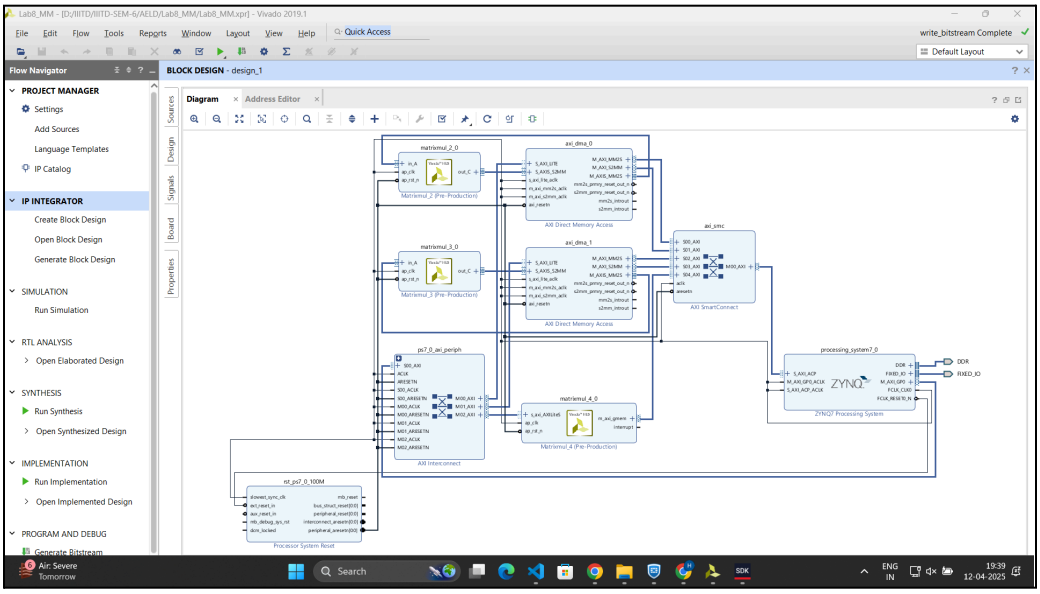
Design of a Custom IP for 8x8 matrix multiplication

Vivado Block Design



Pragmas Used for Optimisations:

- ***#pragma HLS PIPELINE:*** Directive used in High-Level Synthesis (HLS) tools like Xilinx Vivado HLS to instruct the compiler to pipeline a specific section of code, typically a loop or function. Pipelining improves performance by allowing the overlapping execution of operations across multiple clock cycles.
  By Default, it tries to reduce the **initiation interval** of a loop to 1.
- ***#pragma ARRAY_PARTITION****:* In hardware synthesis, arrays are typically mapped to memory structures like block RAMs (BRAMs). When multiple operations try to access the same memory in a single clock cycle, memory access conflicts arise, which can limit parallelism and pipelining.

Array partitioning removes this bottleneck by splitting the array into smaller, independently accessible memory blocks or registers, enabling parallel access to different parts of the array. This is essential for achieving high throughput in pipelined designs.

Example :

Matrix A[2][1] = [[1,2,3],[4,5,6]]

If we use *#pragma ARRAY_PARTITION variable = A complete dim = 1,* then the array is split into two arrays, [1,2,3] and [4,5,6]. Now, each row can be mapped to a different BRAM or register block, allowing concurrent access to elements from different rows, which would not be possible with a single BRAM.

Results

Method	Avg Execution Time (Micro-Seconds)
PS	26.074459
Pipelined MMUL in PL	31.892307
Pipelined + AP MMUL in PL	7.283692
Pipelined + AP + Memory Mapped MMUL in PL	6.415385

Reasons :

The table shows average execution times for different methods of matrix multiplication (MMUL) on a processing system (PS) and programmable logic (PL). Here's an analysis of the reasons for the differences in execution times:

PS (26.074459 µs):

Reason: The PS (Processing System) refers to a general-purpose processor (ARM Cortex-A9). It executes the matrix multiplication sequentially using software. While this method benefits from high clock speeds and optimized CPU instructions, it lacks parallelism, leading to a moderate execution time compared to PL-based methods.

Pipelined MMUL in PL (31.892307 µs):

Reason: This method uses pipelining in the programmable logic (FPGA). Pipelining breaks down the matrix multiplication into stages, allowing concurrent processing of different parts of the computation. However, the execution time is higher than PS because the PL clock frequency is typically lower than the PS, and the overhead of setting up the pipeline (e.g., data transfer between PS and PL, pipeline stalls) may outweigh the benefits for the given matrix size or implementation.

Pipelined + AP MMUL in PL (7.283692 µs):

Reason: Array Partitioning (AP) in the PL involves dividing arrays into smaller blocks that can be processed in parallel. This enables better resource utilisation on the FPGA, such as using multiple DSP slices or LUTs concurrently. Combined with pipelining, this parallelism significantly reduces execution time compared to both the PS and the basic pipelined method by allowing simultaneous computations on different array segments.

