Main Memory Characterizer

for PynqZ2

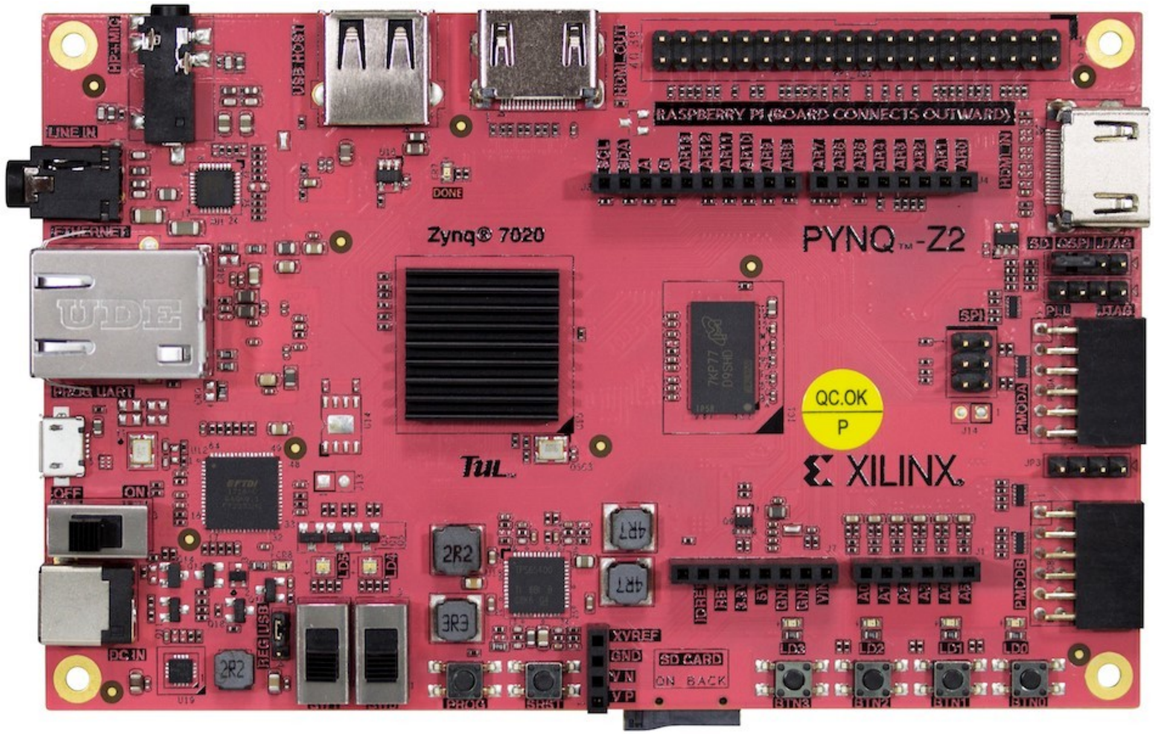
Valentino Guerrini

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

Pynq Z2 What??

-Dual-core ARM Cortex-A9 processor
-Xilinx Zynq-7020 SoC device
-2GB DDR3 memory1
-6MB Quad-SPI flash memory
-10/100/1000 Ethernet port
-USB 2.0 ports
-HDMI input and output
-40-pin expansion header
-Micro SD card slot
-On-board power and user buttons



POLITECNICO MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

Goals:

- Analyze the performance characteristics of DDR memory modules of the PYNQ-Z2 board
- Read Performance
- Write performance

Yes but...HOW??

- **Designing** the **IP block** using Vitis HLS that compute the bandwidth performance of the main memory.
- **Integration** of the IP block into the overall block design using Vivado
- **Developing** the **Host Code** that communicate with the IP block

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
void  ddrbenchmark2(TYPE*  input,  TYPE*  output,  const  int  modo)
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
void  ddrbenchmark2(TYPE*  input,  TYPE*  output,  const  int  modo)
```

