iy8dX3i6BJRqt/brbTMZCUKDZjC9j+7Wt+gD50+DmpeMpfjn4p/tHQ7WAXUynXNkyn7EwSQxhPnO7c3Bxu/CvRPj/wD8kL8Q/wC7b/8ApRFXU6L4N0nQfEes65p6Sre606Pdl5CykrnGB2+8aseJ/Den+L/Dd1oesrI9ldhRKsb7WO1w4we3KigDznxpbTXf7J4itonlcaFZOVRSTtURMx47AAk+gBp3gP4k6Zbfs/2Wpz6lpkd9pemGD7O9woxJErLEjDOQzhFOOpzxXp1lpVpY6Hb6RFEHsre2W1WOX5w0artAbPXgc+tea3H7N/w7uLqSb7BdxeY5by47pgq5OcAdhQBT/Zigni+E0008Txrc6pNLEWUgOu1FyPUblYZ9QfSvYqq6Zpllo2l2+naVbR2tnbII4oY1wqgf5696tUAFFFFABRRRQAUUUUAFFFFABWF4z8VWHgvwnd67q9vcXFnbbRJHbIruQzBeAxAx83PPSt2qup6ba6xpVzp2oxCa1uomimjJxuVhgigD52tvhx8LPi0k+reFNXn0C+nWRzp0kkW6Ng2PMaHcSFyRwGAxjpmsf4QzX3gv9oGLwbpniNNX0mXzopzDnymKQu/CnhWDLglSe/Jrt7v9lTwtPdySW2s6nbRM2VhGxgg9MkZNdd8Ovgr4d+HN/NqFlJNqF/INsdzdBd0KkchQBgZ7nr2oA8kt30Jf2xtSXxJHG8Msxjt/O2+WJmhULu3HBBG4Ac/MV4qX9qZLC38QeH57APFq7B2Z4VUZAK7SSPm35xj2qvN4J0zx/wDtQ+LdF1lpkh+zvLG8LbWSQJEFb3xuPFemeHP2fPD2i+KItf1LU9S1u9hkWaJr2QHDr0YkctjA6+goA5T4keM5bj4zpoXhDwXp+reJLKFUW/vgN0bFN6lMsFAUPnLdz+fBaIfETftM6A/jO3srfWJLmNp1s2hIOVbBfyiQGI555xg9xXuPjr4E6D438UDxAb+90vUCF82S0I/eMoAVuehAAGR6CsXTf2aNA0nWNO1Oy13VY7mxkWUvlcysHLZPHHGF49PegDk73SLfWv22JYLvJjh8m4C4BDNHaIwBB7ZFejftE/8AJD9Z/wCulv8A+j0rXj+F2nx/F6X4gC/uTeyR7Da4Xyx+5EXXGegz9a2/GnhOy8b+E7zQNTklit7oLmSI4ZCrBlIz7gUAef6R4i07w3+yzZanqtmNStItJjjlswR+98xvLCt6KS3J9M9elePapqvjbWPhTcT6d4S0HRPB0kbSShPJTewOPMUSv5m/JAXaOeAM5r2/wz8CtC8P+Gtb0K51C+1Sx1iKKN47hgPJEbMylMdDufd9RXNw/ss+HEXy59f1aaABysJKBVdlwGwB1B2n32gGgDkNE/5Mp1z/AK+//bmKvSfgn4T0a7+A9nbz2UbrrMEwvGI+aQF2X73UYCjHoeat6Z8ENM0z4a6v4MTWtQlstUlSVpX27oSrK2FGMclBmuz8HeGLfwb4RsNAs55LiGyRlSWUAM2WLc44/ioA8V/ZL/5AfiX/AK+YP/QXq7p17+0OddtRqOm2q6eblBOwazyItw3Hh8/dz05r0D4Z/C7T/hjZ6hb6Zf3N6t9IjubgKCpUEcYHvXcUAfG3wv8AAXhrxR448Qaf461B9OmsvMMdk7eSzsC29ix6bMA7epznoDXW/s2TDT/ir4k0PStVkv8ARUtZJInAKpMVmjVZNp6Hax/OvSvHvwC8M+Ote/th5rjS7uQf6Q1oFxO3ZmBH3vcde9b3w6+Fug/Dawmi0hXuLu4P769nA8x1zwvHRR6DqeTQB2lFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUotF0uDVJNTg02zjv5RiS7SBRK444LgZPQd+wq7RRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH/9k=)

Interface between PC USB port and Rx,Tx of FPGA

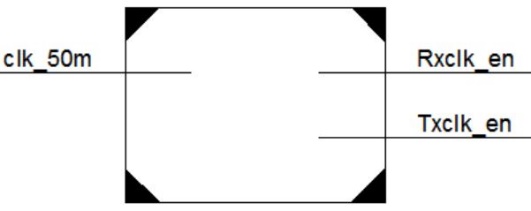
In the FPGA, there is a EXAR USB-UART bridge which can enable communication with the board using COM ports. EXAR part delivers the data to the FPGA using software flow control and 2 wires.

The UART module consists of 3 sub modules

1. Baud rate generator
2. Transmitter
3. Receiver

### 3.3.1 Baudrate Generator

This module generates separate clocks for Transmitter and Receiver modules. The input clock signal for this is a 6.25MHz clock which is generated by dividing the main clock (100MHz) by 16.

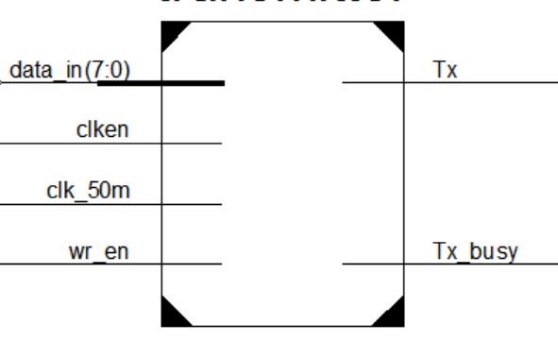


|  |  |
| --- | --- |
| **Control Signals** | **Description** |
| Clk\_50m | Clock signal for enabling the baudrate generator |
| Rxclk\_en | Divided clock signal for the transmitter |
| Txclk\_en | Divided clock signal for the receiver |

### 3.3.2 Transmitter

Transmitter consists of 4 main stages in transmitting process.

1. TX\_STATE\_IDLE - Transmitter is idle and Tx pin is set to high
2. TX\_STATE\_START - Transmission starts
3. TX\_STATE\_DATA - Parallel data is sampled and sent through the Tx serially
4. TX\_STATE\_STOP - Stops the transmission and sets the state back to idle

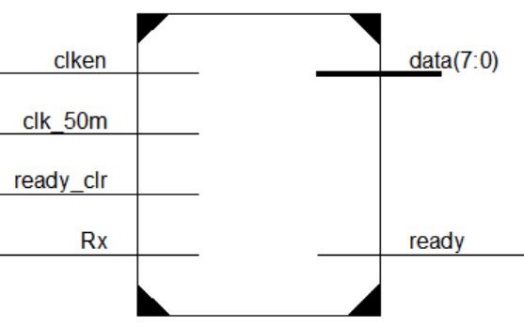


|  |  |
| --- | --- |
| **Control Signals** | **Description** |
| data\_in | Input data to the transmitter which is to be sent through Tx |
| clk\_en | Divided clock for the Transmitter generated by baud rate generator |
| clk\_50m | Input clock for enabling the transmitter |
| wr\_en | A flag which should be high to start transmission process |
| Tx | Wire to transmitter pin |
| Tx\_busy | Checks whether transmitter is in IDLE state |

### 3.3.3 Receiver

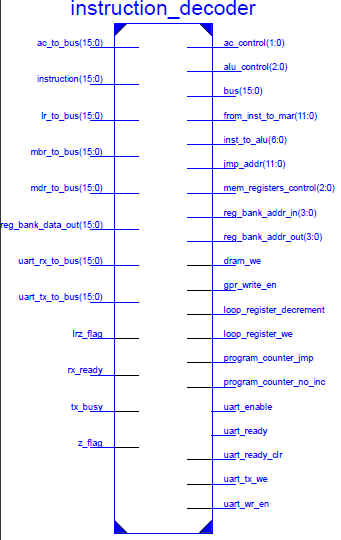
There are 3 states in the receiver

1. RX\_STATE\_START - Starting the receiving process. The *ready* flag is set from high to low
2. RX\_STATE\_DATA - Collecting the incoming serial data and converting into parallel form
3. RX\_STATE\_STOP - Ends the receiving process. The *ready* flag is set to high again



|  |  |
| --- | --- |
| **Control Signals** | **Description** |
| clk\_en | Divide clock generated by baud rate generator to run the states in receiver |
| Clk\_50m | The main clock to enable all processes of receiver |
| Ready\_clr | Flag to enable the transmitter |
| Rx | Wire connected to receiver pin |
| data | Serially incoming data is converted into parallel form and assigned to this wire |
| ready | When ready clear is set to high this becomes low and starts transmitting. At the end again this becomes high |

## 3.4 INSTRUCTION\_DECODER



Instruction Decoder receives the binary instruction from the Instruction Register and issues all the control signals required for executing the instruction. Other than control signals it also issues immediate constants extracted from the instruction to other modules. It also includes the Main Bus multiplexer which drives the Main Bus. In total, Instruction Decoder issues 19 control signals and 6 immediate constant outputs. Other than the instruction itself, outputs of AC, LR, MBR, MDR, UARTTX, UARTRX and Register Bank (registers which will be driving the bus) and the 4 flags (Z flag, LRZ flag, TXBUSY flag, RXREADY flag) are given as inputs to this module. Instruction Decoder is a combinational module. A summary of the outputs is given below.

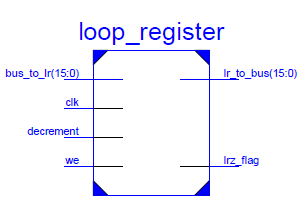
**Immediate constants:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reg\_bank\_addr\_out [4 bits] | Reg\_bank\_addr\_in [4 bits] | Inst\_to\_alu [7 bits] | Jmp\_addr [12 bits] | From\_inst\_to\_mar  [12 bits] |
| Address of the general purpose register which is read into the main bus | Address of the general purpose register which is written from the main bus | Immediate constant from the instruction to the ALU | Jump address from the instruction to the Program Counter | Immediate LOAD/STORE address offset from the instruction |

**Control signals:**

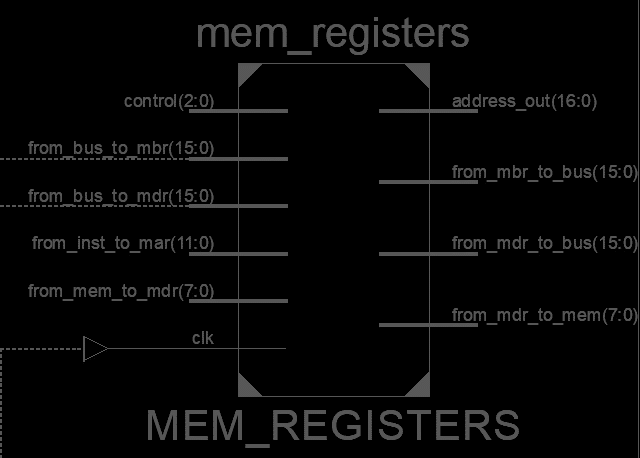
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INSTRUCTION | ACCUMULATOR | | ALU | | | MEMORY REGISTERS | | | REGISTER BANK | PROGRAM COUNTER | | LOOP REGISTER | | UART | | | | | DATA RAM |
| AC WRITE ENABLE | AC INPUT SELECT (0:BUS, 1:ALU) | ALU OPERATION SELECTION (0:ADD, 1:SUB, 2:MUL, 3:DIV, 4:SHR, 5:SHL) | | | MBR WRITE ENABLE | MDR WRITE ENABLE | MDR INPUT SELECT (0:BUS, 1:MEMORY) | REGISTER BANK WRITE ENABLE | PROGRAM COUNTER JMP | PROGRAM COUNTER NO\_INCREMENT | LOOP REGISTER DECREMENT | LOOP REGISTER WRITE ENABLE | UART READY | UART READY\_CLR | UART WR\_EN | UART ENABLE | UARTTX WRITE ENABLE | DATA RAM WRITE ENABLE |
| NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| UARTSEND TXBUSY = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| UARTSEND TXBUSY = 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| UARTREAD RXREADY = 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| UARTREAD RXREADY = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| ADD | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SUB | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MUL | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DIV | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SHR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SHL | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOAD | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| STORE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| JUMP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| JMPZ Z=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| JMPZ Z=1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| JMPDEC LRZ=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| JMPDEC LRZ=1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MOVE | WRITE ENABLE SIGNAL OF THE CORRESPONDING REGISTER WILL BE 1 DEPENDING ON THE DESTINATION. ALL OTHER SIGNALS WILL REMAIN 0 | | | | | | | | | | | | | | | | | | |

## 3.5 LOOP\_REGISTER



Loop register is a normal 16-bit data register except it has its own subtractor and associated LRZ (Loop Register Zero) flag. This register is sensitive to the positive edge of the **clk.** The included subtractor will decrement the value in the register in each clock cycle until the **decrement** signal remains high. **bus\_to\_lr** is the input from the Main Bus and when **we** is high, the register value will be replaced by this value.

## 3.6 MEM\_REGISTERS



MBR and MDR are the two registers come under this. MBR consists of the base address which is used in construction of the address and the data in that address will be loaded to the MDR replacing any value which it contains. MDR is an 8-bit register. We cannot directly connect the 16-bit bus to MDR. Therefore, we add 8 zero bits before MDR when connecting to the bus.

The address to be stored or loaded isn’t stored in a specific register. In the LOAD or STORE instruction, the address is given as a 12-bit address. It is added to the twice of the value in the MBR to get the address to be loaded from the data memory. That is, the value in the MBR is shifted 1 bit left and added to the given address in the instruction. Therefore, if we increment MBR by 1, it increases the address by two. When we run the down sampling loop, it is easy to skip pixels using this way.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

Address from instruction

(+)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

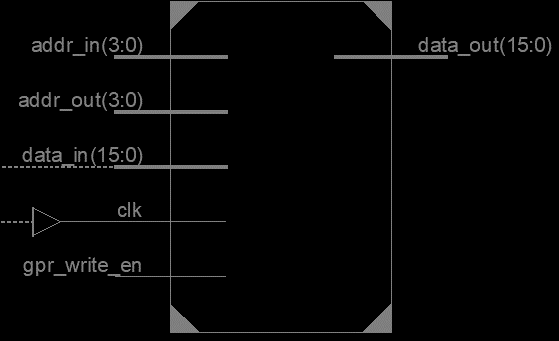
MBR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Address loaded from memory

|  |  |  |  |
| --- | --- | --- | --- |
| Control signal | operation | | description |
| Control[0] | Input selecting | 0 | Input is taken from the bus |
| 1 | Input is taken from the memory |
| Control[1] | MDR write enable | | if(control[1]): mdr<=mdr\_input\_wire  at the next positive edge of clock |
| Control[2] | MbR write enable | | if(control[2]): mbr<=from\_bus\_to\_mbr  at the next positive edge of clock |

## 3.7 REG\_BANK



There are 16 general purpose registers. Data of the register specified by the address given through ‘addr\_out’ will always available at the ‘data\_out’.

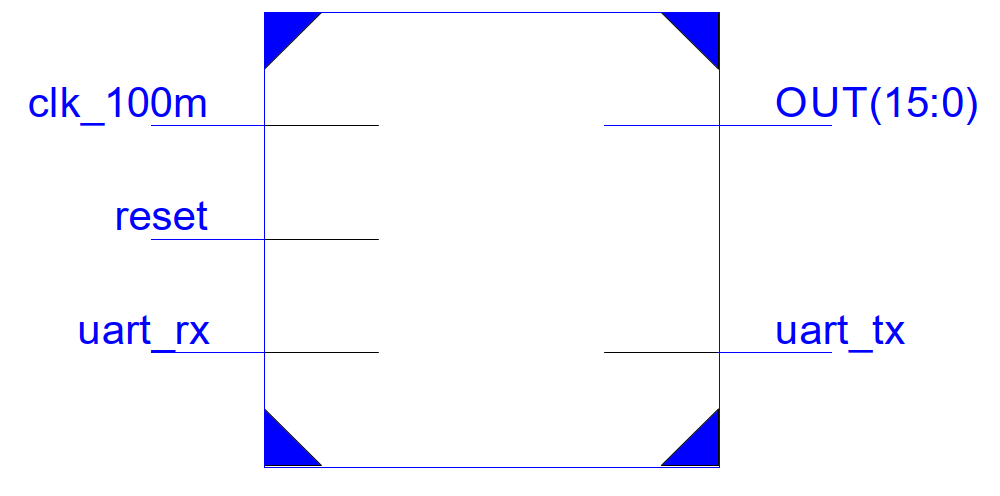
|  |  |
| --- | --- |
| Control signal | description |
| gpr\_write\_en | When high, writes data from ‘data\_in’ to the register given by the address ‘addr\_in’, at the positive edge of clock |

## 3.8 IRAM and DRAM

IRAM (instruction memory) and the DRAM (data memory) were constructed using the in-built Ip Core Generator tool of the Xilinx ISE Design Suite. These were constructed as block memory modules. Each module has its **clk (clock), we (write enable), addr (address), din (data in)** and **dout (data out)** pins. The on-board 100MHz clock was directly given to the clock pins of these RAMs resulting a low-latency data in/out cycle.

**Dimensions:** IRAM – 16 bits x 4096 words, DRAM – 8 bits x 131072 words

## 3.9 PROCESSOR\_TOP\_MODULE



This module wraps all sub modules (processor, RAM and UART transceiver) together. **clk\_100m** pin is connected to internal built-in clock of FPGA and internally divided into 6.25MHz clock because timing analysis indicates that maximum clock speed we can use is 8.602MHz. It is done by incrementing clkreg a 4-bit register at positive edge of clk\_100m and by taking 4th bit as divided clock signal. **reset** pin is connected to reset button in order to restart the program. Data communication between PC and DPUT Processor is done through **uart\_tx** and **uart\_rx** pins of this module. These modules are connected to UART interface of the development board. **OUT** is a 16-bit port which is connected to 8 LEDs of the development board. it can be connected to any internal wire/register and can be used to observe states of connected wire/registers. By default, it is connected to wire **ac\_to\_bus**. [Appendix I](https://github.com/damithkawshan/Image-Downsampling-Processor/blob/master/Documentation/top_module_inside.pdf) shows the internal wiring and sub modules of this module.

## 3.10 TEST\_BENCH

Test Bench was used for the simulation and debugging process of the processor in ‘ISim’, a Verilog simulation software. This module includes an instance of the **processor\_top\_module** and an instance of the **UART** module placed externally. This allowed to carry out a complete test regarding the functionality of all the modules comprising the processor. Test Bench does not have any inputs or outputs.

# 4.0 Design Summary

## 4.1 Resource utilization

|  |  |  |  |
| --- | --- | --- | --- |
|  | processor\_top\_module Project Status (05/03/2019 - 15:50:36) | | |
| Project File: | processor\_top\_module.xise | Parser Errors: | No Errors |
| Module Name: | processor\_top\_module | Implementation State: | Placed and Routed |
| Target Device: | xc6slx45-2csg324 | Errors: | No Errors |
| Product Version: | ISE 14.7 | Warnings: | 32 Warnings (1 new) |
| Design Goal: | Balanced | Routing Results: | All Signals Completely Routed |
| Design Strategy: | Xilinx Default (unlocked) | Timing Constraints: | All Constraints Met |
| Environment: | System Settings | Final Timing Score: | 0 (Timing Report) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Device Utilization Summary | | | | | [-] |
| Slice Logic Utilization | Used | Available | Utilization | Note(s | ) |
| Number of Slice Registers | 202 | 54,576 | 1% |  |  |
| Number used as Flip Flops | 202 |  |  |  |  |
| Number used as Latches | 0 |  |  |  |  |
| Number used as Latch-thrus | 0 |  |  |  |  |
| Number used as AND/OR logics | 0 |  |  |  |  |
| Number of Slice LUTs | 843 | 27,288 | 3% |  |  |
| Number used as logic | 827 | 27,288 | 3% |  |  |
| Number using O6 output only | 651 |  |  |  |  |
| Number using O5 output only | 24 |  |  |  |  |
| Number using O5 and O6 | 152 |  |  |  |  |
| Number used as ROM | 0 |  |  |  |  |
| Number used as Memory | 12 | 6,408 | 1% |  |  |
| Number used as Dual Port RAM | 12 |  |  |  |  |
| Number using O6 output only | 0 |  |  |  |  |
| Number using O5 output only | 0 |  |  |  |  |
| Number using O5 and O6 | 12 |  |  |  |  |
| Number used as Single Port RAM | 0 |  |  |  |  |
| Number used as Shift Register | 0 |  |  |  |  |
| Number used exclusively as route-thrus | 4 |  |  |  |  |
| Number with same-slice register load | 0 |  |  |  |  |
| Number with same-slice carry load | 4 |  |  |  |  |
| Number with other load | 0 |  |  |  |  |
| Number of occupied Slices | 295 | 6,822 | 4% |  |  |
| Number of MUXCYs used | 336 | 13,644 | 2% |  |  |
| Number of LUT Flip Flop pairs used | 881 |  |  |  |  |
| Number with an unused Flip Flop | 693 | 881 | 78% |  |  |
| Number with an unused LUT | 38 | 881 | 4% |  |  |
| Number of fully used LUT-FF pairs | 150 | 881 | 17% |  |  |
| Number of unique control sets | 19 |  |  |  |  |
| Number of slice register sites lost to control set restrictions | 46 | 54,576 | 1% |  |  |
| Number of bonded IOBs | 20 | 218 | 9% |  |  |
| Number of LOCed IOBs | 11 | 20 | 55% |  |  |
| Number of RAMB16BWERs | 68 | 116 | 58% |  |  |
| Number of RAMB8BWERs | 0 | 232 | 0% |  | |
| Number of BUFIO2/BUFIO2\_2CLKs | 0 | 32 | 0% |  | |
| Number of BUFIO2FB/BUFIO2FB\_2CLKs | 0 | 32 | 0% |  | |
| Number of BUFG/BUFGMUXs | 2 | 16 | 12% |  | |
| Number used as BUFGs | 2 |  |  |  | |
| Number used as BUFGMUX | 0 |  |  |  | |
| Number of DCM/DCM\_CLKGENs | 0 | 8 | 0% |  | |
| Number of ILOGIC2/ISERDES2s | 0 | 376 | 0% |  | |
| Number of IODELAY2/IODRP2/IODRP2\_MCBs | 0 | 376 | 0% |  | |
| Number of OLOGIC2/OSERDES2s | 0 | 376 | 0% |  | |
| Number of BSCANs | 0 | 4 | 0% |  | |
| Number of BUFHs | 0 | 256 | 0% |  | |
| Number of BUFPLLs | 0 | 8 | 0% |  | |
| Number of BUFPLL\_MCBs | 0 | 4 | 0% |  | |
| Number of DSP48A1s | 1 | 58 | 1% |  | |
| Number of ICAPs | 0 | 1 | 0% |  | |
| Number of MCBs | 0 | 2 | 0% |  | |
| Number of PCILOGICSEs | 0 | 2 | 0% |  | |
| Number of PLL\_ADVs | 0 | 4 | 0% |  | |
| Number of PMVs | 0 | 1 | 0% |  | |
| Number of STARTUPs | 0 | 1 | 0% |  | |
| Number of SUSPEND\_SYNCs | 0 | 1 | 0% |  | |
| Average Fanout of Non-Clock Nets | 5.35 |  |  |  | |

## 4.2 Timing Summary

Speed Grade: -2

Minimum period: 119.206ns (Maximum Frequency: 8.389MHz)

Minimum input arrival time before clock: 5.158ns

Maximum output required time after clock: 4.240ns

Maximum combinational path delay: No path found

## 4.3 Technical Aspects

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Value** | **Units** |
| Input clock speed | 100 | MHz |
| Processor clock speed | 6.25 | MHz |
| Average clock cycles per instruction | 1 | Clock cycles |
| Instruction memory size | 16 x 4096 | Bits |
| Data memory size | 8 x 131072 | Bits |
| Register width | 16 | Bits |
| I/O method | UART | - |
| UART speed | 9600 | Bauds |
| UART flow control | None | - |
| UART parity | None | - |
| UART stop bits | 1 | - |
| UART no of data bits | 8 | Bits |
| USB-UART bridge model | XR21V1410 | - |

Additional software required for communication:

* XR21V1410 USB-UART bridge driver – can be downloaded from [www.exar.com](http://www.exar.com).
* A serial communication terminal (Eg: Putty)

**Typical instruction cycle:**

Previous instruction (N-1) is executed. Program Counter increments to N.

Nth Instruction is loaded into Instruction Register.

Nth instruction is executed. Program Counter increments to N+1.

Clock signal (6.25MHz)

# 5.0 Simulation

Simulations of the processor were carried out using the ISim simulation tool which is packaged up with Xilinx ISE 14.7.

Simulation result of executing the program given below is shown in the diagram:

nop

move zr ac

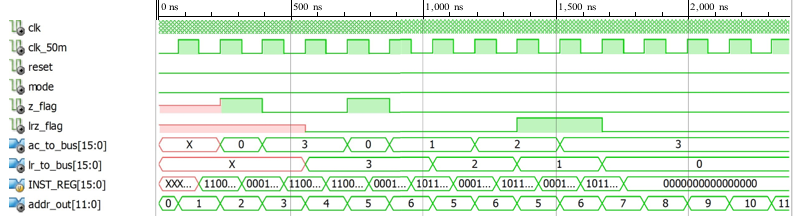
add zr 3

move ac lr

move zr ac

add zr 1

jmpdec 5



Clock running at 6.25MHz

1.Program Counter updates

2.Instruction Register updates

3.Instruction executed

z\_flag goes high when AC is zero

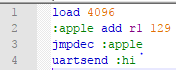
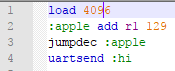
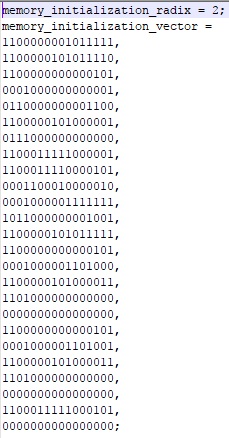
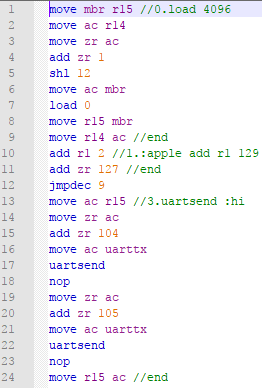
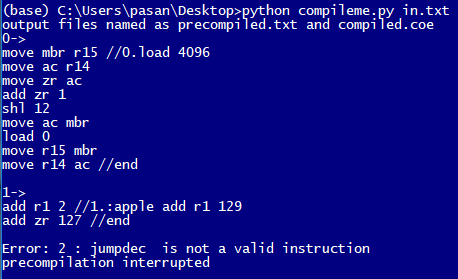
Loop Register is decremented and Program Counter jumps

Clock running at 100MHz

# 6.0 Supplementary Tools

## 6.1 Compiler

An assembly-to-machine code compiler (compileme.py) was written using Python (version 3) in order to convert human readable instructions into machine code. This compiler includes features such as label resolution, immediate addressing resolution for LOAD and STORE instructions, string resolution for UARTSEND and immediate constant resolution for ALU instructions. A file including instructions written in assembly is given as the input. Output is two files, one in assembly itself but with resolved labels, addresses, strings and constants, and the other includes instructions in machine code. An automatic commenting and error detection mechanism is also included. A compilation example is given below.



Assembly code with incorrect instruction

Assembly code with corrected instruction

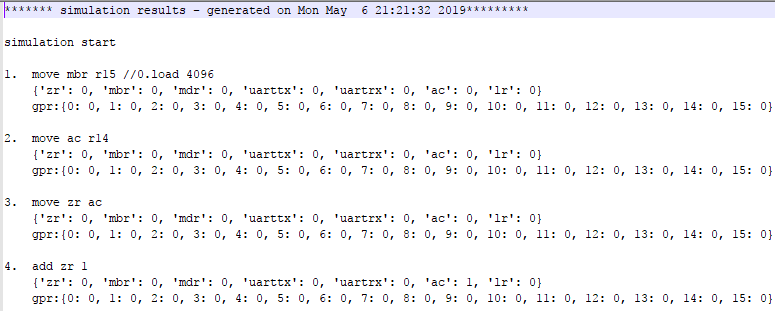
Compiler reporting about the incorrect instruction

Output file 1: in assembly (with resolved labels, addresses, strings and constants)

Output file 2: in machine code

## 6.2 Simulator

A simulator (tryme.py) was written in Python (version 3) for the purpose of testing and verifying the algorithms. This was based on the compiler mentioned above. The processor architecture was implemented using a dictionary resembling the register stack. Memory was also implemented using a dictionary. Values of these dictionaries were written into an output file by the simulator after the execution of each instruction. This allowed to investigate the values of each register and memory location through each step of the algorithm. Following image shows a part of an output file generated.



# 7.0 Down-sampling algorithm

The down-sampling algorithm consists mainly two stages. If we directly down sample the original image, we might not be able to obtain an image equal to the original image because it will skip some pixels which contribute for the details in the image by a considerable amount. Therefore, we first filter the image using a suitable filter in order to smooth the image. After that we pick the relevant pixels to make the resulting image.

## 7.1 Filtering

In this we use a 3\*3 kernel with the weights of each position as given below.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

Applying the kernel for each and every pixel is not needed as we are going to skip one pixel after the other each time. Therefore, we can skip the even numbered rows and columns in both directions when applying the kernel. Let the pixels of the image have values as shown below.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | . | . | . |  |  |
| 0 | aa | ab | ac | ad | ae | . | . | . |  |  |
| 1 | ba | bb | bc | bd | be |  |  |  |  |  |
| 2 | ca | cb | cc | cd | ce |  |  |  |  |  |
| 3 | da | db | dc | dd | de |  |  |  |  |  |
| 4 | ea | eb | ec | ed | ef |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

If we apply the above-mentioned kernel to the (1,1) cell and average by dividing by the sum of the kernel we obtain the result as shown below.

After obtaining the result we are going to replace the value of (1,1) cell with the result ‘r’. We do this averaging only for the cells with both coordinates are odd numbers as we pick only those at the down sampling stage.

We store the original values of each pixel that came through UART in the data memory as a stack of pixel values.

|  |  |
| --- | --- |
| 0 | aa |
| 1 | ab |
| 2 | ac |
|  | … |
| 256 | ba |
| 257 | bb |
|  | … |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 65535 |  |

|  |
| --- |
| 1 |
| 2 |
| 1 |

Applying of above calculation for the values in the stack is little bit difficult and the code may be longer. Therefore, we implement the same thing by applying two kernels of 1\*3 and 3\*1 in the rows and columns respectively.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 1 |

By applying the first filter along each row for the odd numbered pixels and replacing the middle cell we obtain the grid as below.

…

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 |  | 4 | . | . | . |  |  |
| 0 | aa |  | ac |  | ae | . | . | . |  |  |
| 1 | ba |  | bc |  | be |  |  |  |  |  |
| 2 | ca |  | cc |  | ce |  |  |  |  |  |
| 3 | da |  | dc |  | de |  |  |  |  |  |
| 4 | ea |  | ec |  | ef |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

After that we apply the vertical filter for the modified columns (odd numbered columns).

|  |
| --- |
| 1 |
| 2 |
| 1 |

Therefore, we can see the result obtained by this method is same as the previously obtained result.

Since the image contain an even number of rows and columns, we are unable to do this calculation for the last column and last row. For that issue we add an extra row and a column at the end containing values as the same as of the last row and last column.

## 7.2 Down-sampling

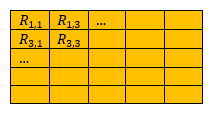
After replacing the pixels with obtained results of the image will be as shown below.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 |  | 4 | 5 | . | . | . |  |  |
| 0 |  |  |  |  |  |  | . | . | . |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  | . |
| . | . |  |  |  |  |  |  |  |  |  | . |
| . | . |  |  |  |  |  |  |  |  |  | . |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | . | . | . |  |  |

In the data memory stack, the result will be as shown below.

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
|  | … |
| 253 |  |
| 254 |  |
| 255 |  |
| 256 |  |
| 257 |  |
| 258 |  |
| … |  |
| 509 |  |
| 510 |  |
| 511 |  |
| 512 |  |
| 513 |  |
| 514 |  |
| 515 |  |
| 516 |  |
| … |  |
|  |  |
|  |  |
|  |  |
|  |  |
| 65535 |  |

Now we have to select these cells and store them separately in the correct order. The initial image stored in the address range from 0 to 65535. Therefore, we store the result from 65537 to 81920.



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 |  | 4 | 5 | . | . | . |  |  |
| 0 |  |  |  |  |  |  | . | . | . |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| . | . |  |  |  |  |  |  |  |  |  | . |
| . | . |  |  |  |  |  |  |  |  |  | . |
| . | . |  |  |  |  |  |  |  |  |  | . |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | . | . | . |  |  |

# 8.0 Results and Verification

## 8.1 Transmitting and receiving down-sampled image

For the transmitting and receiving process at the PC end, *Pycharm* IDE was used. Reasons for using Pycharm was, image processing sections and serial communication process could be easily implemented with it.

The ***Pyserial Library***is included for serial communication with COM ports and FPGA. Since the data which should be transmitted and received was integers between [0-255], ASCII values could not be used in the code (ASCII values are only up to 127 starting from 0). A built-in library in Python could address this issue. By the method ***pack*** included in ***struct class***, integers could be encoded (similar to Unsigned char in C) and sent and received through COM ports.