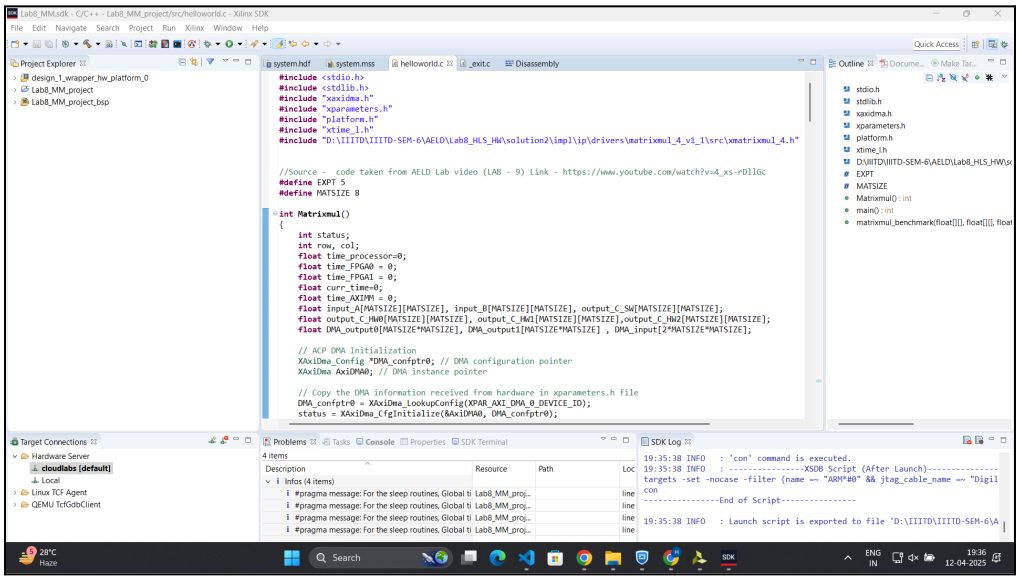
Pipelined + AP + Memory Mapped MMUL in PL (6.415385 µs):

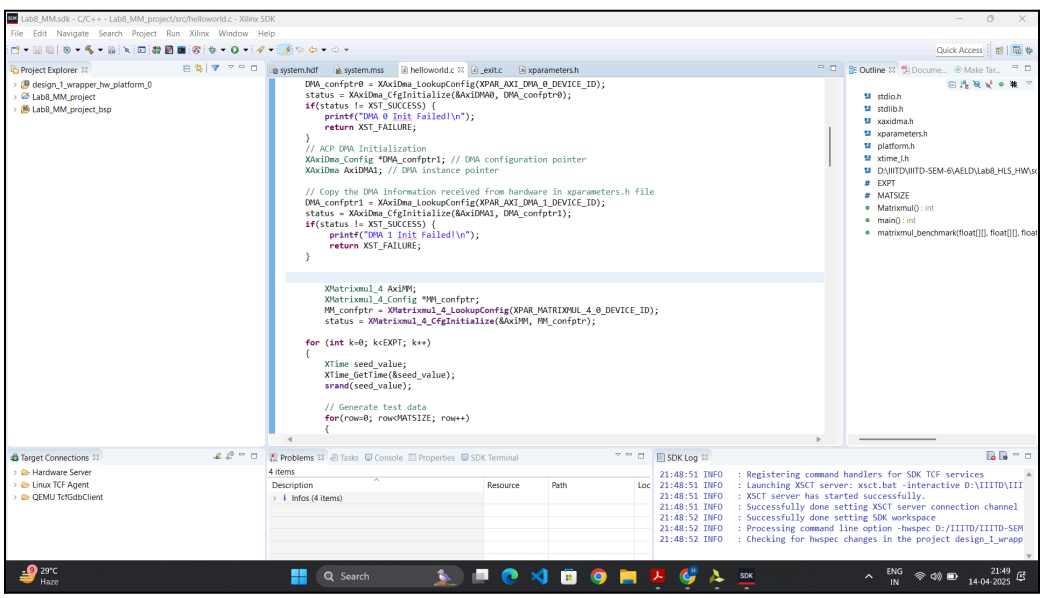
Reason: The addition of Memory Mapped maps the PL directly to memory (without using DMA). This reduces data transfer bottlenecks between the PS and PL, allowing faster access to matrix data. The slight improvement over the previous method (7.283692 µs) is due to reduced latency in memory operations, making the overall process more efficient.

Results:

- Achieved 4 times speedup, reducing running time from 26.07µs to 6.41µs using memory-mapped interface.