```
#pragma HLS INTERFACE m_axi port = input depth=arr_depth offset =slave bundle = gmem
#pragma HLS INTERFACE m_axi port = output depth=arr_depth offset =slave bundle = gmem
#pragma HLS INTERFACE s_axilite port = modo bundle=control
#pragma HLS INTERFACE s_axilite port = input bundle=control
#pragma HLS INTERFACE s_axilite port = output bundle=control
#pragma HLS INTERFACE s_axilite port = return bundle=control
```

POLITECNICO
MILANO 1863

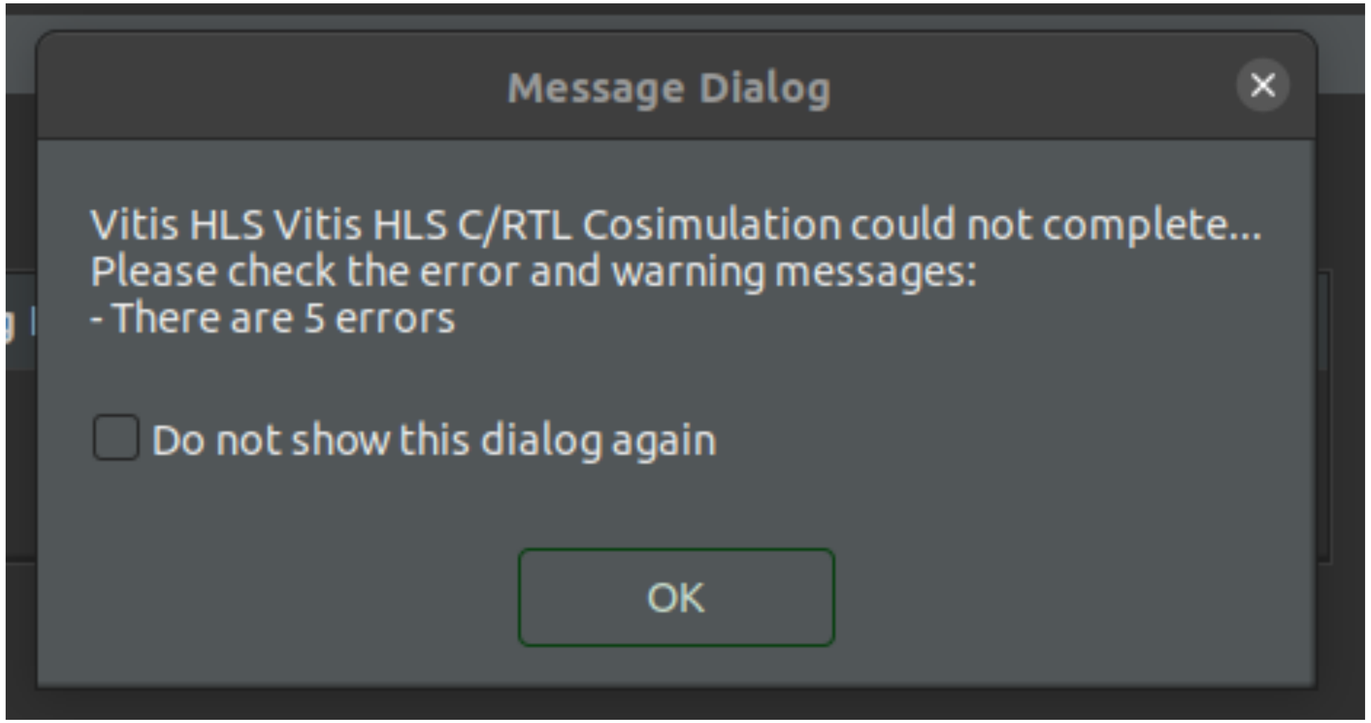
POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
void  ddrbenchmark2(TYPE*  input,  TYPE*  output,  const  int  modo)

#pragma HLS INTERFACE m_axi port = input depth=arr_depth offset =slave bundle = gmem
#pragma HLS INTERFACE m_axi port = output depth=arr_depth offset =slave bundle = gmem
#pragma HLS INTERFACE s_axilite port = modo bundle=control
#pragma HLS INTERFACE s_axilite port = input bundle=control
#pragma HLS INTERFACE s_axilite port = output bundle=control
#pragma HLS INTERFACE s_axilite port = return bundle=control
```

IP block:


Message Dialog
Vitis HLS Vitis HLS C/RTL Cosimulation could not complete...
Please check the error and warning messages:
- There are 5 errors
Do not show this dialog again
OK

