Matrix Multiplication 2-Verification

Digital Design & Logic Synthesis

Project: Matix Multiplication

Block: matmul

Digital High Level Verification

Version 0.2

Version 0.1 4 June 2007

Revision Log

Rev	Change	Description	Reason for change	Done By	Date
0.1	Initial document			Roee Shahmoon	14, Apr, 2024
0.2	Digital Changes			Noam Klainer	14,Apr, 2024

Table of Content

LIST OF FIGURES

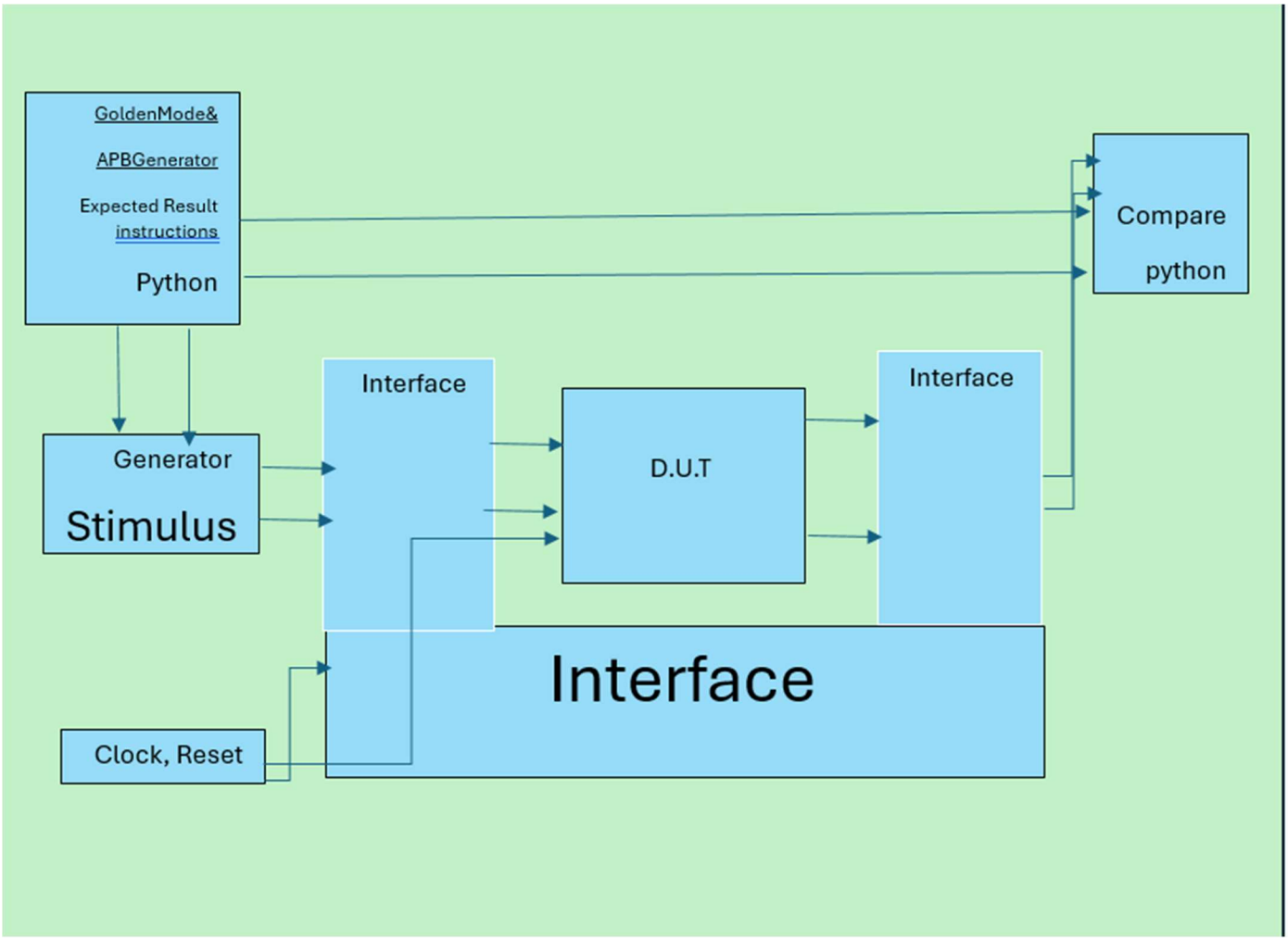
LIST OF TABLES

1. VERIFICATION PLAN

1.1 Verification Test Objectives

1.2 Test Bench High Level Diagram and Architecture

Test bench (overall)