Xilinx SDK (Software Development Kit Code)





Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
for (int k=0; k<EXPT; k++)
{
    XTime seed_value;
    XTime_GetTime(&seed_value);
    srand(seed_value);

    // Generate test data
    for(row=0; row<MATSIZE; row++)
    {
        for(col=0; col<MATSIZE; col++)
        {
            input_A[row][col] = ((float)rand()/(RAND_MAX/5));
            input_B[row][col] = ((float)rand()/(RAND_MAX/5));
        }
    }

    // to store the time at which certain processes starts and ends
    XTime time_PS_start, time_PS_end;
    XTime time_PL_start, time_PL_end; // PL time calculations

    // Matrix Multiplication on PS
    XTime_SetTime(0);
    XTime_GetTime(&time_PS_start); // Capture the timer value at the start

    // Call software benchmark function
    matrixmul_benchmark(input_A, input_B, output_C_SW);

    XTime_GetTime(&time_PS_end); // Capture the timer value at the end
    curr_time = ((float)1.0 * (time_PS_end - time_PS_start) / (COUNTS_PER_SECOND/1000000));
    time_processor = time_processor + curr_time;

    printf("****************************************\n");
```

Outline
- stdio.h
- stdlib.h
- xaxidma.h
- xparameters.h
- platform.h
- xtime_l.h
- D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\sc
- EXPT
- MATSIZE
- Matrixmul() : int
- main() : int
- matrixmul_benchmark(float[][], float[][], float

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

SDK Log

```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

28°C Haze | Search | ENG IN | 19:36 12-04-2025

Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
    printf("Execution Time for PS in Micro-Seconds for %d iteration: %f\n", k, curr_time);
    ////////////////////////////////////////////////////////////////////


    // Generate DMA Input
    int index = 0;
    for(row = 0; row < MATSIZE; row++)
    {
        for(col = 0; col < MATSIZE; col++)
        {
            DMA_input[index] = input_A[row][col];
            index = index + 1;
        }
    }

    for(row = 0; row < MATSIZE; row++)
    {
        for(col = 0; col < MATSIZE; col++)
        {
            DMA_input[index] = input_B[row][col];
            index = index + 1;
        }
    }

    // Matrix Multiplication on PL using DMA 0
    XTime_SetTime(0);
    XTime_GetTime(&time_PL_start); // Capture the timer value at the start

    status = XAxiDma_SimpleTransfer(&AxiDMA0, (UINTPTR)DMA_output0,(sizeof(float)*MATSIZE*MATSIZE), XAXIDMA
    status = XAxiDma_SimpleTransfer(&AxiDMA0, (UINTPTR)DMA_input,(sizeof(float)*MATSIZE*MATSIZE*2), XAXIDMA
    // We have only configured the DMA to perform these two transactions.. DMA might not have started the t
    status = XAxiDma_ReadReg(XPAR_AXI_DMA_0_BASEADDR, 0x04) & 0x00000002;
    while(status != 0x00000002) {
```

Outline
- stdio.h
- stdlib.h
- xaxidma.h
- xparameters.h
- platform.h
- xtime_l.h
- D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\sc
- EXPT
- MATSIZE
- Matrixmul() : int
- main() : int
- matrixmul_benchmark(float[][], float[][], float

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

SDK Log

```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

28°C Haze | Search | ENG IN | 19:36 12-04-2025

Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
        printf("Execution Time for Pipelined MMUL in PL in Micro-Seconds for %d iteration: %f\n", k, curr_time)

        // Receive stream output C from hardware function
        index = 0;
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
                output_C_HW0[row][col] = DMA_output0[index];
                index = index + 1;
            }
        }
        // Compare benchmark and hardware functions output
        // Receive stream output C from hardware function
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
                float diff = abs(output_C_HW0[row][col] - output_C_SW[row][col]);
                if(diff > 0.001)
                {
                    printf("Pipelined MMUL Error at row index %d and column index %d\t\n", row, col);
                    printf("Hardware output %f\n", output_C_HW0[row][col]);
                    printf("Software output %f\n\n", output_C_SW[row][col]);
                    break;
                }
            }
        }

        // Matrix Multiplication on PL using DMA 1
        XTime_SetTime(0);
        XTime_GetTime(&time_PL_start); // Capture the timer value at the start
```

Outline: stdio.h, stdlib.h, xaxidma.h, xparameters.h, platform.h, xtime_l.h, D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\sc, EXPT, MATSIZE, Matrixmul() : int, main() : int, matrixmul_benchmark(float[][], float[][], float

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

SDK Log

```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

28°C Haze | Search | ENG IN | 19:37 12-04-2025

Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
// Matrix Multiplication on PL using AXI MM
        XTime_SetTime(0);
        XTime_GetTime(&time_PL_start); // Capture the timer value at the start
        XMatrixmul_4_Set_Matrix_In(&AxiMM, (u32)DMA_input);
        XMatrixmul_4_Set_Matrix_C_HW(&AxiMM, (u32)output_C_HW2);
        XMatrixmul_4_Start(&AxiMM);
        while(XMatrixmul_4_IsDone(&AxiMM) == 0);
        XTime_GetTime(&time_PL_end); // Capture the timer value at the end
        curr_time = ((float)1.0 * (time_PL_end - time_PL_start)) / (COUNTS_PER_SECOND / 1000000);
        time_AXIMM = time_AXIMM + curr_time;
        printf("Execution Time for Pipelined + AP + Memory Mapped MMUL in PL in Micro-Seconds for %d iteration:


        // Compare benchmark and hardware functions output
        // Receive stream output C from hardware function
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
                float diff = abs(output_C_HW2[row][col] - output_C_SW[row][col]);
                if(diff > 0.001)
                {
                    printf("Pipelined + AP + Memory Mapped MMUL Error at row index %d and column index %d\t\n",
                    printf("Hardware output %f\n", output_C_HW2[row][col]);
                    printf("Software output %f\n\n", output_C_SW[row][col]);
                    break;
                }
            }
        }
    }

    printf("Avg Execution Time for PS in Micro-Seconds : %f\t\n" , time_processor/EXPT);
```

Outline: stdio.h, stdlib.h, xaxidma.h, xparameters.h, platform.h, xtime_l.h, D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\s, EXPT, MATSIZE, Matrixmul() : int, main() : int, matrixmul_benchmark(float[][], float[][], floa

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

SDK Log

```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

28°C Haze | Search | ENG IN | 19:37 12-04-2025

Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
        // Receive stream output C from hardware function
        index = 0;
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
                output_C_HW1[row][col] = DMA_output1[index];
                index = index + 1;
            }
        }
        // Compare benchmark and hardware functions output
        // Receive stream output C from hardware function
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
                float diff = abs(output_C_HW1[row][col] - output_C_SW[row][col]);
                if(diff > 0.001)
                {
                    printf("Pipelined + AP MMUL Error at row index %d and column index %d\t\n", row, col);
                    printf("Hardware output %f\n", output_C_HW1[row][col]);
                    printf("Software output %f\n\n", output_C_SW[row][col]);
                    break;
                }
            }
        }


// Matrix Multiplication on PL using AXI MM
        XTime_SetTime(0);
        XTime_GetTime(&time_PL_start); // Capture the timer value at the start
        XMatrixmul_4_Set_Matrix_In(&AxiMM, (u32)DMA_input);
        XMatrixmul_4_Set_Matrix_C_HW(&AxiMM, (u32)output_C_HW2);
```

Outline
- stdio.h
- stdlib.h
- xaxidma.h
- xparameters.h
- platform.h
- xtime_l.h
- D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\s
- EXPT
- MATSIZE
- Matrixmul() : int
- main() : int
- matrixmul_benchmark(float[][], float[][], float

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

SDK Log

```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

28°C Haze | Search | ENG IN | 19:37 12-04-2025

Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Window Help

Project Explorer
- design_1_wrapper_hw_platform_0
- Lab8_MM_project
- Lab8_MM_project_bsp

system.hdf | system.mss | helloworld.c | _exit.c | Disassembly

```c
        }

        // Matrix Multiplication on PL using DMA 1
        XTime_SetTime(0);
        XTime_GetTime(&time_PL_start); // Capture the timer value at the start

        status = XAxiDma_SimpleTransfer(&AxiDMA1, (UINTPTR)DMA_output1,(sizeof(float)*MATSIZE*MATSIZE), XAXIDMA
        status = XAxiDma_SimpleTransfer(&AxiDMA1, (UINTPTR)DMA_input,(sizeof(float)*MATSIZE*MATSIZE*2), XAXIDMA
        // We have only configured the DMA to perform these two transactions.. DMA might not have started the t
        status = XAxiDma_ReadReg(XPAR_AXI_DMA_1_BASEADDR,0x04) & 0x00000002;
        while(status!=0x00000002)
        {
          status = XAxiDma_ReadReg(XPAR_AXI_DMA_1_BASEADDR,0x04) & 0x00000002;
        }
        status = XAxiDma_ReadReg(XPAR_AXI_DMA_1_BASEADDR,0x34) & 0x00000002;

       while(status!=0x00000002)
       {
         status = XAxiDma_ReadReg(XPAR_AXI_DMA_1_BASEADDR,0x34) & 0x00000002;
       }

        XTime_GetTime(&time_PL_end); // Capture the timer value at the end
        curr_time = ((float)1.0 * (time_PL_end - time_PL_start) / (COUNTS_PER_SECOND / 1000000));
        time_FPGA1 = time_FPGA1 + curr_time;

        printf("Execution Time for Pipelined + AP MMUL in PL in Micro-Seconds for %d iteration: %f\n", k, curr_

        // Receive stream output C from hardware function
        index = 0;
        for(row = 0; row < MATSIZE; row++)
        {
            for(col = 0; col < MATSIZE; col++)
            {
```

Outline
- stdio.h
- stdlib.h
- xaxidma.h
- xparameters.h
- platform.h
- xtime_l.h
- D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\
- EXPT
- MATSIZE
- Matrixmul() : int
- main() : int
- matrixmul_benchmark(float[][], float[][], flo

Target Connections
- Hardware Server
  - cloudlabs [default]
  - Local
- Linux TCF Agent
- QEMU TcfGdbClient