IP block:

```
ERROR: System recieved a signal named SIGSEGV and the program has to stop immediately!
This signal was generated when a program tries to read or write outside the memory that is allocated for it, or to write memory that can only be read.
Possible cause of this problem may be: 1) the depth setting of pointer type argument is much larger than it needed; 2)insufficient depth of array argument; 3)null pointer etc.
Current execution stopped during CodeState = CALL_C_DUT.
You can search CodeState variable name in apatb*.cpp file under ./sim/wrapc dir to locate the position.

ERROR: [COSIM 212-360] Aborting co-simulation: C TB simulation failed.
ERROR: [COSIM 212-320] C TB testing failed, stop generating test vectors. Please check C TB or re-run cosim.
ERROR: [COSIM 212-5] *** C/RTL co-simulation file generation failed. ***
ERROR: [COSIM 212-4] *** C/RTL co-simulation finished: FAIL ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 8.55 seconds. CPU system time: 1.01 seconds. Elapsed time: 8.62 seconds; current allocated memory: -1023.719 MB.
command 'ap_source' returned error code
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
ERROR: System recieved a signal named SIGSEGV and the program has to stop immediately!
This signal was generated when a program tries to read or write outside the memory that is allocated for it, or to write memory that can only be read.
Possible cause of this problem may be: 1) the depth setting of pointer type argument is much larger than it needed; 2)insufficient depth of array argument; 3)null pointer etc.
Current execution stopped during CodeState = CALL_C_DUT.
You can search CodeState variable name in apatb*.cpp file under ./sim/wrapc dir to locate the position.

ERROR: [COSIM 212-360] Aborting co-simulation: C TB simulation failed.
ERROR: [COSIM 212-320] C TB testing failed, stop generating test vectors. Please check C TB or re-run cosim.
ERROR: [COSIM 212-5] *** C/RTL co-simulation file generation failed. ***
ERROR: [COSIM 212-4] *** C/RTL co-simulation finished: FAIL ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 8.55 seconds. CPU system time: 1.01 seconds. Elapsed time: 8.62 seconds; current allocated memory: -1023.719 MB.
command 'ap_source' returned error code
```


```
ERROR: System recieved a signal named SIGSEGV and the program has to stop immediately!
This signal was generated when a program tries to read or write outside the memory that is allocated for it, or to write memory that can only be read.
Possible cause of this problem may be: 1) the depth setting of pointer type argument is much larger than it needed; 2)insufficient depth of array argument; 3)null pointer etc.
Current execution stopped during CodeState = CALL_C_DUT.
You can search CodeState variable name in apatb*.cpp file under ./sim/wrapc dir to locate the position.

ERROR: [COSIM 212-360] Aborting co-simulation: C TB simulation failed.
ERROR: [COSIM 212-320] C TB testing failed, stop generating test vectors. Please check C TB or re-run cosim.
ERROR: [COSIM 212-5] *** C/RTL co-simulation file generation failed. ***
ERROR: [COSIM 212-4] *** C/RTL co-simulation finished: FAIL ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 8.55 seconds. CPU system time: 1.01 seconds. Elapsed time: 8.62 seconds; current allocated memory: -1023.719 MB.
command 'ap_source' returned error code
```


