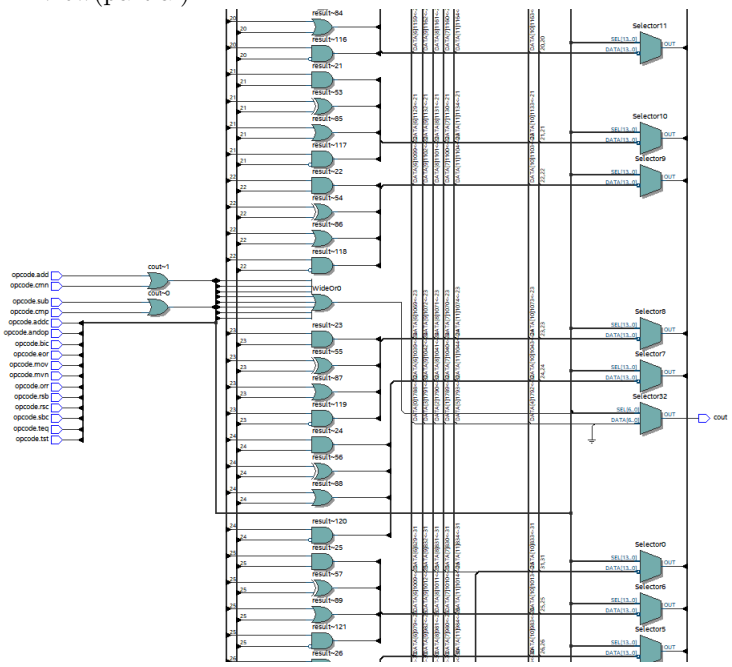
COL216 Assignment 2 Stage 1

Chinmay Mittal

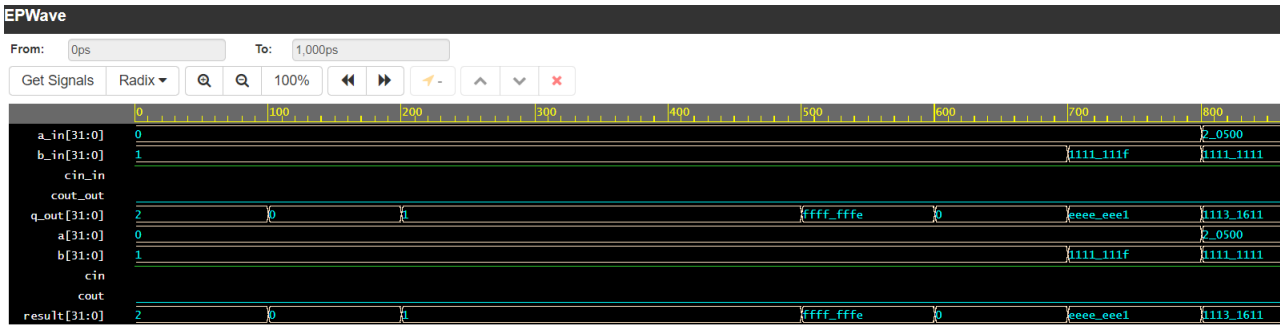
February 2022

ALU

To test ALU code load, ALU.vhd , types.vhd , testbenchALU.vhd. Choose appropriate tool and also remember to set top level entity as testbenchALU.
RTL view(partial):



Simulation view:

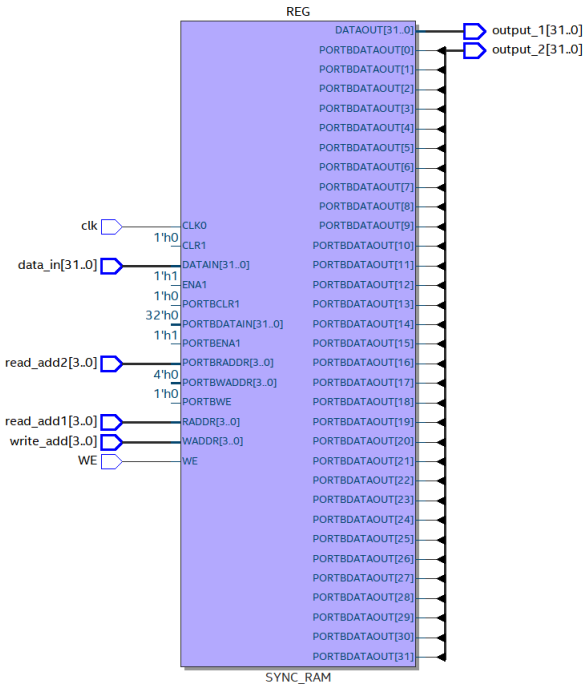


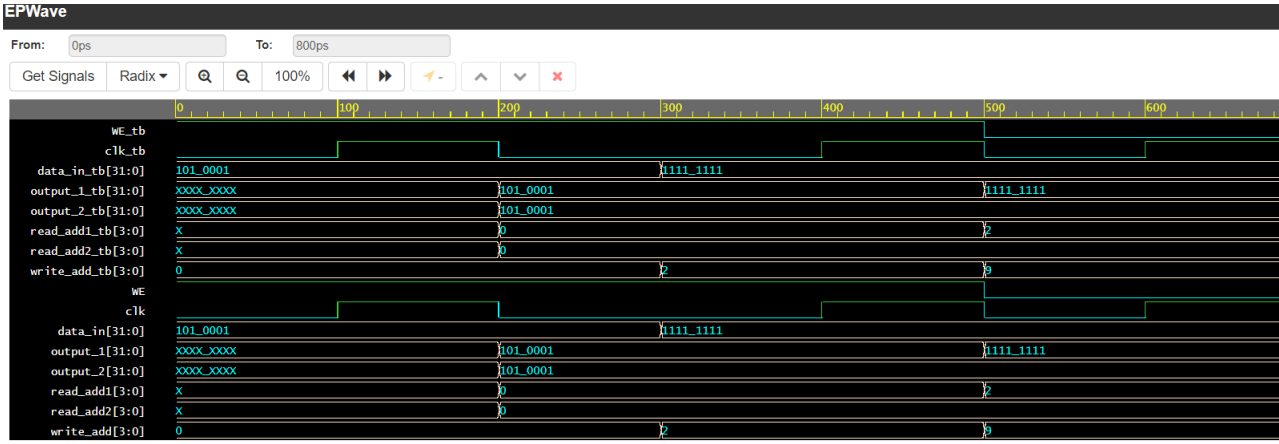
Resource utilization:

```
# Info: ************************************************************
# Info: Device Utilization for 7A100TCSG324
# Info: ************************************************************
# Info: Resource                          Used    Avail   Utilization
# Info: ------------------------------------------------------------
# Info: IOs                               114     210      54.29%
# Info: Global Buffers                    0       32        0.00%
# Info: LUTs                              104     63400     0.16%
# Info: CLB Slices                        26      15850     0.16%
# Info: Dffs or Latches                   0       126800    0.00%
# Info: Block RAMs                        0       135       0.00%
# Info: DSP48E1s                          0       240       0.00%
# Info: ------------------------------------------------------------
# Info: ****************************************************
# Info: Library: work    Cell: ALU    View: arch
# Info: ****************************************************
# Info:  Number of ports :                          114
# Info:  Number of nets :                           362
# Info:  Number of instances :                      282
# Info:  Number of references to this view :          0
# Info: Total accumulated area :
# Info:  Number of LUTs :                           104
# Info:  Number of Primitive LUTs :                 105
# Info:  Number of LUTs with LUTNM/HLUTNM :           2
# Info:  Number of MUX CARRYs :                      32
# Info:  Number of accumulated instances :          282
# Info: ******************************
```

Register File

To run load RegisterFile.vhd and testbenchRegister.vhd. Also choose top level entity as testbenchRegister.
16*32 Memory, with two read address ports, write enable, two data out ports and a clock, writing is clock triggered whereas reading is combinational and continuous.
RTL view:





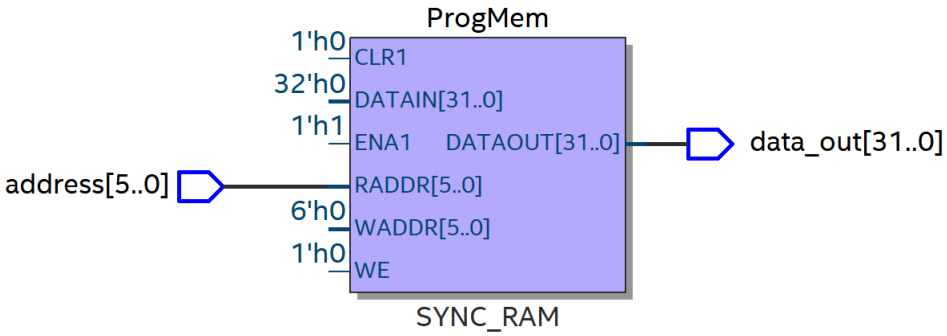
Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Resource Utilization:

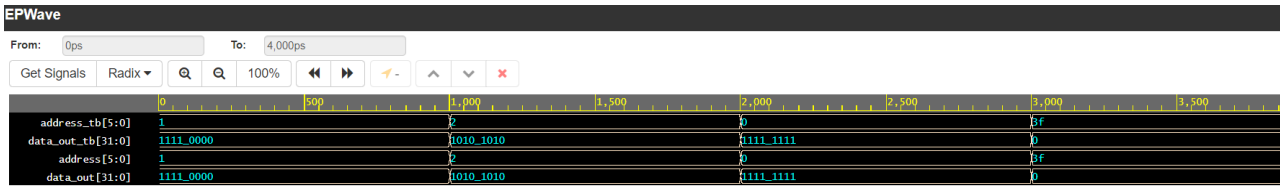
```
# Info: Device Utilization for 7A100TCSG324
# Info: ***********************************************************
# Info: Resource                          Used    Avail   Utilization
# Info: -----------------------------------------------------------
# Info: IOs                               110     210      52.38%
# Info: Global Buffers                    1       32        3.12%
# Info: LUTs                              48      63400     0.08%
# Info: CLB Slices                        12      15850     0.08%
# Info: Dffs or Latches                   0       126800    0.00%
# Info: Block RAMs                        0       135       0.00%
# Info: Distributed RAMs
# Info:    RAM32M                         10
# Info:    RAM64M                         2
# Info: DSP48E1s                          0       240       0.00%
# Info: -----------------------------------------------------------
# Info: ***********************************************************
# Info: Library: work    Cell: RegisterFile    View: RegisterArch
# Info: ***********************************************************
# Info:  Number of ports :                          110
# Info:  Number of nets :                           220
# Info:  Number of instances :                      111
# Info:  Number of references to this view :          0
# Info: Total accumulated area :
# Info:  Number of LUTs :                            48
# Info:  Number of Primitive LUTs :                  48
# Info:    Number of LUTs as Distributed RAM :       48
# Info:  Number of accumulated instances :          123
```

Program Memory

To run load ProgramMemory.vhd, testbenchProgMem.vhd and also set top level entity as testbenchProgMem.
Program Memory is 64*32 ROM, with one address port for data output. There is no clock and output is available continuously.
Output of synthesis



Some dummy values where initialized in the program memory for testing, corresponding simulation results is as shown:

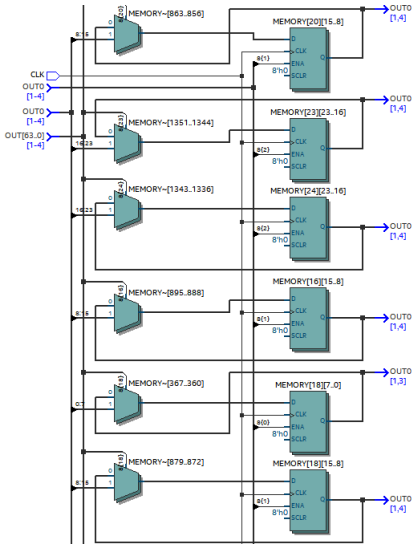


Resource Utilization

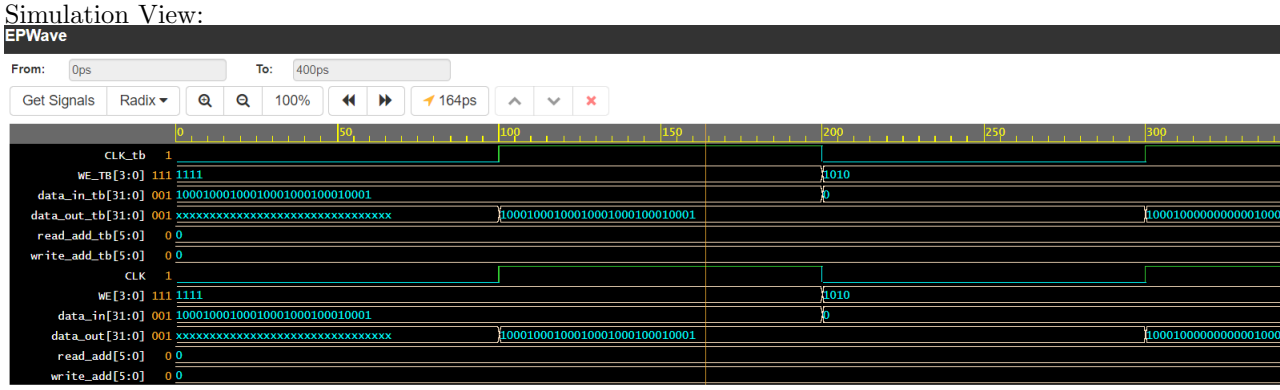
```
# Info: ************************************************************
# Info: Device Utilization for 7A100TCSG324
# Info: ************************************************************
# Info: Resource                          Used    Avail   Utilization
# Info: ------------------------------------------------------------
# Info: IOs                               38      210      18.10%
# Info: Global Buffers                    0       32        0.00%
# Info: LUTs                              4       63400     0.01%
# Info: CLB Slices                        1       15850     0.01%
# Info: Dffs or Latches                   0       126800    0.00%
# Info: Block RAMs                        0       135       0.00%
# Info: DSP48E1s                          0       240       0.00%
# Info: ------------------------------------------------------------
# Info: ************************************************************
# Info: Library: work    Cell: ProgramMemory    View: ProgramMemoryArch
# Info: ************************************************************
# Info:  Number of ports :                          38
# Info:  Number of nets :                           49
# Info:  Number of instances :                      43
# Info:  Number of references to this view :         0
# Info: Total accumulated area :
# Info:  Number of LUTs :                            4
# Info:  Number of Primitive LUTs :                  4
# Info:  Number of accumulated instances :          43
```

Data Memory

To run load DataMemory.vhd, testbenchDataMem.vhd and also set top level entity as testbenchDataMem. 64 register of 32 bits each, one write address port, one read address port, one data out and one data in port, Write Enable is a 4 bit vector for controlling write to 4 parts of each word. Writing is clocked and reading is synchronous. RTL View (partial):



Simulation View:



Resource Utilization:

```
): ********************************************************************
): Device Utilization for 7A100TCSG324
): ********************************************************************
): Resource                          Used    Avail   Utilization
): --------------------------------------------------------------------
): IOs                               81      210      38.57%
): Global Buffers                    1       32        3.12%
): LUTs                              64      63400     0.10%
): CLB Slices                        16      15850     0.10%
): Dffs or Latches                   0       126800    0.00%
): Block RAMs                        0       135       0.00%
): Distributed RAMs
):    RAM64X1D                       32
): DSP48E1s                          0       240       0.00%
): --------------------------------------------------------------------
): ********************************************************************
): Library: work    Cell: DataMemory    View: DataMemoryArch
): ********************************************************************
):  Number of ports :                          81
):  Number of nets :                           162
):  Number of instances :                      82
):  Number of references to this view :         0
): Total accumulated area :
):  Number of LUTs :                           64
):  Number of Primitive LUTs :                 64
):    Number of LUTs as Distributed RAM :      64
):  Number of accumulated instances :         113
```

1 Foot notes

The RTL diagrams are generated by intel quartus. Resource utilization table and simulation was done by edaplayground. The mode for simulator was Aldec Riviera Pro and to synthisize Mentor precision was used.