APPENDIX B: Cycle accurate Simulator

Cycle Accurate Simulator

The Cycle-Accurate Simulator (CAS) was developed as a specialized tool for verifying the design and functionality of a 5-stage pipelined MIPS processor. Unlike traditional simulation tools like ModelSim, which rely heavily on waveform-based debugging, the CAS provides a clock-by-clock visualization of the processor's internal state. This includes tracking data transfers between pipeline stages, instruction progression, and the values of key control signals, offering a clearer and more structured view of the processor's operation.

The primary motivation behind creating this simulator was to enhance the verification process by closely mimicking the hardware design implemented in Verilog. The CAS reproduces the behavior of the pipelined processor cycle-by-cycle, ensuring that the simulated execution aligns with the intended hardware implementation. This alignment allows engineers to quickly identify discrepancies between the hardware design and its expected behavior, particularly in scenarios involving data hazards, control hazards, or branch mispredictions.

By visualizing the data flow through the pipeline and the changes in pipeline registers at each clock cycle, the CAS provides insights that are difficult to extract from waveform simulations alone. This makes it easier to debug complex behaviors and validate the correctness of the design under various operating conditions. The simulator also allows users to verify the impact of design decisions, such as branch prediction strategies or hazard detection mechanisms, in a highly detailed and precise manner.

In summary, the CAS serves as a robust verification tool that complements traditional simulation workflows by providing a more intuitive, cycle-accurate representation of the hardware's operation.

How to Operate

1. **Download the Simulator**:
   Begin by downloading the zip folder containing the Cycle-Accurate Simulator (CAS).

2. **Open the Command Prompt**:
   Press **Windows + R** to open the *Run* dialog box. Type cmd and press **Enter** to launch the terminal.

3. **Navigate to the Simulator Path**:
   In the terminal, use the cd command to navigate to the folder where the simulator executable is located. For example:

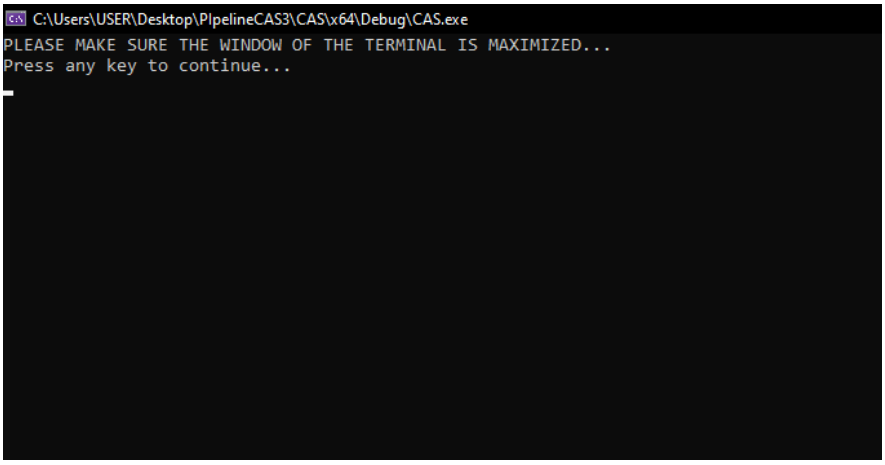
```
cd path\to\simulator\folder
```

4. **Run the Simulator**:
   Once in the correct directory, type the following command to launch the simulator:

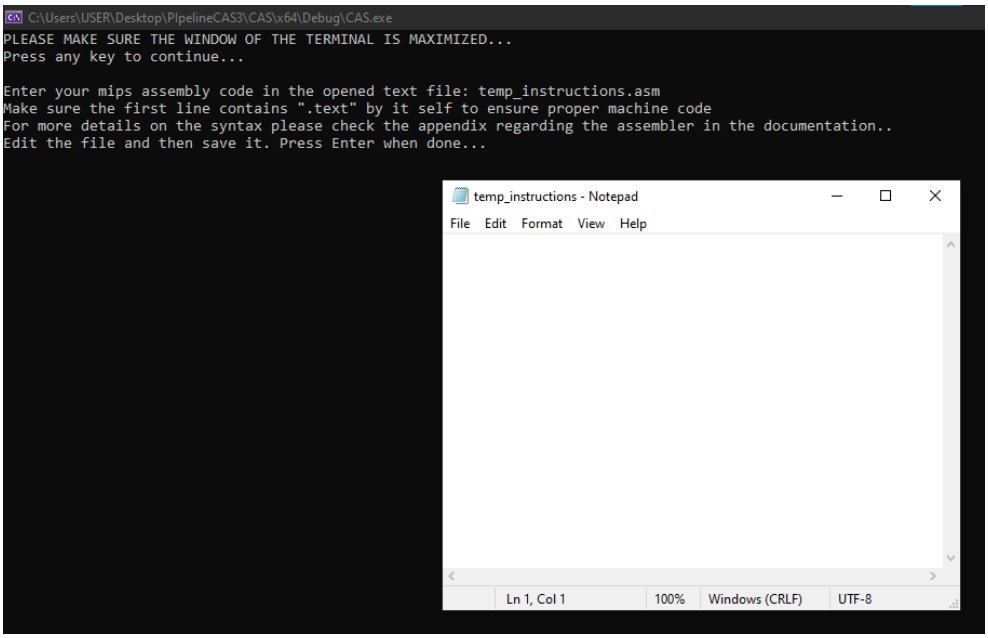
```
CAS
```

5. The simulator will start, and you can follow the on-screen instructions to interact with it.

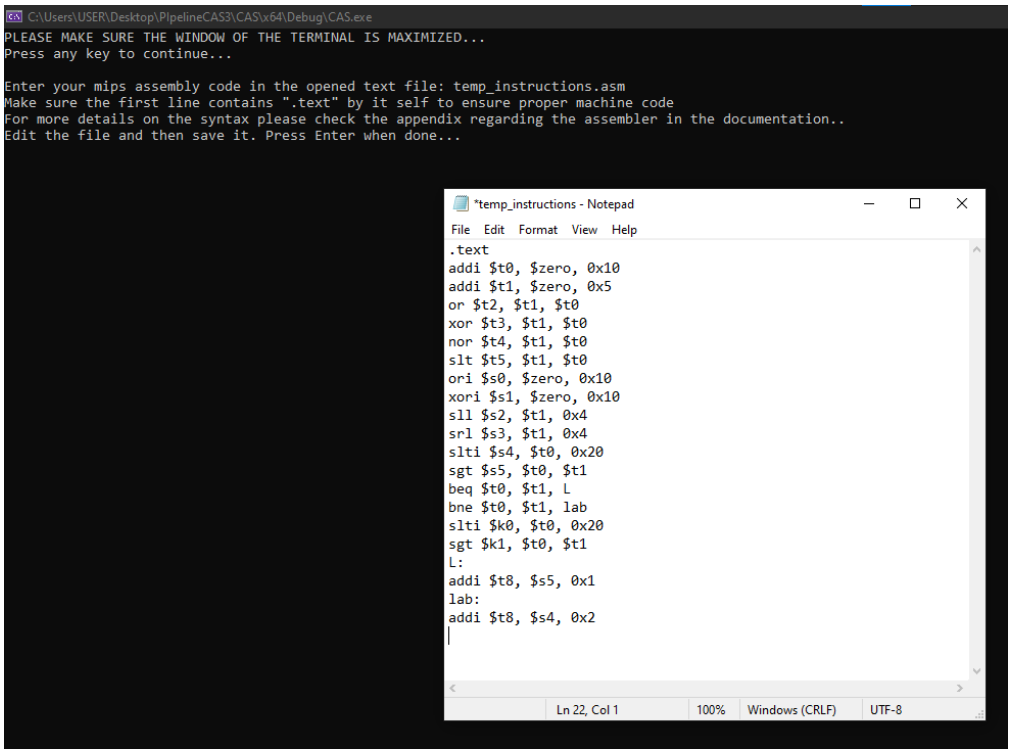
For correct screen printing make sure that you maximize the window of the terminal.