POLITECNICO
MILANO 1863

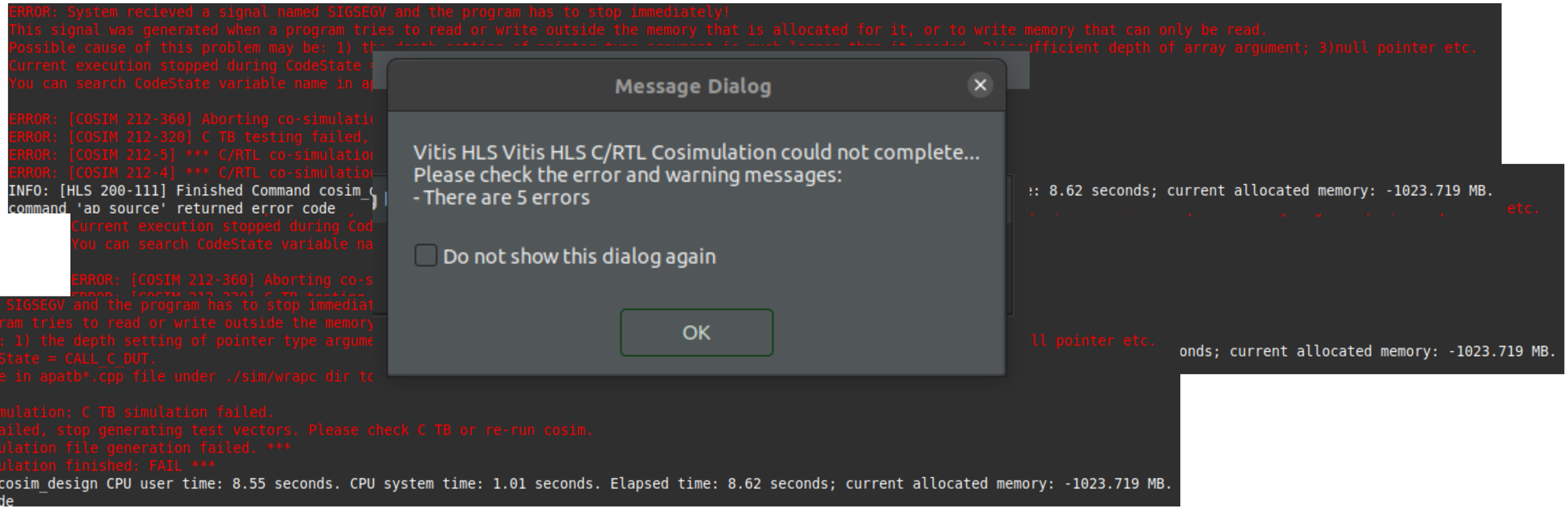
POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
ERROR: System recieved a signal named SIGSEGV and the program has to stop immediately!
This signal was generated when a program tries to read or write outside the memory that is allocated for it, or to write memory that can only be read.
Possible cause of this problem may be: 1) the depth setting of pointer type argument is much larger than it needed; 2)insufficient depth of array argument; 3)null pointer etc.
Current execution stopped during CodeState = CALL_C_DUT.
You can search CodeState variable name in apatb*.cpp file under ./sim/wrapc dir to locate the position.

ERROR: [COSIM 212-360] Aborting co-simulation: C TB simulation failed.
ERROR: [COSIM 212-320] C TB testing failed, stop generating test vectors. Please check C TB or re-run cosim.
ERROR: [COSIM 212-5] *** C/RTL co-simulation file generation failed. ***
ERROR: [COSIM 212-4] *** C/RTL co-simulation finished: FAIL ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 8.55 seconds. CPU system time: 1.01 seconds. Elapsed time: 8.62 seconds; current allocated memory: -1023.719 MB.
command 'ap source' returned error code
```

IP block:



POLITECNICO MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
TYPE temp[ARRAY_SIZE];
TYPE write;
TYPE outData=123;
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

IP block:

```
TYPE temp[ARRAY_SIZE];
TYPE write;
TYPE outData=123;
if(modo == 0){
    ReadWrite:for(int i=0; i < ARRAY_SIZE; i++){ //read&write
                        #pragma HLS PIPELINE
                        temp[i] = *(input+i);
                        *(output+i) =temp[i];
    }
}else if(modo == 1){
    read:for(int i=0; i < ARRAY_SIZE; i++){
                        #pragma HLS PIPELINE
                        write = *(input+i);//solo lettura
                        }
}else{
    write:for(int i=0; i < ARRAY_SIZE; i++){
                        #pragma HLS PIPELINE
                        *(output+i) = outData; //solo scrittura
                        }
}
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

TestBench:

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

TestBench:

```
TYPE ref[ARRAY_SIZE];
TYPE out[ARRAY_SIZE];
int seed = 3456;
std::default_random_engine rng(seed);
std::uniform_int_distribution<int> rng_dist(0, RANGE_UPPER_BOUND);
for(int i=0;i<ARRAY_SIZE;i++){
   ref[i]=static_cast<TYPE>(rng_dist(rng));
   out[i]=0;
}

...

#ifdef AVERAGE_REPS
     execMultiTest(ref,out);
#endif
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

TestBench:

```
for (int i = 0; i < REPETITIONS;i++){
        auto start = std::chrono::high_resolution_clock::now();
        ddrbenchmark2(ref, out,0);  //HW function
        auto end = std::chrono::high_resolution_clock::now();
        double durationb = std::chrono::duration_cast<std::chrono::microseconds>(end - start).count();
        testCheck0(ref,out, &resultb);
        tot_duration += durationb;
        std::cout << "RW result: " << i << "SUCCESS="<< resultb << std::endl;
        std::cout  << std::endl;
        std::cout << "RW time:" << durationb << std::endl;
        rep++;
        resultb = true;
        }
        std::cout << "simultaneous RW average:" << tot_duration/rep << std::endl;
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

HLS Synthesis:

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

HLS Synthesis:

INFO: [HLS 200-42] -- Implementing module 'ddrbenchmark2_Pipeline_ReadWrite'
INFO: [HLS 200-10] --
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-61] Pipelining loop 'ReadWrite'.
INFO: [HLS 200-1470] Pipelining result : Target II = NA, Final II = 1, Depth = 3, loop 'ReadWrite'
INFO: [SCHED 204-11] Finished scheduling.

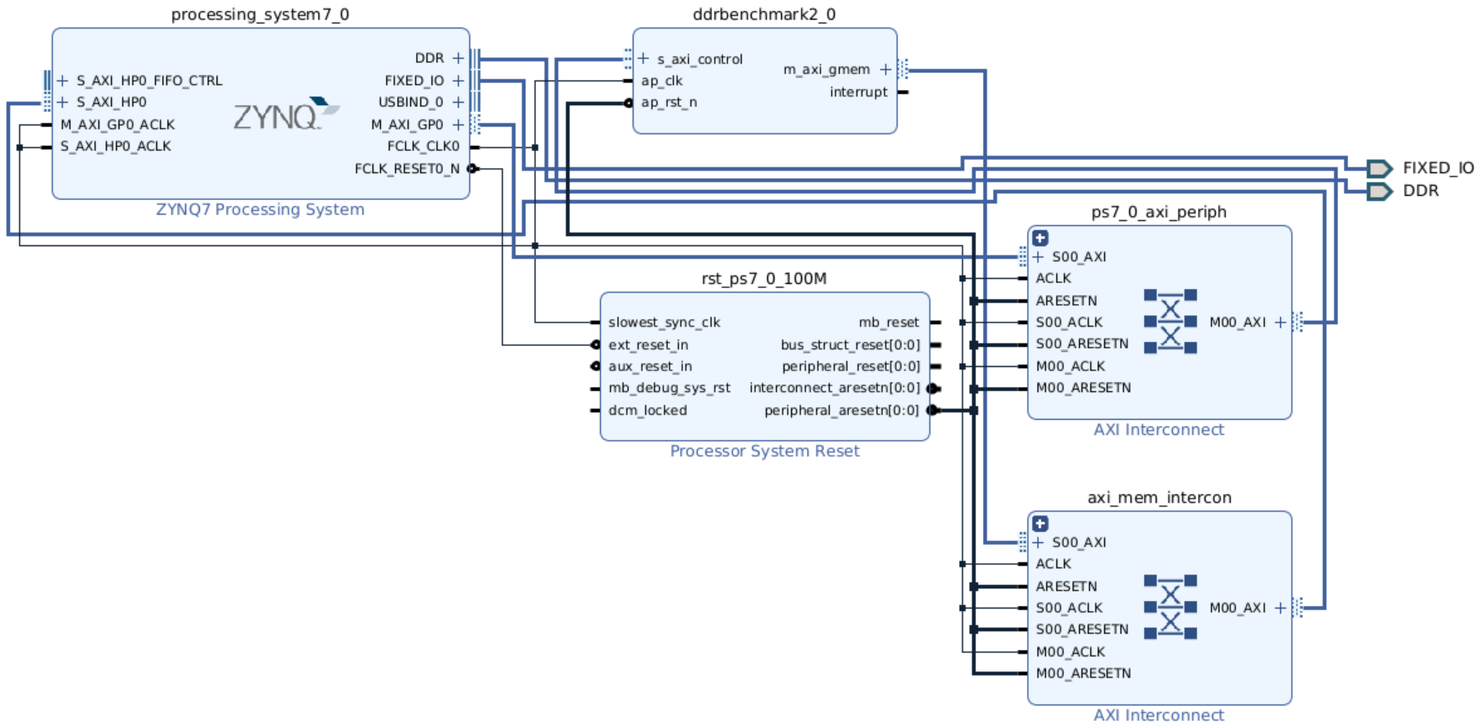
INFO: [HLS 200-42] -- Implementing module 'ddrbenchmark2_Pipeline_write'
INFO: [HLS 200-10] --
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-61] Pipelining loop 'write'.
INFO: [HLS 200-1470] Pipelining result : Target II = NA, Final II = 1, Depth = 2, loop 'write'
INFO: [SCHED 204-11] Finished scheduling.