![A screenshot of a social media post

Description generated with very high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAkACQAAD/4RDiRXhpZgAATU0AKgAAAAgABAE7AAIAAAAIAAAISodpAAQAAAABAAAIUpydAAEAAAAQAAAQyuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFRoaWxpbmEAAAWQAwACAAAAFAAAEKCQBAACAAAAFAAAELSSkQACAAAAAzUzAACSkgACAAAAAzUzAADqHAAHAAAIDAAACJQAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE5OjA1OjA4IDExOjI2OjI0ADIwMTk6MDU6MDggMTE6MjY6MjQAAABUAGgAaQBsAGkAbgBhAAAA/+ELGmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTktMDUtMDhUMTE6MjY6MjQuNTI3PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPlRoaWxpbmE8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgC8QKUAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+jo5opgTFIjgddrA4p9UNPRUuLxUUKBJwAMdq4fwV4h1jUfibrdl4h0e/wBHumsYbhbSXV0u7dEDFAY0RAELck/Mc46VrTpuak10VxN2Vz0eivLNS+MWo6Prur2WoeCro2+k3cUE9xb3qSEpLnyWCYGWfK4QEkbjkjHNe8+NWradprXd94FuENrqX9nahDHqMbvbyEIU2AL+8LBjwMDI685raODrytZb+a628/Nfeu4OSW/9b/5M9borgdJ+I2qL4gstL8beEp/DR1LK2Fw16lzHK4GSjFAPLbHQHrz6VQb4t6ncW8+s6L4H1DUfC1s7CTVVuo0kdEOHkjgPzOo5wcjODnGDifqtW9rfirffewXR6bRXAeLPie/h3S9F1uw0RtU0DUzFv1BLoIYPMI2/utpZzgk4Hpir+meK/FOpabqV23gS6tPLiEumxXN/Cj3gPRXXrA2MHDZxyM8cr6vUUedrT1X+f9bhzK52FFeSfCnxZ431HSbGHUPCs11p8l1cLLrU+sxuyDzXyPLI3kKfkHPQZHFet1NajKjPlbXyaf5AncKK5rxh4n1DwydMax0Yaqt/diz2JdeXIsjKSmFKFSvynJLLtHPNGm+IdYurrVtMvdHtLbV7CGOeKNL9pLe4STcFPm+UGX5o3BHlnGARnNYX0uM6WiuPXxrfzeENA8Q2+kW7W2ptbLcxvfMr2/nyJGuzEREmC+Tkp09+M/UviHrWl6lq8MnhaGe20q6gt5JodTG6Uz7BFsRox82XXcCQFzwzdKqzvbzt/X3gegUVw1/491XTtKe5m8OxT3FtqsemXdtbX5c7pPL2NCWiUSZEq8N5eDnnHNa1l4g1eHVrSx8TaNbaf9vJS1ms783KGQIzmN90cZVtqsRgMDtOSDjKWu39aX/UHpudHRXEr48vofDviW/1TSbKzutBlaM2h1B280AAqxbyQVDggphWznHByB1WkXN7eaPa3Gq2SWF5LGHltUm80RE/w79q5I78dc9etH/A/ENv67Fyiq9/PNa6dcT2tsbuaKJnjgV1QysBkLubAGTxk8CvN/CfxkfXvGFloGqaLZ2Ml+sn2eWx1y31Ab0XcVcRfcBAOCepGPUjanRqVU3Bbb7CbUVdnqFFeX+LvjFN4e8VXmi6boFvfNYqnnSXutQWDOzLuAjST5pBgjkd8jHQnv8Aw/qVzrHh6x1C+0+XTLi5hWSSzmzvgYjlTkA5HuB9KJ0KkIKpJaPzX5bhdXsaNFRXTXCWsrWUUctwEJijlkMaM3YFgrED3wfoa4bSPiHq+oaPpmt3fhhLTR765itWl+375o2kcRhxH5YDR+YQNxYNj5tuMZx3dh9LnfUVx8vivxDJ4u1HQNM8N2lxJZxxTi6m1NooTFJvC7iIWZZCUOFAYYySwwAdzQdZbWLacXNo1lfWc32e7tjIJBHJtVvlYfeUq6sDgHBGQpyALUHoalFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBRSyuVsrkJciK7uFbEoQMImIwCFPXHoetedQ/DHxxD4kk10fE3OoTQLbSy/2BB80SsWC434HJPIGa9ToralWnSvy217pP80xNXVmefTfDnWbh9ckm8WK8upajbX9u50qP/RWhYFV+9+8+VEXnGNucZJrifil4W1fS7i9+wf27q1trurQ6i0Gj6ZuaxMW0M3mhid5XG3gDK17vRW1LF1Kc1Le3Sy8vL+6vu82Dimn/AF3/AM2ea+E/BWraxcaT4i8a+I9T1f7KvnWOnXunR2JtnIKlpY1J3OAeM9OeuaY/wm1i2gn0fRPHmoab4YuJGLaWtpHJJGjHLxxzk7kU84GDjPOcnPptFS8XV5rq1u1lb7rW+e4lGx554t+GWo65p+kaX4f8T/2FpOlLCYbP+zkuMyRHKSbywPHHHQ45rqvDGma3pWlvB4k8Qf29dGUstz9iS22pgYTahIOCCc+9bNFZyr1JQ5Jbeiv9+4cqPONN+GOt6Xd/ZbTx5qFv4dW7e6TTLa0jjlG5zJs+0fe27jyMcjI716PRRSqVZ1Lc35L+mO2tznvFOganrlxpMmm6naWK6deC7K3Fk05lYKVAyJU2jDN69umOXWuhanD43vtbn1K0ktLq1jthaJZMsiLGzsp83zSCcytn5Bnjpg536Ky6W/rUb1/rtqeft8PdeHh+z0O38V28On6bLDLZINKyzeVKskazsZf3igLj5BGScHPUGS88B61ftrjXHiCwDapc2t0mzSnAhe3eMpnM53KRGAR8pycgjpXeUUXf9f15AeX+NtG1HT7CcNeXdwNU1qyvQdJ0iaSa0aMwiWTKmQYAiDKCvXI+eun0jStV1O603VfEGrQXsVqDNaQQ6VJYsJGQpvkWSR23BWYBcLjccgnGOpooWit/W1v0G3dnl2qw2fif4taO8FnrcNmkRbUGl0m5it7uSF91sjs8QHyszuGJ2nAGTkV6jRRQtI2E9Xcx/Fvh2PxZ4T1DQ57mW1S8i2edF95TkEcdxkcjuMjvXnFh8ENU07WNN1a18XWVtf6fMzpLaeGrW3DIy7WUhMZJBYZYnGeADzXr9FdFLE1aKcYPR+Sf5iklJWZ5l4l+F/iPxBqd+/8Awm8f9m3Vz9oi0/UNDhvktjt24QysdvGfugdT6k13PhnQYPC/hmx0W0nnuIbKIRpJOwLsM55xgd8AdhgVqUUp16k4Km9l5L+mFlfmIrpbh7WVbKWOK4KERSSxmRFbsSoZSR7ZH1FcXH4G1yHwHp/hyHX9P3WNxFKLltKch1ikWVF2faBg7l5OeQcAA813NFYDPMy+sP8AFjU30q8vLCWW0tbXzLrw/dTWNyyecXO4FApXK7W8zadxHzHGO40HR30m3uWu7oXl9ezm4u7hYvLV3KhQFTJ2qFRVAyTheSTknUooWgPVhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcB4M8bzeJfHus2U0Wt2EcdrFNDpuq6ZHbGAZ2lw4cu+488gAY4rurr7R9jm+xeX9p8tvK83OzfjjdjnGcZxXlNv4a+LcPjSfxL5ngs3VxaJZyR7rvyxGrFgQMZ3ZJ749q68PGElJSaWmlyZX5dDWvPjX4e03Wb/TtT0zXLRtOuBBczyWOY4w2dshKsSEbA2nGW3DAPOIJ/jr4etNOS+vdG8Q20Au2s7ky2AX7HINp/e/N8uQ2QBknB44pl54H8W38/iS4uV8OtLqGqWd7afNPhkt2XCTcf3UU4X+Jm5xjHD/ABXiu/ClxqMeqarp9nba7rtvqVg8cMs00Xl7Vkd1K7CFG07ckn3rso0cNUlGHV26+Ub9PN/cKV9Wv63/AOB956l4e+KOk67rkWk3Ol61oV5coXtE1my+zi7A5byzk7sDB7dapXXxo8O213P5On65eaXay+Tc61aae0llCwOG3SZzgZGSAeoxnNZ3h3RvE/j280HxF4t1XRJ9HsT9rsI9HgmQ3TlSoeQygMuOuB1zyOBUMPgL4gaNolx4R8PazoI8Myl447m7tpDeW8Mhy6hR+7cjLYLdc9uMQ6WHU7S37Xdt3fW2+2n/AAwJt69P62/r79zqfE/xL0fwqulz3trqFzp+p7PL1K1hVrWIMRtMkhYBcg578A0aZ8R7PWNN1K+0/QPEUlvZxCaBzpjKNQQ9Gt8n58jBAO04I46457xl8P8AxRd+HdC8N+EZND/sjSRbyE6q03mySwnjOwFShGMjr16V3Hhj/hJv7Lf/AITP+yft/mnZ/ZPm+V5eBjPmc7s59ulYyjQjS5o6u769OmgXldX8vv6nCfDX4pX/AIgsrKz1fw/4kuLq4uZkbVP7MVLRFEj7QzqQBtUBTx1B69a9UrzHQfCPxE0Bl0TT9a0K28PpeyTpdrbySXpjaUyFCrfu+SSpPYHI5r06oxXs+fmp2s+39aeg1e7TM3Wtbg0WCFpIbi6uLiQRW1paoGlnfBOFyQBgAksxCgDJIFZVt40efUp9Nl8Na1bajFatdpazLb5mQMq/JIspjJy3TeMY5xkZs+I9Hvby70zVdGa3/tHS5XaOK6ZlinR12vGzKCUJGCGAbBXoQTU2kyeJZ755NcttJsrVY8JBZ3ElzI75+8ZGSMKAONuxs5zuGMHk/r+vn/XUp7f1/X9fIw/+FlQnwrY68nhvWmt766W1hgH2Xzt7MEXKmfjLkrjO4FTkADNaGrW2t37tcnXH8NaZDaCUmJIHnEnJfzWlV41RAB93qdx3AAZyrPwPfQeO/tc1xbN4dt7qXUrO0UnzEvJUCsSNu3YMysOc7pD6CpPEujeJ9W8RKwstJ1LQYAjw2NzqMtt5kw5LzKsEgkAONq5Cg8kE7Spul5/h/Wv4bD2f9d/8rfibnhG/v9T8J6feauE+1yx5do0KLIMkLIFPK7lAbB6bsVs1S0ltTbTI212O0jvizmRLN2eJRuO0BmAJO3bk4GTngCrtN7krRHBav8WtP0LU57fVfDfie2srecQy6rJphW0XLBd+8tnZk9cc9ga2PFnjqw8JPZQS2Opare3xYwWWlW3nzMijLPtyBtGR37/XHkOt/AXxPrE2oveTeHr66mZmi1i6nvftcp3fLvUMYl+X5eFYAAADuOy8X+DvHHiew0qO4t/B18YrOSK9tdQhlaISscb4XC+ah2gDIZefXjHpujhfctL1+7T8fT5Cu+Z9jrfB3jW08aW99JaadqemyWM/2ee31K3EMittDfdDHsR15rpK4n4Y+D9X8G6Lc2usXlsySyhrewsnmkt7JQMFY2mYuQx+Yg8A5xXbVxYhU1Uap7f1/X+YRvbU5++8WGDULm00vQtU1p7PAuXsRCqQsRuCbpZE3NggkJuIyM4yM58vxFtxpmlahaaBrN3bapctawsiQxmOUOyBZFllQoSUOMj0BwTipY9M8RaBqepN4ftdL1Gx1C5N5svb2S1kgkYAOoKxSB1JXcPukZI5GMWtV0LUPEHgefTNYuLVdTmj3pPaRsscEyndGy7ixOxgvPfGcDOBh0T9P+Cv8i+tv68n/mLZ+K/tnjG88PJoupJJZxrJNeMYDAFfdsPEpf5tjYG3PcgDmugrnfBuhX+kWN3da/Jbza1qdy1zfSWpYxA4CIiFgDtVFQcjrk966KntoTvqFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRXN6F4ytPEPiXUtO06TTrm0tIo3iu7TU4bhpi33gYkJZNp4yetVGEpJtdA2OkorKj8VeH5r1rOHXdMkull8hoFvIy4k5+QrnO75W468H0qNfGPhl0tnTxHpLLdSGK3YX0REzggFV+b5iCRwPUU/Zz7Bc2aKzdL8R6JrkkqaJrGn6i8IBlW0uklMeemdpOOh60T+JNDtdXTSrrWdPh1GQqEs5LpFmYt0AQncc9uOaOSV7W1A0qKz7/X9H0q7t7XU9VsbK4uTiCG4uUjeU5xhVJBbkgcU0eI9DZb5l1nTyunHF6RdJi1OSMSc/J0PXHQ0uWVr2A0qK47wd8TNA8W6fbv/aOm2moXE0sSaf8A2hHJKdrsoIXgncF3Djoe/Wuxpzpzpy5ZqzFdMKKr32oWel2Ml7qd3BZ2sQzJPcSCNEGccscAckCoNL17SNcQvouq2OoqqhibS5SUAEkA/KTxlWH1U+lQMv0VkWnizw5qGqtplhr+l3OoKzKbSG9jeUFfvDYDnIwc8cYqxPr2kWuqLpt1qtjDfvGZltZLlFlZACSwQnO0BSc4xwfSgC/RWZD4l0K40aTV7fWtOl0yIkSXqXSNCmOuXB2jqO9M8M6o+saIt5JfaXf75pQk+lTebAyCRgnzZOWC7d3bdnHFAGtRWRH4s8Oy6ydIi1/S31MOYzZLexmYMOq7M7sjB4x2qHxfrWpeHvD91q2m6daX8dlBJcXCXF61uQiKW+TET7icHg7R70m7K40m3Y3aKjtpvtFrFNt2+YgfGc4yM1JVNWdmSmmroKKKKQwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAiup/s1nNP5Uk3lRs/lxLud8DOFHcnsK8O0HxFFoXxMvta0r4Z+KtP0u706O1W3tdAEREokLM7KpC4II5znivdqK6KNZUr3V7+YpK6seA674V0/WtU8Y37+DNZjZtXsQm3StrSwB1Wd4MHkuVkLMMZVlJPJFY3jXw5bWFv4h0xPCEqR/8JLbSae32UW8MsUiqjQwzEfLuK87eBwe1fS9c54r8AeGfG7Wp8Uab9uNoGEP7+SPZuxn7jDP3R1rso49xlHmvZW/Dl9P5fxYpRvf+u/8An+B5xpGmrrvjzQLzwd4C1HwYmmM7ahe3dktmJo2Ujygg/wBdkgfMemPesFdAXTvDGpeF9d+F+p614nuppQNbjtlliuHkbKTfazzEOQSOgxzjLY980jSbLQtIttL0qHyLO1jEcMW9m2qO2WJJ/E1cqHjWnZLT1d9297+e23z1EotanhvxK028j8IeFtEvvDms67rdktrJcatplgZmjRDiVFmHzK5xkdM8Emu78DaR4Vv/AA5N/Z3gWTQoGH2Sa21bTEimuEAB+fO4yKc9WJyQc129FZTxTlT5LW1b37+QKNmrdNPuPDvBEK+GJINGHwqvLnxBbX8wXVW0+KK3EZlZg63R/uxngDrgKK9xoorOvW9s7ta/MpKzbOW8dGzNlpyX0Gs5F6ssF5o9kbqSzkVWIkZAj8EZXlGGWHA6jA0vT9U1O28WR6S90W1CxEcGtanpQsLyS42MuxgIoy6INhDlBjcQC2Pl9Iormto0VezTX9a3PJEshcaXY6VdXfjgT2kkQi0xNCtES3kjYBdlwLQQqFxkOsoGOh7Vs/EGTTb7xN4ds9U0rVL+zsrx7i8EGk3VxDtMEgTLRxlXG8qCuT7jGa9Coqm7ij7uxkDVtMs9JfWY7O9SK6cPIItKnNxI2AgZ4Vj8zOFAyV6AdsVzvw71eMw31hLY6tbTz6pfXUf2rSbqBDG87up3vGFGVIOCc9sZruaKS3B6q39df8zxnxHda7q/hy5i1J/FU+qfaN8uk2mhJ9kgCS7gVlaAmQAKCDHKzseVA6DtfH+swv4D1Oygs9VuLnVNLnW2jt9JuZSWeMhVfbGfLOSOH2kd67Gik1ePKUnaXMZHhvV7fVdLjFvDfQmBER1vLCe1OdvYSou76jIrXooqm7u5EVyqwUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBXtLoXUZypSRDtdD2NVrbWPtOvX2l/wBnX8P2NI3+1ywbbefeM4jfPzEdxjikjuYrManc3LbIYSZJGwTtULknA56CvCPB3jT4X+D/AIpalc+H7/7Hot1psUaN5NzJvuPMJYYZSw42+1deHw7rKVk20tLK/wDXkTJ8sbn0TRXzXrGleFLjxB4z1bSdWuLF7PXLFDLA86G0MjBLiUf3mLNMo64wcAAg1l+J9G8PaZBrej2C3y3eh+JIfsMVrJK81vBME3eVk4Llk4yScgetdMMAp2XM9bdO/L5/3l+PzUpWv/Xf/I+qKK8D8JnwqPH+i/8AClf7Q5LDXdwn+zeTtOPN83pJu+7t4+9WBK3hQaff/wDCbP4gHxQE8ghMf2j7QJt37n7PsPl+X0255xnHG2l9R96136W977r7ed+wc39f59j6bor59+LN5pcHhvwcfF921p41gFpK86JIzQx7h5r4AMZIZc4IJyOBiu58C+Gfh3r/AIa1WXwzJdataan/AKLql1c3NyHvHADFnDlfmO7O5QPvHHpWUsKoUvaNvdrRaadb32/XTzDm1S72O08NeIrTxToaarp8c0cDyyxBZ1AbMcjIeASMZU456Vq185/DyP4ceHdQtILvRtQm8cWeozRLaW0Vy8qDzWCttyI9qxkEk9gTya+jKjFUY0p+7ez2urfdq7rzGm7tMKK474kyaK2i2Vnr+tWWkx3F4vlNqMXmWk7IrN5c67lDIQCcMyjcF5PQ43w8stEl1PXrTT7XQtSsp7eFJr7RY2WwlU7wbfyTJJGrDJZgjYYSgsAT83ItblPRf13O/wBP1O01WGWWwm81IZ5LdztK4kjYo45A6MCM9D2q3XknhKH4e+Hry9t9Q03RINftdauvs9tHYI98imZmiMcaKZNoQqQVGAvPAFWfiRJ4Pk8deG49Y1bTdNvFnkS9nF3Hb3C25t5SI3kyHWNiQOozu4IJo6J9/wDIdvea7X/M9SorzbxVovhWD4bnUfDVlpradDqNrqMctkqvDCEniEsse3KoAiNuKY/iJ6mlt9f8IX3xk0y60S602W4u9Ou4Zr+2C7buQNbFIhMOJXVQTtBJUdhmmtXb+tr/APAF0uekVUvdTtNPms4ryby3vp/s9uNpO+Taz44HHyoxycDivMYR4CmsL0fEaXSoPE3nzC6kv5FS8Qhz5Zty37wJt2mPyuPT5s1a8Y+H/C0OleDL/VdNtJdOsbqC2nvNWs1VktfIlVBMZFBVd5TIYABiMgGkunqvxG9L+j/A9PorA8JXPheWxng8FQ2cdhFJlm0628u2kcjko6qEkPGCVJwRg4Nb9MQUUUUgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCG3tIbRWFumwMcnkn+dTUUUAFcp438AWfjtbJNQ1fWLBLN/MRNOuRErPkFWYFTllI4PbJrq6KuE5U5KUXZoCnpGnf2TpFtYfbLu9+zxhPtN7L5k0nu7YGT71cooqW23diSsrBRRRSGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVT1d7uLR7qTTpYYrqOIvG88RlQEDPKhlJ/wC+hXFWPiXxj/whFt4w1NdFFj9iW+uNNhglEvk+XuZlnMhXdjLBDHj+Hd/HQtW/63/4Ydmeg0VzFxqmu6tr95p/huXT7GDTwguLu+t3uDLI6hwiRrJHgBWBLFjycbeM0WutazqHhvUwosrHWtLma3nZomuLdnVFk3KoZGwyOpwSCpOPmxkpuybfQEuZq3U6eiuP8Iar4u8QaZpGtajDpNnY3sCSSWaq7TBWjyJRIHKjLY/dlThTkvn5azPD48a3HinxNC3iLSmS1voUKy6VM6gG3ibEY+1fuxzyOctk98VVmpcr/rUm+lz0OiisrxHrH9haJLf+bpsXlso3apffY4Bk45l2Pg+gxyeKV7D3NWivPLT4nfabyGD7f4EPmyKmIfF29zk4+Vfsw3H0GRmvQ6fS4BRWP4o0Q+INENipQEzxSHf0wsisR0PYEVm3FjN4X03xXrFuUaa4L3cOOSNsQxnI/vBj34o0s3/XT/g/cZSnKMrW07/f/kvv8jqqK87tfhppWseFYbp7q4OsX1sskmqNO8jMXAJBXcAVwduPSpINCs/GPirWIPEckt1b6S0dtbWXmsiqCgJlO0gktzz6fhjRwSbV9tznWIqtJ8nxba+Teummi6XPQKK8U1yxnl26Lb3cly+la1HbaddSPuaNZIy3lk99rKo9sY9qvXl+vjLxb4a1CZT9jhuI7douQPPKGSTn/ZIQfhVKle2v9Pb8zB4+1046+vVNp9Olvmep32p2GlxrJqd9bWaOdqtcTLGGPoCSKsRSpNEksLrJG6hkdDkMD0IPcV5JcjVtS8SeItafRdN1W30ud4GXUSWPlJyUhX7qtgFixB+9x3B7qfTIPENn4bvrCOOC1tZo7xIiu3anlnCgAYBBK8dOKnkSSb8vxNoYmU5SSjte3nZ2Z0dFY/hjQzoGlS2jFWL3U02VJPDOSvUdduAa2KzZ2QbcbyVgooopFBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBS1lpl0S8+y2kt5MYWVIIWQO5IxgF2Ve/ciuF8GeCbw+EbHTNbn8QadaW0UUVxo9zPaSQTFVXeFeMPJ5RYH5DIOMrtC/LXo9FC0bfp+APVJGFqXhcXerPqWm6xqGi3c0YjuHsRCwuAv3d6yxuuVyQGABwcEkAAIvhSODw7JpWn6pf2bzyNLc30flPcXDt99nMkbLlvZRgABdoAFb1FKytYd3e5heGvDUvhu1htF1/U9Rs7e3W3gt7xLcLEqgBcGOJGJAGOSfz5qC+8Gpc6ve31nrmraYNQ2G8t7KSJUnKrs3bmjMiEqAuY2U8AjB5rpKKpu7uxLTYZDEtvbxwoXKxqFUyOXYgDHLMSSfckk0+iikAUUUUAFNdFljZJFDowIZWGQQe1OooA48/DTSuYE1LWI9OLZOmJfMLYjOSu3GcE+9XtY8Fafqt8l9Dc32l3ixiJrjTrjyXkQdFbg5A/PgeldFRV88u5z/VqKTXKtTkbn4baRNYWNpa3mpWEdk5lVrW4CtJKcfvXJBy/HB4xnA4p958PdOury0uI9Q1K0+yFWSO1mWNGkHWQjb99s8t1NdXRR7SXcTwtB/ZOY1bwFpurancXjXmo2guwou7e0ufLiuscfOuOeOOCPz5ro7e3itLaK3tkEcMSBEReiqBgCpKKnmdrGsaUIycorVhRRRSNAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKWsrM2iXn2W7ls5hCzJPCqF0IGcgOrL27g1x/hv4im48JaTqWr6Xq32SS1h+1a00EaW/mFF3uUDiQJvJG8R7O+dvzV2Orpdy6PdR6dFDLdSRFI0nlMSEkY5YKxH/fJrirLwz4xPgi38H6lJoxsfsSWU+pwzSed5PlhWVYDHtLYyocuB/Fs/goju7+X63G7WXz/Q6vV9ZvdMmVLTw5qmrIU3NLZSWwVDn7pEsyMT34BHPWqd340sbbw7Ya3Ba3l3YXs8UO+JFRoDI4jBdZGVgA5CkAEj0rE8Q+CrvUfEl3fHQ/D3iCGeGOO2/tx3P9n7QQQkflurKThjgxkngnABBaeDdYT4WDw3N/ZsF9aTpLatbu/wBnk8udZkBG0GMErtKjdtHQt0oXn3/DqLW50w18L4sXQZtOvIXkt3uILtjEYZlQxhwuHLggyqPmUd8ZrlfFfjuY+ENVvtCsddgtYIJHg1y2toHi3IpIIjkLSNGWG0v5RXGWB2/NV6PS/Ft740sNY1a30WKyhtbizktLe7ld41kMTeYJDEokJMeNm1MDnc2cCld+HPF8Pgm58IaQui/Yvsb2NrqNxcyiVYSm1Q0AjILBfl3CTGfm2/wVLvy+f/B/yt/SKjbm1O10qeS50aynmbdJLbxu7YxklQSaXUtRh0qxa7ukuXjUgFbW1luH5OOEjVmP4DiotDhvLbQbKDU44IrqGFY5Ft5TKmVGOGKqTwM/dH9al1LTodVsWtLp7lI2IJa1upbd+Dnh42Vh+B5rSpbmfL3M4X5VzGPF430qaZIktNeDOwUF/D1+qgn1YwgAe54roq52LwRpUMySpd68WRgwD+Ib9lJHqpmII9jxXRUuhRheLdRubDTLZbO6Sxa8vIrVrx1DC3DnG4BuNxOFXORuYZB6GroerPB4tvvDVzrP9rzQ2yXSySLGs0WTho5PLCr3Rh8oOH5zwTsa1DcXGlSwWun2WomUbJLa/mMcToeoJEb5+m3msnwv4Y/sm5mvp7LT9PlkTyYbHTBi3to9xY4O1dzMxyzbV6KMcEt6FOVL6rJTWvTa7en/AG8rfd89HEr30MG38XR69qkjN4mu9JtxcCCCKxsVdFy2F+0XEkTxq7/KVQFCAyg7iRXQahPf6r4pk0Ww1ObSorO0jup5raKN5ZTI0iqgMisqqPLYn5STlcEYOcmDwjq1ho954Y09dOXQ7qeVhctK6zwxSsWePywmHPLAOZBwRkHbg7epadqlt4g/tnQYbO5lntltbm3u52gDKjMyOrqj8gu4I287hyNuD1Vnh+f9zbZ8t+Xys3p1V9JXafXqLXX+uv8AkZ//AAk93D4OnnZ45dSt78aV5zxkI0puBCspUHp8wcqCO4BHWrNnLf6L4ls9L1HWLjVo9Rt5Xje5hiR4pItuQDGqjawfoQSCvU54RfCkjeD5tOluIxqE9wb97hUJRbrzvOBCk5KK4UY4yoqWx07Vr3xBBq3iGCxtmsreSG2gs7l5wWkKl3ZmjTHCAAYPVueeMpSocs+W1ry7X2923W1+3Tcev9ev+R0NFFFeSWFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFU4bm9fWLq2lsPLso4o2hvPOU+c7Ft6bOq7cLyeu7jpVqSRIo2kldURAWZmOAoHUk15boHiHwBpXxT1RtE1fw3ZWt3plqM2lzBGk0/nTbvukBnwVz35HtQtZWB6Rb/AK3R6pRXgviK28GvrHjW7t5NLS6tdU08mW3uFRrdGeFLlwVI2ZLOruME8hjxW6ZdFto9S0i2ms4fDltr6SajZ2sgWOGyktEZWKIflgaYgseEIL543ULX+vT/AD/AdtGz12ivJ9EtPBN38Y9NPg+DTZ7W30u6mP2BUe0in8yAbo9vyLLtOGK84K56ivWKOif9b2F1sFFcf4o0HVdS1cT2Fj58XlKu/wD4Su/07nJ48qBGT/gWcn8KseEtG1LS7i5bUbP7OsiAKf8AhJLzU8nP924RQn1Xk0LUHodRRRXmd0fB0niPW0+J50cXv2nFl/bbIE+ybBsMHm/KOd24pzuHzdFpdbf1/X/BH0uemUV5brXhDw7efDWx1LRdNGrrosrX9m2pWxeW4iErSSx5lUMyurPtJ4PyNk8GtDwFpmja94j1fx9aabZqt/N5Om3C26q7Qxgq82cZ3SOXyepVVqktX5f0vv8A0Yuif9ef3HoVU9Tub21t4n06w+3yNPGjx+csWyNnAeTJ67VJbb1OMCqdp4s0W+8WXvhq1vd+r2ESzXNt5Tjy0YKQdxG0/eXoT1rjfizfeD4rSBtVutDTXbW7snhNzJCLqGP7VGWK7vnVdu4kjjGaXWPm1+Ydz0qivLviHqXgfxAPCk95qeiX0D6wIxP9riYNF5b+YocH7mfLDDODlQeuKyfDH/COaNr2nXEE1haXUXibUNOmfzlVoodtwILY5Pyp8keyPgZAIGaaV1/Xl/mD0/r1/wAj2eivHIW+Gt38UNfOoar4f+wyQWrfYmu4Ftrq5LTb3ePOJZFyPvZwWzjOCL3jPTLe08R2FnrFxodl4bttPjhsH8Q2Ml7arMCwcMzTIiybAm0vkkbgpHzZlaobVrnqtFeYaNe22ieBPEk2lX2leI9PhwUtdPsZYtPh3jEiqS8qtGAdzrGSFw3ygmsfw5e6FpHxG0RNI1vwmYbxZYJx4c082cMrlcxpI6zPHI+4fKh+cZJHBOatrYnpc9iupnt7WWWK3kuXRCywRFQ8hH8ILELk+5A96p6BrUHiLQLPV7SKaGG7jEiRzgB1HocEjP0Jrz5x4KF5qp+I76ePEIu5jH9sc/aRBvY2/wBkz84Gzbjyed+7+PNb/wAI7y1u/hXon2O4WfyYBFLhwzRuOqv6MMjIPPNC1V/Qb0djodL1y21e91K3sklKadP9mknYARvIFBZUOcnbkAnAGeBnBwsWswTeIrjRlimE9vAs7SMBsZWJAwc5zkHt2rn/AIWEr4FSGb/j8gvryO8yuG877RIWznucg/QiuqWzt0v5L1YwLiSNYmkyeVUkgfmxoVupMlK/u9/w/wAyeivLfEx0IeM9Q/4WH9s+y7E/sr/W+Tt2jft8v+Pd1z7e1ReEbfw1Z/Di/vPEqypbX140V08xlLybXJQME5Bz1xj3rRU7x5v6/wCHOF4z966dl13fbq1bRPp5anpel6h/adkLn7JdWeXZfKu4vLfg4zjPQ9RVyvN/hj4j0CK3m0OyusTS3072sHlycxdVO4jH3R3Oa5oXXgZ/GurJcajPBo1wqFreMTBbmbJ3ByBuwCc4yBnHpVeyblb9CPr0VSjO6d3Z6/8AA/pHqfiTxNF4ZtkuLnTtRvImDM72UAkEIXGS5JG0c/oa1bS5S8s4bmIMEmjWRQ3UAjIz+dcZ421fQvDfgu40E3DQST2Dx2cLCSQsMEAbjn6cmtbwZ4j0rXdFhg0q68+Szt4knXy3XYduMfMBnoelRy+42kbxrr2/s5SWq28zoqpanqX9mxwP9ju7vzplh22sW8pn+JueFHc0ljrNhqN7e2llP5k9i4juE2MNjHOBkjB6HpmuU8e+DNAudF1nXJ7DdqK2ryCfzpB8ypwdu7b2HapVk/eNas5ezbpWb9e3ye3Y7mivJ/Gen6YsHhHUL+O5AkhWG5ktWbeYBFkjAPTlskc4zRpRg/tC/Hwt+1f2f/Z8vnbg/lfacfu9hk53+vbGK0dKyeu1/wADl+uvnUXHt1117K2tuuu2p6xWdreswaFp63d1FNLG0qRYhAJBdtoPJHGSK8W1KLwlcaXYwWI1BtdluYk1NZzKHOWAk83d8v3sdOc4r246RYnS4dOMANpB5flxliduwgrznJwVFEqahZvv/wAP/wAAdHEzxHNGKSaXe+rva+nS2pcopksqQQvLMwSONSzseigck1Q8P+IdM8U6LDq2hztc2M5PlTNC8W/BwSA4BxkHnGKxPQNKivJoR4CmsL0fEaXSoPE3nzC6kv5FS8Qhz5Zty37wJt2mPyuPT5s1d8SaBoGhWXhjximmxyxaCIIp7i8swJxaFQgkcMoYPGdj8gFcPgDJFC7v+v6/rYb3sj0yiuF+GXh3T7TTbrxLDpNpY3evTvdqIbdY2jtmI8qPgcZRVdh/eZjXRaJ4s0XxFf6pZaNe/abjSZ/s96nlOnlSZI25YAN908jI4p2ez3F+RW1bxxouiaitjqK6os7yCKPytHu5UlcqW2o6RFXOAThSeh9DUk3jPRLXU9K0+7nuLa61dd1kk9lPH5n+ySyAI3+yxB5HHIrm/Hnizw7ZeJPDNrea/pdvcWesiS5hlvI1eBTbTAM4Jyo+ZeT/AHh61R8e3nh3xF4h0HT28Q6bbyXdhcyWNwLuPKSl4Gt5U55+dMrj720gZ5ojqrvv+gPe3l+rOyuPGeh23iOfQWuJ5dUt7b7XLbW1nNMyRYzklEIyey5ySQAORUGm+P8Aw/qslottNfRpesEtprrS7m3hlY9FWWSNUJOOBnJ7ZrifDV1p3hv4isPE3iDRv7euLC7udVMV4oSOUyQsqANhlVYYxgMAcIT6mqPh/ULi7+H+g3N7rmj6j4VsZIX1JLCAiez8sh4/NfzXBVXCF8KpCjOMZoS91N/1q0D3aX9aJnpF/wCNtH07WJtLnTVJLyFFkeO20a7nAVujBo4mUjgjIPUEdQafF4x0ifQ31e3N/LaxzNA4j0y5aZHU4YNCI/MGCOSVwK4TxTrGiah49vIo/iJbeHN2j2w86C9tlEuZZHAJf5h8rKRsZThwckEUni34iaJP4N1Sx0HUdHsp9U1F9NtroX8ccUoZEaa5MgHygB2BYBsNt6k4pa203/4Nh6c3lp+Vz0Hw14s0Xxdp8l74fu2uoI3COWgkiYEqGHyuobBVlIOMEHisyP4leGpBKxl1KOGCUxTzzaNeRwwMDg+ZI0QVAM8liAOpxXKaXrKeD/ihbafrJ0HR4NY0+K2SztNW88rLCMQuweKMqGQlBgEEoo61a8JePfCNlB4i+0+I9KZpNZupY4Eu43knU4xsQEs+7oAoOTwM0O19NrP8Gl/wRdPmvxTZ6VHIksayRMro4DKynIIPQg0k88VtbyT3MqQwxKXkkkYKqKBkkk8AAd6wvAlhd6X4D0ey1CNoriG2UNCx5iHUIf8AdBC/hSeOre4uPCFwLS3kujHNBNLBGNzSxJMjyKAOWJRWGByenenKyYLUhi+InhqS4tIWu7qA3siRWj3OnXMKXLOwCiN3jCvncD8pPHzdATSz/EHQLe7vLaX+1d9i5S5ZdFvGSEgZyziIqBjnOcYOc45rndb8aQavqemQeFPEXhfWFuL22VdNNsbu5j+cFpvknG3yxl+U+XbyRVQ6vql/4w8Y2vgjxF4fa+EsZFhMnmzyFII0cqwmAXB+X5kYBhg+y1s36/oM7rWfFekaDpMeqahNO1hJGZRc2lpNdIExu3sYlYKuOdxwKl07xJpmr+HzrWmyzXNkA+SltKZMoSrL5W3fuBBG3bn2rzHXvE3h+98O+GfBHhTUNNf7VLFFNZ6hfi3ZYoCMwSkKzLI7hU27SW+cY71r+DvEKaX8Std8N6xc6PaXWpOuoW9lZakLjy5iAsyEskbB2wsgXb0LEd6drtpedvl/X4C2V/vOqk8daDF4Yt/ELzXn9mXMixxSrptwWZmO1f3Yj3gE8AkYJIweRmxqPizStLlghuWvJLieLzlt7XT57iZY+m5o40ZkGeMsByCOoNcTpuo+F3+L1xpkPiCxltoVW7tNOW9iMaahI8iS7VBzvAXOzs0jtjLZq/4h1fw1D43uY7vxRc+E9VhtYo5bwywRQ3cRLMqZnV0ZkJJ6BgH6kE0uiff+vz/AfVrt/X5fidJa+M/D154bm1+DU4hpcLsklzIrRhWU7SuGAOd3GMZJ4HNR2PjfQr/VYNMjnuoL+4J8m1vbCe1kkAVmLKsqKSoCN83TPGckA8pa3VxL4X0zWriNJ7HTtckuZ7uyt/LF9bhZEF2Y1+9lnWQkDDbSyjBFdA3jbwhf6jbyWt9YaobMPNJf20kc0OmpsOZJJs7YtwyoGdzZPBAYh6f16C/r8Wv69Sa2+IHh258SR6CLm7g1OU4SC6065t93DEYaSNV5CPjnnacZqzrfi/SfD15DbamNQEk5AiNvpdzcK7HOFDRxsu7g/LnPtXkM2rXut+E9b8aWMnh3y5dTjvra+udbMclr9nIEEDR+SVDFM/IZBkznpmuy8QfEHwnqNn4Vvo/EWlxJNqEF3slvYlaOPZICWG7jB+U+hBFJdL90n87f8H7hvS/o/wADuNR1yy0rQn1i++0JZxxiVytpK7op5yY1UuMDrkcc5xg1mW3jzQ7m8tLYDVIXvZBFbtdaNeQRyOQSAHkiVckAnr2rnvEnxC0DVNC8V2Nhqmm3Fna6IZTfw30bxmSUSoIuOA2VXHOTvAx60dE8R6Bp8tqD41tPF91f3VlDa2U15bNJZt9wvGkQAyN5JwoPHJ704q7+a/F2Jlovv/BJ/qegR65bP4mm0Jklju47ZbpC6jZNGWKkoQedpADAgY3L1zWlXG+IQZvit4OS2DNNBFfSz7T9yAxqvzD0Mhjx7iuypdCmFFFFAgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqO4iee1lijnkt3dCqzRBS0ZI+8NwK5HXkEeoNSUUAYeleGBYasdUv9W1DWL4QG3imvfJXyYywZlVYo0X5iqkkgn5RzityiigAooooAKKKiF1btePaLPEblIxI0Icb1QkgMV6gEqQD7H0oAlooooAKKKKACio7i4htLaW5u5o4IIULySyMFVFAyWJPAAHenqyuoZCGVhkEHIIoAWiiigAooooAKKKKAKdppNlYX17eWcPlTX7rJclXbbI6qFDbc4BwACQATgZzgVcoooAxtd8Of27JC/wDbGrad5QIxp915QfP94YOcY/WrejaPZ6DpcWn6cjLDHk5ZtzOx5LMe5J5q9RT5naxn7OHPz21CiiikaBRRRQAVzWr+CbbW9QluL7V9YNvMV8ywS8227AY+XYB0OOea6Wq+oX9rpWm3F/qM629rbRtLNK/RFAyTTUuXUidONVcslcwtd8EWuu31rdPqmqWLWiBIEsrgRrH1+YDacHBxkdq6KKPyoUj3M+xQu5zlmx3J9aSCZLm3jni3BJUDrvQo2CMjKkAg+xGRVe11exvr2+tLWfzJdPdY7nCnbGxUMF3YwTtIJAJIyM4yKbcvhYo04RfPFbmGfAlnNqq3l/qusX8aT+elndXheBWzkYXHQZ4GfzrqKyNJ8V6Hr1/cWejalDfS2yK8hgy8e1iQCHHytyCOCcEYNO1bxLpei3CW97LM1w8ZlEFray3MgQHBcpErFVzxuIAzxWzo13P2Ti79rO/3ChThBtxW+4viDSX17RL3SWne3t76BoJJYW2yIrcMVJBHIJFP03SxpGn22m6cscVjaqscKjqiKOFx07YzT11vTG0Mayt9AdNMXnC63jZsx1zUWl+IdO1ieWCze4S4iUO8F3aS20m0kgMElVWK5BG4DGQRUexqWcuV2W+m3qaXTSNOiisrXfE+i+GbYT67qMNmGUsiMcySAY3bUGWbGcnAOBzUwpzqSUIK7fRajNWiojdQLZm7eVUtxH5hlc7VVcZySegxzzWVpni7RtXvks7K4mE8kZmhWe1lgE6DGWjMiqJF5HK5GCD3pxpVJJyjFtLfTb1FdWubVFY+p+KtI0e6e3vZ5vMjjEk3k2ssywIc4aVkUiNeCcuQMAnoDVq/1nT9LsYry+uVjtppY4kmALJukYKmSAQASwG48c9aJU5xSlJNJ7efoPrYvUUVT0zVbLWbRrnTpvOjSWSF8qVKOjFXUqwBBBB6iswLlFFFABRRRQBh3934qjvpF0vRtHubUY8uW51eWF245yi2zgc5/iP9Kl0q58RzXTLrelaXZ2+wlZLPU5Lhy2RgbWgjAGM85/D016KEAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBTi0qyh1i41VIf8ATriJIZJixJKIWKqAThRlmPGMk85q5RRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAhG5SDnBGODivOfD3hifVNR8S21z4j11LGx1JoLGGLUpvMgYwQsXMrOWkwW4RyUHOVbPHoksUc8LwzxrJFIpV0dcqwPBBB6isS08CeELC8iu7HwroltcwsHimh06FHRh0IYLkH3FC0lf+t0O/u2OS1Ma5r3jzV9OWJZEsFiFtbHxNdaYxiZFJm2W8RMgLll3MxA2YAU5LTyarrGneA1g1HV7ZZI9UTTrvUbO+Fy9lAzhSZJWjQeau4IWZOMhjk5Ndnq/h7RdfWJde0iw1NYSTEL22SYIT1xuBxnA6elT2+mWFppg021sraGxVDGLWOJViCHquwDGOTxihaIXX+uxzqeHLLRfE+lTWviXUrbzHkH9m32qzXK3x8tvuiaRmyud3y8ccjgEYljbJqvg258Uat4l1Sx1KPzZZpor94odOeMnMX2fPlEIV2kSIxbnJORXZaT4V8PaDcPPoWhaZps0i7Hks7OOFmXOcEqBkZHSm3HhPw5d6wNWu9A0ufUldXF5LZRtMGXG07yN2RgYOeMCh6jWjucdqFtqGu+IvBbXmoanpT6npssmp2tpdzQK5RIm2BQ37shnILLh8ZGe41NFkl03TfFunya5Jb2+mXbx2t/qM3nG0R7aKUF3kOXCNIxG8njAJwK6O68O6JfatDql7o9hcajb48m8ltUeaPByNrkZGCSeDUMHhLw5bLeC28P6XCL9St2I7KNftIJyRJgfOOT1z1pt3vbz/MS0tfyOJ0yw1PQPE/h8a7Jfwi6laFDaeJZ7+O7m8ljmWGeNSqYDt+6OA23cMAETt4W0y5+L2rC5utWj8zSLe4/da1dxcmefdjbKMIOPlHyrngDNdfpXhLw5oV011onh/S9NuGQo0tnZRxOVJBK5UA4yBx7VLq/hvQ9fMR13RtP1Mw58o3tqk3l5xnG4HGcDp6UdU/X8Q6Nen4O5zXjqW1u/hNf6joOrTslnaNNZXtjqUvLICoJkR8yDPUMSCRzzWpZeEvs6SyX2t6xe3NzbvFdMb6SOORmIO5EVsQlcEL5e04PJY81pap4d0TXIIYda0ew1GKA5iju7VJVj4x8oYHHHpRpHh7RdASVNB0iw0xZiDILK2SEOR0ztAzjJqWr3HfRI86TU0b4WeCdQvdbu49VluLCKJv7SlR7pmniWVXUOBMdobIYNjnpzXS3MQ8Q+PNQ0jVb6+tYLG0hmtLWzvpLQzhyweYtEyuwUrsxu2jqRkgjWbwX4WeS7d/DWjs99n7UxsIibj5g/z/L83zANznkA1Z1Pw5oetW8NvrOjafqENv8A6mO7tUlWPjHyhgQOB2qm7/fcW2xxPiDQrb/hVvi+yHiK+1r7Kl1IhGoSebbMtvgW8jI+XA6lX+9uywJ5PQ+HLDQ9FNgtvqly15qFqDBBeazPcecoUMxjjlkYcccqOAfSt+y06y02wjsdOs7e0tIwVS3giVI0B5ICgYHU1U0nw1oWgPK+haLp2mNMAJWs7RITJjpnaBnqetC0/D8Aetvn+NjTrj/FH9vf2uP7J/4SryPKX/kE/wBleTnJz/x9fvN3r26Y712FFIDl/CX9s/aLn+2P+Ej2bBs/tn+ztuc87PsnOfXdx6V1FFFMDzm08NvrHjLxdbXOva7HZ20sX2W3g1W4j8iSSBXZwwfcRk8JnYOfl5GM7UL6bUtI+HN/qfiC+0x9XVIL6e3u3gWffbM20qCEVmfADgbgT8pBxjv18IeGkkvJF8PaUr36lbthZR5uQTkiQ7fnBPJznmsPW/AH2ptMh8O/2HpWm2DSudPl0bzoZmkjeNtypLGu0q54xye/al/wPw3+/wD4FrD6/eWNMtzofjhNH0++vruynsJLqeG9vJLprZ1dFjYPIWcBwZPlLEZjJUD5s9BFrGmz6tPpcGo2kmoW6B5rRJ1MsSnGCyA5AORyR3FR6PoOk6BbNDoul2OnJIQ0q2VqkCu2MZIUfzqSLR9Ng1afVINOtI9QuECTXaQKJZVGMBnAyQMDgnsKfkI5bxnaQXesQW9pcapc6zc2+y2sLXVri0ghUMd1xL5LqQo3YJOS2AqjNZmvG78N+B9K0vTNZl1FX1EWmo6neas8TJneWVrjEjxZcLGMZZQwAYMQw7HVfCXhzXboXWt+H9L1G4VAgmu7KOVwo5AywJxyePerNnoek6fpTaZYaXZ2unuGDWkNuiRMG+8CgGOe/HNJbf13G9/67HI+H7LXdD8V2sN9LaWdnexSKbOfxLc6jLMyjcHiW4iDAj+LD7cHJXIBqGDQJtQ+JOvWD65rEWkxW1rcvZxajOC8shm5WXfvjQbfuRlQSFzwMHrdJ8L+H9Amkl0LQ9N0ySVdsj2dpHCXHXBKgZFUz8PvBjTmY+EdCMpbeZDpsO4tnOc7eue9D1sJaJmZe2cl34q07w1qGsalHYw6aJozHdNbzahKrbGLzR7WOwbGIQrkyZORgUmm3MmnN4p0uPX3Sw02FDBqV/IJjYSujlkeRz84T92/zkkB8E4xjqNV0XS9dtVtdb02z1K3Vw6xXkCyoGAIDYYEZwTz702HQtIt9FOjwaVZRaYyshskt0EJViSw8sDbgknIxzmh6r+v6/r5D0TucDplhqegeJ/D412S/hF1K0KG08Sz38d3N5LHMsM8alUwHb90cBtu4YAI3PiW2dG0iCbb9juNcsYrvccAxmYHB9iwQfjW1pXhLw5oV011onh/S9NuGQo0tnZRxOVJBK5UA4yBx7VZ1rR7TX9FutL1FWa3uo9jlG2svcMp7MCAQexAp32fZ3/G4l+lv+CXq474WN5vgVLiXaby4vryS8IOT532iQNn6YA+gFdbBG0NvHE8rzsiBTLJjc5A+8cADJ68AD2qjpeh22kXupXFk8oTUZ/tMkDEGNJCoDMgxkbsAkZIzyMZOTqHQy7f/kq1/wD9gS2/9Hz1PquqrbapJZaBZw3evXEaeYSMJBGCdrzuOQoy21fvMc7QBuZXJ4G8JRTLNF4W0VJVYMrrp8QYEcgg7etT6h4U8O6vdm61bQNLvrggKZrmzjkcgdBuYE16Dq4d1E3dpJLbtbz1XzRNndtGFq9ha6f8NDZWUqXmy4jUXAdVQXJulJkbqAqzEsy9gpWrGnxajF4+Qa5cWd9cHTJDBNZ2zW6wp5qblZGkfJY7SDuH+rYY6mugg0nTrbSv7MtrC1h08o0f2SOFVi2tncuwDGDk5GOcmq1loOm6JE8fh7S7HTfPYeabS2SIHGeSFAzjnGfWj61H2c4vVtt7Lrbr026b7PQXLoWTq2nLq66U2oWo1Fo/OWzMy+cUzjdsznb74xWb44/5J74i/wCwXc/+imqhpPhWWx8Za1r88wm1HUfLjE5iwkFuigLEvJzlgWY8denAJ29S0HSddjh/t7SbHUDCCUF1bpMIycZ27gcZwPyFcmHnGFWE59Hc0vZjHk0+LwoH1o24sFtF+0G6x5ezaM7s8Y+tZdo8mua/Y6rqEQ0+zgZ10u3uPkuLmRkIaRlPKjZvxH97GWYAjA1LfwxoFpp9xYWmh6bBZ3P+vtorSNY5f95QMN+NR2HhHw3pV6l5pfh7SrK5jzsnt7KON1yMHDKoIyCRXTGrRjGdm7u9tO69dL7PfQyUWopHPXUGrSa14qfRL2w02IeWLn7dbmZpX+zr+8UiRfLXZtXkOMoxx1BtanFYT/BW4SW3a1sToJIhkO5oEEGQCTjlcDn1Fbmo+GtC1i6S51fRdPv7iNQqS3VqkrqAcgAsCQMkn8aXXtDtvEWlnTr95RavLG8scTAecqsG8tsg/K2MEDBIyM81GJrxrUVTXl07K3z767bI0j7s1LsR+FZ7q58G6LPqJY3cthA85f7xkMals++c1h+FiY/iR42t4CwtRLZylAMKs7wfvMe5VYifrXZVm6JodtoVtcRWzyzPdXMl1PNOwLyyOcknAA4GFAA4VQO1csneTYoq0bGlRRRUjCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiqGuPqUehXkmhmD+0EhZrcXETSRs4GQpVWU89ODxnPPSvP5vio02iTWel3VrJ4it9Mt78u9i5tZzKU2xIom3Bv3sQPzHG8HnoBauy/rf8AyD+vy/zPT6KjgEy20S3TpJOEAkeNCis2OSFJJAz2yceprF8WeI/+EbsIbjz9Fh82Xy86xqn2GM8E/K/lvubjpgcZOaHoC1N6iuF0T4if2trVtY/bfBknnvt22Hij7TMeM/JH9nXcfbIruqfS4EV3d29haS3V9cRW1vCpeSaZwiIo6kseAPepQQQCDkHoa5j4gyahbeC9Tu7A6ZJFa2c01za6lYtcx3KKhbZgSJjOMEnd16UzxfresaP4Jj1fQ/sP2lWt90N3E7RyCR1TaCrApy4O7DdOhpLUO3z/AEOqorzyDxV4rt9XubfUm0aWDTtUtNPuTb20qNcG48shkzIRHs81eu/fg/crRXWvE+rw32qeHjpKafZ3EsEVrdQyNLdmFykh80OoiyysB8j9AT1wE2kub+uj/VDs/wCv68jsqKwvBWp6hrfgvStW1hrZrm/tkuStrC0aIHUMFwzsSRnGc8+grdqmmnZkp3VworzeDxp4mbVLSG+On6a1xerBJZ6hpF7AkC78FVvSTDNIVGFACqzMMHs0k9lqcPxk1NfC0el2El1pFrNeXdzbGQMRLOo/doyF2OANxYYC9+BSWtv66XG9E/L/ADseiUVyUfiyVvAj6nf3FppV9HcSWTO1vJdRtPHM0JCRIyu+8qdqA7uQOcc1vC/izVNW1u80a8ZJJls/tMF42i3engHdt2tDOcsAcHcr85xhcZJu7L+uobLX+un5nbUVwPh7V/Hmu6I+pq/h+MQ3E0K27WswN55UzoW3iU+Ru24A2y4xnnO0RX/xFmvZtPh0DfbC406HUZZpNFu9SKLLnZH5dvjB+ViWZx0AAbJKg7HodFc54O8QX2t213HqdrNHPayhRcHTrizjuUIyGWOdQykcgrlgMA7jnAzfDOlafJ8R/GGpyWNs9/FewxR3bQqZUQ2kOVD4yBz0zT628r/l/mI7WiuL8V6PZx+LPDGqgTNdvq4jzJcyOiKbebIRCxVM7VztAzjmup1TS7DWNPks9Xsba/tXwWguoVlRiOQSrAjg1rXpxp04zi73V9rdWu7/AEEvisW6K5X4YgD4WeGwBgDT4sD/AIDXVVnJWbQ0FFFFSAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXDW/wAIvDVtLYSQm8DWOpvqUZ81Tudtv7tvl5jXy48L28teeK7mihaO4dLBRRRQAUUUUAYfifw1J4msZbJ9c1LTrS4geC4gs1t8TK4wcmSJ2HBI+UiqF74Hl1Hw6dHvfFWtywmWOQSFLMOojIKrxbgY3BT0zx1xkHq6KNgOUuPAUVxNqEv9u6qkl/NbXLsgt/kngEYWVcxEZPlLkHK9cKKyvEmgX9vcXdvokHiee1v1MtzBp9zYx20krcPuaYiaPdgFvKwOSR8xNegUUmrq39f1/kh3MHRvDstj8PLHw5cXkkMsOmpZPdWTlHRhGELxsRwQeQcenFXNH0b+yPD0Olf2lqF95SMn2y9n8y4fJJyz4GSM8HHYVpUVUnzNt9RLS1uhycPgVvs9raar4p13V7G2aNvsl81uVlMZBTzHSFZHwwVuX5I+bIyDeTwsY/F114gXWtR865tltTbbYPKRF3Fdv7rdkM7Nyx5PORgVvUUgOUt/AUMGimwbXdWmkW+bUILxzAs1vOzMzMu2IIQS75DKwwxGMYw+y8INpevHX21zW9Uvlt2ikinkg23CdVQII1VMEAjZsyfvE11FFH9foH9fr+Z5l4O8I6u+nXlrqEviXw2k9xPNPbpc2Tw3Almkb5GXzZIzsZQ20pzyOctXX33hK0uJLSbS7u70S6s4RbRXGnFAfJHSJkkV0ZR1GVJU9CMnO9RRtsG7M3R9HfSlna41XUNUnnYF5r2RTgAYAVEVY0H+6oJ6nJrM0jStY03xprl08FjJpeqTxzrMLpxPGVgjj2mLy9pBKE539D0rpaKOtwOe1vwpLreo290/iHVbQWswnt4LdLbZFIEKbhvhZjkM3DEjn6Y0LhNTs9FEeneXql6gChtQnFuJOeSzRxEA49Ex9K0aK0qVZVIKnLZeS/PcVtbmL4O0e48P+C9I0i+aJ7iytUhlaFiULAYOCQCR9QK2qKKhu7uMKKKKQBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQXtsl5YzW8rSqkqFWaGZonAPo6EMp9wQa878E+MNch8DeHb3WdJeTTZ1t7R9QmvjJdO8hWNZWiKHKGRgNxk3YOdtei3iXEllMljLFDcMhEck0RkRW7EqGUsPbcPrXGp4H1uHwBpfhuDXrANp8sDi6fS3IkWGRZIxsE4wdyDJ3cjOAOtC0bv5frcOh3FUNW1C5063SSz0e91ZmbaYrN4VZBj7x82RBjtwSfarsYcRqJWVnwNxVcAnvgZOPzNUtW0LSNet0g13S7LUoY23pHeW6TKrYxkBgcHB60MEUbLxBqV1exQT+ENaso3OGuJ5rIpH7kJcM35Ka3awrLwN4T0y9ivNO8L6LaXUJ3RzwafFG6H1DBcit2n0A5f4iTTQ+CdTY6Raatp62kz39vcX8lqWiVCSEKRuSSAe6/WpPE/iS78O+EotZsNLjvwGhEls115LBZGVRsJUhmyw4O0deRT/ABfomseIdHutL0zVbGwtb21ktrj7Rp73EhDqVJQiZAvBPUNWfqnhXxLq/hM6Nda/pSv5sLLPHo8gASNgwUqbk5JZV5z0yMc5CXW/kD6fP9CvbePNX/tSS01Hw5BbJaX9vYXssWpeZskn2eX5Y8sbwPMTdu2Yz8u/FXZvFmrzm8u9C8PxahpdjK8MkzX/AJc0zRsVk8mLYysFII+Z0yVPbBNa58EatPc6pcR65ZJJf3lnfru012Ec8AiHP74bkbyvu8EbvvHFZ2uxah4ca803StUvIrS/8y5NvaeGrm8kjeQkyCKeM+XHubcwEgbaWJJ24Al35fP/AIC/W5Stp/Xf/gHW+EtaufEfhTT9ZvLOKya+gW4SCK4M21GAK5YovODyMceprZrnPDujalp/wy03RhcDT9Ug0qO2E4VZvs8wiC7sHh9rc46HFaOj2OqWXh6Gz1bV/wC09RRGWTUPsyw+YxJw3lr8owCBj2rSdlJ2IjdpXOQ074m3Oq6vBZWGm6TNLPc+SbEa7GuoW6BsO81syDaVUMxVXZuAMdSHXF9rWn/FjUbLQbSXUhc6Xb3LxXupyR2tu3mTKzDIk2FgqAKiYOCTjqbsnhPxHqtpY2PifxHp9/aW00M8j2+jm3uJXiYOreZ5zKhLKMlUHGQNuci/H4e1WPx1e6+NVszBc2aWiWpsH3IELsrGTzcE7pGz8oyMDjqVs18/y/zG9U/l+f8AkLB4pN14Ll1xILWzlgMkc8WpXfkQwyRyGOQNMFbChlbDbeeOBnih4X8fJ4k1G90yH+x7i+t7f7RG2laut5buCdoVpAisjZ6goeCCCeQIovAupnw8LG81qymuodWbVbaZdMIiWRpWlKyRNM29dztjDKRhTnIzVqy8P67ZeKD4i1jxBb3yraPBNaW+lMihB8w8rEjOGyMnd5megC0tL67f8D/MOmn9a/5FPR/GfifWtLfUbTwhb/Z4JpYZlOq4llaORkcwKYgHHy8F2jycjgfMU1r4n6ZZSafFp11o++9s0vlfWNWWwjEL/cwSrszHB4C4AByR8oON4GtfEV1o95ZabrF1pUUl1cyvDqnh64WSASTyEGCWQxrypDYKybWJzx8o6qTwfJp0tnceEr+PSri2s47FkuLY3MM8EedgdQ6NuUscMHH3jkHjAr21/rf/AIA3a7t/X9Il8HeMrPxdbXf2eSya6spRHcLY3qXcPIyrJKuMqR6hWBBBA4JpeHoDd/EDxTdXF3fSNZ3cMFvCb2byI0a1iYgQ7vLzlic7c5Nb2j2erWyzvrmrR6hNKwKrb2gt4YQBjCqWd8nqSzt7AdKx9Dg1LT/HXiP7RpFz9i1K5iuIL9ZITFhbaNCpXzPMB3IR9zHvT05vl/l/wSegxbQ+KfEGtpd3+o2semTpaWyWV49uFJhjkMh2Eb2zJjD7lAUfLy2bekSDxd4DsZ9Re4ja5gVpWtLmS2cuvUh4mVgCRnAPQ4p9/oOpjU7u70DVodP+3qv2pZrMz/Oq7RJGQ67X2gA7gw+ReODm0YR4c8NQWmkaZc38dpEkEVrbNEJGUADOZHRfc5IrsxVSnLD8sXd6WWullr06vXT1YK/MUPh1c3F58NtAubyeW4nlsYmklmcu7nbySx5J9zXSVgeBNNu9H8AaJp2pQ+Rd2tnHHNFuVtjAcjKkg/ga365Z/E7AtgoooqBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFcyfiJ4ZFxDE17Oq3Fwttb3DWE4t55WbYFjm2eW5J/useAT0BIPIDpqKxtU8WaVpF/8AYrlrye6CCR4bHT57to1JIBcQo2wHBxuxnBxnBqQ+JNLPh0a5BNJdaey7hJaW8k7Yzg/Iil8g5BGMrg5xg0rq1x2d7GrRXP6b468O6veW1tpt+1w10B5Mi20vlMxj8zZ5hXYJNnzeWSGA6iugpiCisfVPF/hvQ7wWmteIdK065KhxDd3scT7T0O1mBxweak0nxPoOvySR6Frem6m8QDSLZ3ccxQHoSFJxRuGxqUUVhap4z0PRr2W1vri48y3QPcNBZTTx2ykZBleNGWIY5+cjjnpzQBu0VjTeLdFg0Sx1Y3Ty2uoqrWYgt5JZbgMu4bIkUu3y5YgLwAScAGpdI8Rabri3H9nyzeZakCe3uLaWCaLIypaKRVcAjODjBwcZxRsBqUVyv/CyvCwjlLXtyrwMyzwtp1yJYAoUs8kZj3RoA65dgF561P4q1+DTBZQNqV1przXUB+0x6XLcxMplVTEzqhSPfnZkkEbgRR28wem50dFV769i06xku7hJ3jjGWW3t3nc844SMFm69gapaR4m0vXJ5oLGWdLmBQ0ltd2ktrMqno3lyqrbSQRuxjIIzwaANWiuYfVNbs/iFZaVd3FhPp2oW1zPEsVo8c0PlGIAM5lZXz5h6KvSuno6XAKKp6vqUOjaLe6ndLI8FnA88ixLucqqliAO5wK5228T+JmuNJe58L2r2GpyKoutP1Q3P2dGUsHkHkqMYGMqzDJ69CRauwPRXOuorA1Dxvoem3lxbTy3kr2vFw9pp1xcxwHG7DyRxsqEAgkMQQCCeCKs6t4n0nRdFj1e9uJG0+RQy3NrbyXKbCMhyYlbCY53HjpzzQBrUVjaZ4s0XWNSex027aaZVd1byJFilVGCuY5SoSTaxAOwnBODVXRdeh1XxXqcEGo3bLDbwkaddaXLamD5pAZQ8iKZA5GOMgbOOtG4PQ6OisnV/EthokyxX0OpOWTfutNKubpQM92ijYA8dCc1Y+1nVtEF14dv7RvtEYe2umjM8JB77VdSw+jCjpcOpeorA8G6vf6zos8urm2a6t7+5tGe1iaJHEUzRhgrMxGQucbjS+KNcudK/s2y0qOKTUtUvFtrcTKWRFALySMAQSFRWOMjJwMjNHbz/AFDv5X/A3qKwNV8b6FourjS9RuLmO+aPzIoEsZ5GnGQMRbUPmEZ5VMkDJIABq3p/iXSNS025v7e8WOC0LC6+1I1u9ttGT5qSBWj4+b5gOCD0INHS4dbGpRWDpvjbQtVvoLS1uLiOW5UtbfarKe2W5AGT5TSIqycc/ITxz05pNQ8caDpWqy6bqNzPb3caB1jayn/fglVAiOzEzZdRtjLHJ6daAN+iuS174iaZo/hu31i2t72+iuLhYFWOxuMofOWJw+IyY2Uk/I4BYrtHNbmka5aa2krWcV/GIiA323Tp7UnPoJUXd07ZxQBo0UUUAFFc3401668NWVhqMVzYwWX2+CC9N5GSBFJIqMwfeoTaCTkhhx2qzpHi7R9bvPstlNcJcNH5scV5ZTWrSp3eMSou9RkZK5AyM9RQtdv66g9NzborCuPGejWmpGyvWvrVhKIfPuNNuYrfeTgDz2jEfJ4B3YJIAzkVcfXbCPxDHokjzLfywNcRobeQI6KQGIk27CRuGQDkZHHNAGjRWPY+K9Hv9Pvr1Lp7a309yl29/byWnkEKH+YSqpA2spz0wa5y++IsEviDw/Z6NPsTUbsQy2+o6Xc28s0ZRm8yFpNgIXaM4V/vr04y1q0l1/UHom+x3dFFFIAopk8y29vJNIHKxqXYRozsQBnhVBJPsASa5zQvHFlrbMgstShk+1y2y5065KHY5UMzmMKmcZIYjb0PSrjTlJNpbFKMmuZLQ6aisPxHrEWnfZIGvrnT3kuIf36afJPGymRQY2cKVQvnbkkEbs1PrXiTS/D32f8AtaaWEXL7ImS2kkDP2XKKQGPYHk9s0KnJpWW4cknsjVorF07xfomq3qWljdu80hZVDW8iDeoy0ZZlAEgHJQncByRTJPGeiRTOr3E4ijk8p7oWcxtlYHacz7PLGDwTuwDkHkU/ZVL25X9w+SXY3aKz9U1yw0cwrfSv5twSIYIYXmlkwMsVjQFiAOpAwM81XbxXo66Hcav9okNpasVuNttK0kBHUPGF3oQOTlRgcnjmkoTaukJQk9kbFFZ8uu2MGp2dhK0yzXozbt9mkMbnazY8zbsDbUY7SQcDpWTJ45sofE11pEtlqZ+zQpIZYtNuZNzF3UgKsR+X5MhwSrZOOhpqlOWyGoSaukdNRWTqPirQNGS3bW9asNL+0pvhW/uFt2ccZ+WQg8ZGRjjvWHrfjuxu9Avz4D8QeHNS1i3gNwlvJerMpROXJWJt33c47ZxmoSu7E21sdlRXhi/GjxRqXw3TX9HtNEh1CwlWPVrO9SbOJWUQSRKGHysGJO49uM16NrPxE0DwXZ2MPjnWbOz1OaFWlhtkklG7HzMqqCwTIOCw5x611VMHWg7NXd2rLV6f8Bp/MhST2Otormb/AOI3hPTfDlnr11rMP9l3sgiguYUeUO5BO3CAkHg5BAwRg4NP0H4g+FfE+s3elaFrMF5e2mfNiVWHAOCVJADgHupI5HqKx9jVs3yuy8h8yOjorjLn4u+A7PX20a58S2iXiuI24cxK3oZQPLGO+W4PBwa1PFevz+HLSx1MLC2mC7ji1BnB3RxSHYsikHHyuyEjB+XPTFTKnOFuZNX2Hpexv0UUVABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA2VnSF2jTzHCkqmcbj2Ge1eaWOlax5lg2i+GNe8NXv2iOSbz9bjn06BN2ZoxALhwVK7lULEuCVI2YyPTaKFo7g9rHnmr+GLq18WanqDadr2sWmpukyjSNdks3t3WNIyjRmeJGUhAQwJbqCMAGtPT7W60bwZeLZ+GtR866mkZbI6ot3cHeMb5JJ5QoPHKrIwHYnmuwopNXi4/IadpJnJfD1b3TvC2maHe+GbzRjYWMUbySPbGKSQKA+0RSscltzZKjPOea62iiqk+Z3ZKVlYx9U0W/1C8E1p4n1XS49oXyLSK1ZM/3sywO2fxxx0qTSdJvdNkka88QalqwcAKl5HbKI/ceVEh/PNalFLYYV59rtjcXeu6lMfB3iPzSQkN9ouupapdKEG2R0+0xfMCSvzI3Cjkjgeg0Uh3PP9Y8Ma7d2fh3UtTa51XUNOtXg1C30zUZLF5zJ5e6SN0aMFgY/usUVgSflIAq74Y0x7PVbvUU8Oa1ZFbXy1bVdca8nmO7dsRDPKirx94upycYxk12dFVcXSzOD8Jtqth4g1ue78F6nanWNSE/2sy2R2RCKNB5m2cscFXOAG4bjqas/EBtYv8ATP7L0nw1qOoH7Ta3H2mGa1SLEc6SMv7yZWzhD/DjJHPeuzopdvK34bB1b7nKeIJNd1nwvG9hpeqafJ9rT7XZLdQRXctuD84jlSRkUnjnepwGAKkg1j+F9K1Sy8fm/Ph7WLPTZ7FrcS6lrQvZI2DBsurTSbAcYHls+erba9DooWjv/W1geqt/W9zg9TvNdl8d6XqsPgvWZLXT7a7t3IuLEGQyNFtZQbgcfuz1weRx1x1Gt6hqtl4dlvdE0VtU1FVQx6c90kBckjIMhyoIBJ7g4wK1KKOlg63MjUbrXP8AhEnutK06Ea21urpZXEoMaynGUZwQCByMj0rkbLw+0viLS7vR/A7eEriG482/vo2tUW4i2ndDiCQtKGbafnUAY3cMAD6LRT63DpY8zPhW60jUb+O50jxNrENxdy3UFzpHiOS3UCSRnKPE1zEqspbGUBDDBODkVd8Q6TqSfCOXw9pXhm5luby0kgFrDqEc4ti2T88s7oX6843c5HIwT39FL7Nh397mMnRr6eawdG8OX2jpaoEhtp2tvnUDhUEUrqAMAYJXtXMWt7rw+IF1rEngrWktbnT7ezGbix3IySyszEC5+7iQdMng8eve0U+txfZscF4j07WZ/E15JPpmu6rZtDGNPXSta+wRQnB3ibbNG5JbB3YkwuMAEENc8IDV/Dvw+trW98O38l7aSPGbSG6gleQFywdZHlUMuG6uVc4OVrsaKXSwdbnF+A31iyS7sdV8M6lp63GoXd4tzNNavGqyTNIqny5mbdhgOFIz3707xuP7O1/wv4jl/wCPPTb2SG7bdgRRzxmMSH2D7MnsCT612VFHbyDv53/E4TxFc31n8V9HudN0r+1Wj0e6EsEciJMEM0A3Rlyqkg4yCy8Z5yACt5oGreJoNdvprH+yp72G1htbK9kRy32eRpQZTEzqA7OUwC2FGep2jpLzwp4e1DVV1O/0HTLrUEKlbuazjeVSv3SHIzx254rWoWkbev4u427u/p+H/DHG3cmteKZtNtLjwzd6PFb3kN5cXV9Pbuo8pw+yMRSuxZiAMsFAUseuFNbVTqn/AAsu01hPBmpX9tp9jPaxTxy2WTI7xkPGHnUgbVYEkA84xya7uijrdef4q35BfS39b3Ob8Zaffa34LePT7RjeCS2ultJHRWYxTJKY92SoYhCuc7c98c1d0jU9W1G6lN7oE2k2ip8hu7qJpnf/AHIi6hcfxF85/hxzWvRQI5nwx4tm8Ta5r1vDpnk6bpN19jivzPu+1SqP3gVNowFPGdxzXTVBZ2Vrp9uLewtobWEMzCOGMIu5iWY4HGSSSfUk1PR0QdWcl8QrHVtT0nT7TQ9LXUJxqVtcMZpUSGNYpVcmQk7sHGPkVz7UiDWPEfiDR7m90GfRLfSZXuZHu54ZHmdoXiEcYid/l/eElmKnhQFOSV66ihafn/X3A9fy/r7zyDWPDnie/wBEube40PXtQ1lpi8t23iERWMoEmR5UCzBcbQoVHiQf3myMnstdi1RfEfh/xDZaJdXy29tcQXFlFLAs8RmETA/PIqHBjKnD9wRkV1tFHYOtzzmDRvEmr2/iRdR0f+zLi71K11O0zdxyQyiJIMQuyksCTCVb5Coz8pcDmXxBdeJtbvvD8sHgm/hh07Uku7nz7u08wgRun7oLMQ33+SzKcYwGyceg0U07W8rfh/wyB6p+d/xOZvvFs0PxE0/wrp+mfa2mtXvL26M+xbOIHapxtO4swIAyOldNUEdlaxXs15FbQpdTqqyzrGA8irnaGbqQMnGemTU9LoHUK5LSv7c0Vr3T4dBkufN1Ge4ivGuoo7cxyyGT5jkyAgMRgRnkdcHI62irjPlurXKUrKxyvjRtTvNP/s7TdBvr0/aLafz4pbdY8RzJIw+eVWzhD/DjOOaZrk2q6l/Yc1v4b1EG21EXE0bzWoaNFRlz/rsHO/jBPQ5xxnraKuNblSSS016/5lKpZWt3/FWOFs9M1e2ubbfpF0Vt9evLhnjmh+aGYThZFzJnA8xcg4b0Bqla+GbnTNLGlXWia9qTwx+Slxa+IZI7adcYUsrXClM/xKEIHO3cMV6PRV/WZdvz/wAy/bSv/wAP3v8AqczLZ3+j6taalZaW9/H9gSyltbadTJBtJYFXlZA6nJBJIbhTg84gl0bUda03xLPPajTp9YsvscFtM6syBUkUPIULLkmQ8AnCqvOcgdbRUe2l8/8Ag3IVRxd1v/kcVNN4gvbzw858L3UFvp9yJLnzLu3MhJgljyirIQVBcEksG6YU84v36alpnjKXVbTSLjVLe50+O2xaywq0bxySN8wkdOCJOoJ+6eK6aim613svx/zDn6W0tb8bmLdabq2rWttI+sXmg3CqfNg01redGJPGWmgYnA9AvU9etZGt6F4gs/DeqGw1jVvEl1LavDFp12bKBHLjbu3pAhBAJPLYOMV2NFZJ2lexCdnc+edQ+GHi/VfAukPDoZ0vW7BIdMu7UX0Mg1C0Qo6ylgdqlXXpnOB16Cu61PSfFfhX4j6r4l8NeHLbxPDrUMSOjXqWs9oYwBtDvwY2xnA5z1xgZ9Mortljqk/iStrpr1tfrfdd/wANDNQS/D8NjxLVPAfiyH4R6zplpolrdap4j1MXk+n2lykUOnplX2qXIDH92AQDjLcZA53rmw8VXnxE8NX6+EFs9Pt9La0upY9QhP2UzBdyhRgt5e3A2jDZ4xXp9FQ8XJqziuvfquXv2Hy/1873+8+aB8KPG2j6bN4fGm+ItW08MyltO8VQWlpOrHP/AB7yRkr15BzkgnpXqnizTpU+FuleDl3f2jqi22mxLJL57Lt2tK7PtXcEjRyW2jJA4GRXodFFfFzrpKaW9+uv42+4FFJ3QUUUVxlBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBBeXtrp1nJd6hcw2ttEN0k08gREHqWPAqjB4q8PXWpRada69pk17MoeO2jvI2kdSu8EKDkgr83056Vd1H/kF3X/XF/8A0E15boc9refBzT/Dll4e1CDUp9PiFuP7OkMKzsgZbr7QF8sYYiXdu3DGMbvlroo0lUTb7r8b6/K3/DA9EvO/6Hpk+uaTa6mum3OqWUN88ZlW1kuEWVkAJLBCc4AUnOMcH0qOHxHolxo8mrQaxp8umxEiS8S6Qwpjrlwdo6jvXJePH0+98QeHbPVdM1PULWyu2nvBBpV1PEVMEgXJjQq4LlQVyevIxmupGq6baaW+sR2d4kd04aQRaXOZ3bAQFohH5mcKBkr0A7YodNKCdnr/AJ/1/WyDw7qjaroovZb7TL1Wll2z6ZN5kBjDsE+bJ+YLt3dt2ccVHH4y8MSvbJF4j0h2u22W6rfREzNnbhPm+Y54wO/FYXgHVohDfWUtlq1vNNqd7dR/adJuYUMbzO6ne8YUEqQcE57YzXOrBay/D3x4V0TU/tmp3V2yK+kXIluC+7yCFMe4gHB44UnJxmtfYRc5KSfT8WC1087fnqd14g8b+H/C99Y2etana2s95IEVJLiNDGpVj5jhmBCZQru9SBWnpmr6brVobrRtQtdQtwxQzWk6ypuHUZUkZ5HFc34mne40/wANaxBaX01vZ6lHc3CJZymdIzDLHu8nb5hwzrkbc4yccVv6RrI1lZpItPv7WBGCxy3tuYPO9SsbYkAB4+dVz2yOaylTSp3S16i5r281/mQSa/nxjDoFrbec4tDd3cxk2i3QttjGMHczMG44wFJz0Bk1fWJNLvtLt0tPPW/ufs7P5m3yvlLZxg54U+lYWk5sfi94igutwbU7G0ubUnGGSLfHIo/3WZSf98V1zwxytG0kaO0bbkLKDtOCMj0OCR+JrBW0b/rX+kE02mouz/4BheL/ABJeeGNOivLXSf7RiaQJKftIi8skgL1BzknHHSotH8U31zri6Tr+hyaPdTRGW2/0lJ0lVfvDcvQjI4/+tmj8T2uJvCwsLPT9QvJbmZCGsYTIYwjqxJx0OOnvUOg+HIrjRr3UrR9cXVri3ktoJ9ecmeAY4wP4RnnPWtIqPs22u/5aficNSdX6zyQelk+lt3fpfa1td/I6xNZ0yXUTp8eo2j3qkg2yzqZAQMn5c56U6TVdOi1BLCW/tUvHGVt2mUSMPZc5PQ15UdJFx4dttAsvA97Z67GyqdSaFVjjdTky/aByw7498DoK0/HkEV14j0yObw3q18LU/wCm3dlZn/SIymNgkUgnk8jI6mq9mrpX/ruR9cqcjk47W79em26/qx3Nz4j0qHRrjUotQs5reAEF1uV2FwMhN2cAnjj3qHQPFWma/ZWr297aC7nhEr2aXKvJHkZIIHPH0FVLbSdJ1TwhOll4dj09buNsWlzZpE+8ZCll9c8g1g+C5VtP7NsYPA91aX8MQivNQntEgUADDMsnWQk4475z2pKMdV1NJVqqnBu1mvPuvLt+p6HXPa94pk03UodK0jTJdX1WZDL9mjkEaxxjjc7nheeB6/lnoa47WP7Q8O+Mn1630y51SwvLZbe4js03zQspJUhe6nPbp1+ucUnLU6MRKUYXj/nZdy5pfjexuFuIdeCaDf2rhJ7a9nUAZyVKucBwQOo/wJ2LfWtLu7xrS01KznuUzuhinVnXHXKg5rzfxHY3OuaN4h8TatpRsovsSwWNvdIPOAV8mRh/A3JA74z25LIdPXUbfSbDQvBt5o2p2s0Tyak8IjSNV++RMOZcjt3zntW3s4vy/Q4PrdWDta66b3f3Ld9Nl12PS7zWtL0+4S3v9Ss7WaQApHPOqMwJxkAnJ5qS51GxspoYby8t7eW4bbCksqq0h9FBPJ5HT1rxvWfDusWniLVTe2t9cC7uXljmttBh1ANGfu/vHYMhA429sZ71qarp00XgrwvaXGjavqj290JHb7BiaO3UnMbBWYpkFQBnnbzjFJU01F33t/XyG8bVTmuS1r/g/wBf6ueoWWpWOpRu+nXtvdojbWaCVXCn0JB4NRw61pdxqDWMGpWct4hIa3SdTICOoKg54rgzYSr4r16w0DQ7vS47zTPs8Nwlp5Nv5wDHduHAzuAz1yDXG2/h3VI0trOSy1i3uIHQCS38NQHZIp4IuBICwz/ETg9TRGnGT3sFTG1YLSF9WvLS3f8Ar1PbdV1mx0e333t3awOysYUuLhYvNIHQFj9PpmqPhTxDL4k0ye5uLEWMsFy9u8InEvK4ydwAB69q5bVbVdO8X6jfeIfC954hgvI4xaS29qtz5KquGQoT8nzHOe+T71a+F6T2On6jptzo19pmy8eeNbiM+XsbACq5+8Rt6/SkoLkb8jT6xUdeMdldq35dPus/+B3dFUdK1P8AtW3kl+xXll5crRbLyLy2bH8QGTlTng0Tan5Os22n/YryTz0Z/tKRZhjx2Zs8E44FZWd7Hdzxtzf12L1FVtRvf7P0+a7+zXF15S7vJto98j+yr3NSW032m0hn8uSLzUV/LlXa6ZGcMOxHcUiuZXsS0VRsdT+23t7bfYry3+yOE824i2JNnPMZz8w460X+p/YLizi+xXlz9ql8rfbRb1h/2nOflX3p2d7E88bc3QvUUyaTyYJJdjybFLbEGWbA6AdzUGmX/wDaemw3n2W5tPNBPkXcflyJgkfMvbpmkVzK9i1RVGPU/M1uXTfsV4vlRCX7U0WIHyR8qvnluemKNW1P+ybIXP2K8vfnCeVZxeY/PfGRwO5p2ZPPFJvsXqy/D+sSa3YzXE1p9laK5lt9nmb87G25zgdSK1KZFDHCpWGNI1LFiEUDJJyT9STmjQbUrppmVpuvfbPEWq6Nc232a6sDHJH+83C4gcfLKOBj5ldSOcFevIrYrjbZft3xqvLq2CtDpuipZ3MgPSaWXzFT6hF3e28etdlSXwp/1v8AruW92v62CiiigQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBXv7KHUrGazuTKIZl2v5MzxNj2dCGH4EVV0PQbHw7p4stKFytsuAkc95LceWAAAqmRmKqAB8owPatKiq5pJct9ACiiipAKKqaZqH9p2ZuPsl1aYlkj8u7i8tzscruxn7rY3A9wQat02mnZgFFRTXVvbvCtxPHE07+XEHcKZHwTtXPU4BOB2BqWkBDLZ2091BczW8Uk9vuMMroC0W4YbaeoyODjrU1U5tWsrfV7XS5ZsXt2jyRQqjMSiY3McDCgbgMnAyQOpovdVs9PubSC7lMct5J5UChGbe2M4yBxx64o32E2oq7LlFYmr+MdB0LUorDVb7yLqZQyR+TI2QTgcqpHUUDxjoJ8Rf2EL7/iZb9nkeTJ97Gcbtu3p71SjJ7IzdaknZyX399jbooqnqWq2WkQxS6hN5KTTLAh2M2XboOAfz6VO+ho2oq72LlFMmmS3gkmmbbHGpdjjOABk0yyvINQsYbyzfzIJ0Ekb4I3KRkHB5FAXV7E1FU31WyTWI9KabF7JCZ1i2NygOCc4x17ZzVygE0726Gbrfh/TPEdmlrrNt9phR/MVfMZMNgjOVIPc1Jo+i2GgaetjpNv9ntlYsE3s3J68sSavUU7u1ifZw5ueyv36hRRWZrniLS/DdpHc61dfZoZH8tG8tny2CcYUHsDSKlJRV5OyNOisXQvF+h+JZpYtEvvtLwqGkHkumAeP4lFbVNprcmFSFRc0HdeQUUUUiwooooAKKr6hf22l6fNfX0nlW8C75H2lto9cAE09bmJ7MXSFmiaPzAQhJK4z93Gc47YzRsriur8vUloqrpupWmsaXbajps63FpdRrLDKoIDKRkHB5H0PIq1RsMKKKKACiiigAooooAht7S2tWma1t4oTPIZZTGgXzHIALNjqcADJ54FTUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFLWYFudEvIneaMNC3zQTPE4wM8OhDDp2Neb6QuPhLaeLbbxNql7rEdikqyvqEjxS3AUKLdoC3lnLYj+7vJOc7zur0vU7e5utLuLexnit55YyiSzRGVVz3KhlJ/76Fc74X8B2OiW9o+qWei6hqVlHHFBqUOlLBOFSMIpZmd2LYHUEemK6qNSMIO76r573+/+kwlsvn+hV8ZWN5ceKPDQ07Vr/S5r64kt7hoLhyhiFvK5AiLeXv44cqSDg84AroLXw7b2umXWni91Oa2uH3DztRmaWIYA2pNu8wDIz94nJOCBwG6j4R8N6xeG71fw9pV/csApmurKOVyB0G5lJqaTw5okujJpEuj6e+mIcpZNaoYV5zwmNo5OelS6icFFP+riMnwCZG0G+hmuLm4EGq3tvG9zcPNII0ndVXe5LHAAHJrjl0u7i8HeLdXXxBrf2nSLq9Omg6lMywLASyqwLnzgSMHzNwxgADnPoOn+D/DWk3YutK8O6TY3AUqJraxjjcA8EZVQcGmx+DPC8VjNZReG9IS0uGV5oFsYhHIy/dLLtwSMnBPStFXjGTkr62/MEu/e/wCehz3jzSLXUdV8J3N3NexmTVEhbyNQngUAwTEYCOAG3YG4fNzjODiun0WPTLL7TpumX73Mtq4NxHPqEl3LCWGQGMjsy5AyATjuKlk0LSJtHGkTaVZSaaqhRZNboYQAcgbMbcA89Kl07TLDSLJbPSbK3sbVCSsFtEsaKScnCqAOtZSqJw5LvQVno32/zOY0VvO+LvihrraZrexsY7YE8iFvNZsD3fOfoK6W+0y31C4sprgNusp/Piwf4trLz7YY/kKhm0O2l8SW2uK8sV3BA9s3lkbZo2IO1wQc7WGQRggk9iQdKsU7WHJKV0zgPiVc3Gl6jomp2YczYuLSPZ13yx4T/wAeFZ/hk3TeOZ7DWbyV30Ownie7mkO9leQFXLHp8hHPtXpFzZWt75X2y2huPJcSR+bGG2OOjDPQ+9NOnWJnnnNnb+bcpsnk8pd0q4xhjjkY7GtIzSja3f8Ar8WcNTCylW9opdU7elv8o/ceYaBs1PUtR0yfx9dpZw3uy02aiv2m5JAwRIScr1G1Rgk+1a3xEuILCGG5h8U3drqNsYYzZR36xiVS43O8YwScEnPA49K66HwvoFvMk1voemxSxsGR0tI1ZSOhBA4NSXnh/RtRuDcahpNjdTEAGSe2R2IHQZIzVe0XMn2/r8TNYSfsZwuru/fTf8uhn64bPxL4TvhpmrM0aoxM2m3IJ3KM7CwzweMj0Nct4Dk0Swt9Gk/4S29ury6gWJdMe982ONyuSvlgZTGDjPAr0Cx06y0yEw6bZ29pEW3FLeJY1J9cAdeKjtNF0uwuXuLHTbO2nkBDywwKjMCcnJAyeaiMlFNdGbToTnONR2uvXujhn1bTtL+JVnJH4ukuLO5W4e6huNSVoIHGNqgZAXknAPpU3iu5WfxtoEVl4mnsodRikSVbe7ATYV+R1XpuYkgNzyBjkV1T+FPDsjs76DpbMxyWNnGST69KnfQtIkmgmk0qyeW3VVhdrdC0QX7oU44A7Y6U1KPu+X/B/wAyHh6rU1datPr5f5fiefDxDe6P4f8AE9pputyahLptxGsFxO63E0cTbd7nH3tpLdemMVSukg/t7QdO0bx7q2oJqE2LkR6hvdAFyCCPudT8pz+lenwaNp1nJPLp9ja2k86kPNDAis2eckgc8881y2ifDptO1q31DUNUhuhbMZIorXTYrMF8YDMU+9gE4B9fqDcKkb3fl89P6fzOetha1lGOq167K9+66O3XRff24GAB1+tUdW1Gy02weTUby3tEcFFaeVUDNg8Ak8mr1Vr7TbHU4li1Kyt7yNW3KlxEsgB9QCOtc7V9Geu7pe7uee6PqVjJ8FfIg1pbS7tLJnY212qSxOGJUHnI3HAx3zisy8j1m302+ZfF19IbXSoNRXy7oEySnIOCOTEAOfUkEmvS4/DWhRQyxRaLpyRzACVFtECuAcjIxzg881N/YullSp02z2tCLcjyF5iHROn3fbpW7qK7a6u55iwc3CEZP4VbS55r4u8Rpf6jbx6bqd/5kVsjSm21yDT4W3DcCu8HzD6kcDgU6DxZfXPw10oyax5Esl4LTUL5GEkttEWYBiR0JAA3e+fevRX8P6NJ5HmaRYv9nAWHdbIfKAOQF44554qOfQLNdJvbPSbe0057xCrSR2iMpJGMsmMP16Gj2kOW1utw+qV+dzc91b8P66/cef2Kq3xDsdN0jxlqmq2cdu10yi/83MinhGZeGU4Hy9ufWubj8S+I70vfyajcQah5jbVbXra3hjIOADauM4GMYJ5616T4Z8Cf2Fqx1G81CO8nSMxwrb2MdpGgb7xKpwzHAGfQfTG7deHdEvrlri90fT7md8bpZrVHZu3JIzVe0jF9/wDhzFYOtUhe/K77X6WS8/P7xkMX/CReEUi1IKv2+zCziCRXA3LztYZB68HkVpRRLDCkUYwiKFUegFOVQqhVACgYAA6Ukil4nRXaMspAdcZX3GcjP1rnk97HrwhZK+9rXOP+G5K2viO3jLfZLfxBeR2q4wqJuDMq+wkaQV2VUNC0a18PaHa6VYbzBbJtDStueQk5Z2PdmYlie5Jq/S6JdrfgW9W2FFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUyOaKYExSI4HXawOKfQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGdZjypL/yIgSr/ACoMLk44HtXg1xdfEXxt421jdpniOyFjLFA9jpHiiC0jswUDYb5CJScltw9cdhXvsNpOlrcfvwlzOCfMCgiNscEA9cV57o3wz8aaPrt3qcXxI3vqFxHNfL/YUI+0bAFxncdvyjHygetd+DqQpuUpWvbS9+/l5fp0uTO/LZf1/X9dDT1fx5qVjr0nhrwf4bu/FOoadDG1/K95HapFuHygyMMM5HJAA6+xAdF8R7jUfAt9rmh+Hbi51HTZmgv9JubhLeS3dBmTLkEEAcjA5HYHina94A1OfxNca94O8V3Hhu9vY1jvlFol1FcbRhG2OQFYDjd6enOYl+G97Y+B7nRdD8T3FrqOoTtPqWrXNqlzJds67XBRiAoIxjByAOp5NH+zuC2vp33636W7W1GtJLt+lvz/AAG+EviNrHjCTTbnT/Bd4mj3Ma/aNSlukRYZCuSERwrSoCQN4xzuwDjnv64rwH4M8ReEI4rPUvGH9s6Vb2wgtrL+zI7fysEYbeGLNgAjB9a7WscT7L2lqVreV/xv1JhzW945f4iTTQ+CdTY6Raatp62kz39vcX8lqWiVCSEKRuSSAe6/WujtJBNZwyKuwPGrBc5xkdKw/F+iax4h0e60vTNVsbC1vbWS2uPtGnvcSEOpUlCJkC8E9Q1XtCs9XsbPyNav7K9KBVia0snttqgY+YNLJk+4x9K5l1uW+lvP9DTrzjxX4h1my+J3hy0udHv7fSXvhb29/bauiRXbyR5IltwhZguGxkjkZzzXo9eb+Kfhz4s8Sa/Dfx+P/sdvZ3n2vT7b+xopPsrYIHzbgXwCfvflXThXBVL1Gred/wBOv4Cl8LsbHjvxxd+C7jRjHoEuqWmo3YtJJYbgK8Tt91VjwS5IDY5A+XkjNL4R8aahrut6hoviHw5LoGqWUSXHkG7S5WSJyQrB0wM5U8frWX4j+HvibxHo+hW1x452Xuk3P2t7z+yIj9omBJjfYGAXaCRjkHOSK0tP8F6tZ/Ey88VT+JftFtdW4tzp32BFwi5KDzA2eGLHOMnODWqWHVKza5tdfe6beWv5b2ZMr7ry/PX8PxOxrmPE3inUPD2sabbQ6J/aUGou0ERt7gifzgjuB5ZTaE+UZcuMZJIwOenrmdf8P67qmvadf6drOnWkWnSNLBDPpjzMWaJo23MJ0yMOSAAMHHJrmpcnN7+3z/QsstrGrWmgX17q+n6bp9xatx52p4tXTCneZjGGUckcx9V9CDWf4W8dQ+I9VudMDaXLcwwC4Emk6ot7AVJ24LhVZWz2K4wQQTyBe1fw/fa1odnBd39qNRs7mO7SdbIm3aRGJG6FpCSuD035BAYEECq2n+HNeh8Vrrepa/a3ZNs1s9tHppijVcggxnzSyncMncXz0G2tV7Fxd7X17/L7/Ml3tp/Wv+Rkp8QNdOmarqMnhaEWmizyxX7rqRJYRt+8aAGIeZhRn5vL54BPOOg1LX71dXXSvD+nQaheLAtzObm7NvFDGxITLBHYsxVsALjCkkjjOSngzW/+ET17RpddsGfWJZ3M66W4EQmz5g2+ec9flORjuGp+s2d/opttag1JY9RaBLK7aDQ572K4VSzIfIicyIVJbDbyPmIOSVxclRbtG349l+t9vyDXW3n+f+VyG3+IU0Wna9f69pEelwaNMlq6vfKXlnZUO0blVFQmRQHL8jkhecR6V8TbS71qz0++ufDrfbX8qJtJ19L1lkP3Q6GNGwemV3c4yAOaj8K+GrjVdC18a7LeEapqYu4LqSA20wKRxBZViYExYkiJRXBO1VznPO/aaZ4oFzb/ANpeI7OW2hYM62mleTLOAPus7SuoB77UU+hWnJUI3TWvq+y9et9xataef/AIPH+p3Vh4citdNuDbXurXkGmwXCnBhMr4Zx7qm4j3ArN8dSXWiXPhO40gahdXC6iLNbVL6RVuVNvKQsuW2t8yKS7BiME+x2PG2i3eteHcaVt/tKxuIr6yDkBXlicOEJPQNgrntuqPX9D1XxBcaDeWd7b6X/Z1wLx7e6szcOz+WybCyTKBgSNnG7nBBwMHhX6/hp/wS3t8n9/9WLOh67eX2qX2lazp0VhqNnHFMVt7k3EMkUm4KyuUQ5zG4IKjGByc1Q8I3tzF4g8R+Hryae5Gm3KT2087lmMFwC6oSeTsYSKCf4Qo7VctND1SDxxe63NqdpJaXVtHbC0SyZZEWNnZT5vmkE5lbPyDPHTBzD4V0e7t9V13XNWg+z3urXQCwbw3lW8I8uIEgkZI3OcHjfjtQv0/X8/+CPv/AF0/4f8AAu+JdS1rS7CGXw7oH9u3DzqkkH2xLby4yDmTc4IOMAbepz7VsVj+JdN1rVLCGLw7r/8AYVwk6vJP9jS58yMA5j2uQBnIO7qMe9bFHT+v6/r0EFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVR04Mst1G0skgRwFMjZPSr1ABRRRQAUUUUAFFee/F3x5qPgvw2Bo2n30l7d4SK+is/OhtSXVctkgbyG+RTkEjms/wAA+J9btdM1rWfHus6wum2MaE/234ej04p1yyeWzGT0xjOSuOtdUcNOVH2t9Pxf9XE3ZqPU9SorhNG+Lmi6rq1nY3Wma5ow1A7bC61SwMMN4x6CN8nJIORnGeO5ApNY+LuiaD4mudD1bTdZt7qJC1uzWYK3zZCqkGG3SMzMAOMZByQBmp+rVubl5XcOZHeUVmeHtbXxDo8eoLp2o6aXYqbbUrYwTIQe6nseoIJFaMjFI2ZUaQqCQi4y3sMkD8zWEouLaY1rsOorkLTWppvHNkdR0jxFpT3drJbww3VxataMVIkLlIpnPmYGA2OhIrr6QBRWfr2r/wBhaJcaj/Z99qPkBf8ARdPh82eTLAfKmRnGcnnoDXP/AA08RzeI/BrXt3dX15cQ3U0Mj3ljHaygq33DGjMBtBC9cnHNaKnJ03U6INreZ2FFeaJ8c9Be1luG0LxIkdrcNBfk6dn7BtxlpsMQg5PHLfKeOmfSIZUngSaFg8cihkYdwRkGnUo1Kes1YV1ew+iormV4LWWWKCS5dFLLDEVDSEfwjcQuT7kD3rl9I+INtrYRtO0TWJUW4+zXbrDGVs5fMMe18Od2CMny9+1SC2AamNOUldD2OtorD1fxHcaRNKZPDuq3NnCu+S9tjbtGq4yx2GUSHHOQEJOOAah1/wAaWXh+0027ls768tNSkWKCazjV/nZcxrtLByW6DCnnrimqU5WstwOiorC03xVbXr6jFf2V3o9xpsazXMN/5eViYEiQNG7qV+Rxw2QVOQOM5d78RYrC1hubnw3ryW93KkNnKYYR9pd3CoADLujzuz+9CcA9+Kaozbsl/TFc7GisK88TSQ3zWen6FqmqTworXK2vkKtuWGQjNLKilsc4UtgYJwGXOhpOqwazp6XdsssYJKvDOhSSJwcMjqehB/A9QSCDUuEkrsLl2iuY0jWb/VfEniGeIvLpmlOLGC0iVA086oHkbc2OfmVACwXhieuQ7TfHFlqegR6jHZXUFxNFJNb6ZPLbpdTpGcMyr5u3APXLDHfFReyv8/6/Mq2tv6/rodLRXF634uuYfBmk+MbRLiy08Swz3tncpGZJLWU7M/KWwy71kGG6DB64HaU7NbiCisfRPEP9tX+qWv8AY+q6f/Z0/k+df2vlR3XJG+E5O9eOvHUVsUgCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooApaekmZ55EMYmfcqnqB71doooAKKKKACiiigDgvij4a8XeLNNg0vwzJosVkZEmuG1BpRJvjcOm3YCNvHORn0p9z4Y8T+MfA+qaH4+n0e2uLllNtNoyysse0hlZll+9hgOOhFdtNIY/L24+ZwpzUldEcRKMFBW0d0+t9P8kJq7ueZDwf4+8S32lW/jzU9B/snTLmO7xpUMnnXcsZ+TeXACDqTs9xjB4h1zwv8Srn4kDxNpLeEilpBLaWSXZudwhZs5cKPv8AbIOOTxXqdFWsVNO6S69NNd/vFyq1v601K9h9s/s22/tTyPtvlL9o+z58vzMfNt3c7c5xnnFTSSJFG0krqiICzMxwFA6kmnUVyt3dylocU/iHR/Enjrw+PDuq2erC0FzJctYzrOsCmPaC5QkLknAzjPOOhrtaKKQGT4k/4SL+yD/wiH9mf2j5i4/tTzPJ2d/9X82emK4r4feGviH4Xu5bbVpPDMmlXV3cXlwbZrgziSTLYXcAu3djrzjPOa9LoraFZwg4JLUTVzyC28A/EBfBPi7R7mfw19o8QXb3KtG9xsTzsiYElcjAC7eDznOa9M8OW+pWfhqwttcNqb+GFY5jZ7vKJHA27uegHWtOiqqYiVRWaX9KwW1v6/iQXj3SWcjafDDPcgfu455jEjH3YKxH/fJrmvB1h4m0a0TTtWsNJW18+4ma4ttRkkcGSV5ABG0Cg4L4zuHTOO1dZRWSnaLjbcZ5xrXgC+1G91cyaL4a1WS/mZ4dV1bfLPbIQAIxEYyMLyBtkUfxYznOjd6F4lk0PwtbQ2mlG40meOa5D6hIqny0ZAEPkEnIbJyBgjHPWu2orb6xOyXb/hhNJ/j+O5yF94b1PWPEWurqMFlHo+q6WuneZFduZwFMp3bDEFGfOPG842984HL65r2p6rNpnh1r/wAO3F3baraCeLTL9ri6m8qdGcmDYPIACs7Es+3btyc5r1eiiFflabV7W+9BJXv/AF0t+hwms+Ao5PE17q1v4d8O68NQKvNHrKbJIXVVTKS+VJlSqj5CowckHnA6LwvpDaNpDW76dpOml5mk+y6TD5cMYOMDOBvbA5fauf7oxWzRUSrTlDkYW1ucp4XsrnRPFfiPTpbaT7JfXX9qWtysZ2N5iqskZboGDpnB5IcEdDji9c+EWuXupanqGm6raQ3H2hk0suXAgtJvONyjAKfmJuHK4yP3cfSvX6Kx7eX9flp9/cq718/6/wCCch460qXUfC1p4W0y2mEepTRWkksSHZa26EPIzHoPkQqAcZZgK6+iij1FsrIx9E/4SX7fqn/CR/2V9k8//iW/YPM8zycn/Xb+N2Nv3eOtbFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQXX/ACx/66rU9QXX/LH/AK6rU9ABRRRQAUUU2V/KheTaz7VLbUGScdgPWhuyuG5lDxd4bbU5NOXxBpZvon8uS1F7H5qNuC7Smcg7iBjHUgVJq/ibQdAeJNe1vTtMaYExi9u44S4HXG4jOMivNrOKKSDRrfw9beIrhYbyKa30PXvD2y2s1LZc+e9upR41Zir+cxyAPnzg6eqWeoaT441i/vNU8R2VvqJia2uNF0uG8XYkar5Tj7PLKpDb2GcId5x826gDvLrWdLsdJGqXupWltp5VXF3NOqRFWxtO8nGDkY55zUcHiLRLnUIrC21iwmvJoRPFbR3SNJJERkOqg5Kkc5HGK5jRWtfD/wAP7sJDr8ttLNN5C3OmZnbfk5W3t4gY0LZOGRSCSSBkVH8MoNGg8JaRpcOjXtpfWVtHcT/btJuIdtz5YSRxLMgDPyVyGJ28D5RTW7Xp+v5A9k/X+vmd5VAa7pB1o6ONVsjqirvNiLhPPC4zny87sY5ziq0/iHyPF1toP9j6rJ9ogM39ox2ubOLG75Hkzw3y9Mdx61ynxH0zXZtWsG8LrdRTavCdJur61B3WSGVJBMSOV2oJgD/edfWp7f1/Wo+/9f1odefE+gi7vLU63pouLCMy3cJu499ugxlpFzlAMjk4602LxX4dneySHXtMkbUM/Ywl5GTc4OD5fPz4II4zzXJ/D661eLSdV17xhY31ver5Nk8f2aSWSVbdApkVFUsweR5WGAcgiofhvHoyafNZXGh6jDqN9f3F1O19o10ivtuHkhJkljCDC7WUZGCeAGJquttxdLnV6h4it4vFGnaPaaxoqXTyk3djc3QF00flsV8qMHJbdtPIxt3Uvinxlong6yiudevobfzpUjijeZEeTLqpZQzDIXeGYjoMmue8Z67AfEHh+KKw1qc6bqwnung0W7lRU+zzLuDrEVf5nUfKT19jWh40aTWvh+1zplrdz7Z7W6EBtpI52SK4jkcCJwH3bUbC4yew5pLVJ+f4aA/it5f5m5pHiHRdfSV9B1ew1NYSBIbK5SYIT0ztJxnBqG08WeHNQ1VtMsNf0u51BWZTaQ3sbygr94bAc5GDnjjFZt7qlx4s0HWNO0C01SxlksZI4NQvbN7VBMykBQsoWTIyDuCbffIxXGpZC40ux0q6u/HAntJIhFpiaFaIlvJGwC7LgWghULjIdZQMdD2ppXdgeiuepalqNtpGlXWo38nl21pC00r+iqMn9BVbStVkuvDdvqurW6aWZIPtEsUkwYQLjd8zYABC9ewOeTjNZHxLsLnUvhrrltYo8s5ti6xp96QIQ5UfUKR+NL4nvtK1jwCLmW21DUNN1BYGQ6XbmeZVZlZJQgByFOHIw3A+63Qrv8v6+f6D7fP9DU0vxPoGtvs0XXNN1FsMdtpdxynC43fdJ6blz6bh61jWfxS8FXmoXlmviXSontZliDy38KrOSitmM7/mA3bT/tAjtWb4Y+0XHjkXcB1LVYPsLRXGpa3oi2VxbneCkUT+RCXRsuWXawBUHcM4aaDVH8NeMPE/2zStWuW1G5guLP7Fp8s6Tj7PHHt8xR5aHchH7xlA4JOOafXXt+v+Qu51l7rWl6beWtpqOpWdpc3jbLaGedUedsgYRScsckcD1FVP+Eu8N/Y7q7/4SHSvs1nIIrmb7bHsgcnAV23YU54wea5r4l3EF/oGnadeaXq032i9s7i4gtbC4nMcKzKZAzwKyghQeA2T2zxV/VLSz1/4e63F4b0ySCa40uXT4I5tPezcgRsI4wsiIwQFzjjaMnHepv7rfb/gf8MUkm0u/wDX/BN99d0mLVbfTJNUsk1C5TzILRrhBLKnJ3KmcsPlPIHY+lRat4n0HQZY4td1vTdNklBaNLy7jhLgdSAxGa4y68RnUL7wh9m0LXxBZXqveXEukTxiAtazRhSrKHb5mGWVSi92GRm/Z3sXhnXtfXX9N1GeTULvzob200ua8We32AJGxhRyuz5l2tjrkZ3Gqas7epC1Sv5fr/l+Jr+IPHfhzw1pcN/qWq2ghuJhFCFuY8yHzBGxXLAEIWy2PugGtDSPEOi6+kr6Dq9hqawkCQ2VykwQnpnaTjODXJ6xpiD4TpB4f0C8sIILmC7j0wxgzLGl0kz7Y1ZuSoZgnUZC4B+UdNpHiGPWrqVLTTdUit40ybq9sntUZv7gSXbITjnOzb/tZ4oXX+uwdDXrGuvGPhmxa4F94j0m2NtKIZxNfRJ5UhBIRst8rEKTg88H0pbTxD9q8WXuh/2PqsP2OJZP7QmtdtpPkL8scmfmYbuRjsfSuSl1Cz0j4y6zfX2k3lz/AMSyziS9s7KS6a3y0xKFI1ZwG2/eAx8mCRxme3mVbc7iXWdMg0catPqNpHppjWQXrzqIdjY2tvJ24ORg55zTdJ13Sdet3n0PVLLUoY22PJZ3CTKrYzglSQDg9K4rfq+neHdRvtDtLzTbLUdZWVEh08vc2lqyoskqWxBO5pFZ9pUkeYWKE5WofCsk3/CzpLqQ+Jb2C6037PHqGraQtsrsj79vyQxlQAWx5qjJJ25zVLV2/ra//A/qwnor/wBb2/4P9XO0vfFXh7TdTTTtR17TLS+fbstZ7yNJW3fdwhOTntxzVbVfEklhqWgJaW1veafrNx9nF0l1goTE8isqhCHUhDzuHUda5vTLuy0Tw7qGgeKNC1O8upp7g3fkaPNdR6lvYkSeZGjIdylRh2BXGDgAGk8Y6Zd32n+D9F0CLUdGmjulZJUtTP8A2fEttKnzyDdGCCyry+TnIJxmkunqv+D939WG+vo/+B/X5nZ6Z4i0TW5p4dG1iw1CW3OJktLpJWi5I+YKTjoevpUEmv58Yw6Ba23nOLQ3d3MZNot0LbYxjB3MzBuOMBSc9Ac7wJqEL6QdKXSL3S7mwd0nSe1uFilfed0sc0q/vQ7ZbJJY7styc1X0nNj8XvEUF1uDanY2lzak4wyRb45FH+6zKT/vin1X9dBdH/XU3dX1iTS77S7dLTz1v7n7Oz+Zt8r5S2cYOeFPpWd4r8T6j4eurGKx0P8AtJL2QQI/2tYcSnomCD1A69K6J4Y5WjaSNHaNtyFlB2nBGR6HBI/E1wvxNBvF0mwfRNT1O2W7S4ufsduzqYxkMuVIIY59vrVQs5JM5MS5wpTlF+n9We5t6B4nutT1e50nWNHfSr+CIT+X9oWdWjJwDuXoc9qvQ6xJL4qutHa02pBbJcC48zO/cSMbccfdPeuL8G6bHb+MvtHhvQtW0TSzbMt7HqSMiyvn5CgZmJI5yc8D0zz6KIYxM0wjQSsoVn2jcQMkDPoMn8zVTUU1YjDSq1Ie89nv3X3L02Q+isHxH4usvDV9o9lcW13d3esXX2W1gtFVmzjLO25lAVRyT29K3qyO4Kw4/EFvd+LxpVjq2jzLDBIbqzW5DXkcoZdvyA8KAW3ZGclfetyuB1XxFbH4haPeLYa49tZWt5DPKmhXrBXdodoGIvmB2NyMjjryK2ow55NW6P8AJiex1t94i0TTLlrfUtYsLOdIvPaK4ukjZY843kE5C54z0oHiLRToh1kavYHSx1vhcp5H3tv+szt+9x168VzHiE2t/wDEXwlJPp19cxW6TzCT+zp2jhkcIIizbNqnIY/MRtIBOODVvwhNG3iDxVHDZ3ttHNqIuIzcWE0CSAwRIzKzqA2XR+nXr0IJt0kqfNre1/xtb9QvqW/Dnj3w14pgtDpOsWT3V1F5i2JuozcKMZIaNWJBA6+lXvEmuxeG/D9zqk0TTmIBYoEOGmkZgqRj3ZmA9s1zXgXVpNN8NaL4em0bWDqFrEttc5sXjih2/KX86TbG68cbGYkHIB5xY+JYaDQdN1Nt32bS9XtL27244hSQbmPsudx/3amvCManLHRX/C+/3BHz/ryN/UdZttH0UXuuXlhpRKhS91chYVlI+7vbbkZ9gTjpUFnr1vbeEbPV/EGq6TEjW8b3F7BcBbMuwAJR2P3Cx4yeeKsavrVro9skt1FeyrKSq/YrCe6PTuIkYge5wK8+0fUwPgVLoz6Zraajb6G1s9u+i3asZGjKBVzF853EfdzgcnjmudvST9P1KirtJncr4x8MOVCeI9JbdP8AZlxfRHM3/PMfN9//AGetT6r4j0TQXhXXNZ0/TWnz5QvLpITJjGdu4jPUdPWvLL3Tke21QWOgXyfafBkVpahdImTDo02+L/VjY33DtOC3y4B4rqbfUrbSdf1vUdW0rVLuLWFiktJ4dGnmZoBEqm3dFQtGVcO22QKD5meu7FSVnb1/O342v/Wqjqr+n5J/rb+tOxsdW07VHuF0zULW8a1lMM4t5lkMLjqrYJ2t7Hmsrwzr+oa/PqEk2m21rY213PaQypeNJLK0UrRktH5ahQdueGasv4TRrF8P440tPsSpf3yrbZB8kfapfk4446cccVZ+Hv8AyBtU/wCw5qP/AKVSUdfl/kJ7fO35/wCQ9/GulXXibTtO0jxL4cmDSyRXls1+jXRbadixKrfe39Qe2cc1d8XazqHh7w3d6tpun21+LKGS4niuLtrf92iFjtIjfLcdDge9VvGf+r0P/sNWv/oRp/xB/wCSaeJf+wTdf+imqG/ccu3+Sf6mkUnNR7/5tfobttN9otYptu3zED4znGRmsPTtf1DUvF2raZHptslhpcqQyXbXjea7tCkgxF5eMfPjO/t0rX03/kFWn/XFP/QRXO+Fv+Ry8af9hGD/ANI4a0atNr1/NGUG3BNl7UPEVvF4o07R7TWNFS6eUm7sbm6Aumj8tivlRg5Lbtp5GNu6tLU5b+Gwd9Itba6ugRtiurloEIzzl1RyOP8AZP4VxvjPXYD4g8PxRWGtTnTdWE908Gi3cqKn2eZdwdYir/M6j5Sevsa7W1vI7/T0uoFmSOVcqs8DwuPqjgMv0IFZyb5HJF/at5f5lLwvrR8R+FdN1k2/2Y31uk5h37/L3DON2Bn64FM8P69/bQv4Z7b7He6ddva3Nv5m/bjlHBwMq6MrA4HXHas74Zf8kt8Of9g+L/0Gq3hJftfj7xnqtuFNnLPbWSSA/wCslgjIlP4FwmfVD6VpJJTaXn+f9feJbG1qOvf2Z4k0vTrq2xbamJI4rvzOFnUbliK4/iUOQc9UxjkVc1e7urDRru70+y+33MELSR2ok2GYgZ2hsHBPQcda5nx8v27U/Cel2wV7uTWobwKTykMALyP9MYX6uB3rrLy8t9PsZ729lWG3t42llkboiKMkn6AVD+Bv1/4f+uxX2kRaTqlrrej2mp6fJ5lrdwrNE3qrDIz6H1HY1brk/hfY3Gn/AA10iK8h8iWRJJ/J/wCeayyNIqn0IVwMdq6yqlo2iFsFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAEF1/yx/66rU9QXX/LH/rqtT0AFFFFABRRRQAUUUUAFFFVNMuby6szJqVj9gm82RRD5wlygchH3Dj5lAbHbOD0p20uBbooqrealaafNaRXcvlvez/Z4BtJ3ybWbHA4+VGOTxxQk3ogLVFFZs2uW0XiS20NUllu54HuW8sDbDGpA3OSRjcxwAMkkHsCQgNKiqd5c3sN7YxWlh9pgnlZbmbzlT7MgRiH2nl8sFXA/vZ7VcoAKKKpG6vv7eW0GnZ082xkN/568S7gBF5fXlctu6cYoAu1DaWdtYWq21jbxW0CZKxQoEVcnJwBxyST+NU7nXLaz8Q2Oj3CSpNfxSPby7R5btHgtHnOQ20lgMYIVueK0qACiiigAooooAKKKKACiiigArEh8I6TB4kl12L7et/MQZW/tO5MbgZwDF5mwqNzYXbgZOAK26KPMPIKKKKACiiigAqGWztp7qC5mt4pJ7fcYZXQFotww209RkcHHWpqKACiiigAooooAozaLp8+u22szW4e/tYXgglZm/do5BbC5xk7RzjOOM4q9RRQAUUUUAFFFFABTJYo54XhnjWSKRSro65VgeCCD1FPooAZDDFbW8cFvGkUMShI441CqigYAAHQAdqfRRQAVm6voNprflfbJtQi8nO37FqNxaZzj73kuu7p3zjnHU1pUUAVtO0600jTobDTYFt7aBdsca9u+c9SSckk8kkk81BZaBo+m6hc3+naTY2l5dkm4uYLZEkmJOSXYDLc88960KKOtw6WMODwfpEOqR6gwvrmeKQyxC81K4uY4nORuSOSRkU4JAIAIBIGKu6toWk69bpBrml2WpQxtvSO8t0mVWxjIDA4OD1q/RR0sHW4iqqKFRQqqMAAYAFUF0DR01ttZTSbFdUddrXwtkE7DGMGTG7GAB16CtCijzAKq6lpen6zYtZ6vY21/auQWguoVlRiDkZVgRwatUUARWlpb2FpFa2NvFbW8KhIoYUCIijoAo4A9hSWtpbWNuLeyt4raFSzCOFAigklicDjkkk+5NTUUAQm0tjei8NvEbpYzEJ9g3hCQSu7rjIBx0yBRd2ltf2klrfW8VzbyjbJDMgdHHoQeDU1FABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBBdf8sf8ArqtT1Bdf8sf+uq1PQAUUUUAFUdbtLa+0K9tr63iuYJIGDxTIHVuM8g8Gr1VtRtGv9OntUuprQzIU8+AIXTPcb1ZfzBoKi0pJs8c0P/hCx8NtPOltar41FhF9lLsBqRujGAgH/LQx5I4/1fl/7Fb3xDk8JyeNPDaazqunaderPIl7Ot3HBcLbm3lIRnyHWNmOOozu4IJrufDmhHw5o8Omrqd5qEFuixQG7EW6KNVCqgMaJkADqcn3rVrvqYlOq5K7369/l+BlFNI5eGDwNpvhOeWI6HB4cvn3vmSL7FIxwvAJ8sZKDgcEgnqTWN8Mdc8MnS73SNK1PSSw1O+eCytriPIh89ypVFP3NuCMDGK9Born9qnGUXd38/8AgFW0R4ZDY+B/+EA8Z6rbT6UbizvLttPuI54/9BYMzW4gIP7vcwyNuCx9cADrvHVr4Z1WHwv4h1S20m9sBfxrcajNHHJGLdopQA0hBHl+Yy9TtyRXotFbSxTclLXTz8kuwrb+d/x/yMLwrc+GZbKeHwbFZpYxSZZtPtvLtncjko6qEkPGCVJwRg4NZeikx/F3xQl1/rZrGxktcryYR5obB9A+c/UV2NU5tJsrjV7XVJYf9NtEeOKZXZSEfG5Tg4YHaDg5GQD1Fckpc0rj2Vv63PPPG+r+CLLx14dvZdQ8P2+s22rBb2dp4FuIovs0wxI2dwXJQc8ZK+1d5c+I9HtfDcuvyajbvpMUJna8hfzYyg6kFc7vwzWnXP8AijwwnibR5NNvpZpbOeaOSaBXCiQI4YIcg/IcAMBgkdCKjXlsVo5Xf9bmpY3tnrmjw3lrmayvYQ6ebCyeYjDjKOAcEHoRXJT+F/D+gePvDUuhaHpumSSm6WR7O0jhLjyc4JUDIrr4raWGIwrMwjKYVwBujPsCMY9ARxj0rKsPCzW2rQ6lqeuanrNxbI6232wQIsO/AYhYYowSQAMtnAzjGTmtL6E621Mz4hAy3XhO3twzXbeILeSJUPzBEV2lb6CPcD9a7KqcmlWU2swarLDvvbeF4IZGYny0cqWwucAnavOM4GM4q5SWit53/JfoN6v+vP8AzCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAZDNHcQrJE25W6Gn1R09Qt1ehQAPN7fSr1ABRRRQAUUUUAQXX/LH/AK6rU9QXX/LH/rqtT0AFFFFABRRRQAUV5k+oeL9PvLWbxDqOr6ZE18kd1P8AY7GfTtplwEj2N9oRX+VBJJnbuJYDqu7rGpX3h/x/ZXWqazNH4dv4GhSAxwiKC5VS3zvs37WRWI+bhlI5DAA6XDudhRXlcuo+LU+GFx4li8Q3i3VzexTadDc21sFW2eZUjSQCEE7lcMSMMOADwc9Ppp1vR/Gtppep67LrFvf6dNcEz28URhlieIHZ5ar8jCb7rbiNo+brl2d7P+tLh6HW0VxGmr4i8V6PJrdj4muNKklllW1sUtIHgiCOyBZt6GRmJU7trp1wAMZPW6ZHeRaRZx6pKk18kCLcyxjCvIFG5hwOCcnoPpSWqB6MtUVleI7250/RJbixm8mZWUB/7Mmv8ZPP7mEh2+oPHU1yNp4q16W8hjk1TcjSKrD/AIQPVYsgn++0uF+p4HU0LV2B6K56HRRWH4tOuDRV/wCEbEpn89PP+zCL7R5P8Xk+d+639Pv8Y3d8UAblFeead4k1XRrLxLPrF1ql3caZp4u4NL1Ozto7gIFc+Z5tuTHIGYFflwV2cjJy1azv/HKfZL2G08SX7syPNb3KaQlpIjfe2eXN5qDBypLvjAzup21B6f16f5nplFcZ4xm8RW/ibw/F4d1j7MNSnktZbe5hjeBFWGRzKMJ5hcbQQN4U4AOOa6PR7C+0+3lh1HV5tWzJuimuIY45FTaPlby1VW53EEKOCBzjJS1G9DQoorgo4/FF94/1nRoPE9xBpdvBb3Xm/Z7drmJpDLiKL91s2fIMlw7YXA6lggsd7RXn2t6v4kHiC38PWbavPJaadFc3V5o8FiklxI7MmcXT7VT5CSFVjlh8ygYa9pmt6xp3hfVpvF0Op2cdqcWt5NFbNdTK/Cjy4GkjaQMQB8oDZX5euX0uLrY7Oorq6t7K1lur2eO3t4ULyzSuERFHJJJ4AHrXB6Hq/iO08YabYammvNZahHKCdbTT9ysi7h5ZtGyO+Q6ntgjBBvWP9v8AigX2pWviKfR0gu57W1sorOF4x5MjJun3qXbcVzhGjwpAHPzEeiDzOts7y11Cziu7C5huraZd0c0EgdHHqGHBFEd5bTXU1tDcRSXFvtM0SuC8e4ZXcOoyAcZ61zPwtLN8LdALlWY2gJKdCcnp7VF8MD9p8Hvq0ykXmqX11dXRbrv85kC/8BVFUey02rSaDpc6u1u7a+txcWVxFcwsWUSQuHUkEqRkccEEH3BpmoajZaTYyXuq3lvZWkWPMnuZVjjTJwMsxAHJA/GuW8JN9k8feM9Ktyos4p7a9SMD/Vyzxkyj8SgfHq59an+JH/Im/wDcRsP/AEshpdl3t+I+p0lnf2mo2Md7p91BdWsq7o54JA6OPUMOCKg0zXNJ1tJH0bVLPUFiIWRrS4SUIT0B2k4rL8d7j4VZMBoZby0iuAehha4jWQH/AGShYH2Jpt8PI+IWhraRqvm2N0k+xP8AlkhiK59gzcem446muynQjOnzdXzW/wC3VfX1IbaNSHX9HuNZl0i31axl1OFd0tklyjTRjjkoDuA5HUdxWhXKat/yVbw1/wBg7UP/AEK2rq6418Kf9btFdQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooArWdq9sZWkm81pG3E7dtWaKKACiiigAoqG1u7a9hMtncRXEYdkLxOGXcpKsMjuCCCOxBFTUbAQ3KlvK2gnEgJwOlTUUUAFFFFABTZY0mheKQbkdSrDPUHrTqKGrqzC9tUczp/w88O6b9kW3hv3hsiht7a51W6ngjKfcxFJIyfLgFeOCARggVn/ABC0HxB4s0xtEtNL0a40uSeCSZ7zUJI5JFR1dkCCBwuduN24kA9K7aijW4LQyda8Oad4j0mPT9Yim+zqyv5VtdywDcvI5jZSQDggHjIBxkCmHwppba5Z6u325r2zi8mFzqVwVCcZBTzNrZ2qSWBLFQTkgVs0Udbh0sc7eeBdBvbq5meK9h+1sXuIrTUrm3hmYjDM0UcioxbuSOe+a3ba2gsrWK1s4I7e3hQJFFEgVEUDAUAcAAdhUtFABRRRQAVla54b0zxELX+1EuC9nKZreS2u5baSNypUkPEyt91iOvetWigDK0jw3pmiG4ayimeW5wJp7u6luZZAM4UySszbRk4XOBk4HJrOX4e+G1IQWlz9lV94sDqFwbMfNuA+zeZ5W0HnbtwPSumoovrcDB1fwXo+uapFqGoHUvtMJzE1vq11AsR2lSVWORVUlSQSACQTmtaxsYtPsY7SBp3ijBAa4uHnc855dyWbr3JqxRR0sBj+GfCmi+DtKbTfDll9jtGlaYx+a8mXbGTlyT2HeqK/D7Qk1GW/STWUuptvmyLr18C4UkqDibkDccDoMn1rpqKOtwMvWPDmma80EmoQyefbk+Tc21xJbzRgjDBZY2VwD3GcHAyOBVeDwboUOl31gbJp4dQAF291cSTyz4GF3SyMXO0fd+b5e2K3KKAOag+H/h631W21QQ30uoWrbobufVLqaVR/c3vISUOTlCdp7g1NqPgnQtVvZ7q7t7kPdAC5jgvp4IrnAx+9ijcJJlcKdwOQADkDFb9FAGboOgad4a0pNN0aKSGzjJMcT3Ekoj9lLsSq+ijgdhUGgeHx4fuNTW1uA1jeXbXcNt5ePs7vzIA2eVZssBgYLNyeMbNFHmHkY/h/Qf7FF/NPc/bL3Ubt7q5uPL2bs8IgGThURVUDJ6Z70nivQ5fEfh6TT7a7SzmM0E0c0kJlVWilSQZQMpIJTH3h1rZoo7eX6B1uU0s5LrRzZ66ba+aWMx3PlwGOKUHgjYzMQMdixqvpXh3TtGmkms1uHmkQI011dy3L7RkhQ0rMQuSTgEDNalFaKpNRcU3Z7roKyOfbQNQuPHNtrl3qVs1pZ280FtaRWbJIBL5ZYvIZCGwY+MIvWugoorPpYYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAVNVe6j0m5k0+WGK5SMtG88RkQEc8qGUn8xXG2PiPxcfBVv4u1JdHFl9iW+n06KGQS+T5e5mWYyFd2MsEKY/h3fx12OrtMmj3ZtrWW8mMTBIIWQM5IxgF2VfzIrh/B3gy7PhOy0zWpte0+1t4oorjSLme1kgmKqu8K6B5PKLA/KZBxldoX5a6qPJyNytuvu1v+n+aCWy+f6Gx4p1vxDpeuaNDoUVhew6pK9uLedGjdHETv5hl342DaMrsJwDg5IrWtYvEX9mXUV7eaYb7f/o11DaSeUVwOXhMmc53DAk5GDkdKqa14Ul1nWLTUB4i1WxezcvbxWqWxSNijIzfvIWJyrMOSRzwBV+TS7t9GSyXXdQjuFPN+sdv5zc55BiMftwg/PmpbjyJK39MRm+B7qe60C5FxDYwzW+o3du32G2MMTskzqX2FmILEFj8x5J5rnE8UeNBoWu6vJ/YjW+g3E8bxi1lVr5ISS5U+aRCdowM+Zk54AxnpfD/hKXw/NIYvEWrXcMsss8lvcpbbGkkYsz5SFWzuJOAce2OKqp4DVfDuraO3iHV5IdWkkkuJWFt5g8zPmBcQgANnng47YrVSpqcm7O9unnqC8+/4a/8AAK/jabX11bwydE1W2soLnUViaOW0kkLN5MzfOVmQMmAPkx1AO7jFdRpkWpRWhXWbu1u7jcSJLS1aBNvYbWkc5685/CqOoeGk1PSLKzuNTvhc2MqTwagnlCdZFBG7GzyzlWYEbMYJ4qzpGjnSlmaXUr/Up52DST3swbOOAFRQsaDH9xVz1OTzWUpRdPlT28v1Fro32/zOfsiNY+LeqPc5KaBZQQ2sbDhZJwzySD32qi59Nw7mtrWtE/tXUNHuQyL/AGfd+e27qV2MMD3yV/Kq/wDYVzbePTrtjJGLa9sxbahC7MCWjJMUicEEjc6kHHBBzxg79YJ2s1/WoTippxez/wAjz34qaZp039iX19YTXri+SB44CxkkiIYlFUEZJI+vvVHwffWWlXfiLUtIsrrTtE0+2/eabcTZm88DJYoSSnA28nn8MDsPEnhC38TXFrNc6lqdobU7o1s5wih+z4Kn5h60tz4N0261SC/lkufOS3+zT4lwLyPGMTDHz/p+grWM0ocvr/X9dzzqmGm8R7WKXT1fz3/ztY4O3+MVwtxHcXjaS9pI43WcC3H2mJSe7snlsVHXGM9q6Lxt41uvD+oW8NjqOhwJJFvZL5Z5JM567YlO0Yx168+lXIvh7b28iC28Q+IoLeMjZax6iwjRR0QcZC4465x3q1qXgq01DWG1KLUtV0+aUIJxY3ZiWfbwN3GenHBFU3SbVl/X9epEaeNVOSlK70/4PX/IzYPHVzd+CdL1K1sY5NT1Ob7JBCXKxeblgWJPIX5Scde3vWPb2msw/FzRrnxDaael3PDPi5sJJCkirHgKVc8Eeoxnd7V0Vt8PNJt9Fm0t7rULi3eUTQma4y1rIMkNEQBtOST3z3zUMXw3so9UttRfXNdmvLZspPNehmK/3Cdv3evAxncaFKCk2vP8UE6OJnGPPq1y9eqd306r+u8XmanD8VbIarbaayXMFwlrPbmXzViUg4YFtmTkdAe9W/Emu+INN8UaZp2k2+nzw6jHIsfnlwyyKuSWIONgyp4BJwR6VBcfDe3udQF7J4k8RCdC3luL4ZiDdQp25A9qv3Xgu3vNU0vUJdW1bztMRUixcjD46lvlySw4YjGRUpx92/T/AIJryV0pqKtdprVeV/yf5GbH401HT9D1iXXLG3l1DS7hLcpZuRHM0m3Zgtyo+YZzVO/8QePdMvtMsru00DzdSl8uJ0MpWMgZKsM5z7jI4NbEfgPTov7Wa4uL7Uk1QEzW93cZQtnIIwAQRgAHPAFcroHg/XpPFOmXmpWl/awae5kD6jqyXhxjASNUUbc8Zz6fgahyN3fl+X+f9d8KyxMUo663St69dO1lrbZ/L1MZwM9e+KyfEfh/TPEOmeRrFt9ojhJlRfMZMMFIz8pHrWtWZruiDXbNLdtR1DTwr7i9hP5TNwRtJwcjnpXP6HrSScbNX8u5w/hLTU0D4XnxD4ftbddVktDLM9y8jJIqMSflB4OAcYxziln8b+L4LKeV9K0xWt7OLUJSWkwsTcbMf3yQT1wAMcmt2z+HlpZaLdaVFretm0uIxFsa7GIlzkhBtwM9DxyCatzeDLS4guIp7/UZFuLBbCXdMDuVTkOcry/J5PqeK6JTi5N7nlU8PXjThCPu2VnZrV9zF8YeOLvQ7q0Szv8AQ7VZoBI0d8J5JAT7RKcDHc9efSpYfHd3deCdK1KzsI5NT1Sf7JDEXKxCXLDcT12/KTjr2z3q2fh9Zi4jng1nXLeURpHM8F8UNyE4UyEDk444xxUE3gGxsfCt7p9t9v1PLfaLaGe8CNFKMkeW+MIcknJzz1pXp8tvP8C+XGc7lfS3423+/wAvvKyeIfGdt4sstF1O30QebG1zLNB5pDRKRuC5OQw56jByK55/jHcvK93B/ZSWisdtjMLj7S6j/bVDGCevt3rW8HeFNah8WjWNZt7u3WG3aH/iYait5NLu6AMoAVRzwecnv22n+Hlmsr/2dreu6XbsxZbSyvikUZPJ2qQcZOT+NV+7i7P+tf8AKxio4upDmptrXrvay9OtzT1CGPxb4JmS0IVNSs8wmTI2llyufocVpQWccWmx2UirJGsIiZSMhgBjGPSpLeEW1tHCrSOsahQ0jl2OO5Y8k+5p0hcROYVVpNp2hmwCe2Tg4H4GueTWqWx60IO6lL4rWOS+G80iaJqOjuxePQtTn023dm3MYU2tGCfVUdV/4DXX1h+ENBk8PaAILySObULmaS7v5ogQslxI25yuf4QTtHThRW5Q/P8Ap9TTq7BRRRSAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKFpcSILv7XNvWA8tsxgAZPArxTV/wBoKa88S3Ft4T1Lw5badGUjgm1e2vmkumI5ZREnygE7cNg8Z6Hj2i0eXyb25ghaQuS8KE7fMIHAyemT3rzLwxP8RNJ8Xa3qNz8OsQ67dwyyH+3Lf/RVVAhPGS/Az0HpXfg4025Smk7LS7S6+fl/WqJm2o6f1/X9dTtvEPxE8NeDFtYvF2sW1jeTRhjBErzHpydqqWC5BAJAzirF5498MWPhNPE0+rwto0hULdwq0oyxwBhASDnggjIPXFcpqek+K/CvxH1XxL4a8OW3ieHWoYkdGvUtZ7QxgDaHfgxtjOBznrjAzW0zw54q8M/DnWha6BbX+r+ILuWeXSrO7SCGxWVNp2s/BK4GQOCTwcDNHsaLgpX1duq+aturd3oNfEl0/wCBv/wDqrP4neDdQ1nT9Ks9dgkvtRiWa2hCOCysNygkrhGI5CthuRxyK6uvNvhFY+IfD+jQaBq/geHQbe3t90moR38Mpu5+ASyJyCRzkk42gelek1jiadOnU5aeq73Tv92xMG2rs5/xnrcWi+G7qR7+70yR4X8u/t9MlvRakDPmOqKwAHX5sCt2FxLBHIjb1ZQwbGMgjriuZ8fvqd14V1PSNI0G+1ObUbCeBJbeW3SOJ2QqN/myoe+flBrT8PXl5daaqX+i3ulPAiptu3gbzMDkr5UjjH1wa5lrct9Pn+hrVw+s/EWwtPiNpfhWz1SwS7eby761ube4MpDIGjETqvl5ORnccY967ivJ/HjePdW8UaU2leA/tNnoeqC8huf7YgT7WoVlxtbBTO7vnp0rpwsIzqWnt6pfn+QpX5XY7vxB418PeFr6wtPEGpJZTagxW2DxuVYggHLAEKBuHLECmeFvHXhvxrHcN4Z1WO++zMFmUI6Mmeh2uAcH1xjg1xPj6Dxn4o0Dww8XgZZbi31JdQvbB9UgIi8okLGWOA28NnIzjGCDWtpVn4lT416pql14cFvo91Yx2i3322JsmMswbyx83JbbjHGM5rVUKfsrt+9r9pW08vPprrurkyk+nl+f9M9ArG1fxboug30VrrF09m0qM6Sy28ghIVWYjzduzdhSdu7PtyK2a4vxdHqN14o0GWDwte6ra6XdtdPLHJahWJhdV2CSZTuV2U5IGMZBrmpRjKVpf5fmWbtt4n0280u5v7f7Y8do/lzxf2fOJ0bCnHklPM6Mp+70OelO0rxLpmszTQ2ck6XECh5Le6tJbaYKejeXKqsVJBG4DGQRnIrP8Rf2zqnhi3fTrK/tZWuI2u7KO5iiumgDfOiSK5RWPByHHGQGU8jE8N6Xqdl46N+fD+sWunTWLW4k1DWReyIwYNlleZ9gOCBsZs9W21qqdNxbvZ69V0/O/SxLbt/Xc1k+JXhZ9+L64URSeXcM+n3Cratu24mJjxDyDy+3pnpWtq/iLTdEkhivpZmnnBMVva20tzMwHVhHErNtGRlsYBIGeRXFJputN4A8YWJ8LXsd5q1xePBC01rmXzwwViRMVG0bd2Tnpjditsrq2n68mvQ6De3qXunxWs1kk9uJ7V42dgfmkEZVvMOcOSCq8HJxUqVJPR/iuy/r8Nwu/wA/zt+Wpf0/xv4e1SC+uLPUM2+noHuriSCSOKH5QxUuyhdwBGVzuXoQDUmn+LtI1LUIrKFry3uJlZoUvtPuLTzsDJCGVFDkDkhcnHPSuK0vSb/xTo3jOAxLpt5Nrsc6RJcFQGjit38t5YuQx2bWZCdrZwTgZ0NP0WSbVNPaTwt4igeGdJXm1PxNJNbxFerKouJDIc9AyAHuVqnRord66dV2T8r/AC7A2+nmdN4t12Tw94dlvLWFbi9kdLezgYkCWeRgkanHONzAnHYGrN82tx6XENNisLm/ICyPcSPDEpxy4UK5PP8ABkcfxd6wviRbzHw5a6nBE039i6jb6lJEmdzxRv8AvMAdSELHHtW/d6o0Wmw3umWNxrCTbSi2UkWWQjIcGR0Ur06HPI4rlpay2T167beq8/uKfT0/r9PvKvhTV7jXPBumateRxi4u7VZpEhBC7iM4UEnH4msttW8U2E2mXWrQ6aLbULpLd9OhR/tFtvzg+bvKylcZYBFGAxBIXJPAx1bTfBlppmoaBeWl1p9kqDzZrcpO4H3VKSMR9WAFUNIk1+XV4NT1zwpqUmpyOsSs9zai106JmAfywszMxC5JfbufGPlGFHq+yhGtVty8t3bVarW1tfve6/B5391f1/X6/lualqGrz+JBo2iyWdmyWgupLm9t2nDguVCIiyJ02ks27jKjHzZC2mq6vq3gd9Q062totWMUohilLNC8qMyjkEEoxXIPHDA1n+JdHW68RJd6l4ZfxPYm1EcNti3dbWQMS77J3VfnBQbgSR5eOAauadMPCPgE3OuEwRWMMs7x795gj3MyQg/xFVKoMddoxXNiFSWFTi1fTtfZ3637brTpoWr+0/ryNHw7rlt4l8OWOsWWRDeQiQK3VD/Ep9wQQfcVDP4s0W28XW3hie926xdwG4htvKc74xuy24DaPutwTniqXw70i70PwBpdlqSiO72PPNGBjy3lkaQp/wAB34/Crs+pa1H4uttPg0DztHkgLzat9sRfJk+bEfkkbmzhfmHHze1cD+KwdDYooopDCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKADp0oqpY3E8zTJciMNE2393nH61boAKKKKACiiigAorK8TeI7Dwn4fuNX1XzTbwYGyGPfJIzEKqKO5JIA6Dnkis7wr40Him4uIf8AhG/EOjGBA27V9PNusmTjCnJBI9K0VOcoOaWiE3bc6aiiisxhRRRQAUVSN1ff28toNOzp5tjIb/z14l3ACLy+vK5bd04xV2gAoorO0LWP7c0tb3+zr/Ttzun2fUIPKlG1iMlcng4yD3FOzauBo1T1TR9M1u1W21rTrTUIFcOIruBZVDAEA4YEZwTz71cooTad0BBY2FppllHZ6bawWlrEMRwW8YjRBnPCjgcmp6KKG23dgFFFFIAooooAKKKKACis2DXLa68R3ejW6SyTWUKS3EoA8uMvnahOc7yAWwBwMZIyM1YfFlnP4oTQ0s9QWZ45XWea0aKJjGVDBS+C33lOVBU561pClOom4q9tRNpbm5RWVq+vR6Xc29pFZ3Wo31yGeK0tAm8ouNzkuyoqjcoyzDJIAyaTTvEljqGi3OpOJbKOzaRbyO7XY9syDLh8Ejgc5BIIIIJBqvYVORTtowvrY1qKwdO8Vx3t9a21zpeoad9uVmspbtIwtyAu4gbHYq235trhTgHjggXI9ctn8TTaEySx3cdst0hdRsmjLFSUIPO0gBgQMbl65qalKdOSjJb/ANfo/uYJpq6NKiiisxhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBR05hJNdyJyjS/K3Y8VepERY0CoAqgYAHaloAKKKKACiiigDxX9ol/CJ0nT014/8AE5EiGzH73/UmVPO+78v3R359KktIvDI+FfiaX4DicXT+Ws7W32jzuDk+WJed+xmxt7++K9kZ1TG443HA+tOrtjiuWiqWujvvo9U7NW2077tsTV5qX9f1+h87aK3g4+JPD/8Awp19XPiBrqM6ru8/a1tx532rzPk6/wB3jd0/hqHxnqHw8j+N9wbrWbrTLWW1lg117VZ91xMHUGEnaSFZVwxjwCARkZr6PorZY5KfNZ7Nb66+dtl0VtGTy6Nen4fr59jE8H6Xoek+FbKHwpG0ekSoJ7ZWlkf5X+bI8wlgDnOPfpWzJGksbRyoro4KsrDIYHqCKdTZFLxsqu0ZYEB1xlfcZBH5ivOnJzk5PW/fUqKscRP4X8P6B4+8NS6Foem6ZJKbpZHs7SOEuPJzglQMiu5rAsPCzW2rQ6lqeuanrNxbI6232wQIsO/AYhYYowSQAMtnAzjGTnfpDOZ+IV14WtvBtwnj19ui3DpFMMSnc24Mo/dfMOVHT0rzr4IeM/CEOmXPhWxv9txNqV3LZ2vkzHNvyyneVx9xe5z+Ne10V0QrRjRlSabv56fdb+kJq9rdD5VsrbwHN8P/ABpqNtdXIn0jUpZND2S3CraiQ/uHUcfMxjwS2SNgzivpTwtq9vrvhPTNTtJzcRXFujCUqQXOME4IB6g1rUVpiMUq8bWe/V36W7dXqTy+9f1/F/oVNVk06LSbltce1TT/ACyLg3hUQ7DwQ+7jBzjmvMvhvY+BNQs4UW/0y+1VL+4lsrdbyMy2caXMjIIEUgxoQAx2/eB5yuAPWKKxp1eSDir6+Zb1Vjxzxhd6TqOp65Lfv4V03UrV2gghutLe41Wbao2SxskiSDOQUKK2AAd3BC2vFV74b1vwx4DudZ1SxZ7m6hBuzdIsjR+UwmAkzkKWwr4PUgHBxXrNFbRxSjy6befl6dSZK9/n+P8AkeaRtp+j+JPF2heBZtOtL5tHimt9NtZERVux56sREvCttEO7j+6T2rB8QQfDu60nSbfREsZ9Tn1GyS7iiw9zIjXEfmi8B+cjP3hLzvx3r2msC48LG/1JJ9T1vUr21juVuY9PkECwK6tuTlIlchTggM55AzmnTxCU1J30t13t8v67hJaO3n+KOG8S2dr/AMJpfr4pu/C9haKsa6V/wkOlG4h8kIu4RO06Rq2/duUDdjaSSu3HZeAQieFxFb6rFqttFO6W88FtJFEIweEjMjuXQHIDBiuOAcCumorKddypqD/T/L9R21ucd4EJTWfGMFz/AMfi648j5XBMTQx+UfcbRgfQ1T1vxb4ct/iXov2jxBpcX2S1vorjfexr5LloMK+T8p+VuDzwfSuxTSbKPWpNWjh23ssC28kquw3opJUMucEgk4JGRkjOCauUYerCk7zi3pbR26cvZhJXb9b/AKnK+KfHOn6DJa2Md9p8V/fxmS3kvrhYreOPp5rOSNwGRhFO5vYZZYdPm0268B63/YFzB4mm2Ti6kB3Je3BiDFcpwQQyrhcgDC54rsKK0jiKcaajGLve7d99fS/3PzDXmued6Dd2d14i0FNO8XDxS0UcnnRM8D/Y18rHmjy1DKd2E/eFjiQ98mtLxCDN8VvByWwZpoIr6WfafuQGNV+YehkMePcV2VU4tKsodYuNVSH/AE64iSGSYsSSiFiqgE4UZZjxjJPOajE11WmpJWSVunm+iS69hRjyqxF4h/s3/hGdT/t//kF/ZJftn3v9TsO/7vzfdz059Ko+Bv8AhHf+EJ03/hCv+QFsb7H/AKz7u85/1nz/AHs9a36K5e5QUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQXX/LH/AK6rU9QXX/LH/rqtT0AFFFFABRRSEblIOcEY4OKAForzjw94Yn1TUfEttc+I9dSxsdSaCxhi1KbzIGMELFzKzlpMFuEclBzlWzxHqY1zXvHmr6csSyJYLELa2Pia60xjEyKTNst4iZAXLLuZiBswApyWFql5q/4Das35f5npdFedyarrGneA1g1HV7ZZI9UTTrvUbO+Fy9lAzhSZJWjQeau4IWZOMhjk5NbCeHLLRfE+lTWviXUrbzHkH9m32qzXK3x8tvuiaRmyud3y8ccjgEP+vwv+ojrKK84sbZNV8G3PijVvEuqWOpR+bLNNFfvFDpzxk5i+z58ohCu0iRGLc5JyKTULbUNd8ReC2vNQ1PSn1PTZZNTtbS7mgVyiRNsChv3ZDOQWXD4yM9wkm/6/r+u4bf12/r/Ox6RRXF6LJLpum+LdPk1yS3t9Mu3jtb/UZvONoj20UoLvIcuEaRiN5PGATgVjaZYanoHifw+Ndkv4RdStChtPEs9/HdzeSxzLDPGpVMB2/dHAbbuGACGtbedvxB6K/r+H9f5npAurdrx7RZ4jcpGJGhDjeqEkBivUAlSAfY+lS15y3hbTLn4vasLm61aPzNIt7j91rV3FyZ592Nsowg4+UfKueAM1f8dS2t38Jr/UdB1adks7RprK9sdSl5ZAVBMiPmQZ6hiQSOeaX2b/ANbv/Idve5f62T/U7eiucsvCX2dJZL7W9Yvbm5t3iumN9JHHIzEHciK2ISuCF8vacHksea4xNTRvhZ4J1C91u7j1WW4sIom/tKVHumaeJZVdQ4Ex2hshg2OenNC1dvRfeI9WqO4uIbS2lubuaOCCFC8ksjBVRQMliTwAB3rkrmIeIfHmoaRqt9fWsFjaQzWlrZ30loZw5YPMWiZXYKV2Y3bR1IyQRkeINCtv+FW+L7IeIr7WvsqXUiEahJ5tsy2+BbyMj5cDqVf727LAnkpuybKSvJI9HVldQyEMrDIIOQRWXq3iXTdFv7OyvmuTc3wcwRW1lNcMwUqGJEaNtA3rycDms3w5YaHopsFt9UuWvNQtQYILzWZ7jzlChmMccsjDjjlRwD6VHq3/ACVbw1/2DtQ/9Ctqtq0+X1/C/wDkZxd43Omu7j7LZzXHlSzeTGz+VCu53wM7VHcnoBTLC7+36bbXfkT232iJZfJuE2SR7gDtdezDOCOxqnq3h6y1qSOS8m1KMxghRZ6pc2oOfURSKG+pzVHwDLLL4IsGuJ5rhwZV8yeVpHYCVwMsxJPAHJNSUdFVO81H7He2Nv8AY7uf7ZK0fmwRbkgwjNukOflU7doPqQO9LY6tp2qPcLpmoWt41rKYZxbzLIYXHVWwTtb2PNcz4qtfsfi3wxf2t3fxS3eqC2niW/m8iSP7PM2DDv8AL6qpztzxR1XmHR+Sf4HT6lqMGlWL3d0ly8SEAi1tZLh+TjhI1Zj+A4pNL1O01rSbbUtNl860uoxLDJsZdykZBwwBH4irT/cb6Vxfgfz/APhS2jfY/tXn/wBlR+X9j8rzs7eNnm/u93pu49aOj8rfr/kHU7WivNv+Ku/6qB/5b9aPiPTdV1jWoI9OvFu1trNftGmNrs2mSwuxOJna1Vi+4KVwcKChK5ycAHcUV5Vca1fw/DnS47G7vX8/VpLHUbzU9UETRkPIHX7VHGdqGRRGrqgOCANjcrs+H7LXdD8V2sN9LaWdnexSKbOfxLc6jLMyjcHiW4iDAj+LD7cHJXIBp2/r8Qen9edv0O8orz2DQJtQ+JOvWD65rEWkxW1rcvZxajOC8shm5WXfvjQbfuRlQSFzwMHsmuLPw7oMb6tqix21pGiSX2oTqmeihnc4GScc8ZJpdNRta2NCiq7rb6pppCTNJbXUXyy207IWRhwySIQRwchlP0NcH4E8P3WpaLFqOpeItamkt9QuFtEW+kCxRRXUi7JAWPn7guC0m4hcBdpGSdbMXS56JRXlMUHiPxTqutuIw1xa30tusaeLLuwe0RWIi/cQwFPmUBwz79270wo17vVbvUfDPh/S5zBq+r6kX2z6brktnbv5IO9zcQqrHjGVVOWz8oC5B0v/AFqHVr1/A7+s/WddsPD9rFc6q8sUMkyQiSO3klCsxwu7Yp2gkgbmwMkc81wfh/WdbsdL8aW1s0d9daPHus7SPVH1LypTExMZmkRZGO5futkjOAcYApeMtK0218B2erw+KdSuZr27s9s0+ptLFqBedGKiEkxL0LDylXAXj5cgtLVL0/H+v+HF0+/8D1hmVELuwVVGSScACsseJdKPhc+IhcsdKEJuPtAgk/1Q/jC7dxXAznHI56c1kfFKaa3+FuvvbsVP2Uq7A42xkgSHP+4WrpYre1GnpbRRxm0EQjVAMoUxgD3GKl3syuxLHIksayRMro4DKynIIPQg1mat4l03Rb+zsr5rk3N8HMEVtZTXDMFKhiRGjbQN68nA5rE+FEskvwu0Yyu8iqkkcLSdTCsrrF/44F/CpdW/5Kt4a/7B2of+hW1V9q3r+TYuhu6rrFlotsk2oSsokcRxpFE8skrHnaiICzHAJwAeAT0Bp2n6tZ6rp5vLGR5IgWVg0To6MpwysjAMrA/wkA+1ZGpkxfELQJZmCwPaXkCE95iYXC/UpHIfoppfDrvN4l8UTI261N7HHGQcgukEayY+hAU+6kdq7HQh7Dn62v5fFy2tbfrv8ib6l7RPEeneIVuv7Na43WkoinjubSW2kjYqGAKSqrfdYHp3rUrlPC3/ACOXjT/sIwf+kcNdXXF0T8l+RQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBBdf8sf+uq1PUF1/wAsf+uq1PQAUUUUAFMlijnheGeNZIpFKujrlWB4IIPUU+igDBtPAnhCwvIrux8K6JbXMLB4podOhR0YdCGC5B9xV3V/D2i6+sS69pFhqawkmIXtskwQnrjcDjOB09K0aKAKtvplhaaYNNtbK2hsVQxi1jiVYgh6rsAxjk8YqppPhXw9oNw8+haFpmmzSLseSzs44WZc5wSoGRkdK1ahtbu2vYTLZ3EVxGHZC8Thl3KSrDI7gggjsQRT13DpYzrjwn4cu9YGrXegaXPqSuri8lso2mDLjad5G7IwMHPGBU114d0S+1aHVL3R7C41G3x5N5Lao80eDkbXIyMEk8GtGikBkQeEvDlst4Lbw/pcIv1K3Yjso1+0gnJEmB845PXPWjSvCXhzQrprrRPD+l6bcMhRpbOyjicqSCVyoBxkDj2rXoo2AzdX8N6Hr5iOu6Np+pmHPlG9tUm8vOM43A4zgdPSl1Tw7omuQQw61o9hqMUBzFHd2qSrHxj5QwOOPSrc15bW80ENxcRRS3LlIEdwrSsAWIUHqcAnA7AmpqAM7SPD2i6AkqaDpFhpizEGQWVskIcjpnaBnGTVVvBfhZ5Lt38NaOz32ftTGwiJuPmD/P8AL83zANznkA1t1D9ttRfixNzD9rMRmFv5g8wx5xv29duSBnpmjqBT1Pw5oetW8NvrOjafqENv/qY7u1SVY+MfKGBA4HarVlp1lpthHY6dZ29paRgqlvBEqRoDyQFAwOpqxRQBmaT4a0LQHlfQtF07TGmAErWdokJkx0ztAz1PWoNY8NDVtZ07VItUvtOutPSWNGtBCRIkhQsrCSN+P3a8jB681tUUdbgZer2+vzSRnQtT02yQKfMF5p0lyWPYgrNHj6YNQ6Zpc3h/w/b6RYyPO8cRVLqVQcyEkl2UYGMnOB9K2qKOlg63OY8GeGB4L8PW+iW2bmTzGmuL4rt+0SMcySMMnDE8AZPGOeKtXvgbwlqV7Leaj4X0W7upjuknn0+J3c+pYrk1u0UPV3AzZdFiTRF0vRZ30OGMARHToYV8oZzhVdGQA/7vemaRo6+HPD+n6Xpskk1vYQrComILyKBgEkADP4AVq0UAVBLcj9+UJiP/ACx2/OB6/X2qlqnhDw3rl4LvWvD2lajchQgmu7KOV9o6Dcyk45PFbFFAFdNPs49NGnJaQLYiLyRaiICIR4xs24xtxxjpiqWk+F/D+gTSS6Foem6ZJKu2R7O0jhLjrglQMitWijrcOljnj8PvBjTmY+EdCMpbeZDpsO4tnOc7eue9bGoadZatYyWWq2dve2kuPMguYlkjfByMqwIPIB/CrNFHSwdblWfStPutLOmXNhbTWBjERtJIVaIoOi7CMYGBxjtWfpvgzwvo18t5pHhvSLC6QELPa2MUTqCMHDKoPIraoo63DpYytW8K+HteuEn1zQdM1KaNdiSXlnHMyrnOAWBIGSeKmv8AQtI1XTo9P1TSrK9soypS2uLdJI02jAwrAgYHAq/RQBnaf4d0TSbgXGl6PYWUywiASW1qkbCIHITKgHbnnHSqj+CPCklxczyeGNGaa6z9okbT4i02TuO47fmyQCc963KKAK72Fo+mtp7W0X2NofINuEATy8bdmBxjHGKzl8NRJ4NPhtdQvhb/AGU2gufMXz1jxgYbbjIXjOM8Zznmtmih63v1BaWt0IrS1gsbKC0s4lht7eNYoo16IqjAA+gFZWseGhq2s6dqkWqX2nXWnpLGjWghIkSQoWVhJG/H7teRg9ea2qKL63DpYrX+nWOq2bWmqWdve2zkFobmJZEbByMqwI60trY21hYpZ6dBFZ28alY47eNUWMf7KgYH5VYoquaXLyX07B1uY3h/w4ugyahM2pXupXOozrPPPeCIMWWNYwAIkRQNqDtWzRRUgFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAQXX/LH/AK6rU9QXX/LH/rqtT0AFFFFABVTVXuo9JuZNPlhiuUjLRvPEZEBHPKhlJ/MVbqnq7TJo92ba1lvJjEwSCFkDOSMYBdlX8yKNehUbcyucdY+I/Fx8FW/i7Ul0cWX2Jb6fTooZBL5Pl7mZZjIV3YywQpj+Hd/HWj4p1vxDpeuaNDoUVhew6pK9uLedGjdHETv5hl342DaMrsJwDg5IrH8HeDLs+E7LTNam17T7W3iiiuNIuZ7WSCYqq7wroHk8osD8pkHGV2hflro9a8KS6zrFpqA8RarYvZuXt4rVLYpGxRkZv3kLE5VmHJI54ArvqOkqrWltfS3Tb8/xMoXtqW7WLxF/Zl1Fe3mmG+3/AOjXUNpJ5RXA5eEyZzncMCTkYOR0qj4Hup7rQLkXENjDNb6jd27fYbYwxOyTOpfYWYgsQWPzHknmtKTS7t9GSyXXdQjuFPN+sdv5zc55BiMftwg/Pms/w/4Sl8PzSGLxFq13DLLLPJb3KW2xpJGLM+UhVs7iTgHHtjiue8XGSbX9fIrov67nNJ4o8aDQtd1eT+xGt9BuJ43jFrKrXyQklyp80iE7RgZ8zJzwBjOn42m19dW8MnRNVtrKC51FYmjltJJCzeTM3zlZkDJgD5MdQDu4xVhPAar4d1bR28Q6vJDq0kklxKwtvMHmZ8wLiEABs88HHbFaWoeGk1PSLKzuNTvhc2MqTwagnlCdZFBG7GzyzlWYEbMYJ4rZ1Kakmrfd5f5id9fO/wDwC9pkWpRWhXWbu1u7jcSJLS1aBNvYbWkc5685/CuasiNY+LeqPc5KaBZQQ2sbDhZJwzySD32qi59Nw7mug0jRzpSzNLqV/qU87BpJ72YNnHACooWNBj+4q56nJ5qj/YVzbePTrtjJGLa9sxbahC7MCWjJMUicEEjc6kHHBBzxg8krc39dh9H/AF1KPiO51Gx8XeHXf+yrmwutQ+zxxy2DG4t2MErGRJvMwCdhX7nRiM11tcvrXgyfW9TgvJfFWtWwtbj7TawQJabIH2MmRugZj8rtwxPX6VV8Xx6+PAtzpOh3V0+qXG22j1OaSOJ0UuBJMSgQAqu4gKATgYGaj7JW8vl/mdhIHMbCJlVyDtZl3AHsSMjP5iuPjOuab440gazPod81/FNB59rpL288aovmbRI08mVJAyuPet/R4203RoLFjd3MtrAAGuZfMmuMD7xdjyxPXJ4J9Kxojq2veLNJvbjQLzSLXTRM8j301uzSs6bVVFhkk9ySxXtjOeK+1oT01IvGrHSfEHhfXbUAXB1JNLmGcebBcfKVPrh1jcem0+prsawda0G41rxJos1w0P8AZemSNeNFk+ZLcgbYuMYCqGduuS23jit6kvh+f9fjd/Mp7/L+v0CiiigQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAAORkciiqOmgxtcQbiyRSbUB7Cr1ABRRRQAUUUUAQXX/ACx/66rU9QXX/LH/AK6rU9ABRRRQAUUUUAFFefad8TbnVdXgsrDTdJmlnufJNiNdjXULdA2Hea2ZBtKqGYqrs3AGOpFjX/iVaaV4hudItbnw9HNZhRcNrOupY/MyhgqKI5GbCkEkhRyACTuwAdzRXMaZ4wPiDwXLrXh6Czu7iEsklu9+PKV0++BNGsgIxyCByCM7ecReF/Fms+JrXTdSTw0bTSb6JWM014BMhMe7d5RQZj3fKG3BjkHZtOaOtv61Dpc6yiiuV1jxo3h/xHdW+tW1raaLa6eb99Sa6dn2hgrDyREckMR0Y8MMZJwD+v1D+v0OqormNN8Uahq+tapaadYafNbWtpDcWtyNQf8A0gygtGrL5P7vhckgtgFeDnil4e8Z694igjuLbwqsdvFdNa3cj3+3DLKY2aANGPNVQu4lvL7hdxBFHWwdLnaUVyXiO6lg8Y+HBf6NaXFo1+YrO9GoSLNBM1vLljCI9pG0OvLnqDil+Imsa/onh+K68OQ2jMbu3jlkuLgxlQ88abVXynDBgxUk4Kg5GTxR29bB1sdZRWLHq+o6fo9/qPiyzsNOgs4jMWs757kFFBLE7oo8EY4Azn2rkLf4w2Imtpb+48NJZzyKhW08SRXF3CGOAXhCBTjI3BJHI5xuxRu7B0uek0Vh+NNdfw14K1TV4QrT20BMIboZD8qZ9txFRT3MngrwD5901zq02n26mWSSUs8z5G52dvurklieiqDgYGKA7HQ0VyHhXxpdeJdWlto7bRri1hhLy32j64l9FG5ICRsNiOGIDnO3bhRySSBR0TX/ABxeeI/EFrLpWiyxWV5FEqNq0iiBTBG5CkWuXzuLZbBBJHQA07a2A72iuc8Y+J7zwra2d7FpQ1GzkuYrecRzsJ1Mjqi+XGI2Ehy3Qsv68Vb/AMTeI9L8Matq1/4ZtYWsLX7XHEdU3CRAGZ0ZhEdsihegDISww55Im6tcdnex1tFc/eeIL601jQoRp1u9hqz+Sbj7Wwlhk8mSXHl+XhlxHjO8HJ6cVDP4i1i8v72Dwvo1pfw2EnkTz3mom2DygZKRhYpC23IBLbRk4GcHDejsJanTUVwPiXxV4oj8C2mr6Xo0emXUt7FDPBf3O2WAG6SILgROrbwTk5G0Nld3FdbpE+tTJL/b2n2FkwI8sWV89yGHfJaKPHb1/CgDRoorj5fFfiGTxdqOgaZ4btLiSzjinF1NqbRQmKTeF3EQsyyEocKAwxklhgAgHYUVzkfimR/Dmo3k1taWF/pj+TeW+oXwighcBWJM4VvkKurBtuSCMhTkCn4S8e2/ibV7nSy2lS3MEIuBLpGqLfwFCduC4VGVwexXGCCCeQDd2QbK519Fcl/wlWu31tPqPh3w7b6jpcTusckmpeVPdBDhmij8tlIJBC73TOOwIJzvFetrYx+DPEOp3Fzoqy3ii8hku5I40R7aV2jkTIViGVR8y5yOMZxQv8vxD/g/gd9RWH4Z1bWtZtXutY0WHSoHJ+zL9qd5nXcQGeNokMeQAQCSecEAjFULa9uNW+KN9bLcyx2WhWcQaBJCqzTz7jucD7wVEGAeMuT1AIAOrorI1qxv7rUNHlsJ5Y44LvfdKkpQPHsbhhkbhu28c9a5n4mrcW66TeW2oazAJLtLaS30uYq0iNknao6vxgVUY3aXcwq1nTjKVtjvaK878CapcDxZeaXLda0tsbVZobfX/wDj5Z92GZTj7uO2ff1rrYLG/TxjeXjzytp8lpGkcRlJRZAzbiFzwcBecd/rVShyuxFLEe1hzJdbGvRRUF490lnI2nwwz3IH7uOeYxIx92CsR/3yazOonormvAs8c3h2dYNNi05odQu4pLeO6adPNE772DsoJDPuPQYz0rGT4ga6dM1XUZPC0ItNFnliv3XUiSwjb940AMQ8zCjPzeXzwCecbqhNycY9PT/MV/zsd9RXIeMda8Radqnh+PQLaxltr2+WGQ3F20bSkxSt5ZHkvtX5Q28HORjGCTXR6ZLqUtoW1m0tbS43ECO0umnTb2O5o0OevGPxqHTaipdwur2LlFcpbXtxq3xRvrZbmWOy0KziDQJIVWaefcdzgfeCogwDxlyeoBDNW8W6xp3jGLQLbw9HfPeW7z2U0V6wHyMisZ8xYiUb+oLk4ACkkCs+39f13KtuddRXG6549bw7Y6fHrNtpmnatfPKFgv8AV0gtkWM4L+eVyVIK7QE3HcMquGKu8MfETTtdnv7W4utJFxYw/aXfTNUS+gaHu28KrAqeCGQdRgnsCOworjJvFvif+xv7asPCEd3p0ieZDEmon7Y6H7rGERFRng4EhYA9C3y1o33iDVJdbuNL8NaRBfy2aqbue9vDawxswBWNSscjM+0hj8oUAj5snFGqA6KiuM1Tx7c2Hg+41qPQ3e40+7S11HT5LjbJCxKj92VVhKf3iMv3chuoPy1f0nxHqsviKPSPEGiRabLc2bXdq8F79oDKjKro/wAi7XHmJ03KcnDccn9fqB0lFef6l8TLi21u902w0/SZ7q3uDbQ6fd67HaX90/AUpA6bdjEgqS4JXkAnCm94z1rxPper+HYdBtLB4L2+EMv2i8aNpD5MrGMgQvtX5Ad4OcjGMEmjs+4PS/l+h2VFZ1rPrb6TLJfafYQ6iN3lW8N88kL8fLulMKlcnrhDj36Vznh7xnr3iKCO4tvCqx28V01rdyPf7cMspjZoA0Y81VC7iW8vuF3EEUdbB0udpRXNWPiLVtQfX7WDSbIX+k3awRxvfuIp1aNJAxcQkodr9NrcjGcc1HF4ykuvDWg3tnpom1PXYUktrE3AVVJTexeQrwijqwUnphSTijpf0/HYDqaK5uy1rxNJqs+naj4at7aRbR7iC6h1BpbWRwVCxFzCrKxyScocAZG7kDK8F674y1aOd9T03SZLePVLq3kmGpuJIkSZl2qgtgH2gYBLKWxk4JoWv9edgeiv/W1zuaK5b4jSzReC5Ps9xcWzSXtnE0ltM0MgV7qJWAdCGGVJHBHWteWC80vRUt9At472aLCompahKMrnktMVlcn6g/Wjpf8Ar+tQNKiuU8HSSQ6rr9hc6Pa6ZdC6S8uPsl/JdJPJMnLZeNCpwgGAMVFdwGx+LOkm3u74R6hY3j3ED3szwsyGAKREzFEI3N90DqaOqX9bXA7CisLxzcTWnw98Q3FrNJBPDplw8csTFWRhExDAjkEHuKsaJYjTfDsUdk1xcSNF5g+3XsszM5XODJIXYDP1x2FHfyDt5/pb/M1aK5TwZcmXU/EkNzpFvpt/FqCG8+zXr3Mc8jW8RDhnRCPk2DAUD5ai8WQG18WeFr62u76KS51MWs0aXswhkj+zztgw7vLJyoOdueBzR287fiHRvtf8DsKKwfG0V63g+/n0m5ltr6yj+2W7RsRveL5wjAfeVtu0g8EMa0dG1JNZ0Kw1SFDHHe20dwqMeVDqGA/Wha3/AK/rqHYu0VynhG9uYvEHiPw9eTT3I025Se2nncsxguAXVCTydjCRQT/CFHajxZe3OieIfDmqQTTtbXF6NLu7UOfLZZ/uSbegZZFUZxnDMKO3n+v9WHbc6uiiigQUUUUAUrCOZZLiS4i8oyPuC7g3b2q7RRQAUUUUAFFFFAEF1/yx/wCuq1PTWRXxuGdpyPrTqACiiigApsqu8LrG/luVIV8Z2nscd6dRQ9UGxxsnhPxHqtpY2PifxHp9/aW00M8j2+jm3uJXiYOreZ5zKhLKMlUHGQNuci/d+HdTt9autS8Matb2DXxDXlve2RuYZHVQokULJGyttUKfmIIA4B5PR0U7gYUmla+3h2a0/t21l1Gd2L3M+nZhRG4KJEkikAdizseuSe0fhPRdb0DTLXTdT1bT7+zs7WO2gFvpz28nyAKCzGZweB0Cjn8q6Gil1bDpYx59N1qTxdbahBr/AJOjxwFJtJ+xo3nSfNiTzidy4yvyjj5feqHi7wTb+Lr3R5rm5aGPT7jzJofLDLdRblcxNzwC8Ubd/u4xzXT0Udv6/r+kByHhHwTc+DPDV7p2maslxdTT74Lm7ti6wxhVSOMoJAWCIgXhlz1qbwn4d8QeHYY7O81vTb2wWWaVki0uSGUtJI0hw5uGAAZz/CeBjPeupoo63DpY5LxB4b8S6zq1lc22vaVawafefa7WKTSJJG3eW6YdhcqGGJGPCr2/HQ1nQbvX/CTaXqF/El8THILu3tisayxyLIjCJnY7dyrld3IzyK3aKNlYOtzn4dA1O/tr638WavDqVteW7WzWdpZC2gCMMFvmeSQuckZ3gYx8uearw6H4shjjsx4stzZRsFEx0oG8KA9DKZDEWxwW8nHtnmuoop3sBk+KdDXxL4U1LRnlMJvIGjSUDPlv1VvwYA/hVby/EeqeEbPZcx6DreImnMkCXSBlYb12hgCr4OCGBAbsem/RSA5vT/DepHxNFrniHU7O8ura2e2t1sLBrVQrspbeWlkZ/uLgZAHJwSQRFceGNZh17U73Qdet9Pg1Vo5LlJdP8+WN1QRlonMgVSVVeGRwCM4IOK6migDmPE/hvWtbttPt7DW7S2js5oLlnvNOM8k0sTh1JKSxqASoyAvrjFXpNHvdV8L3+k+JL23uWvoZLd5rG1a3AjddvCtJJyMnnP4Vs0UraNdx3d0+xxJ8H+J57vRrm+8VWcjaPMskMMWkFIpR5TxsZAZixfa/BVlUc5RsjGg/h3WbDVL658MaxZWcF/L9ontr7TmuVWXADNGUmjKhsAkHdzkjGTXTUVV7u4v6/r7zn9T8NXWreDW0e+1iSa9LJKNQeBBiVJBIh8tdo2hlA25ztGCxPzVZ0iw1yC6luNd1uG93JsS3tLEW0Kd92GaRy3b7+3H8Oea16KQGPaabrUPiy9v7rX/tGkTRKttpX2NE+zOAuX80Hc+cNwem72rjZZNVk+LmqyaLdXunNNbW1sr3mg3NxZ3BTzi+XGwArlcP5gU7iPm4x6VRR28h33ORufBE1zpG2TU4n1dtSTVJLuSzDQyToAqgw7vuBFVQN+4bVbcWGTJpvhvXoPF667qviC1vN1s1tJaxaaYUVMhh5Z81mU7hk7i+e22uqooWn9eVvyE9f687/mcnD4Z8QaTDNZeG/EFlaaa8kkkMV3pZnltt5LFUcTIu0EkqGRsdCSABU+t+E5ddm0KO/vorqy0ybz7iK7s1kkvH8p4xuIKoow7EgRkE4xgV0tFAGD4X0TVtBhls77V4NQsEZvsUYtGjkt49x2xtIZW3hVwoJAPHJNQRaTe6d8R59UtIjNp+rWaRXZDKDBNCT5bYOCQyuV4yQVXjByOlooAK5rxX4Y1HxDdWMtjrn9mpZSCdE+yLNmUdHySOgPTpXS0U02ndEVKcakXCWzOZ0nwtqEGux6t4g159XuLeJo7YC1S3WIN94kKfmJwP89OldgiMxzhRk4GaWim5N7ip04001Hr5t/mcZ4s1HXpvFHh3TPDlxLZ2Ukj3eqX6W6yKtumAIwzKVDOzfUAE9q6jUFvprNl0e7tba53DElzbtOgHcFVdDnH+1U8cEUTOY0ClzlvelhhjgTbCgVc5wKSdjTqc34Y8Pa9oLzx3es6deWs9zPdOkWmSQv5krlzhjOw2gt025x371STwZrf/AAievaNLrtgz6xLO5nXS3AiE2fMG3zznr8pyMdw1dpRWvt53v+i6Bt+Zz+reH7/VNJ0xf7St4dU024S5iuhaEws4VkIMXmZ2lXYY35Bwc8Vf0iz1S1WZ9Z1ZdQmlYELFarBDCB2Rcs/PU7nbnpgcVo0VLqSas/yQrJW8jmotJvdO+I8+qWkRm0/VrNIrshlBgmhJ8tsHBIZXK8ZIKrxg5Fe/8OeJp/F41yx17SYPKtpLWGCbR5ZdsbujncwuV3NmMcgAdeK62is+3kVfcwtW8P3eoPYahaaillrVihRLoW2+GRX2+YjxFslGKg4DhgQPm65dYaVrEjXI8S6ta6jBNCYfslpYfZ4cH7zNueRyxBx94Lj+HPNbdFG+gtjk7Tw34nsLGHS7PxVbx6dbhYopDpe+9WJeg80ymMsAMbjCfUgnmrV74f1WHWrjUvDerW1i94ii8hvrJrqORkG1ZFCyxlW2/KeSCAvAIJPRUUasDjdR8D3934Tn0u21qFL2+vUvb++uLIyiaRWVsJGsi7FHlooGWwq4OSd1aEuhazN4v0zWX1ax8mztJLaS3Gnvul8wxl2D+dheYlwNpwMg7uo6Kinf+vlb8gON1Hwr4n1jS73SNV8Tabc6ZemRJd2i4uBC7E7Ffztm5VO0MYz0BIJrV8QeHZdVstNGnXq2d5pd0lzazTxGdNyoyEOu5SwKuw+8DnBzW7RSDf8AH8dzFtNP8QQafdm5123udQmIMTNp+22t8dliD7yCOu6U89MDiqHhPw74g8Owx2d5rem3tgss0rJFpckMpaSRpDhzcMAAzn+E8DGe9dTRR1uHSxyt14W1ddc1O70TX49OttWeN7pDYCWZGVBGWikLhVJRV+9G4BGfaoLLwTe2fhfQ7JNZiGqaCAllfpZEIyBPLKSRGQlgy/ewy8gEbcV2NFG39dgMfSbDX4r57jXdct7xPL2R21lp/wBmiBzku295HLdhhlXGcgnBGVa+FNc0+5u4dL8SRWel3V7JeGNNOD3UZkbe6rK7lMFiesRIBxnODXW0Udbh0sc549sL7UvCMkOl2j3tyl3azrBG6KzrHcRyMAXZVztU9SK0ZdaWz0VNR1Gw1C23Y3WyWzXU0ZJ6FbfzM/UZHvWlRRsrAcv4XeS/8Ra7q6Wl3b2d2beO3a7t3t3k8tCGPluA4GWx8wGcHHGCW3EGpX3xO027/si5g0/TrO6ia9lkh8uZpTCV2Kshf+Bs7lXpXVUUdU/62sHQxvGNhc6r4H1zT7CPzbq60+eGGPcF3u0bBRkkAZJHWl02XU7vwyqpZyaNfrGI0TUEjnCkADcVhlIZfbeD9K2KKO/mHby/r9DkdD8NeJdK1+/1C517SbmLUrhJ7qGPSJI2+WNYwEY3LbeEHUNzmpPEsGpaj4p8OQ2mkXL2tjqAvJ78yQiFV8mZNuDJ5hbLr0THPWuqoo7eX6B0a7mH4yttSv8Awne6fosQe6v0+yb2YKIEkO15TkjO1SzYGSSAMVp6dYQaVpdpp9oCtvaQpBECckKqhR+gqzRRsBzXhXR7u31XXdc1aD7Pe6tdALBvDeVbwjy4gSCRkjc5weN+O1HiPR7vXvEegQNBt0uwuTqNxPvALSxjEMQGc/eYuTjHyAZ5rpaKO3l+n9XH3CiiigQUUUUAV7+9h06xmvLkSmGFdz+TC8rY9kQFj+ANYVn8QvDV+bdrW+leC4ZUiuzZTrbFmHC+eUEYbJ27S2d3y43cVu6j/wAgu6/64v8A+gmvOdBOuar8JNO8OL4bliF5pMVsuoxzwfZUieIDzSN4l3BTnbs+8MZx81dNGnCcW5dGuqW9+/oEtEvO/wCn+Z2ur+LdF0G+itdYuns2lRnSWW3kEJCqzEebt2bsKTt3Z9uRT7XxNpt7pl1fW32x47R/LniNhOJ0bAOPJKeZnDKfu8g56Vg+LItRuPE2gPB4XvtWtdJumuXmSS12sTC6LtEkyncrspyQMYyDXRSapdpoyXq6FqElwx5sFkt/OXnHJMoj9+HP58UnCCgmt35ruIreFdSXU9Ca5GpS6li5nUzSWTWrpiVh5ZiYAgoMJyMnbnvVBPiV4Wffi+uFEUnl3DPp9wq2rbtuJiY8Q8g8vt6Z6VX8EPq9kl5Z6n4a1GyFxf3d4s8s1qyBZJWdVOyZm3YYDpjPfvWQmm603gDxhYnwtex3mrXF48ELTWuZfPDBWJExUbRt3ZOemN2K19nTc5cz006rqwWv3/hrqdL4m8aWvhjUdMtLiyvrhr64ERa3sp5VjUo7ZBSNgzZTGwHdg7ugrY0zU4NWtDcWsd1GgYri7s5bZ8j/AGJVVsc9cYrA8QW2p32m+H9RtNJuHuNOv47qbTzLEsxXy5I2Abf5e4eZn7+CAea2tIvtSv1ml1LSG0uPcBBHLcJJMw7l1TKLz0w7ZHXB4rKUYezut+uq/IV27Py/zKX9uXNz48/sOxjiNtZ2Yub+Z1JYNISsUa4IAJ2uxJzwAMc5E+tajfWOoaPFZRxPFeXfkTl1JKrsZsrgjB+U9c1iWI/sf4uarHc/LHr9lBNaSFuGkgDJJH9drI2O43Hsa7KsFpZ/1uE4uSaTt/wxleIH1yOwV/Dn9n+erZk+379uzB6bOc5xWB4d8Wao3ho+IvF0mmWulyRq0TWkcpkUltvzA5/TNb/iG71S008f2LpH9qzSMUeP7SsOxSD82W6844965DSLbxTa/Dq60Gfwztmis2ghb7fEfOLkgnA4XAbPJ5xirj8L+X63/Q460pKsrN7Po2r6W2Xr1N0/EbwmsXmHWoQu9U+4+ckZHGM4x1PQdDg1a1rxr4f8PXEUGr6isEkqeYiiN5Mr6/KDXL3fhbUJrG/jg0MRPNoMNpB/pEZ8uRSSYu2eoO7gfLUOr6b411G5tP8ARdQhtfIjVYdO1iO2+ztjD+Y2wmTpkY4A96vkhfR/iv6/r78PrOJUW3HXyi/8/wCvvt3L+ItJj0Fdae/hXTmTetwTwR6Adc9sYznjGayrT4jeFL65t7e11ZZJrmQRRIIJASxIAHK8dRycD8q5K28O65Z+BLaDULC1tZNCvftkYvbtGiu1y5IZhwn3uC3f0p+iaxP4m+KNtdTWlnaldOljItLxLogZ+8zpwOTgDr19RT9nG7tra/Xyuvv/AAE8XW9xNWbtunu3Z9Va3nuddP8AEDwtbaqdOm1mBbgNsPDFFPoXxtHvzx3rowQQCDkHoa8Y/wCED8S6daS6SLPV7+0ywLWmuxQQSqTn/VOhI9855zXrWj2r2OiWVpIGDwQJGQ0nmEEADlsDd9cD6VM4xUbxZvhq9epNxqxt8n+ez+RLf39tpmnzXt9J5VvAheR9pbaB3wOaxE+IPhZ1s2XV4sXrFYAUcEnO3kEfKM8ZbA4PpWzqU93a6dNNp1l9uuUGY7fzRH5hz03HgVxXgKy13RJmtb/wpHaLdTSS3Goi9iZuSWVSq5JAzjr3zUwimm2aV6tSM4xh18m7fd/XU39U8deGtF1H7DqWqxRXPG5FRn2Z7MVBCn2OKuap4k0jRtKj1LUL6OOzlIEcyAyB8jIxtBzxzxXnOoeCvEen6zqR0+LVby2vrh7gNp+spaLhj910dTlh65xjFa1xoPiWx8C6Vp2iQ3cQjZ/tVrFqUa3IUklQs5QKAD1wPYZ5NVyR5U79jnWJxHNNOG17aP5eT+VjsNE8S6R4jtpbjRr1LmOFtsh2shQ4zyGAOPfpwfSuR8TfFCytkCeF9R067uIpds8U8cp3ruC/u2GFY5OevTkVR0Hwv4kjudbtb+zuoodZshEb661JLmSBgjABiAC33scDAx1NN1nS/F2o+EofD6+E7NBZGPZdJex7ZNhAzGpwUJ65PbI6mrUIKa1006/eY1MRiZ0HZNS1+y9+ll0v3d0dZ4p8YWug3dpYi+tbe9nkRil1DMwaIsQdpRT83HGePWtXWtf0zw9aR3OsXP2eGSQRq3ls+WIJxhQfQ1yXi5/FeuaOmn23hTAk8mZ5P7Ri+R1YMUwcZ6YzU3iYeJvEXgS5sz4Z+z3txMsZt/t8T7YwQ2/dwDyMY696zUVZX766rbT/AIP5nTKvNSnyp7Jr3Zb66fl23t0Og0vxboWtalPYaXqMVxcwZ3xqCOhwSCRhh7jNUrv4ieFLK+ls7nWYlmhzvCo7KCBkjcAQT7A5zx1rNmtddn8XaHdL4eFvaQWTQTut3H+4MgGQAOTs29uueMVxet3t5oPgu98Hz2mjyrBkfa0v497fMGB8j7/mdBn156DNUqcW/wDg+f8Awz87mVTF1acW2refK+yaVvW6v0tqtT2DStVs9a02K/0ybz7abOyTaVzg4PBAPUVZlljgheWd1jjjUs7ucBQOpJPQVBpismk2iuCrLAgIIwQdoqj4q0eTX/C99pkEoiluI8IzE7cgggHHODjB9jWUklJpbHoRlJ0lJ6uxn23xG8J3d2La31iNpS4RV8qQbmLBQASuDyR07c9Oas6v448OaDfCz1XVI4bjGTGqM5X/AHtoO38cVgMnibVbfT9MvfBdjbRWs0TrcSagphjMf8SxphumQBk9RnIzQLDxH4b1LWE0rw/a63b6pO1wLhrpYmTd/BIG+8B2A7E+vGnJH+mvL7upw/WK9r27a8stN+m76bdze1Dxz4b0ueOK91WJGlh8+MojOrx8/MGUEHoe/NRP8QvCqfaP+JxE32ZFklKI7YUkAYwOeWGQM471gaX4V1XS77QrSbTY7u3sYbl3u47hVjhllydqo3zELjA/3s8Yqk+l6hoGmaPdT6VFZ29tpM9neNJeRAQsx4YnPzFiM4HdsUcsO/8AWv8AkvvF9YxO7Vvk9Ph/zb+T+XVW/wARfCd3fQ2kGsxNNNjYDG6qSRkAsVwD7E9eOtReM9Z8QaCsd9piaa+nK0ccwuBIZtzPt+XBC4wR1964vQf7a8QfD+Lw3p2jaXLayRqsmoLeIyw5+bLxD5xIP/QueldH4ytfEd1pKaFpOgtfW0YgYXzX0aFyhUkFG5/h6571bhGM0vP8DOOJrVcPKWt+W6snv2639djs9R1Kz0mxkvNSuY7a3jGWkkOB9B6n2HJrO0PxfoPiSSSPRtRjuJIxloyrI2PUKwBI9xXO+JdI1/xl4XiNxpjaTfWd4J0tVvlZplVf4ZFGEbk4yOCOaoeF/DWuJ4us9Q1ew1iMWyuBPfa7FcgAgjbtWPJzn1A4/CojCNnzPU3qYmsqsVCPuu3R9/wt5o0R8U9Cg8SajZ39/HFZ24RYZBbylmk5EgOAehA7D8a6LV/Fmh6FawXGq6hHbpcAGIFWZnBGchQC2PfFYuoxeItH8YX2paLoiatBf28SH/TEhMLJuHIbrnd2rH8T+FPE15q1lrkXmTXv2QW9xDpN/wDY2RtxO5XcHKc4I65xQowdiZVsRBTsrtbe69r+uunRHc6f4h0nVNJbU7HUIJLNATJMW2iPHJ3ZwV455xXI3PxKt7zxFpVp4bvbC7tbi5S3uUlilWZdx+8mcAjA688kVn6d4I1e68N63b3EF1ZXd88Lj+0tSW7EzRtu+YogwD0OST7ccyazF4w1i40e5Hgy3tzpN0kwUahEWkA/hU8BF45Bz/D6VUYQU/8Agr+mZ1MRiJUtrPyT7/hprre97Haa1qN9Y6ho8VlHE8V5d+ROXUkquxmyuCMH5T1zTfFuqX2h+GbrVdOhiuGsQJ54ZAcvApzKFIIw+zcRnIyAMc1rxlmjUuuxiASuc4PpmsDx3qA07wRqZVPNuLqE2lrCBkzTy/u40A75Zh+GT2rmk2o6b/1oetTjed29H+BuWtzDe2cN1ayCWCeNZI3XoysMg/iDUtZ/h/TG0Xwzpelu4kaxs4rcuOjFEC5/StCrlZSaQRvbUKKKKkYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBBeWVrqNnJaahbQ3VtKNskM8YdHHoVPBqDS9E0rQ4Hg0TTLPTopG3vHaW6RKzYxkhQMnA61eop8zta+gBRRRSAKKKKACiiigAooooAKKKKACiiigBk8EVzA8FzEk0UilXjkUMrA9QQeoqvYaTp2lq40ywtbMSYLi3hWPdjpnAGetW6KLsXKm7tBRRRQMKKKKACiiigAooooAKKKKACqD6FpMmofb5NLsmvAwf7S1uhk3Dod2M5GKv0U02thOKkrNBRRRSGFFFFABUN3Z21/bNb31vFcwPjdFMgdWxzyDxU1FAb6FWx0uw0uNk0yxtrNHOWW3hWMMfUgAZq1RRRdvcSSirIKKKKBhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAf//Z)