A temporary text file will appear on the screen, here you can input your assembly code



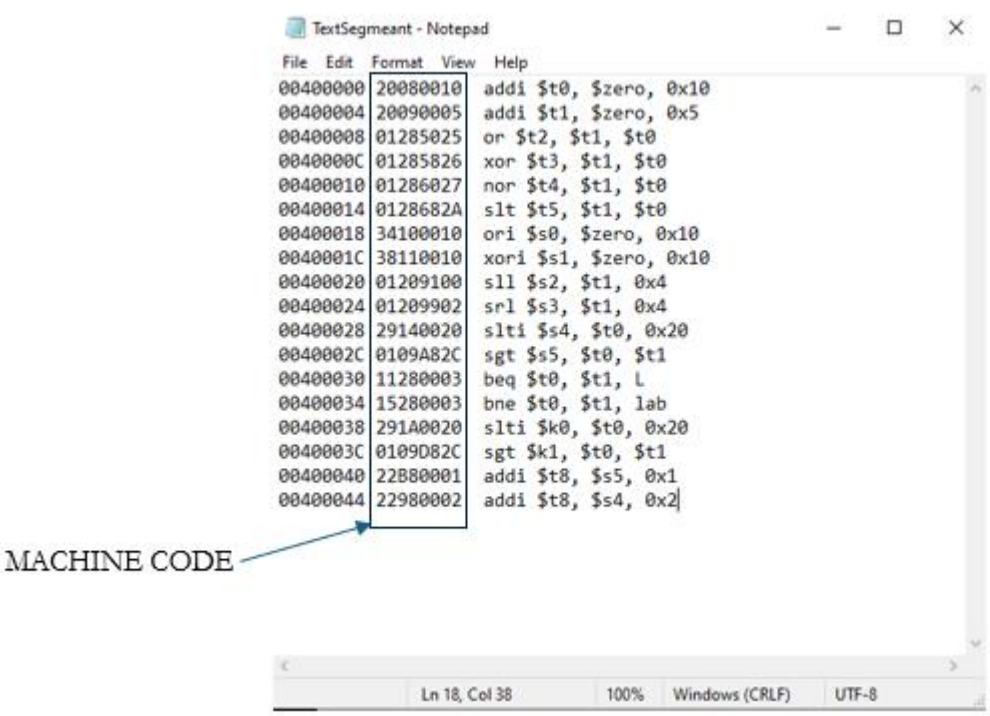
Please make sure that the file starts with .text as shown,



Refer to the assembler appendix for more details on the syntax.

When done save the file and press enter to activate the simulator

A file containing the instructions and their machine code will appear in the same directory as the executable for the Cycle accurate simulator.



Memory Addressing in the Simulator

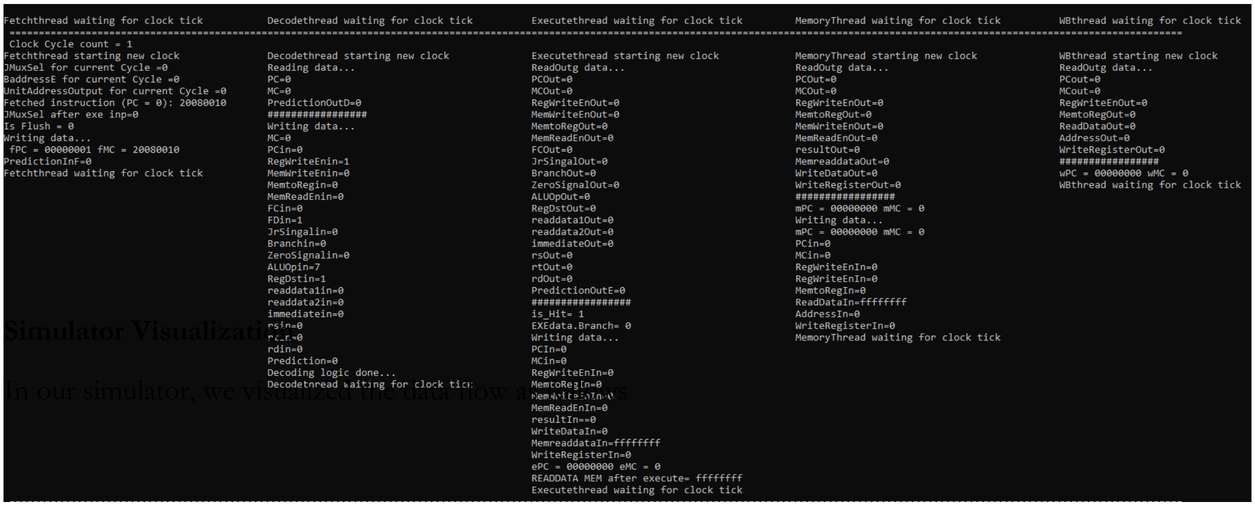
In the simulator, the values displayed on the left represent the addresses of the machine code in memory. However, it's important to note that these addresses are primarily for **visualization purposes** and do not reflect how memory addresses are actually indexed in the hardware or the CycleAccurateSimulator.