Problems | Tasks | Console | Properties | SDK Terminal

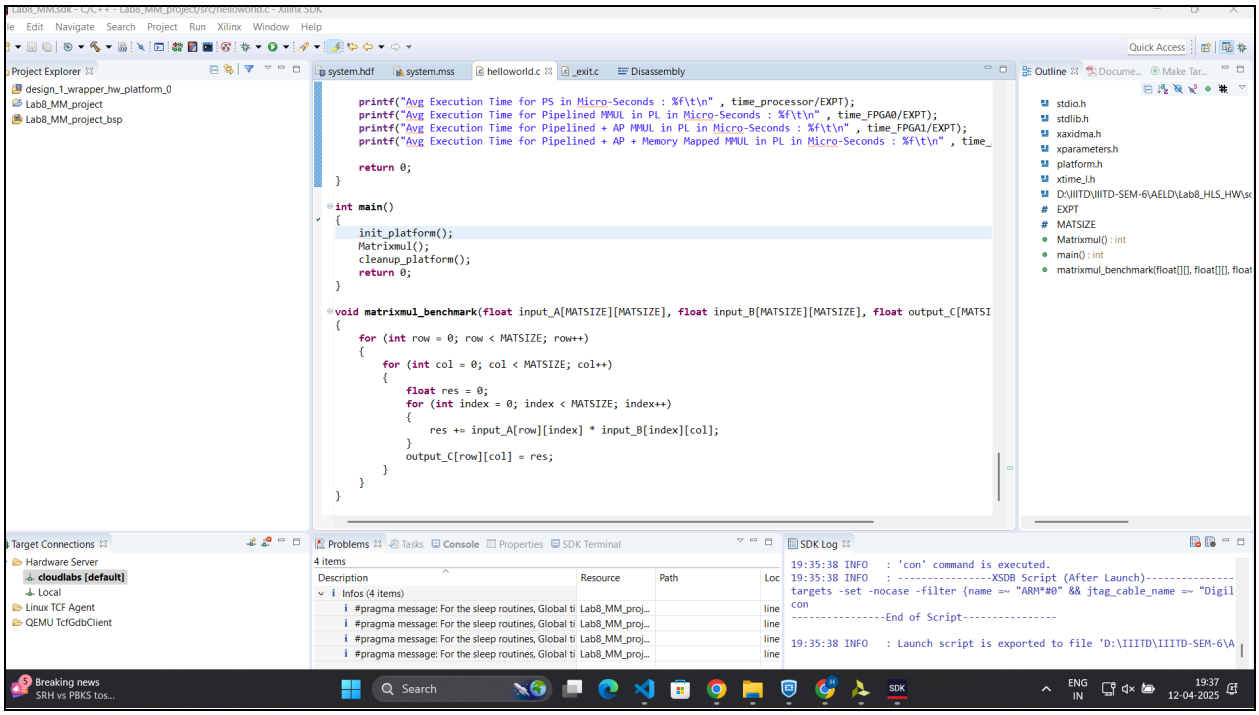
4 items

Description	Resource	Path	Loc
Infos (4 items)			
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line
#pragma message: For the sleep routines, Global ti	Lab8_MM_proj...		line