A person talking on a cell phone

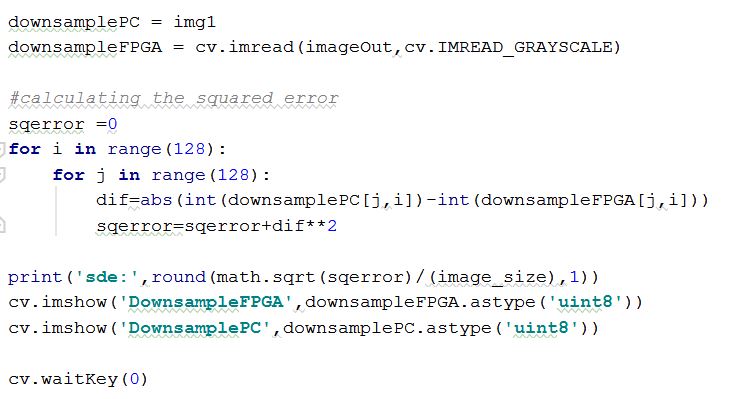
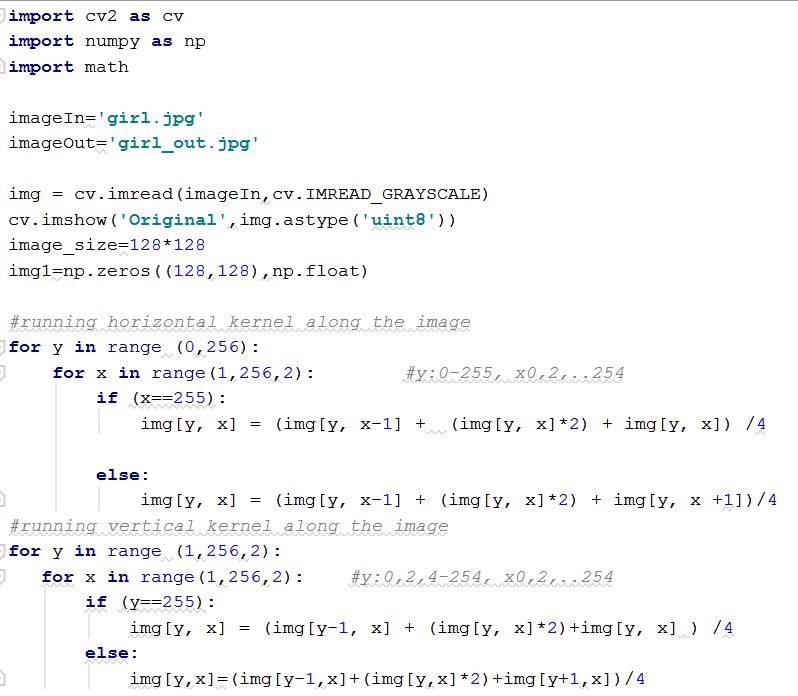
Description generated with high confidenceA close up of a person

Description generated with very high confidence

Original Image (256x256) sent to FPGA

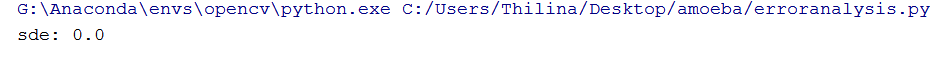
Down sampled Image (128x128) by FPGA

## 8.2 Comparison between images down-sampled by the computer and FPGA



The algorithm which was implemented in the processor to down sample images was implemented in python to compare the difference between expected accuracy and actual accuracy.

The square root of sum of squared error was zero.



# 9.0 Reference

[1] Atlys FPGA Board Reference Manual - <https://reference.digilentinc.com/_media/atlys:atlys:atlys_rm.pdf?_ga=2.122795851.1975576103.1563257482-996813489.1548924697>

[2] Atlys Spartan-6 FPGA Trainer Board schematics - <https://reference.digilentinc.com/_media/atlys:atlys:atlys_sch.pdf?_ga=2.122795851.1975576103.1563257482-996813489.1548924697>

[3] Xilinx ISE 14.7 In-Depth Tutorial - <https://www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ise_tutorial_ug695.pdf>

[4] Designing Digital Computer Systems With Verilog by D. Lilja and S. Sapatnekar (2005)

# 10.0 Appendix

## 10.1 Verilog codes

### 10.1.1 PROGRAM\_COUNTER

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 11:41:24 03/31/2019

// Design Name:

// Module Name: program\_counter

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module program\_counter(

input clk,

input reset,

input no\_inc,

input [11:0] jmp\_addr,

input jmp, //jump enable control signal

output reg [11:0] addr\_out

);

initial addr\_out<=12'b000000000000;

always @(posedge clk) begin

    if (~reset) begin

    if (jmp)

        addr\_out<=jmp\_addr;

    else

        addr\_out<=addr\_out+{11'b0000000000,~no\_inc};

    end

    else

        addr\_out<=12'b000000000000;

end

endmodule

### 10.1.2 AC\_ALU

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 18:38:15 03/14/2019

// Design Name:

// Module Name: ac\_alu

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module ac\_alu(

    input clk,

    input [1:0] ac\_control,

    input [2:0] alu\_control,

    input [15:0] bus\_to\_ac,

    input [6:0] inst\_to\_alu,

    output [15:0] ac\_to\_bus,

    output z\_flag

);

// ac\_control //

// 0-> ac input select 0-bus 1-alu

// 1-> ac write enable

// alu\_control //

// 0->ADD

// 1->SUB

// 2->MUL

// 3->DIV

// 4->SHR

// 5->SHL

reg [15:0] ac;

wire [15:0] ac\_input;

wire [15:0] alu\_out;

wire [15:0] const\_from\_inst;

assign const\_from\_inst[6:0]=inst\_to\_alu;

assign const\_from\_inst[15:7]=0;

assign ac\_input=ac\_control[0]? alu\_out:bus\_to\_ac;

assign ac\_to\_bus=ac;

assign z\_flag=ac? 1'b0:1'b1;

assign alu\_out=mux(bus\_to\_ac,ac,const\_from\_inst,alu\_control);

function [15:0] mux(

    input [15:0]bus\_to\_ac,

    input [15:0] ac,

    input [15:0] const\_from\_inst,

    input [3:0] alu\_control

    );

    case(alu\_control)

        3'b000: begin mux=ac+(bus\_to\_ac+const\_from\_inst); end       //add

        3'b001: begin mux=ac-(bus\_to\_ac+const\_from\_inst); end       //sub

        3'b010: begin mux=ac\*(bus\_to\_ac+const\_from\_inst); end       //mul

        3'b011: begin mux=ac/(bus\_to\_ac+const\_from\_inst); end       //div

        3'b100: begin                                                           //shr

                    case (const\_from\_inst[3:0])

                        0:begin mux=ac; end

                        1:begin mux=ac>>1; end

                        2:begin mux=ac>>2; end

                        3:begin mux=ac>>3; end

                        4:begin mux=ac>>4; end

                        5:begin mux=ac>>5; end

                        6:begin mux=ac>>6; end

                        7:begin mux=ac>>7; end

                        8:begin mux=ac>>8; end

                        9:begin mux=ac>>9; end

                        10:begin mux=ac>>10; end

                        11:begin mux=ac>>11; end

                        12:begin mux=ac>>12; end

                        13:begin mux=ac>>13; end

                        14:begin mux=ac>>14; end

                        default:begin mux=ac>>15; end

                    endcase

                 end

        3'b101: begin                                                           //shl

                    case (const\_from\_inst[3:0])

                        0:begin mux=ac; end

                        1:begin mux=ac<<1; end

                        2:begin mux=ac<<2; end

                        3:begin mux=ac<<3; end

                        4:begin mux=ac<<4; end

                        5:begin mux=ac<<5; end

                        6:begin mux=ac<<6; end

                        7:begin mux=ac<<7; end

                        8:begin mux=ac<<8; end

                        9:begin mux=ac<<9; end

                        10:begin mux=ac<<10; end

                        11:begin mux=ac<<11; end

                        12:begin mux=ac<<12; end

                        13:begin mux=ac<<13; end

                        14:begin mux=ac<<14; end

                        default:begin mux=ac<<15; end

                    endcase

                 end

        default: mux=ac;

    endcase

endfunction

always @(posedge clk) begin

    if(ac\_control[1]) ac<=ac\_input;

end

endmodule

### 10.1.3 Baudrate Generator

//TestBench\_clk = 50MHz

//BuadRate = 9600

module baudrate (input wire clk\_50m,

                        output wire Rxclk\_en,

                        output wire Txclk\_en

                        );

parameter RX\_MAX = 6250000 / (9600 \* 16);

parameter TX\_MAX = 6250000 / 9600;

parameter RX\_WIDTH = $clog2(RX\_MAX);

parameter TX\_WIDTH = $clog2(TX\_MAX);

reg [RX\_WIDTH - 1:0] rx\_acc = 0;

reg [TX\_WIDTH - 1:0] tx\_acc = 0;

assign Rxclk\_en = (rx\_acc == 5'd0);

assign Txclk\_en = (tx\_acc == 9'd0);

always @(posedge clk\_50m) begin

    if (rx\_acc == RX\_MAX[RX\_WIDTH - 1:0])

        rx\_acc <= 0;

    else

        rx\_acc <= rx\_acc + 5'b1; //+=00001

end

always @(posedge clk\_50m) begin

    if (tx\_acc == TX\_MAX[TX\_WIDTH - 1:0])

        tx\_acc <= 0;

    else

        tx\_acc <= tx\_acc + 9'b1; //+=000000001

end

endmodule

### 10.1.4 Transmitter

module transmitter( input wire [7:0] data\_in, //input data as an 8-bit regsiter/vector

                            input wire wr\_en, //enable wire to start

                            input wire clk\_50m,

                            input wire clken, //clock signal for the transmitter

                            output reg Tx, //a single 1-bit register variable to hold transmitting bit

                            output wire Tx\_busy //transmitter is busy signal

                            );

initial begin

     Tx = 1'b1; //initialize Tx = 1 to begin the transmission

end

//Define the 4 states using 00,01,10,11 signals

parameter TX\_STATE\_IDLE = 2'b00;

parameter TX\_STATE\_START    = 2'b01;

parameter TX\_STATE\_DATA = 2'b10;

parameter TX\_STATE\_STOP = 2'b11;

reg [7:0] data = 8'h00; //set an 8-bit register/vector as data,initially equal to 00000000

reg [2:0] bit\_pos = 3'h0; //bit position is a 3-bit register/vector, initially equal to 000

reg [1:0] state = TX\_STATE\_IDLE; //state is a 2 bit register/vector,initially equal to 00

always @(posedge clk\_50m) begin

    case (state) //Let us consider the 4 states of the transmitter

    TX\_STATE\_IDLE: begin //We define the conditions for idle or NOT-BUSY state

        if (wr\_en) begin

            state <= TX\_STATE\_START; //assign the start signal to state

            data <= data\_in; //we assign input data vector to the current data

            bit\_pos <= 3'h0; //we assign the bit position to zero

        end

    end

    TX\_STATE\_START: begin //We define the conditions for the transmission start state

        if (clken) begin

            Tx <= 1'b0; //set Tx = 0 after transmission has started

            state <= TX\_STATE\_DATA;

        end

    end

    TX\_STATE\_DATA: begin

        if (clken) begin

            if (bit\_pos == 3'h7) //we keep assigning Tx with the data until all bits have been transmitted from 0 to 7

                state <= TX\_STATE\_STOP; // when bit position has finally reached 7, assign state to stop transmission

            else

                bit\_pos <= bit\_pos + 3'h1; //increment the bit position by 001

            Tx <= data[bit\_pos]; //Set Tx to the data value of the bit position ranging from 0-7

        end

    end

    TX\_STATE\_STOP: begin

        if (clken) begin

            Tx <= 1'b1; //set Tx = 1 after transmission has ended

            state <= TX\_STATE\_IDLE; //Move to IDLE state once a transmission has been completed

        end

    end

    default: begin

        Tx <= 1'b1; // always begin with Tx = 1 and state assigned to IDLE

        state <= TX\_STATE\_IDLE;

    end

    endcase

end

assign Tx\_busy = (state != TX\_STATE\_IDLE); //We assign the BUSY signal when the transmitter is not idle

endmodule

### 10.1.5 Receiver

module receiver (input wire Rx,

                        output reg ready,

                        input wire ready\_clr,

                        input wire clk\_50m,

                        input wire clken,

                        output reg [7:0] data

                        );

initial begin

    ready = 1'b1; // initialize ready = 0

    data = 8'b0000\_0000; // initialize data as 00000000

end

// Define the 4 states using 00,01,10 signals

parameter RX\_STATE\_START    = 2'b00;

parameter RX\_STATE\_DATA     = 2'b01;

parameter RX\_STATE\_STOP     = 2'b10;

reg [1:0] state = RX\_STATE\_START; // state is a 2-bit register/vector,initially equal to 00

reg [3:0] sample = 0; // This is a 4-bit register

reg [3:0] bit\_pos = 0; // bit position is a 4-bit register/vector, initially equal to 000

reg [7:0] scratch = 8'b0; // An 8-bit register assigned to 00000000

always @(posedge clk\_50m) begin

    if (ready\_clr)

        ready <= 1'b0; // This resets ready to 0

    if (clken) begin

        case (state) // Let us consider the 3 states of the receiver

        RX\_STATE\_START: begin // We define condtions for starting the receiver

            if (!Rx || sample != 0) // start counting from the first low sample

                sample <= sample + 4'b1; // increment by 0001

            if (sample == 15) begin // once a full bit has been sampled

                state <= RX\_STATE\_DATA; //  start collecting data bits

                bit\_pos <= 0;

                sample <= 0;

                scratch <= 0;

            end

        end

        RX\_STATE\_DATA: begin // We define conditions for starting the data colleting

            sample <= sample + 4'b1; // increment by 0001

            if (sample == 4'h8) begin // we keep assigning Rx data until all bits have 01 to 7

                scratch[bit\_pos[2:0]] <= Rx;

                bit\_pos <= bit\_pos + 4'b1; // increment by 0001

            end

            if (bit\_pos == 8 && sample == 15) // when a full bit has been sampled and

                state <= RX\_STATE\_STOP; // bit position has finally reached 7, assign state to stop

        end

        RX\_STATE\_STOP: begin

            /\*

             \* Our baud clock may not be running at exactly the

             \* same rate as the transmitter. If we thing that

             \* we're at least half way into the stop bit, allow

             \* transition into handling the next start bit.

             \*/

            if (sample == 15 || (sample >= 8 && !Rx)) begin

                state <= RX\_STATE\_START;

                data <= scratch;

                ready <= 1'b1;

                sample <= 0;

            end

            else begin

                sample <= sample + 4'b1;

            end

        end

        default: begin

            state <= RX\_STATE\_START; // always begin with state assigned to START

        end

        endcase

    end

end

endmodule

### 10.1.6 UART

module uart(input [15:0] bus\_to\_uart\_tx, //input data

                input clk\_50m,

                input Rx,

                input wr\_en,

                input ready\_clr,

                input tx\_we,

                output Tx,

                output ready,

                output Tx\_busy,

                output [15:0] uart\_tx\_to\_bus,

                output [15:0] uart\_rx\_to\_bus

                );

reg [7:0] data\_in;

assign uart\_tx\_to\_bus={8'b00000000,data\_in};

always @(posedge clk\_50m) begin

    if(tx\_we) begin

        data\_in<=bus\_to\_uart\_tx[7:0];

    end

end

wire [7:0] data\_out;

assign uart\_rx\_to\_bus={8'b00000000,data\_out};

wire wr\_en\_mod;

assign wr\_en\_mod=(~Tx\_busy) & wr\_en;

wire Txclk\_en, Rxclk\_en;

baudrate uart\_baud( .clk\_50m(clk\_50m),

                            .Rxclk\_en(Rxclk\_en),

                            .Txclk\_en(Txclk\_en)

                            );

transmitter uart\_Tx(    .data\_in(data\_in),

                            .wr\_en(wr\_en\_mod),

                            .clk\_50m(clk\_50m),

                            .clken(Txclk\_en), //We assign Tx clock to enable clock

                            .Tx(Tx),

                            .Tx\_busy(Tx\_busy)

                            );

receiver uart\_Rx(   .Rx(Rx),

                        .ready(ready),

                        .ready\_clr(ready\_clr),

                        .clk\_50m(clk\_50m),

                        .clken(Rxclk\_en), //We assign Tx clock to enable clock

                        .data(data\_out)

                        );

endmodule

### 10.1.7 INSTRUCTION\_DECODER

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 17:36:40 03/31/2019

// Design Name:

// Module Name: instruction\_decoder

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module instruction\_decoder(

    input [15:0] instruction,

    //main bus drivers

    input [15:0] mbr\_to\_bus,

    input [15:0] mdr\_to\_bus,

    input [15:0] uart\_tx\_to\_bus,

    input [15:0] uart\_rx\_to\_bus,

    input [15:0] ac\_to\_bus,

    input [15:0] lr\_to\_bus,

    input [15:0] reg\_bank\_data\_out,

    //flags

    input z\_flag,

    input lrz\_flag,

    input tx\_busy,

    input rx\_ready,

    //instructiojn operand digestion

    output [15:0] bus,

    output [3:0] reg\_bank\_addr\_out,

    output [6:0] inst\_to\_alu,

    output [11:0] jmp\_addr,

    output [11:0] from\_inst\_to\_mar,

    output [3:0] reg\_bank\_addr\_in,

    //control signals

    output [1:0] ac\_control,

    output [2:0] alu\_control,

    output [2:0] mem\_registers\_control,

    output gpr\_write\_en,

    output program\_counter\_jmp,

    output loop\_register\_decrement,

    output loop\_register\_we,

    output uart\_ready,

    output uart\_ready\_clr,

    output uart\_wr\_en,

    output uart\_enable,

    output uart\_tx\_we,

    output dram\_we,

    //set outside the LUT

    output program\_counter\_no\_inc

);

//main bus mux

wire [4:0] bus\_mux\_select;

assign bus=bus\_mux(

    bus\_mux\_select,

    mbr\_to\_bus,

    mdr\_to\_bus,

    uart\_tx\_to\_bus,

    uart\_rx\_to\_bus,

    ac\_to\_bus,

    lr\_to\_bus,

    reg\_bank\_data\_out

    );

function [15:0] bus\_mux(

    input [4:0] bus\_mux\_select,

    input [15:0] mbr\_to\_bus,

    input [15:0] mdr\_to\_bus,

    input [15:0] uart\_tx\_to\_bus,

    input [15:0] uart\_rx\_to\_bus,

    input [15:0] ac\_to\_bus,

    input [15:0] lr\_to\_bus,

    input [15:0] reg\_bank\_data\_out

    );

    case(bus\_mux\_select)

        5'b00000:begin bus\_mux=16'b0000000000000000; end

        5'b00001:begin bus\_mux=mbr\_to\_bus; end

        5'b00010:begin bus\_mux=mdr\_to\_bus; end

        5'b00011:begin bus\_mux=uart\_tx\_to\_bus; end

        5'b00100:begin bus\_mux=uart\_rx\_to\_bus; end

        5'b00101:begin bus\_mux=ac\_to\_bus; end

        5'b00110:begin bus\_mux=lr\_to\_bus; end

        default:begin bus\_mux=reg\_bank\_data\_out; end

    endcase

endfunction

//operand digestion

wire reg\_addr\_mux\_select;

assign bus\_mux\_select=reg\_addr\_mux\_select? instruction[10:6] : instruction[11:7];

assign reg\_bank\_addr\_out=reg\_addr\_mux\_select? instruction[9:6] : instruction[10:7];

assign inst\_to\_alu=instruction[6:0];

assign jmp\_addr=instruction[11:0];

assign from\_inst\_to\_mar=instruction[11:0];

assign reg\_bank\_addr\_in=instruction[3:0];

//lookup table for control signals

wire [18:0] decoder\_out;

wire [4:0] reg\_addr;

assign reg\_addr=instruction[4:0];

assign {    ac\_control[1:0],

            alu\_control[2:0],

            mem\_registers\_control[2:0],

            gpr\_write\_en,

            program\_counter\_jmp,

            loop\_register\_decrement,

            loop\_register\_we,

            uart\_ready,

            uart\_ready\_clr,

            uart\_wr\_en,

            uart\_enable,

            uart\_tx\_we,

            reg\_addr\_mux\_select,

            dram\_we} = decoder\_out;

assign decoder\_out =    (instruction[15:12]==4'b0001)?19'b1100000000000000000:

                            (instruction[15:12]==4'b0010)?19'b1100100000000000000:

                            (instruction[15:12]==4'b0011)?19'b1101000000000000000:

                            (instruction[15:12]==4'b0100)?19'b1101100000000000000:

                            (instruction[15:12]==4'b0101)?19'b1110000000000000000:

                            (instruction[15:12]==4'b0110)?19'b1110100000000000000:

                            (instruction[15:12]==4'b0111)?19'b0000001100000000000:

                            (instruction[15:12]==4'b1000)?19'b0000000000000000001:

                            (instruction[15:12]==4'b1001)?19'b0000000001000000000:

                            (instruction[15:12]==4'b1010 & z\_flag==1'b0)?19'b0000000000000000000:

                            (instruction[15:12]==4'b1010 & z\_flag==1'b1)?19'b0000000001000000000:

                            (instruction[15:12]==4'b1011 & lrz\_flag==1'b0)?19'b0000000001100000000:

                            (instruction[15:12]==4'b1011 & lrz\_flag==1'b1)?19'b0000000000100000000:

                            (instruction[15:12]==4'b1100 & reg\_addr==5'b00001)?19'b0000010000000000010:

                            (instruction[15:12]==4'b1100 & reg\_addr==5'b00010)?19'b0000001000000000010:

                            (instruction[15:12]==4'b1100 & reg\_addr==5'b00011)?19'b0000000000000000110:

                            (instruction[15:12]==4'b1100 & reg\_addr==5'b00101)?19'b1000000000000000010:

                            (instruction[15:12]==4'b1100 & reg\_addr==5'b00110)?19'b0000000000010000010:

                            (instruction[15:12]==4'b1100 & reg\_addr[4]==1'b1)?19'b0000000010000000010:

                            (instruction[15:12]==4'b1101)?19'b0000000000000010000:

                            (instruction[15:12]==4'b1110)?19'b0000000000000100000:

                            19'b0000000000000000000;

assign program\_counter\_no\_inc=tx\_busy | (~rx\_ready);

endmodule

### 10.1.8 LOOP\_REGISTER

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 12:20:18 03/31/2019

// Design Name:

// Module Name: loop\_register

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module loop\_register(

    input clk,

    input [15:0] bus\_to\_lr,

    input decrement,

    input we,

    output [15:0]lr\_to\_bus,

    output lrz\_flag

);

reg [15:0] lr;

assign lrz\_flag=(lr==16'b0000\_0000\_0000\_0001)? 1'b1:1'b0;

assign lr\_to\_bus=lr;

always @(posedge clk) begin

    if(decrement) lr<=lr-16'b0000000000000001;

    if(we) lr<=bus\_to\_lr;

end

endmodule

### 10.1.9 MEM\_REGISTERS

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 16:54:52 03/09/2019

// Design Name:

// Module Name: mem\_registers

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module mem\_registers(

    input clk,

    input [2:0] control,

    // 0=> mdr input select 0-from bus, 1-from mem

    // 1=> mdr write enable

    // 2=> mbr write enable

    input [11:0] from\_inst\_to\_mar,

    input [15:0] from\_bus\_to\_mbr,

    input [15:0] from\_bus\_to\_mdr,

    input [7:0] from\_mem\_to\_mdr,

    output [16:0] address\_out,

    output [15:0] from\_mbr\_to\_bus,

    output [15:0] from\_mdr\_to\_bus,

    output [7:0] from\_mdr\_to\_mem

);

reg [15:0] mbr;

reg [7:0] mdr;

wire [16:0] mbr\_alu\_1;

wire [16:0] mbr\_alu\_2;

wire [7:0] mdr\_input\_wire;

assign from\_mbr\_to\_bus=mbr;

assign mbr\_alu\_1[11:0]=from\_inst\_to\_mar;

assign mbr\_alu\_1[16:12]=5'b00000;

assign mbr\_alu\_2[16:1]=mbr[15:0];

assign mbr\_alu\_2[0]=1'b0;

assign address\_out=mbr\_alu\_1 + mbr\_alu\_2;

assign mdr\_input\_wire = control[0]? from\_mem\_to\_mdr : from\_bus\_to\_mdr[7:0];

assign from\_mdr\_to\_bus = {8'b00000000,mdr[7:0]};

assign from\_mdr\_to\_mem = mdr;

always @(posedge clk) begin

    if(control[1]) begin

        mdr<=mdr\_input\_wire;

    end

    if(control[2]) begin

        mbr<=from\_bus\_to\_mbr;

    end

end

endmodule

### 10.1.10 REG\_BANK

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 12:05:33 03/31/2019

// Design Name:

// Module Name: reg\_bank

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module reg\_bank(

input [15:0] data\_in,

input clk,

input [3:0] addr\_in,

input gpr\_write\_en,

     input [3:0] addr\_out,

     output [15:0] data\_out

     );

//REG bank

reg [15:0] R [15:0];

//set data out

assign data\_out=R[addr\_out];

//assign data to a register

always @(posedge clk) begin

    if (gpr\_write\_en) begin

        R[addr\_in]<=data\_in;

    end

end

endmodule

### 10.1.11 PROCESSOR\_TOP\_MODULE

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 03:36:55 04/05/2019

// Design Name:

// Module Name: processor\_top\_module

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module processor\_top\_module(

    input clk\_100m,

    input reset,

    input uart\_rx,

    output uart\_tx,

    output [15:0] OUT

);

wire clk;

wire [15:0] bus;

wire [1:0] ac\_control;

wire [2:0] alu\_control;

wire [6:0] inst\_to\_alu;

wire [15:0] ac\_to\_bus;

wire z\_flag;

assign OUT=ac\_to\_bus; //connect to top module output

ac\_alu AC\_ALU(

    .clk(clk),

    .ac\_control(ac\_control),

    .alu\_control(alu\_control),

    .bus\_to\_ac(bus),

    .inst\_to\_alu(inst\_to\_alu),

    .ac\_to\_bus(ac\_to\_bus),

    .z\_flag(z\_flag)

);

wire loop\_register\_decrement;

wire loop\_register\_we;

wire [15:0] lr\_to\_bus;

wire lrz\_flag;

loop\_register LOOP\_REGISTER(

    .clk(clk),

    .bus\_to\_lr(bus),

    .decrement(loop\_register\_decrement),

    .we(loop\_register\_we),

    .lr\_to\_bus(lr\_to\_bus),

    .lrz\_flag(lrz\_flag)

);

wire [2:0] mem\_registers\_control;

wire [11:0] from\_inst\_to\_mar;

wire [7:0] from\_mem\_to\_mdr;     //from dram - data

wire [16:0] address\_out;            //to dram - address

wire [15:0] from\_mbr\_to\_bus;

wire [15:0] from\_mdr\_to\_bus;

wire [7:0] from\_mdr\_to\_mem;     //to dram - data

mem\_registers MEM\_REGISTERS(

    .clk(clk),

    .control(mem\_registers\_control),

    // 0=> mdr input select 0-from bus, 1-from mem

    // 1=> mdr write enable

    // 2=> mbr write enable

    .from\_inst\_to\_mar(from\_inst\_to\_mar),

    .from\_bus\_to\_mbr(bus),

    .from\_bus\_to\_mdr(bus),

    .from\_mem\_to\_mdr(from\_mem\_to\_mdr),

    .address\_out(address\_out),

    .from\_mbr\_to\_bus(from\_mbr\_to\_bus),

    .from\_mdr\_to\_bus(from\_mdr\_to\_bus),

    .from\_mdr\_to\_mem(from\_mdr\_to\_mem)

);

wire program\_counter\_no\_inc;

wire [11:0] jmp\_addr;

wire program\_counter\_jmp;

wire [11:0] addr\_out;               //to iram - address

program\_counter PROGRAM\_COUNTER(

.clk(clk),

.reset(reset),

.no\_inc(program\_counter\_no\_inc),

.jmp\_addr(jmp\_addr),

.jmp(program\_counter\_jmp), //jump enable control signal

.addr\_out(addr\_out)

);

wire [3:0] reg\_bank\_addr\_in;

wire gpr\_write\_en;

wire [3:0] reg\_bank\_addr\_out;

wire [15:0] reg\_bank\_data\_out;

reg\_bank REG\_BANK(

.data\_in(bus),

.clk(clk),

.addr\_in(reg\_bank\_addr\_in),

.gpr\_write\_en,

     .addr\_out(reg\_bank\_addr\_out),

     .data\_out(reg\_bank\_data\_out)

     );

wire dram\_we;

dram DRAM(

.clka(clk\_100m), // input clka

.wea(dram\_we), // input [0 : 0] wea

.addra(address\_out), // input [16 : 0] addra

.dina(from\_mdr\_to\_mem), // input [7 : 0] dina

.douta(from\_mem\_to\_mdr) // output [7 : 0] douta

);

wire [15:0] iram\_dout;

iram IRAM (

.clka(clk\_100m), // input clka

.wea(1'b0), // input [0 : 0] wea

.addra(addr\_out), // input [11 : 0] addra

.dina(16'b0000000000000000), // input [15 : 0] dina

.douta(iram\_dout) // output [15 : 0] douta

);

wire tx\_busy;

wire rx\_ready;

wire uart\_ready;//control signal never used

wire uart\_ready\_clr;

wire uart\_wr\_en;

wire uart\_enable;

wire uart\_tx\_we;

wire [15:0] uart\_tx\_to\_bus;

wire [15:0] uart\_rx\_to\_bus;

uart UART(

    .bus\_to\_uart\_tx(bus), //input data

    .clk\_50m(clk),

    .Rx(uart\_rx),

    .wr\_en(uart\_wr\_en),

    .ready\_clr(uart\_ready\_clr),

    .tx\_we(uart\_tx\_we),

    .Tx(uart\_tx),

    .ready(rx\_ready),

    .Tx\_busy(tx\_busy),

    .uart\_tx\_to\_bus(uart\_tx\_to\_bus),

    .uart\_rx\_to\_bus(uart\_rx\_to\_bus)

);

wire [15:0] instruction;

instruction\_decoder INSTRUCTION\_DECODER(

    .instruction(instruction),

    //main bus drivers

    .mbr\_to\_bus(from\_mbr\_to\_bus),

    .mdr\_to\_bus(from\_mdr\_to\_bus),

    .uart\_tx\_to\_bus(uart\_tx\_to\_bus),

    .uart\_rx\_to\_bus(uart\_rx\_to\_bus),

    .ac\_to\_bus(ac\_to\_bus),

    .lr\_to\_bus(lr\_to\_bus),

    .reg\_bank\_data\_out(reg\_bank\_data\_out),

    //flags

    .z\_flag(z\_flag),

    .lrz\_flag(lrz\_flag),

    .tx\_busy(tx\_busy),

    .rx\_ready(rx\_ready),

    //instructiojn operand digestion

    .bus(bus),

    .reg\_bank\_addr\_out(reg\_bank\_addr\_out),

    .inst\_to\_alu(inst\_to\_alu),

    .jmp\_addr(jmp\_addr),

    .from\_inst\_to\_mar(from\_inst\_to\_mar),

    .reg\_bank\_addr\_in(reg\_bank\_addr\_in),

    //control signals

    .ac\_control(ac\_control),

    .alu\_control(alu\_control),

    .mem\_registers\_control(mem\_registers\_control),

    .gpr\_write\_en(gpr\_write\_en),

    .program\_counter\_jmp(program\_counter\_jmp),

    .loop\_register\_decrement(loop\_register\_decrement),

    .loop\_register\_we(loop\_register\_we),

    .uart\_ready(uart\_ready),

    .uart\_ready\_clr(uart\_ready\_clr),

    .uart\_wr\_en(uart\_wr\_en),

    .uart\_enable(uart\_enable),

    .uart\_tx\_we(uart\_tx\_we),

    .dram\_we(dram\_we),

    .program\_counter\_no\_inc(program\_counter\_no\_inc)

);

reg [15:0] INST\_REG;

initial INST\_REG<=16'b0000000000000000;

assign instruction=INST\_REG;

always @(negedge clk) begin

    INST\_REG<=iram\_dout;

end

reg [3:0] clkreg;

initial clkreg=0;

always @(posedge clk\_100m) clkreg=clkreg+1;

assign clk=clkreg[3]; //running @ 100MHz/16=6.25MHz

//assign clk=clk\_100m;

endmodule

### 10.1.12 TEST\_BENCH

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 11:24:27 04/08/2019

// Design Name:

// Module Name: test\_bench

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module test\_bench(

);

wire clk, reset, rx, tx;

wire [15:0] out;

reg [3:0] clkreg;

initial clkreg=0;

always @(posedge clk) clkreg<=clkreg+1;

wire clk\_50m;

assign clk\_50m=clkreg[3];

wire [15:0] out\_uart\_tx\_to\_bus;

wire [15:0] out\_uart\_rx\_to\_bus;

reg [15:0] out\_bus\_to\_uart\_tx;

reg send,wr\_en;

wire tx\_busy;

always @(send,posedge tx\_busy) wr\_en<=~tx\_busy;

uart UART(

    .bus\_to\_uart\_tx(out\_bus\_to\_uart\_tx), //input data

    .clk\_50m(clk\_50m),//running @ 6.25MHz

    .Rx(rx),

    .wr\_en(wr\_en),

    .ready\_clr(),

    .tx\_we(1'b1),

    .Tx(tx),

    .ready(),

    .Tx\_busy(tx\_busy),

    .uart\_tx\_to\_bus(out\_uart\_tx\_to\_bus),

    .uart\_rx\_to\_bus(out\_uart\_rx\_to\_bus)

    );

processor\_top\_module PROCESSOR(

    .clk\_100m(clk),

    .reset(reset),

    .uart\_rx(tx),

    .uart\_tx(rx),

    .OUT(out)

    );

endmodule

## 10.2 Python codes

### 10.2.1 Compiler

import re

import sys

ops = {

'nop': 0b0000,

'add': 0b0001,

'sub': 0b0010,

'mul': 0b0011,

'div': 0b0100,

'shr': 0b0101,

'shl': 0b0110,

'load': 0b0111,

'store': 0b1000,

'jump': 0b1001,

'jmpz': 0b1010,

'jmpdec': 0b1011,

'move': 0b1100,

'uartsend': 0b1101,

'uartread': 0b1110

}

reg = {

'zr': 0b00000,

'mbr': 0b00001,

'mdr': 0b00010,

'uarttx': 0b00011,

'uartrx': 0b00100,

'ac': 0b00101,

'lr': 0b00110

}

def getValToAc(x):

num = '{:016b}'.format(x)

acommand = ''

zeros = 0

ones = False

for i in range(0,16):

c = num[i]

if c == '1':

if zeros and ones: acommand = acommand + 'shl ' + str(zeros) + '\n'

acommand = acommand + 'add zr 1\n'

zeros = 0

if i == 15: ones = False

else: ones = True

zeros += 1

if zeros and ones: acommand = acommand + 'shl ' + str(zeros) + '\n'

return acommand

argCount = len(sys.argv)

infileName = ''

oufileName = 'precompiled.txt'

compfileName = 'compiled.coe'

if argCount < 2:

print("no input file\nusage:\nchill.py <input file name(required)> <output file names(optional)>")

exit()

elif argCount == 2:

infileName = sys.argv[1]

print('output files named as precompiled.txt and compiled.coe')

elif argCount == 3:

infileName = sys.argv[1]

oufileName = sys.argv[2]

print('output file named as compiled.coe')

else:

infileName = sys.argv[1]

oufileName = sys.argv[2]

compfileName = sys.argv[2]

# precompiling

infile = open(infileName, 'r')

oufile = open('temp.txt', 'w')

lineNo = 0

command = ''

jumpLines={}

writeLineCount=0

alllines=infile.readlines()

try:

while(lineNo<len(alllines)):

line=alllines[lineNo]

line = line.lower().strip()

x = (re.split("\s+", line))

command=''

if len(x[0])<1: lineNo += 1

elif x[0][0]==':':

jumpLines[x[0][1:]]=writeLineCount

x=x[1:]

else:

# warn if x[0] is not in ops

if not (x[0] in ops.keys()):

print('Error:', lineNo, ':', x[0], ' is not a valid instruction')

break

if (x[0] == 'nop'):

command = x[0] + '\n'

elif (x[0] in ['add', 'sub', 'mul', 'div']):

if x[1][0] == 'r':

# warn if index is not numeric

if not x[1][1:].isdecimal():

print('Error:', lineNo, ':', x[1], 'is not a valid register')

break

regNum = int(x[1][1:])

# warn if index>number of registers

if regNum < 0 or regNum > 15:

print('Error:', lineNo, ':', x[1], 'exceeds valid register index range')

break

else:

# warn if not a valid registers

if not x[1] in reg.keys():

print('Error:', lineNo, ':', x[1], 'is not a valid register')

break

if int(x[2])<128:

command = x[0] + ' ' + x[1] + ' ' + x[2] + '\n'

else:

num = int(x[2])

command = x[0] + ' ' + x[1] + ' ' + str(num % 127) + ' //'+ str(lineNo) + '.' + line + '\n'

for kk in range(0, int(num / 127)):

command = command + x[0] + ' zr ' + str(127) + '\n'

command = command[:-1] + ' //end\n'

elif (x[0] in ['shr', 'shl']):

# warn for shift range

if int(x[1]) < 0 or int(x[1]) > 15:

print('Error:', lineNo, ':', x[1], 'exceeds valid shift range')

break

command = x[0] + ' ' + x[1] + '\n'

elif (x[0] in ['load', 'store']):

num = int(x[1])

#warn for address range

if num < 0 or num > 131071:

print('Error:', lineNo, ':', x[1], 'exceeds valid address range')

break

if num>4095:

command = 'move mbr r15 //' + str(lineNo) + '.' + line + '\nmove ac r14\nmove zr ac\n'

command = command + getValToAc(int(num / 2))

command = command + 'move ac mbr\n'

command = command + x[0] + ' ' + str(num%2) + '\n'

command = command + 'move r15 mbr\nmove r14 ac //end\n'

else:

command = x[0] + ' ' + x[1] + '\n' #TODO: correct getValToAC()

elif (x[0] in ['jump', 'jmpz', 'jmpdec']):

# warn for address range

if not x[1][0]==':':

if not x[1].isdecimal():

print('Error:', lineNo, ':', x[1], 'is not a valid immediate accessible address')

break

num = int(x[1])

if num < 0 or num > 4095:

print('Error:', lineNo, ':', x[1], 'exceeds valid immediate accessible address range')

break

command = x[0] + ' ' + x[1] + '\n'

elif (x[0] == 'move'):

breakAll = False

for i in range(0, 2):

if (x[i + 1][0] == 'r'):

# warn if index is not numeric

if not x[i + 1][1:].isdecimal():

print('Error:', lineNo, ':', x[i + 1], 'is not a valid register')

breakAll = True

break

regNum = int(x[i + 1][1:])

# warn if index>number of registers

if regNum < 0 or regNum > 15:

print('Error:', lineNo, ':', x[i + 1], 'exceeds valid register index range')

breakAll = True

break

else:

# warn if not a valid registers

if not x[i + 1] in reg.keys():

print('Error:', lineNo, ':', x[i + 1], 'is not a valid register')

breakAll = True

break

if breakAll:

break

command = x[0] + ' ' + x[1] + ' ' + x[2] + '\n'

elif (x[0] in ['uartsend', 'uartread']):

if x[0]=='uartsend' and len(x)>1 and x[1][0]==':':

string = x[1]

command = 'move ac r15 //' + str(lineNo) + '.' + line + '\n'

for char in string[1:]:

asciival = ord(char)

command = command + 'move zr ac\nadd zr ' + str(asciival) + '\nmove ac uarttx\nuartsend\nnop\n'

command = command + 'move r15 ac //end\n'

else:

command = x[0] + '\n'

else:

# warn about errors

print('Error:', lineNo, ': \'', alllines[lineNo].strip(), '\' : invalid command.')

break

#print(thisLineNo, '.', x, REGS, GPR, MEM)

#dd=str(thisLineNo)+'.\t'+line+'\n\t'+str(REGS)+' \n\tgpr:'+str(GPR)+' \n\tmem:'+str(MEM)+'\n\n'

dd=command

writeLineCount=writeLineCount+len(dd.strip().split('\n'))

oufile.write(dd)

print(str(lineNo) + '->\n' + dd)

lineNo = lineNo + 1

except IndexError:

print('Error:', lineNo, ': \'', alllines[lineNo].strip(), '\' : invalid command.')

exit()

# precompilation ends

if lineNo==len(alllines):

oufile.close()

infile.close()

# replaceing labels

infile = open('temp.txt', 'r')

oufile = open(oufileName, 'w')

for line in infile:

line = line.lower().strip()

y = (re.split("\s+", line))

if (y[0] in ['jump', 'jmpz', 'jmpdec']):

if y[1][0]==':':

if not y[1][1:] in jumpLines.keys():

print('label', y[1][1:], 'not found.')

print('precompilation interrupted')

exit()

command = y[0] + ' ' + str(jumpLines[y[1][1:]]) + '\n'

else:

command = y[0] + ' ' + y[1] + '\n'

else:

command = line + '\n'

oufile.write(command)

oufile.close()

infile.close()

print('precompilation end\n')

oufile.close()

infile.close()

# compiling

infile = open(oufileName, 'r')

oufile = open(compfileName, 'w')

oufile.write("memory\_initialization\_radix = 2;\nmemory\_initialization\_vector =\n")

lineNo = 0

for line in infile:

lineNo = lineNo + 1

line = line.lower()

x = (re.split("\s+", line))

command = '{:04b}'.format(ops[x[0]])

if (x[0] == 'nop'):

command = command + '000000000000'

elif (x[0] in ['add', 'sub', 'mul', 'div']):

if (x[1][0] == 'r'):

regNum = int(x[1][1:])

command = command + '1' + '{:04b}'.format(regNum)

else:

command = command + '0' + '{:04b}'.format(reg[x[1]])

command = command + '' + '{:07b}'.format(int(x[2]))

elif (x[0] in ['shr', 'shl']):

num = '{:012b}'.format(int(x[1]))

command = command + '' + num

elif (x[0] in ['load', 'store']):

command = command + '' + '{:012b}'.format(int(x[1]))

elif (x[0] in ['jump', 'jmpz', 'jmpdec']):

command = command + '' + '{:012b}'.format(int(x[1]))

elif (x[0] == 'move'):

for i in range(0, 2):

if (x[i + 1][0] == 'r'):

regNum = int(x[i + 1][1:])

command = command + '01' + '{:04b}'.format(regNum)

else:

command = command + '00' + '{:04b}'.format(reg[x[i + 1]])

elif (x[0] in ['uartsend', 'uartread']):

command = command + '000000000000'

else:

# warn about errors

print('unknown compile error!')

break

print(lineNo, '.', x, command)

oufile.write(command + ",\n")

if not infile.readline():

print('compilation end')

oufile.write("0000000000000000;\n")

else:

print('compilation interrupted')

# compilation ends

else:

print('precompilation interrupted\n')

### 10.2.2 Simulator

import re

import sys

import time

ops = {

'nop': 0b0000,

'add': 0b0001,

'sub': 0b0010,

'mul': 0b0011,

'div': 0b0100,

'shr': 0b0101,

'shl': 0b0110,

'load': 0b0111,

'store': 0b1000,

'jump': 0b1001,

'jmpz': 0b1010,

'jmpdec': 0b1011,

'move': 0b1100,

'uartsend': 0b1101,

'uartread': 0b1110

}

reg = {

'zr': 0b00000,

'mbr': 0b00001,

'mdr': 0b00010,

'uarttx': 0b00011,

'uartrx': 0b00100,

'ac': 0b00101,

'lr': 0b00110

}

argCount = len(sys.argv)

infileName = ''

oufileName = 'results.txt'

if argCount < 2:

print("no input file\nusage:\ncompileme.py <input file name(required)> <output file name(optional)>")

exit()

elif argCount == 2:

infileName = sys.argv[1]

print('output file named as results.txt')

else:

infileName = sys.argv[1]

oufileName = sys.argv[2]

localtime = time.asctime( time.localtime(time.time()) )

infile = open(infileName, 'r')

oufile = open(oufileName, 'w')

oufile.write("\*\*\*\*\*\*\* simulation results - generated on "+localtime+"\*\*\*\*\*\*\*\*\*\n\nsimulation start\n\n")

lineNo = 0

#registers

REGS = {

'zr': 0,

'mbr': 0,

'mdr': 0,

'uarttx': 0,

'uartrx': 0,

'ac': 0,

'lr': 0

}

GPR = {0:0,1:0,2:0,3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0}

#GPR = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}

MEM = {}

alllines=infile.readlines()

round=0

while(lineNo<len(alllines)):

line=alllines[lineNo]

thisLineNo=lineNo+1

line = line.lower().strip()

x = (re.split("\s+", line))

# warn if x[0] is not in ops

if not (x[0] in ops.keys()):

print('Error at line', lineNo+1, ':', x[0], ' is not a valid instruction')

break

command = '{:04b}'.format(ops[x[0]])

if (x[0] == 'nop'):

command = command + '000000000000'

lineNo = lineNo + 1

elif (x[0] in ['add', 'sub', 'mul', 'div']):

operand=0

toac=0

if (x[1][0] == 'r'):

# warn if index is not numeric

if not x[1][1:].isdecimal():

print('Error at line', lineNo+1, ':', x[1], 'is not a valid register')

break

regNum = int(x[1][1:])

# warn if index>number of registers

if regNum < 0 or regNum > 15:

print('Error at line', lineNo+1, ':', x[1], 'exceeds valid register index range')

break

command = command + '1' + '{:04b}'.format(regNum)

operand=GPR[regNum]

else:

# warn if not a valid registers

if not x[1] in reg.keys():

print('Error at line', lineNo+1, ':', x[1], 'is not a valid register')

break

command = command + '0' + '{:04b}'.format(reg[x[1]])

operand=REGS[x[1]]

command = command + '' + '{:07b}'.format(int(x[2]))

if x[0]=='add':

toac=REGS['ac']+(operand+int(x[2]))

elif x[0]=='sub':

toac = REGS['ac'] - (operand + int(x[2]))

elif x[0]=='mul':

toac = REGS['ac'] \* (operand + int(x[2]))

elif x[0]=='div':

toac = REGS['ac'] / (operand + int(x[2]))

else:

toac = REGS['ac']

REGS['ac']=int(toac)

lineNo = lineNo + 1

elif (x[0] in ['shr', 'shl']):

# warn for shift range

num = '{:012b}'.format(int(x[1]))

if int(x[1]) < 0 or int(x[1]) > 15:

print('Error at line', lineNo+1, ':', x[1], 'exceeds valid shift range')

break

command = command + '' + num

if x[0]=='shr': REGS['ac']=REGS['ac']>>int(x[1])

else: REGS['ac']=REGS['ac']<<int(x[1])

lineNo = lineNo + 1

elif (x[0] in ['load', 'store']):

# TODO: warn for address range

num = int(x[1])

if num < 0 or num > 4095:

print('Error at line', lineNo+1, ':', x[1], 'exceeds valid immediate accessible address range')

break

command = command + '' + '{:012b}'.format(int(x[1]))

mar = num + REGS['mbr'] \* 2

if x[0]=='store':

MEM[mar]=REGS['mdr']

# oufile.write('storing mem data at address ' + str(mar) + '.\n')

# print('storing mem data at address', mar, '.')

else:

if mar in MEM.keys():

REGS['mdr']=MEM[mar]

# oufile.write('reading mem data at address ' + str(mar) + '.\n')

# print('reading mem data at address', mar, '.')

else:

REGS['mdr']=7

# oufile.write('no mem data at address '+str(mar)+'. so 0 assumed.\n')

# print('no mem data at address',mar,'. so 0 assumed.')

lineNo = lineNo + 1

elif (x[0] in ['jump', 'jmpz', 'jmpdec']):

# warn for address range

num = int(x[1])

if num < 0 or num > 4095:

print('Error at line', lineNo+1, ':', x[1], 'exceeds valid immediate accessible address range')

break

command = command + '' + '{:012b}'.format(int(x[1]))

if x[0]=='jump': lineNo=num; #oufile.write('jump to '+str(num)+'.\n')

elif x[0]=='jmpz':

if REGS['ac']==0: lineNo=num; #oufile.write('jmpz to '+str(num)+'.\n')

else:lineNo=lineNo+1

else:

oldLineNo=lineNo

if REGS['lr']>0:

REGS['lr']-=1

lineNo=num

#oufile.write('jmpdec to ' + str(num) + '.\n')

if REGS['lr']==0: lineNo=oldLineNo+1

elif (x[0] == 'move'):

breakAll = False

sourceVal=0

isGpr=[False, False]

for i in range(0, 2):

if (x[i + 1][0] == 'r'):

# warn if index is not numeric

if not x[i + 1][1:].isdecimal():

print('Error at line', lineNo+1, ':', x[i + 1], 'is not a valid register')

breakAll = True

break

regNum = int(x[i + 1][1:])

# warn if index>number of registers

if regNum < 0 or regNum > 15:

print('Error at line', lineNo+1, ':', x[i + 1], 'exceeds valid register index range')

breakAll = True

break

command = command + '01' + '{:04b}'.format(regNum)

isGpr[i]=True

else:

# warn if not a valid registers

if not x[i + 1] in reg.keys():

print('Error at line', lineNo+1, ':', x[i + 1], 'is not a valid register')

breakAll = True

break

command = command + '00' + '{:04b}'.format(reg[x[i + 1]])

if breakAll:

break

if isGpr[0]: sourceVal = GPR[int(x[1][1:])]

else: sourceVal=REGS[x[1]]

if isGpr[1]: GPR[int(x[2][1:])]=sourceVal

else: REGS[x[2]]=sourceVal

lineNo=lineNo+1

elif (x[0] in ['uartsend', 'uartread']):

command = command + '000000000000'

lineNo=lineNo+1

if x[0]=='uartsend':

print('uartsend:',REGS['uarttx'])

oufile.write('uart output sent. value:'+str(REGS['uarttx'])+'.\n')

else:

done=False

while(not done):

uinput=input('enter uart read number within range:0 to 255\n')

#uinput=str(70)

if uinput.isdecimal() and int(uinput)>-1 and int(uinput)<256:

uinput=int(uinput)

REGS['uartrx']=uinput

oufile.write('uart input given. value:'+str(uinput)+'.\n')

done=True

else:

print('invalid uart input!\n')

else:

# warn about errors

print('unknown compile error!')

break

#print(thisLineNo, '.', x, REGS, GPR, MEM)

#dd=str(thisLineNo)+'.\t'+line+'\n\t'+str(REGS)+' \n\tgpr:'+str(GPR)+' \n\tmem:'+str(MEM)+'\n\n'

dd=str(thisLineNo)+'.\t'+line+'\n\t'+str(REGS)+' \n\tgpr:'+str(GPR)+'\n\n'

if round%(5000)==0:oufile.write(dd);print(dd)

if lineNo>(len(alllines)-1):pp=str(thisLineNo)+'.\t'+line+'\n\t'+str(REGS)+' \n\tgpr:'+str(GPR)+' \n\tmem:'+str(MEM)+'\n\n'; oufile.write(pp)

round=round+1

if lineNo==len(alllines):

print('simulation end')

oufile.write('simulation end.')

else:

print('simulation interrupted')

oufile.write('simulation interrupted.')

## 10.3 Assembly code for down-sampling

nop

move zr mbr

move zr mdr

move zr ac

add zr 127

add zr 1

move ac r1

mul r1 0

mul zr 2

move ac lr

move zr uarttx

:thilalpha uartread

nop

move uartrx mdr

store 0

uartsend

nop

uartread

nop

move uartrx mdr

store 1

uartsend

nop

move mbr ac

add zr 1

move ac mbr

jmpdec :thilalpha

move zr ac //processing image

add zr 64

shl 2

move ac lr

:chanbeta move lr r1

move zr ac

add zr 127

move ac lr

:chanalpha move zr ac

add zr 64

shl 2

sub r1 0

shl 8

move ac r2

move zr ac

add zr 127

sub lr 0

shl 1

add r2 0

shr 1

move ac mbr

move zr ac

load 1

add mdr 0

shl 1

load 0

add mdr 0

load 2

add mdr 0

shr 2

move ac mdr

store 1

jmpdec :chanalpha

move zr ac

add r2 127

add zr 127

shr 1

move ac mbr

move zr ac

load 1

add mdr 0

shl 1

add mdr 0

load 0

add mdr 0

shr 2

move ac mdr

store 1

move r1 lr

jmpdec :chanbeta

move zr ac

add zr 64

shl 1

move ac lr

:chandelta move lr r1

move zr ac

add zr 127

move ac lr

:changamma move zr ac

add zr 2

add zr 127

sub r1 0

shl 1

sub zr 1

move ac r2

move zr ac

add zr 127

sub lr 0

shl 9

add r2 0

shr 1

move ac mbr

move zr ac

load 257

add mdr 0

shl 1

load 1

add mdr 0

load 513

add mdr 0

shr 2

move ac mdr

store 257

jmpdec :changamma

move zr ac

add zr 127

shl 9

add r2 0

shr 1

move ac mbr

move zr ac

load 257

add mdr 0

shl 1

add mdr 0

load 1

add mdr 0

shr 2

move ac mdr

store 257

move r1 lr

jmpdec :chandelta

move zr ac

add zr 64

shl 1

move ac lr

:chantheta move lr r1

move zr ac

add zr 64

move ac lr

:chaneeta move zr ac //load address

add zr 64

sub lr 0

shl 1

move ac r2

move zr ac

add zr 127

add zr 2

sub r1 0

shl 8

sub zr 127

sub zr 1

add r2 0

move ac mbr

load 1

move mdr r3

load 3

move zr ac //store address

add zr 127

add zr 1

sub r1 0

shl 6

move ac r2

move zr ac

add zr 1

shl 15

add zr 64

sub lr 0

add r2 0

move ac mbr

store 2

move r3 mdr

store 1

jmpdec :chaneeta

move r1 lr

jmpdec :chantheta

move zr ac //sending image

add zr 1

shl 15

move ac mbr

add zr 64

move ac r1

mul r1 0

mul zr 2

move ac lr

:thilibeta load 1

move mdr uarttx

uartsend

nop

uartread

nop

load 2

move mdr uarttx

uartsend

nop

uartread

nop

move mbr ac

add zr 1

move ac mbr

jmpdec :thilibeta

## 10.4 Schematics