For comparison with tools like MARS, which use byte-addressable memory, these values are shown as byte addresses. In contrast, our hardware design and the cycle-accurate simulator both use **word-addressable memory**, starting at index 0. This means that in the actual design, each address corresponds to a word (typically 4 bytes), while the simulator's memory visualization shows addresses in byte units for easier alignment with MARS.

This distinction ensures that the design is properly validated, as we can directly compare our word-addressable memory implementation with the byte-addressable system in MARS, making sure our word alignment and addressing are correct.

Simulator Visualization

In our simulator, we visualized the data flow as follows



```
Fetchthread waiting for clock tick           Decodethread waiting for clock tick          Executethread waiting for clock tick         MemoryThread waiting for clock tick          WBthread waiting for clock tick
 ========================================================================================================================================================================================================
 Clock Cycle count = 1
Fetchthread starting new clock               Decodethread starting new clock              Executethread starting new clock             MemoryThread starting new clock              WBthread starting new clock
JMuxSel for current Cycle =0                 Reading data...                              ReadOutg data...                             ReadOutg data...                             ReadOutg data...
BaddressE for current Cycle =0               PC=0                                         PCOut=0                                      PCOut=0                                      PCout=0
UnitAddressOutput for current Cycle =0       MC=0                                         MCOut=0                                      MCOut=0                                      MCout=0
Fetched instruction (PC = 0): 20080010       PredictionOutD=0                             RegWriteEnOut=0                              RegWriteEnOut=0                              RegWriteEnOut=0
JMuxSel after exe inp=0                      #################                            MemWriteEnOut=0                              MemtoRegOut=0                                MemtoRegOut=0
Is Flush = 0                                 Writing data...                              MemtoRegOut=0                                MemWriteEnOut=0                              ReadDataOut=0
Writing data...                              MC=0                                         MemReadEnOut=0                               MemReadEnOut=0                               AddressOut=0
 fPC = 00000001 fMC = 20080010               PCin=0                                       FCOut=0                                      resultOut=0                                  WriteRegisterOut=0
PredictionInF=0                              RegWriteEnin=1                               JrSingalOut=0                                MemreaddataOut=0                             #################
Fetchthread waiting for clock tick           MemWriteEnin=0                               BranchOut=0                                  WriteDataOut=0                               wPC = 00000000 wMC = 0
                                             MemtoRegin=0                                 ZeroSignalOut=0                              WriteRegisterOut=0                           WBthread waiting for clock tick
                                             MemReadEnin=0                                ALUOpOut=0                                   #################
                                             FCin=0                                       RegDstOut=0                                  mPC = 00000000 mMC = 0
                                             FDin=1                                       readdata1Out=0                               Writing data...
                                             JrSingalin=0                                 readdata2Out=0                               mPC = 00000000 mMC = 0
                                             Branchin=0                                   immediateOut=0                               PCin=0
                                             ZeroSignalin=0                               rsOut=0                                      MCin=0
                                             ALUOpin=7                                    rtOut=0                                      RegWriteEnIn=0
                                             RegDstin=1                                   rdOut=0                                      RegWriteEnIn=0
                                             readdata1in=0                                PredictionOutE=0                             MemtoRegIn=0
                                             readdata2in=0                                #################                            ReadDataIn=ffffffff
                                             immediatein=0                                is_Hit= 1                                    AddressIn=0
                                             rsin=0                                       EXEdata.Branch= 0                            WriteRegisterIn=0
                                             rtin=0                                       Writing data...                              MemoryThread waiting for clock tick
                                             rdin=0                                       PCIn=0
                                             Prediction=0                                 MCin=0
                                             Decoding logic done...                       RegWriteEnIn=0
                                             Decodethread waiting for clock tick          MemtoRegIn=0
                                                                                          MemWriteEnIn=0
                                                                                          MemReadEnIn=0
                                                                                          resultIn==0
                                                                                          WriteDataIn=0
                                                                                          MemreaddataIn=ffffffff
                                                                                          WriteRegisterIn=0
                                                                                          ePC = 00000000 eMC = 0
                                                                                          READDATA MEM after execute= ffffffff
                                                                                          Executethread waiting for clock tick
```

Simulator Visualization

In our simulator, we visualized the data flow as follows

Where each stage is represented by a column.