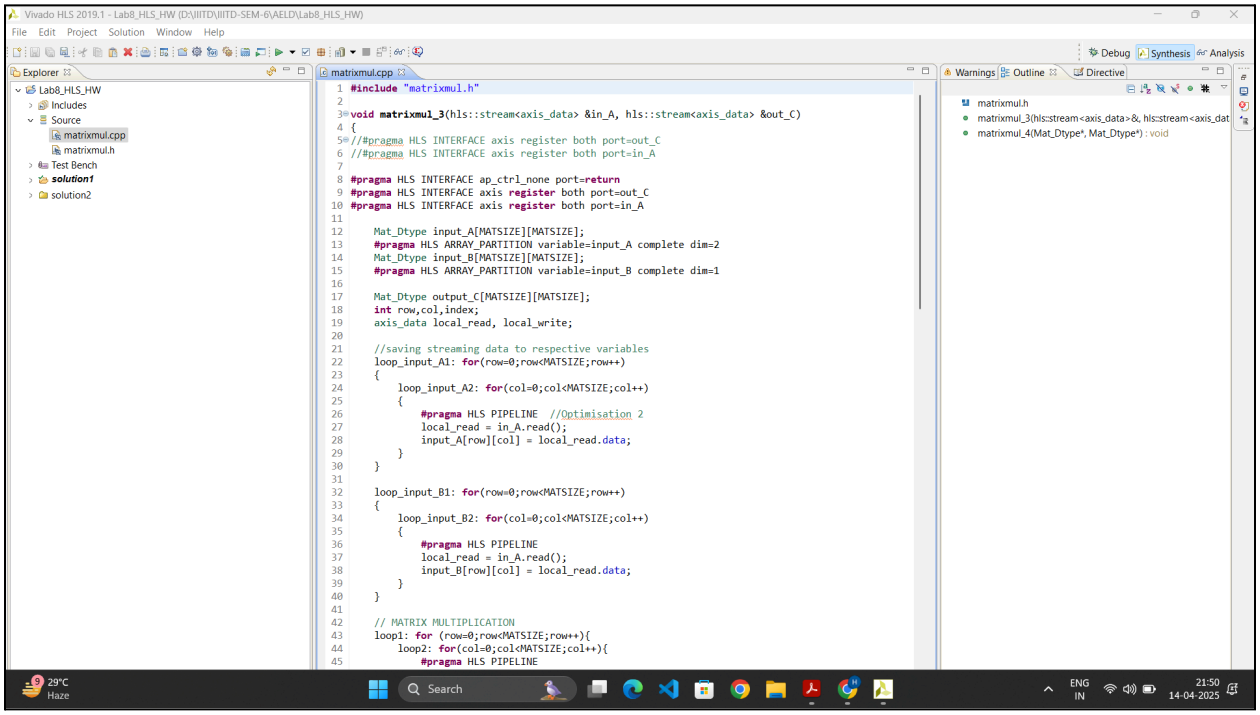
SDK Log

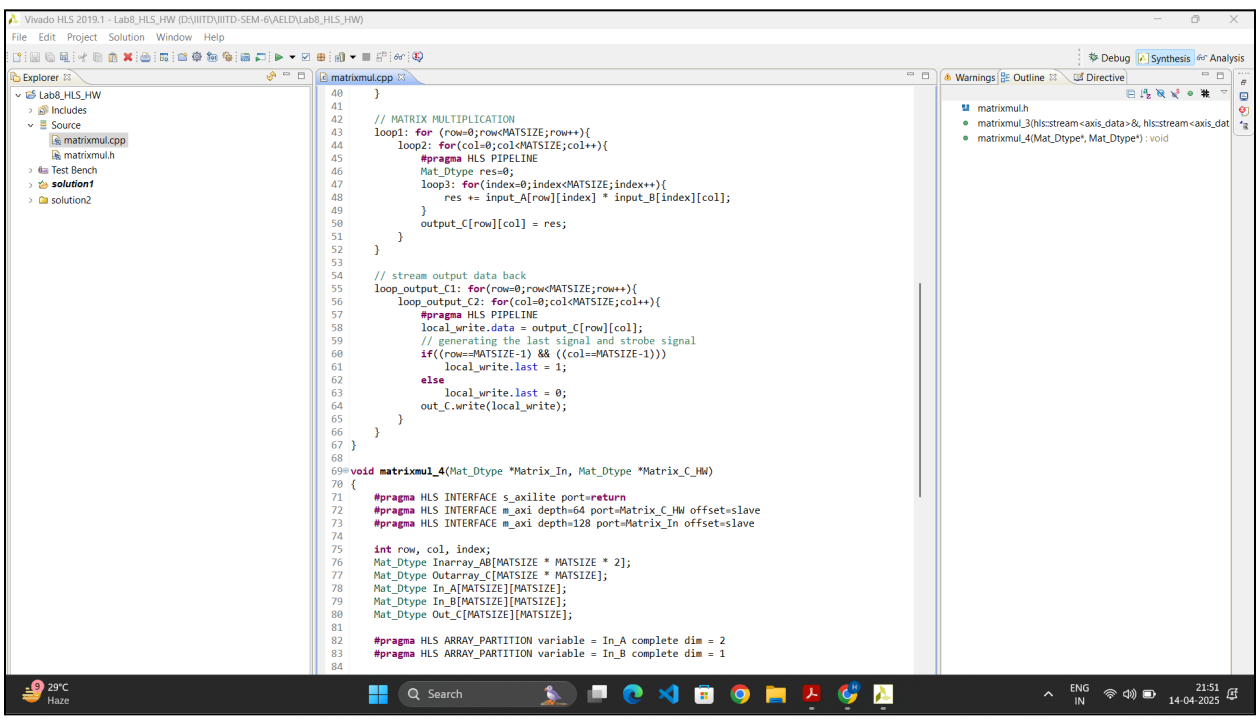
```
19:35:38 INFO   : 'con' command is executed.
19:35:38 INFO   : ----------------XSDB Script (After Launch)----------------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------------End of Script----------------

19:35:38 INFO   : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
```

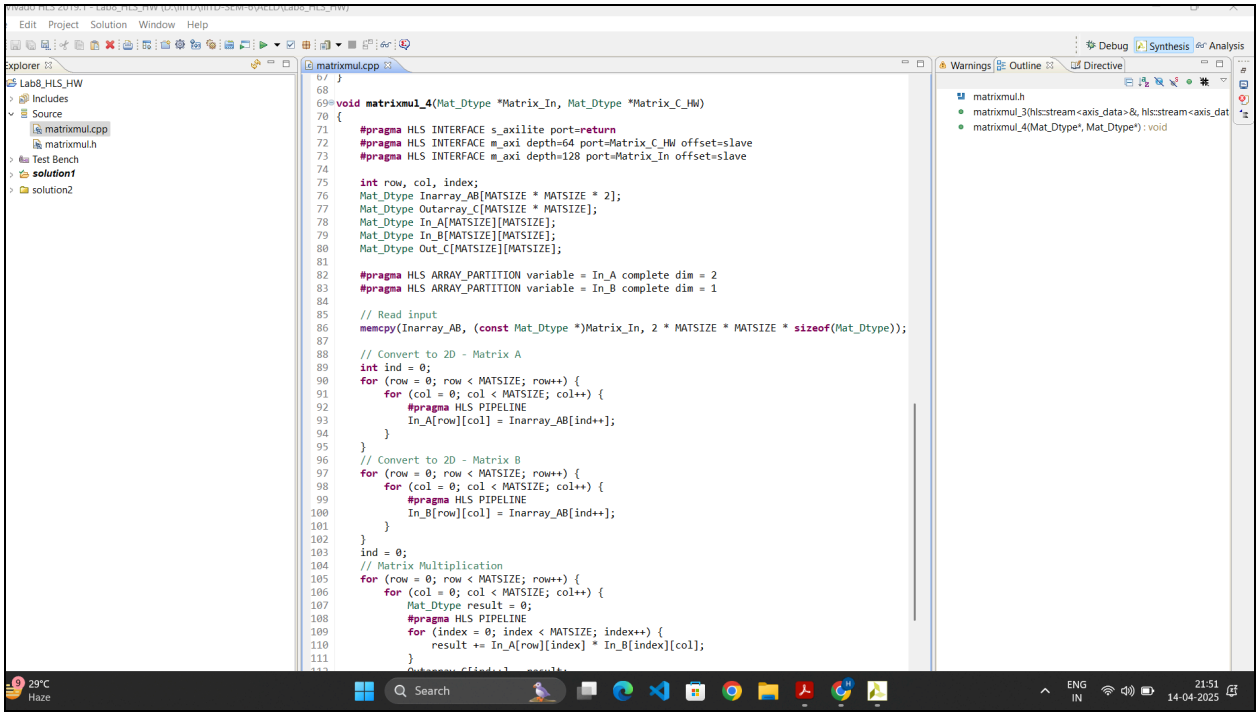
28°C Haze | Search | ENG IN | 19:37 12-04-2025