Final Block Design and Bitstream!!

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

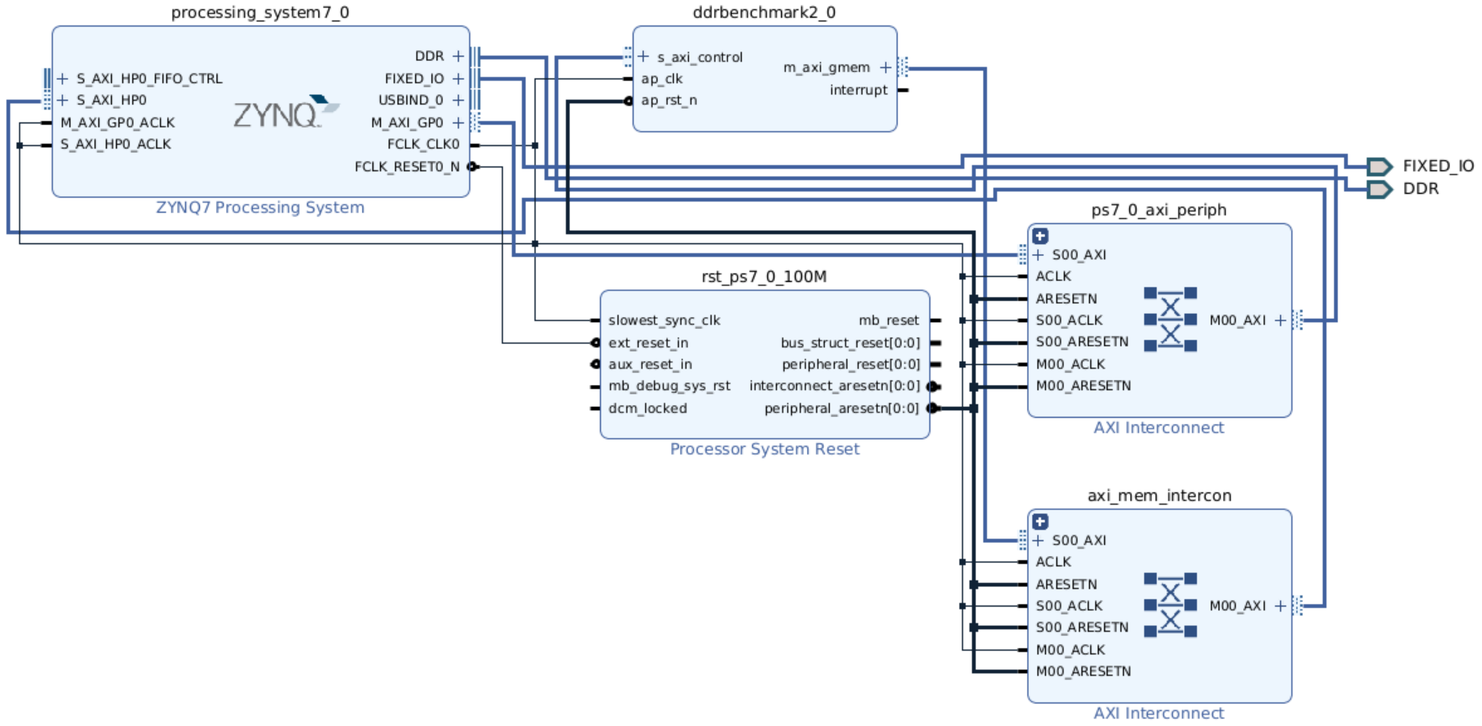
Final Block Design and Bitstream!!