Each stage displays the values its reading from the previous pipe, and after it processes it displays what it is about to write to the next pipe in the next clock cycle.

Mind that the implicit write back stage here is shown in the simulator, but it behaves as expected and explain in the pipeline design chapter.

When the execution is done the simulator runs for some extra clock cycles to make sure that the pipeline is drained.

```
==========================================================================================================================================
 Clock Cycle count = 21
Fetchthread starting new clock          Decodethread starting new clock         Executethread starting new clock        MemoryThread starting new clock         WBthread starting new clock
No more instructions to fetch. Halt inserted Reading data...                    ReadOutg data...                        ReadOutg data...                        ReadOutg data...
DMuxSel after exe inp=0                 PC=0                                    PCOut=0                                 PCOut=0                                 PCout=12
Writing data...                         MC=0                                    MCOut=0                                 MCOut=0                                 MCout=22980002
 fPC = 00000000 fMC = 0                 PredictionOutD=0                        RegWriteEnOut=1                         RegWriteEnOut=1                         RegWriteEnOut=1
PredictionInF=0                         ################                        MemWriteEnOut=0                         MemtoRegOut=0                           MemtoRegOut=0
Fetchthread waiting for clock tick      Writing data...                         MemtoRegOut=0                           MemWriteEnOut=0                         ReadDataOut=ffffffff
                                        MC=0                                    MemReadEnOut=0                          MemReadEnOut=0                          AddressOut=3
                                        PCin=0                                  FCOut=0                                 resultOut=0                             WriteRegisterOut=18
                                        RegWriteEnin=1                          JrSingalOut=0                           MemreaddataOut=ffffffff                 ################
                                        MemWriteEnin=0                          BranchOut=0                             WriteDataOut=0                          wPC = 00000012 wMC = 22980002
                                        MemtoRegin=0                            ZeroSignalOut=0                         WriteRegisterOut=0                      WBthread waiting for clock tick
                                        MemReadEnin=0                           ALUOpOut=7                              ################
                                        FCin=0                                  RegDstOut=1                             mPC = 00000000 mMC = 0
                                        FDin=1                                  readdata1Out=0                          Writing data...
                                        JrSingalin=0                            readdata2Out=0                          mPC = 00000000 mMC = 0
                                        Branchin=0                              immediateOut=0                          PCin=0
                                        ZeroSignalin=0                          rsOut=0                                 MCin=0
                                        ALUOpin=7                               rtOut=0                                 RegWriteEnIn=1
                                        RegDstin=1                              rdOut=0                                 RegWriteEnIn=1
                                        readdata1in=0                           PredictionOutE=0                        MemtoRegIn=0
                                        readdata2in=0                           ################                        ReadDataIn=ffffffff
                                        immediatein=0                           is_Hit= 1                               AddressIn=0
                                        rsin=0                                  EXEdata.Branch= 0                       WriteRegisterIn=0
                                        rtin=0                                  Writing data...                         MemoryThread waiting for clock tick
                                        rdin=0                                  PCIn=0
                                        Prediction=0                            MCin=0
                                        Decoding logic done...                  RegWriteEnIn=1
                                        Decodethread waiting for clock tick     MemtoRegIn=0
                                                                                MemWriteEnIn=0
                                                                                MemReadEnIn=0
                                                                                resultIn==0
                                                                                WriteDataIn=0
                                                                                MemreaddataIn=ffffffff
                                                                                WriteRegisterIn=0
                                                                                ePC = 00000000 eMC = 0
                                                                                READDATA MEM after execute= ffffffff
                                                                                Executethread waiting for clock tick
==========================================================================================================================================
 Clock Cycle count = 22
Fetchthread starting new clock          Decodethread starting new clock         Executethread starting new clock        MemoryThread starting new clock         WBthread starting new clock
No more instructions to fetch. Halt inserted Reading data...                    ReadOutg data...                        ReadOutg data...                        ReadOutg data...
DMuxSel after exe inp=0                 PC=0                                    PCOut=0                                 PCOut=0                                 PCout=0
Writing data...                         MC=0                                    MCOut=0                                 MCOut=0                                 MCout=0
 fPC = 00000000 fMC = 0                 PredictionOutD=0                        RegWriteEnOut=1                         RegWriteEnOut=1                         RegWriteEnOut=1
PredictionInF=0                         ################                        MemWriteEnOut=0                         MemtoRegOut=0                           MemtoRegOut=0
Fetchthread waiting for clock tick      Writing data...                         MemtoRegOut=0                           MemWriteEnOut=0                         ReadDataOut=ffffffff
                                        MC=0                                    MemReadEnOut=0                          MemReadEnOut=0                          AddressOut=0
                                        PCin=0                                  FCOut=0                                 resultOut=0                             WriteRegisterOut=0
                                        RegWriteEnin=1                          JrSingalOut=0                           MemreaddataOut=ffffffff                 ################
                                        MemWriteEnin=0                          BranchOut=0                             WriteDataOut=0                          wPC = 00000000 wMC = 0
                                        MemtoRegin=0                            ZeroSignalOut=0                         WriteRegisterOut=0                      WBthread waiting for clock tick
                                        MemReadEnin=0                           ALUOpOut=7                              ################
                                        FCin=0                                  RegDstOut=1                             mPC = 00000000 mMC = 0
                                        FDin=1                                  readdata1Out=0                          Writing data...
                                        JrSingalin=0                            readdata2Out=0                          mPC = 00000000 mMC = 0
                                        Branchin=0                              immediateOut=0                          PCin=0
                                        ZeroSignalin=0                          rsOut=0                                 MCin=0
```

