INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

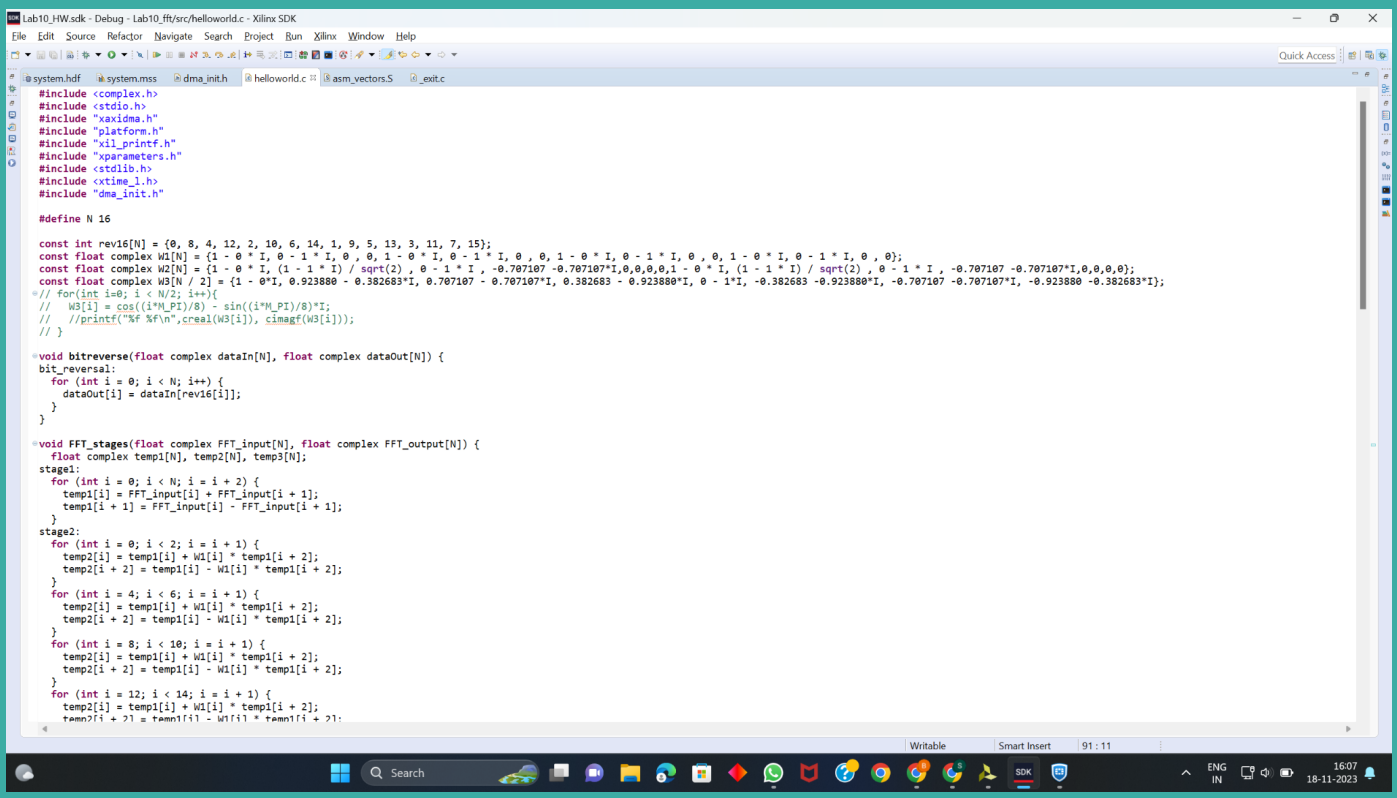
Department
of
Electronics & Communication Engineering

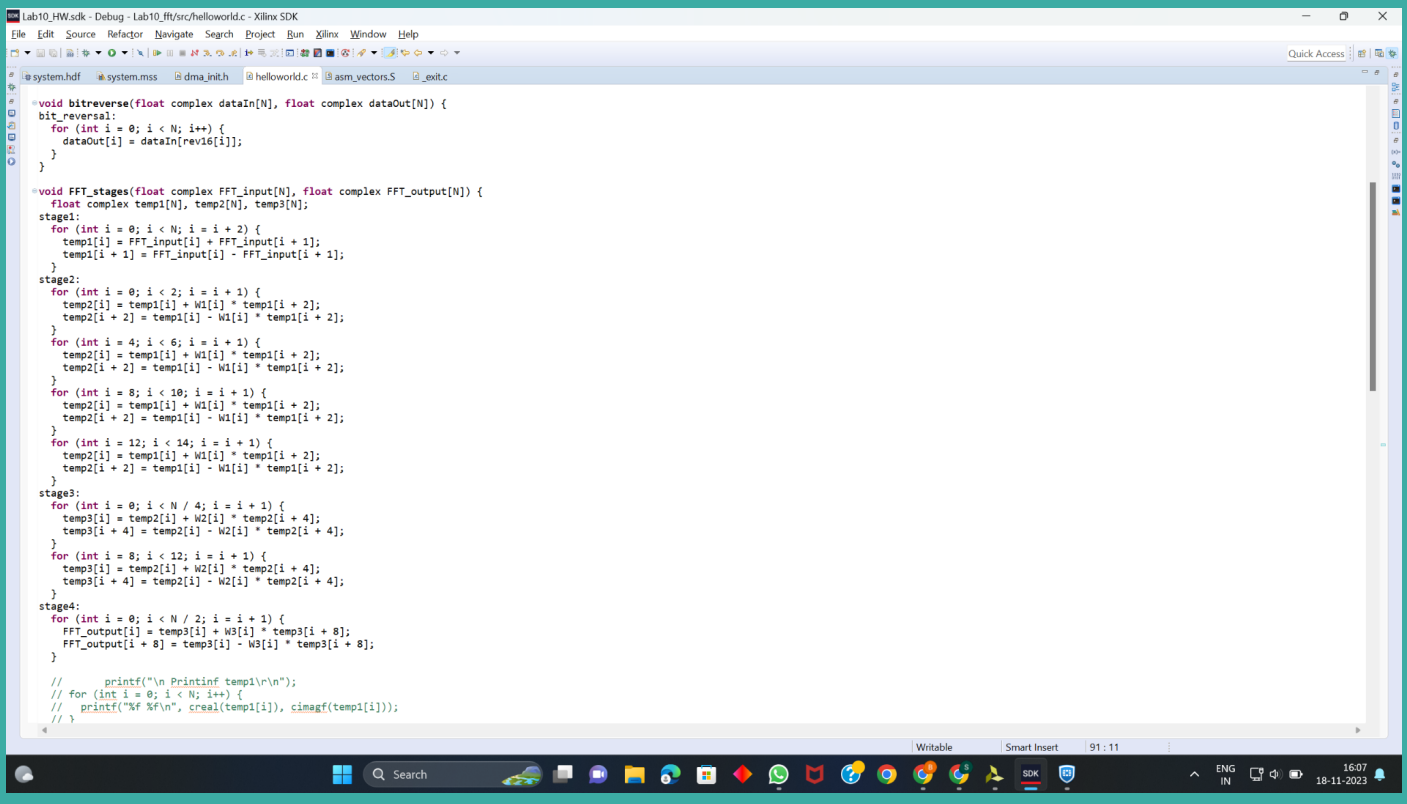
Embedded Logic Design