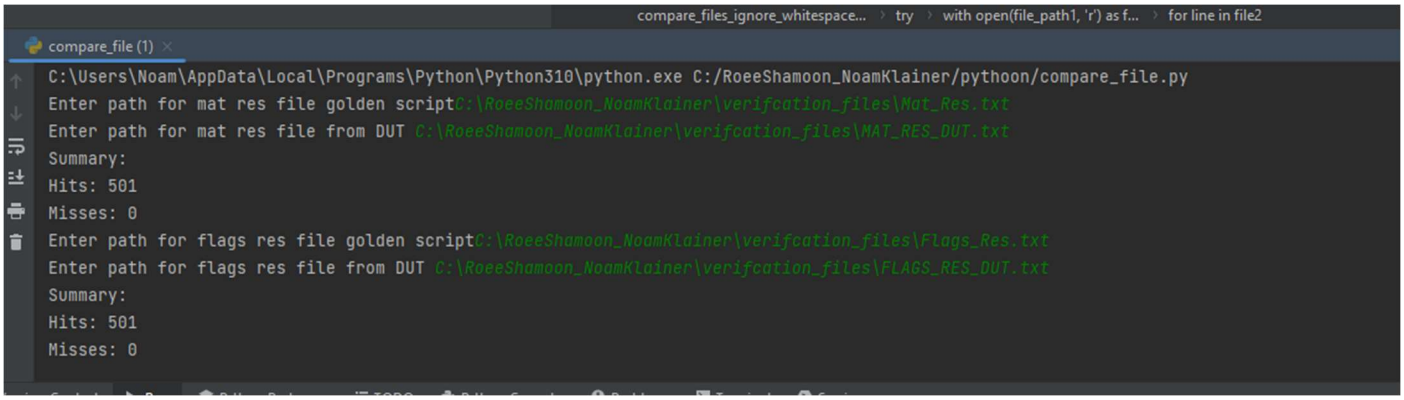
The purpose of the stimulus is to write data to design and read data, to do so it contains and APB master which is responsible to toggle APB bus properly. It also contains a sequence generator where we decide what APB transactions are sent to DUT and in what order, the stimulus also contains tasks to create 2 files for result, MAT_RES_DUT.txt, FLAGS_RES_DUT.txt. those files are for comparing with the golden.

The decided to connect clock and reset signals from a generator to all blocks in the testbench, and not from stimulus because we want to control those signals from the testbench overall without changing stimulus.

Because we wanted to read all the data for the testbench from file, we randomize all the data on the python script. We wrote those data into files; the stimulus read instructions from file and generate APB master to write this data into the design. In this file we did a lot of tests together. After we finish writing to design all the data for the specific test, the stimulus going to sleep until the design is done. When the design is done, we read the result into 2 files: mat res, flag res. We also doing this in the stimulus with the APB master.

In the end we are comparing between the result files from DUT to golden script.

We print to the screen how much hits, and where we missed.



Example for compare script.

1.3 Test Bench Low Level Architecture and Functionality

The stimulus is feeding the design with all is information he need. When the stimulus is going to write to the design start bit is high (writing to control register) the stimulus realize that the design will start operation next cycle so we are going to sleep and waiting to done signal

```
if ( paddr_o == 0 && pwdata_o[0]) begin
    @(posedge clk)
    penable_o = 1;
    @(posedge clk)
    penable_o = 0;
    pwrite_o = 0;
    wait(done_i == 1);
    counter_test = counter_test + 1;
    write_mat_res_file(counter_test);
    write_flags_file(counter_test);
```

Address 0 is for control register and pwdata_o[0] is for the start bit.

So after the design done we write into result file for comparing and continue to the next test.