And when the execution of the given program is finished, we dumb the values of the register.

```
==========================================================================
 Clock Cycle count = 24
Fetchthread starting new clock                Decodethread starting new clock
No more instructions to fetch. Halt inserted Reading data...
JMuxSel after exe inp=0                       PC=0
Writing data...                               MC=0
 fPC = 00000000 fMC = 0                       PredictionOutD=0
PredictionInF=0                               ################
Register File Contents:                       Writing data...
R0: 0                                         MC=0
R1: 0                                         PCin=0
R2: 0                                         RegWriteEnin=1
R3: 0                                         MemWriteEnin=0
R4: 0                                         MemtoRegin=0
R5: 0                                         MemReadEnin=0
R6: 0                                         FCin=0
R7: 0                                         FDin=1
R8: 10                                        JrSingalin=0
R9: 5                                         Branchin=0
R10: 15                                       ZeroSignalin=0
R11: 15                                       ALUOpin=7
R12: ffffffea                                 RegDstin=1
R13: 1                                        readdata1in=0
R14: 0                                        readdata2in=0
R15: 0                                        immediatein=0
R16: 10                                       rsin=0
R17: 10                                       rtin=0
R18: 50                                       rdin=0
R19: 0                                        Prediction=0
R20: 1                                        Decoding logic done...
R21: 1
R22: 0
R23: 0
R24: 3
R25: 0
R26: 0
R27: 0
R28: 0
R29: 7ffffffc
R30: 0
R31: 0


C:\Users\USER\Desktop\PIpelineCAS3\CAS\x64\Debug\CAS.exe (process 34924) exited
To automatically close the console when debugging stops, enable Tools->Options-
Press any key to close this window . . .
```

If there was memory access in the program, a file containing the memory elements after execution

Our simulator is yet to support, memory initialization. The User must use SW instruction to store data in the memory.

```
*DATAMEMORY - Notepad
File  Edit  Format  View  Help
0x00000000:    +0x0: 0x00000001    +0x4: 0x00000002    +0x8: 0x00000003    +0xc: 0x00000004
0x00000008:    +0x0: 0x00000009    +0x4: 0x0000000a    +0x8: 0x0000000b    +0xc: 0x0000000c
```