Lab8_MM.sdk - C/C++ - Lab8_MM_project/src/helloworld.c - Xilinx SDK
Edit Navigate Search Project Run Xilinx Window Help
Quick Access
Project Explorer
design_1_wrapper_hw_platform_0
Lab8_MM_project
Lab8_MM_project_bsp
system.hdf
system.mss
helloworld.c
_exit.c
Disassembly
printf("Avg Execution Time for PS in Micro-Seconds : %f\t\n" , time_processor/EXPT);
printf("Avg Execution Time for Pipelined MMUL in PL in Micro-Seconds : %f\t\n" , time_FPGA0/EXPT);
printf("Avg Execution Time for Pipelined + AP MMUL in PL in Micro-Seconds : %f\t\n" , time_FPGA1/EXPT);
printf("Avg Execution Time for Pipelined + AP + Memory Mapped MMUL in PL in Micro-Seconds : %f\t\n" , time_
return 0;
}
int main()
{
init_platform();
Matrixmul();
cleanup_platform();
return 0;
}
void matrixmul_benchmark(float input_A[MATSIZE][MATSIZE], float input_B[MATSIZE][MATSIZE], float output_C[MATSI
{
for (int row = 0; row < MATSIZE; row++)
{
for (int col = 0; col < MATSIZE; col++)
{
float res = 0;
for (int index = 0; index < MATSIZE; index++)
{
res += input_A[row][index] * input_B[index][col];
}
output_C[row][col] = res;
}
}
}
Outline
Docume...
Make Tar...
stdio.h
stdlib.h
xaxidma.h
xparameters.h
platform.h
xtime_l.h
D:\IIITD\IIITD-SEM-6\AELD\Lab8_HLS_HW\sc
EXPT
MATSIZE
Matrixmul() : int
main() : int
matrixmul_benchmark(float[][], float[][], float
Target Connections
Hardware Server
cloudlabs [default]
Local
Linux TCF Agent
QEMU TcfGdbClient
Problems
Tasks
Console
Properties
SDK Terminal
4 items
Description
Resource
Path
Loc
Infos (4 items)
#pragma message: For the sleep routines, Global ti
Lab8_MM_proj...
line
SDK Log
19:35:38 INFO : 'con' command is executed.
19:35:38 INFO : ----------XSDB Script (After Launch)----------
targets -set -nocase -filter {name =~ "ARM*#0" && jtag_cable_name =~ "Digil
con
----------End of Script----------
19:35:38 INFO : Launch script is exported to file 'D:\IIITD\IIITD-SEM-6\A
Breaking news
SRH vs PBKS tos...
Search
SDK
ENG
IN
19:37
12-04-2025