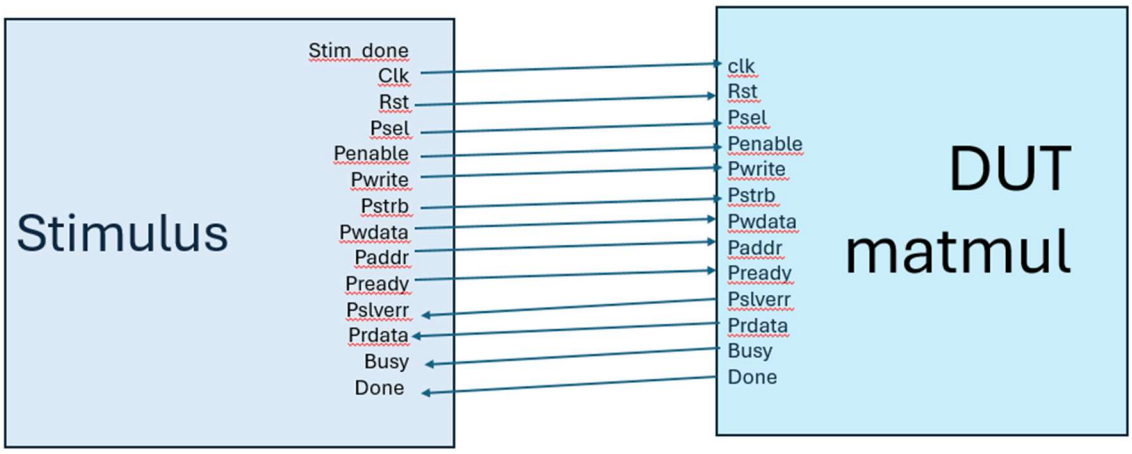


Figure 2: Test Bench Block Diagram

1.4 Golden Model

First in order to test our design we create golden model in python.

The script asking from the user to give a path to the verification files, to control the process.

Is easy to run these tests from different computer environments, just change the path.

```
script_path = input('Enter a path for verifacation files')
# Define the path to the directory
directory_path =os.path.join(script_path, 'RoeeShamoon_NoamKlainer\\verifcation_files')
```

in our golden model script we create 7 files:

1. Bushflies- each line contain data for pwdata, paddr, pstrb signals
2. Mat_A-randomize result of mat A in each test iteration.
3. Mat_B-randomize result of mat B in each test iteration.
4. Mat_Res-result of A*B+C from the golden model for comparing.
5. SP-scratch pad memory from the golden model for comparing.
6. Param_File – randomize value for BUS_WIDH, DATA_WIDH parameters with the constraint BW=BUS_WIDTH, DW=DATA_WIDH.
   $\frac{BW}{DW} \leq 4$ and $(BW \,\&\%DW = 0)$ and write the value of the control register from our golden model for comparing.
7. FLAGS_RES –result of flags register from our golden model for comparing.

We randomize values for parameter, matrix A and B, control register and based on those values we are doing A*B+C and keep the result in file. We are doing this for every test, parameter we randomize only once at start.

After we run.

Example of randomize:

```
# Parmeters Randomised
BusWidth = np.random.choice([16, 32, 64])#randomize BUS WIDTH parameter for DUT
DataWidth = np.random.choice([8, 16, 32])#randomize DATA WIDTH parameter for DUT
AddressWidth = np.random.choice([16, 24, 32])#randomize ADDR WIDTH parameter for DUT
SPN = np.random.choice([1, 2, 4])#randomize SP NTARGETS parameter for DUT
Max_Dim = BusWidth // DataWidth
while DataWidth > BusWidth // 2 and Max_Dim <= 4:#constraint c{2*DataWidth <= BusWidth && Max_Dim <= 4}
    DataWidth = np.random.choice([8, 16, 32])
    Max_Dim = BusWidth // DataWidth
```

Figure 1 set of codes randomize parameter for DUT

```
ControlRegStr = controlreg(Max_Dim, SPN)
```

```
def controlreg(Max_Dim: int, SPN: int) -> str:
    N = np.random.randint(Max_Dim)
    K = np.random.randint(Max_Dim)
    M = np.random.randint(Max_Dim)
```