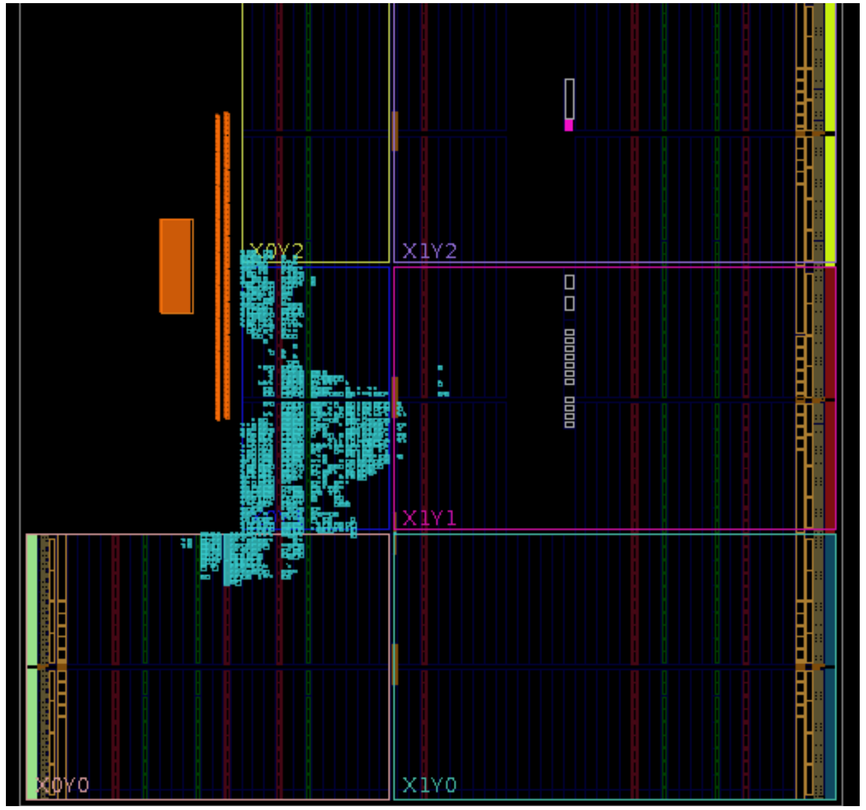


POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

Final Block Design and Bitstream!!





POLITECNICO MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

System Host Code:

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

System Host Code:

```
TEST_NUM = 100
DATA_SIZE = 16000
MB_CONV = 1000000
MICRO_CONV = 1000000

#bitstream upload
overlay = Overlay("./ddrbench_golden_wrapper.xsa")

ddrbench_ip =  overlay.ddrbenchmark2_0

buf_in = allocate(DATA_SIZE,np.int64)
buf_out = allocate(DATA_SIZE,np.int64)
multipleTests(buf_in,buf_out,TEST_NUM).
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

System Host Code:

```
def multipleTests(buffer_in,buffer_out,repetition):
    sum =0.0
    rep=0

    for i in range(repetition):
        input=np.random.randint(low=0,high=8192,size=(DATA_SIZE),dtype=np.int64)

        buffer_in[:]=input.ravel()[:]
        buffer_out[:]=0
        ddrbench_ip.write(0x10,buffer_in.physical_address)
        ddrbench_ip.write(0x1c,buffer_out.physical_address)
        ddrbench_ip.write(0x28,0)
        start = time.time();
        ddrbench_ip.write(0x00,1)
        while(ddrbench_ip.read(0x00) & 0x04 !=0x04):pass
        end = time.time()
        buf_out.invalidate()
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

System Host Code:

```
if(np.all(input == buffer_out)): //check
        if(i==0 or (end-start)*MICRO_CONV<best):best=(end-start)*MICRO_CONV;
        if(i==0 or (end-start)*MICRO_CONV>worst):worst=(end-start)*MICRO_CONV;
        print("test ",i,": ","time:",(end-start)*MICRO_CONV,"µs","  Bandwidth:",(DATA_SIZE*64/8)/((end-start)*MB_CONV),"MB/s")
        sum = sum + end-start
        rep = rep+1

    print(" ")
    print("Executed",rep,"tests","Average_time:", sum/rep*MICRO_CONV,"µs")
    print("Best_time:", best,"µs","  Worst_time:",worst,"µs" ,"  Avg_Bandwidth: ",(DATA_SIZE*64/8)/((sum/rep)*MB_CONV),"MB/s")
    print(" ")
```