Lab 10 Submission

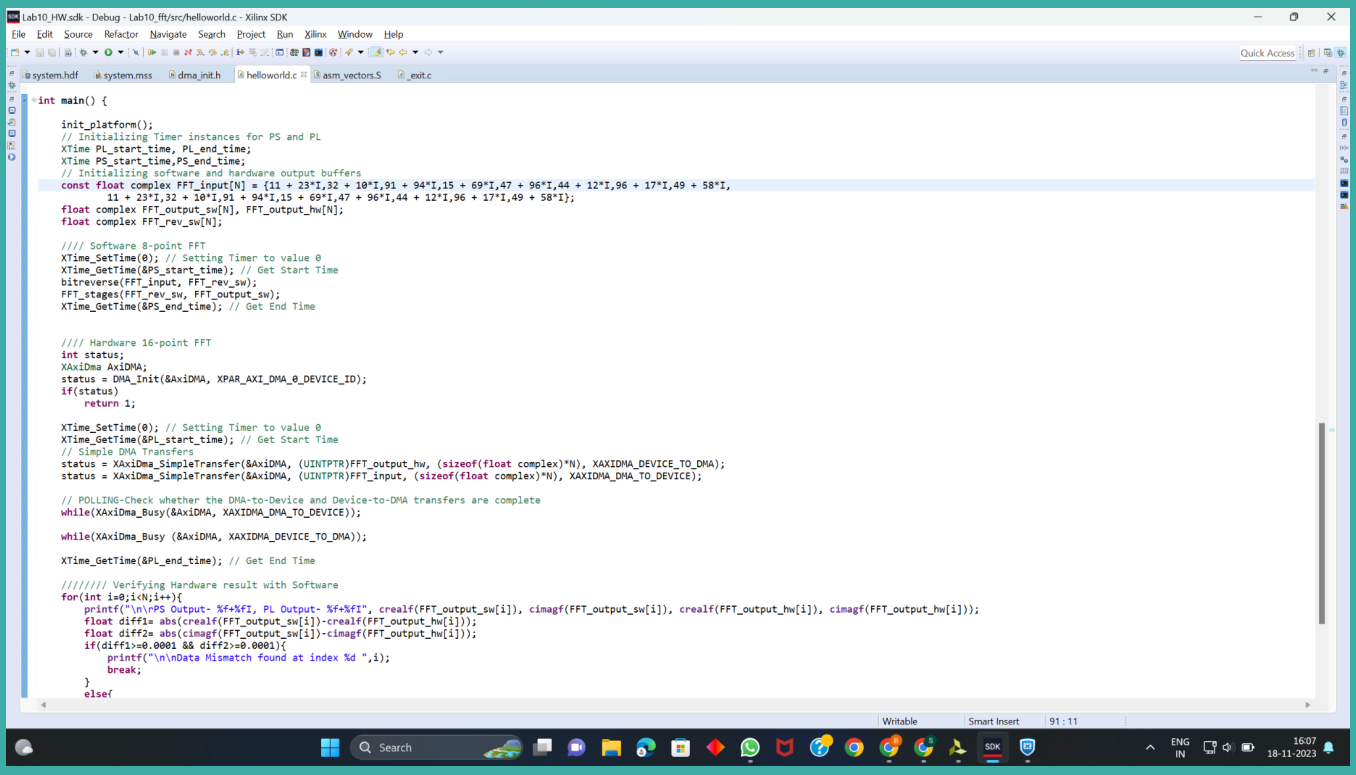
SHIVAM SHUKLA
2022478

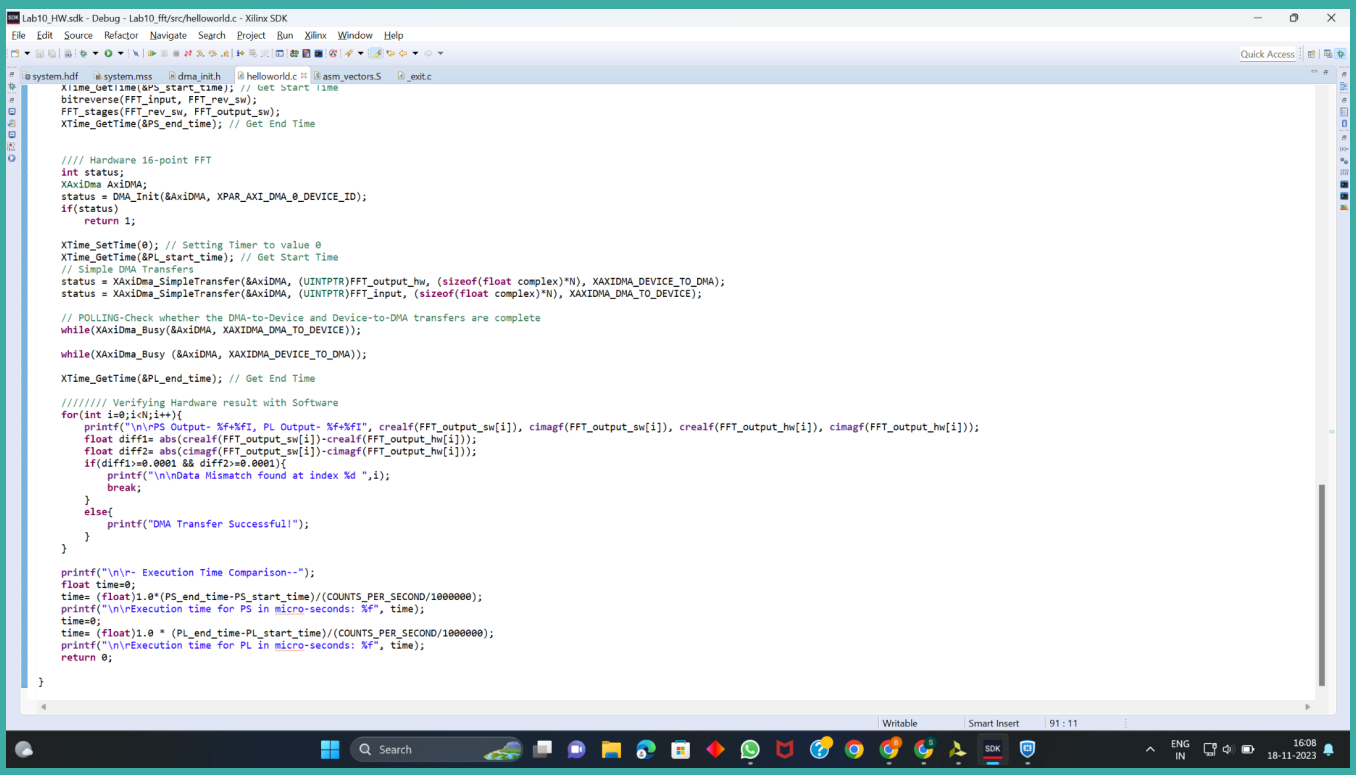
Source Code

16 Point FFT:

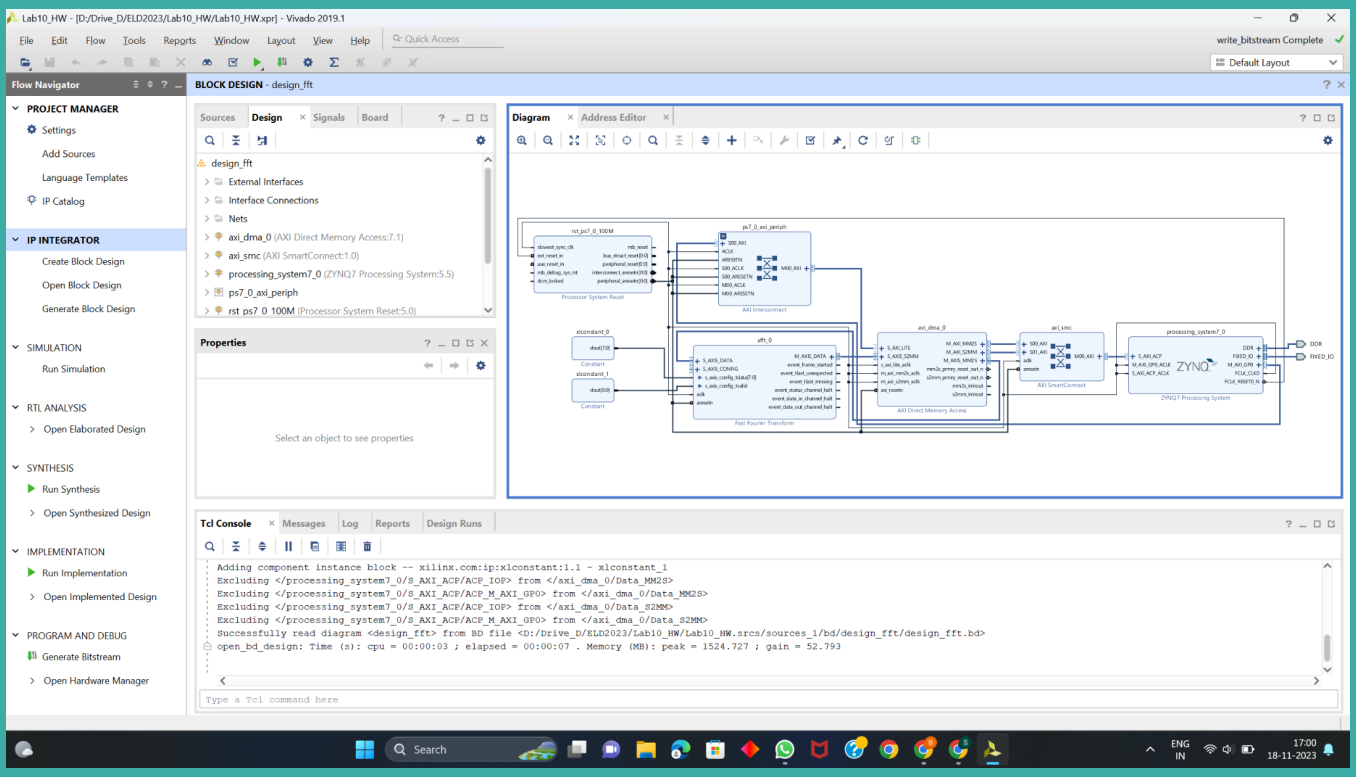