Figure 2 set of codes randomize dimension N, K, M for DUT

```
# Set the start bit (bit 0)
ControlReg |= (0b0)

# Set the mode bit (bit 1)
ControlReg |= (np.random.randint(2) << 1)
#set mode bit to 1 to chek flags
ControlReg |= (1 << 1)
# Set the write target bits (bits 2 to 3)
ControlReg |= (np.random.randint(SPN) << 2)
#set write target always to 1 to make overflow to chek flags because we read also from 1
ControlReg |= (0 << 2)
# Set the read target bits (bits 4 to 5)
ControlReg |= (np.random.randint(SPN) << 4)
# set read target always to 1 to make overflow to chek flags because we write also from 1
ControlReg |= (0 << 4)
# Not in use (bits 6 to 7)
ControlReg |= (0b00 << 6)

# Dimension N (bits 8 to 9)
ControlReg |= (N << 8)
```

Figure 3 set of codes to randomize values for control register, start bit is not random.

2. VERIFCATION RESULTS

```
Enter path for mat res file golden scriptC:\RoeeShamoon_NoamKlainer\verifcation_files\Mat_Res.txt
Enter path for mat res file from DUT C:\RoeeShamoon_NoamKlainer\verifcation_files\MAT_RES_DUT.txt
Summary:
Hits: 16
Misses: 0
Enter path for flags res file golden script
```

Compare results for matrix.

```
Enter path for flags res file golden scriptC:\RoeeShamoon_NoamKlainer\verifcation_files\Flags_Res.txt
Enter path for flags res file from DUT C:\RoeeShamoon_NoamKlainer\verifcation_files\FLAGS_RES_DUT.txt
Summary:
Hits: 16
Misses: 0

Process finished with exit code 0
```

Compare results for flags.

2.1 Test 1

```
:Mat A Random in Test 1 is

0
----------------------------------------------------------------------------------------
```

Matrix A in test 1

```
:Mat B Random in Test 1 is

23773-,0
```

Matrix B in test 1

```
:Mat Res in Test 1 is

0,0
0,0
----------------------------------------------------------------------------------------
```

matrix from DUT test 1

```
:Flags in Test 1 is

0,0
0,0
----------------------------------------------------------------------------------------
```

flags from DUT test 1

```
:Flags in Test 1 is

0,0
0,0
----------------------------------------------------------------------------------------
```

flags from golden model test 1

2.2 Test 2

```
                                                                                :Mat A Random in Test 2 is

                                                                                                      1794,0
                                                                                                     30550,0
--------------------------------------------------------------------------------------------------------------
```

Matrix A in test 2

```
--------------------------------------------------------------------------------------------------------------
                                                                                :Mat B Random in Test 2 is

                                                                                                     31171
                                                                                                       986
--------------------------------------------------------------------------------------------------------------
```

Matrix B in test 2

```
--------------------------------------------------------------------------------------------------------------
                                                                                :Mat Res in Test 2 is

                                                                                                  55920774,0
                                                                                                 952274050,0
--------------------------------------------------------------------------------------------------------------
```

Output from DUT

```
                                                                                :Mat Res in Test 2 is

                                                                                                  55920774,0
                                                                                                 952274050,0
--------------------------------------------------------------------------------------------------------------
```

Output from golden model.

2.3 Test 3

```
---------------------------------------------------------------------------------------------
                                                        :Mat A Random in Test 3 is

                                                                        1794,12973
                                                                        30550,2647
---------------------------------------------------------------------------------------------
```

Matrix A in test 3

```
---------------------------------------------------------------------------------------------
                                                        :Mat B Random in Test 3 is

                                                                        31171
                                                                        986
---------------------------------------------------------------------------------------------
```

Matrix B in test 3

```
---------------------------------------------------------------------------------------------
                                                        :Mat Res in Test 3 is

                                                                        124632926,0
                                                                        1907158042,0
---------------------------------------------------------------------------------------------
```

Output from DUT

```
                                                        :Mat Res in Test 3 is

                                                                        124632926,0
                                                                        1907158042,0
```

Output from golden model.

3. APPENDIX

3.1 Terminology

LSB - Least Significant Bit

TBR - To Be Reviewed

TBD - To Be Defined

IF - Inteface

3.2 References