POLITECNICO
MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

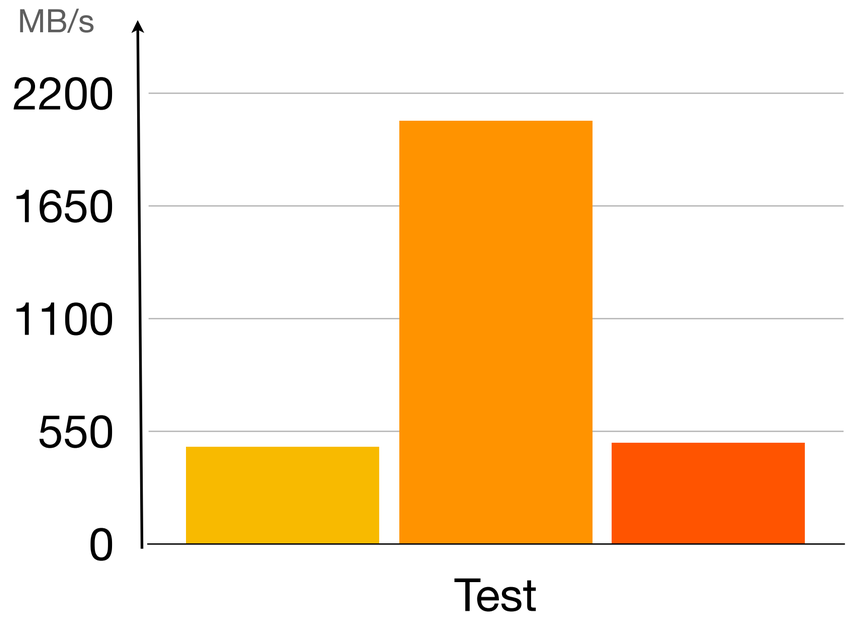
Experimental Result

.....
test 96 : time: 270.843505859375 µs   Bandwidth: 472.5976338028169 MB/s
test 97 : time: 259.8762512207031 µs   Bandwidth: 492.5421211009174 MB/s
test 98 : time: 267.26722717285156 µs   Bandwidth: 478.9214201605709 MB/s
test 99 : time: 273.2276916503906 µs   Bandwidth: 468.4737452006981 MB/s

Executed 100 tests Average_time: 271.37041091918945 µs
Best_time: 259.63783264160156 µs   Worst_time: 355.00526428222656 µs   Avg_Bandwidth: 471.6800168685919 MB/s
.....
test 96 : time: 60.79673767089844 µs   Bandwidth: 2105.376125490196 MB/s
test 97 : time: 61.27357482910156 µs   Bandwidth: 2088.991875486381 MB/s
test 98 : time: 61.98883056640625 µs   Bandwidth: 2064.888123076923 MB/s
test 99 : time: 61.511993408203125 µs   Bandwidth: 2080.895007751938 MB/s

Executed 100 tests Average_time: 62.050819396972656 µs
Best_time: 59.36622619628906 µs   Worst_time: 72.479248046875 µs   Avg_Bandwidth: 2062.825297779144 MB/s
.....
test 96 : time: 257.9689025878906 µs   Bandwidth: 496.1838373382625 MB/s
test 97 : time: 257.73048400878906 µs   Bandwidth: 496.642841813136 MB/s
test 98 : time: 255.82313537597656 µs   Bandwidth: 500.3456775396086 MB/s
test 99 : time: 257.25364685058594 µs   Bandwidth: 497.5634031510658 MB/s

Executed 100 tests Average_time: 263.48352432250977 µs
Best_time: 254.8694610595703 µs   Worst_time: 295.1622009277344 µs   Avg_Bandwidth: 485.79887615031714 MB/s



POLITECNICO MILANO 1863

POLITECNICO MILANO 1863
NECST laboratory

THE END