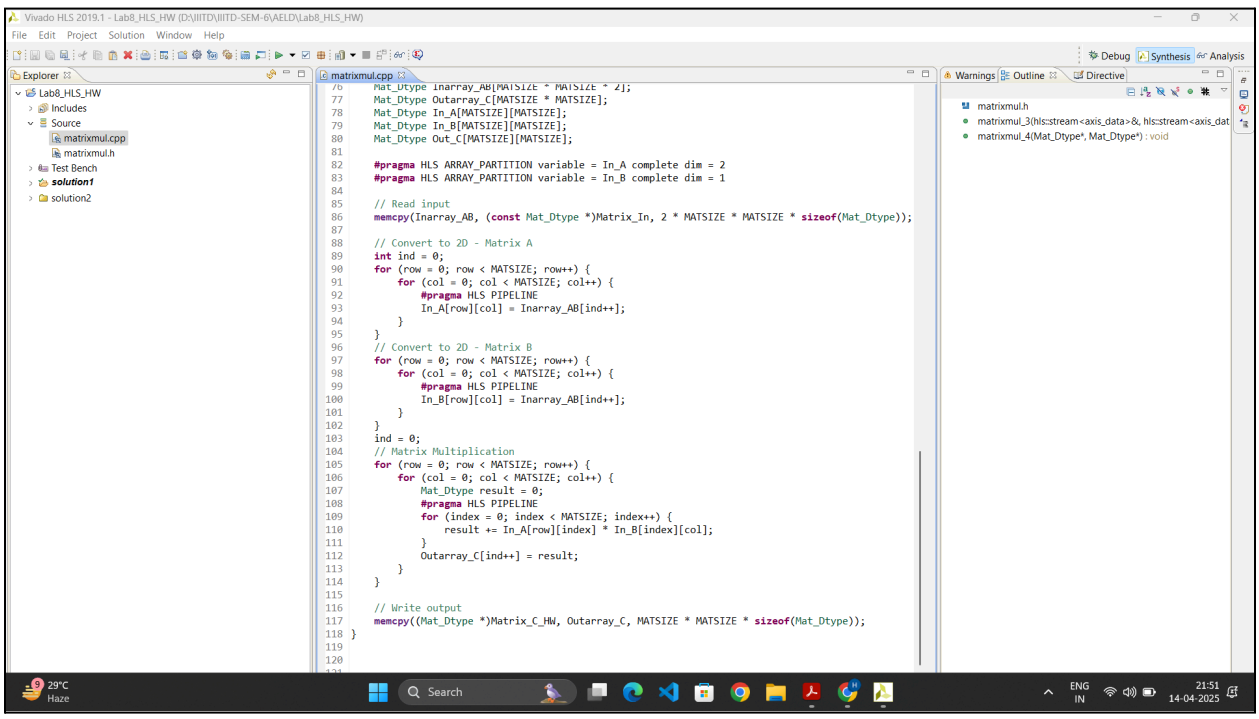
Vivado HLS (high-level synthesis) C code








```
}

void matrixmul_4(Mat_Dtype *Matrix_In, Mat_Dtype *Matrix_C_HW)
{
    #pragma HLS INTERFACE s_axilite port=return
    #pragma HLS INTERFACE m_axi depth=64 port=Matrix_C_HW offset=slave
    #pragma HLS INTERFACE m_axi depth=128 port=Matrix_In offset=slave

    int row, col, index;
    Mat_Dtype Inarray_AB[MATSIZE * MATSIZE * 2];
    Mat_Dtype Outarray_C[MATSIZE * MATSIZE];
    Mat_Dtype In_A[MATSIZE][MATSIZE];
    Mat_Dtype In_B[MATSIZE][MATSIZE];
    Mat_Dtype Out_C[MATSIZE][MATSIZE];

    #pragma HLS ARRAY_PARTITION variable = In_A complete dim = 2
    #pragma HLS ARRAY_PARTITION variable = In_B complete dim = 1

    // Read input
    memcpy(Inarray_AB, (const Mat_Dtype *)Matrix_In, 2 * MATSIZE * MATSIZE * sizeof(Mat_Dtype));

    // Convert to 2D - Matrix A
    int ind = 0;
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            #pragma HLS PIPELINE
            In_A[row][col] = Inarray_AB[ind++];
        }
    }
    // Convert to 2D - Matrix B
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            #pragma HLS PIPELINE
            In_B[row][col] = Inarray_AB[ind++];
        }
    }
    ind = 0;
    // Matrix Multiplication
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            Mat_Dtype result = 0;
            #pragma HLS PIPELINE
            for (index = 0; index < MATSIZE; index++) {
                result += In_A[row][index] * In_B[index][col];
            }
```




```
    Mat_Dtype Outarray_C[MATSIZE * MATSIZE];
    Mat_Dtype In_A[MATSIZE][MATSIZE];
    Mat_Dtype In_B[MATSIZE][MATSIZE];
    Mat_Dtype Out_C[MATSIZE][MATSIZE];

    #pragma HLS ARRAY_PARTITION variable = In_A complete dim = 2
    #pragma HLS ARRAY_PARTITION variable = In_B complete dim = 1

    // Read input
    memcpy(Inarray_AB, (const Mat_Dtype *)Matrix_In, 2 * MATSIZE * MATSIZE * sizeof(Mat_Dtype));

    // Convert to 2D - Matrix A
    int ind = 0;
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            #pragma HLS PIPELINE
            In_A[row][col] = Inarray_AB[ind++];
        }
    }
    // Convert to 2D - Matrix B
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            #pragma HLS PIPELINE
            In_B[row][col] = Inarray_AB[ind++];
        }
    }
    ind = 0;
    // Matrix Multiplication
    for (row = 0; row < MATSIZE; row++) {
        for (col = 0; col < MATSIZE; col++) {
            Mat_Dtype result = 0;
            #pragma HLS PIPELINE
            for (index = 0; index < MATSIZE; index++) {
                result += In_A[row][index] * In_B[index][col];
            }
            Outarray_C[ind++] = result;
        }
    }

    // Write output
    memcpy((Mat_Dtype *)Matrix_C_HW, Outarray_C, MATSIZE * MATSIZE * sizeof(Mat_Dtype));
}
```

Output [On JTAGTERMINAL]