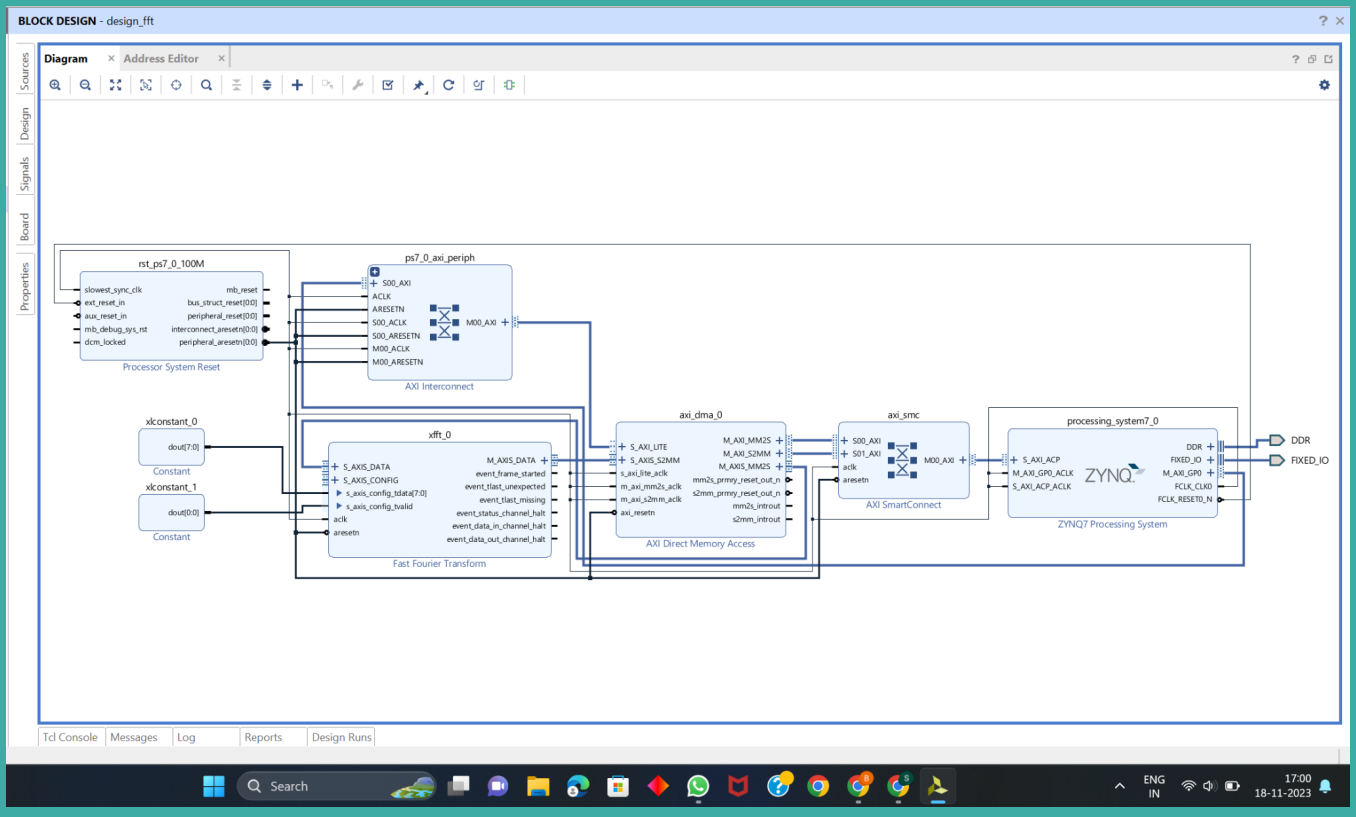



Lab10_HW.sdk - Debug - Lab10_fft/src/helloworld.c - Xilinx SDK
File Edit Source Refactor Navigate Search Project Run Xilinx Window Help
Quick Access
system.hdf system.mss dma_init.h helloworld.c asm_vectors.S _exit.c
int main() {
init_platform();
// Initializing Timer instances for PS and PL
XTime PL_start_time, PL_end_time;
XTime PS_start_time,PS_end_time;
// Initializing software and hardware output buffers
const float complex FFT_input[N] = {11 + 23*I,32 + 10*I,91 + 94*I,15 + 69*I,47 + 96*I,44 + 12*I,96 + 17*I,49 + 58*I,
11 + 23*I,32 + 10*I,91 + 94*I,15 + 69*I,47 + 96*I,44 + 12*I,96 + 17*I,49 + 58*I};
float complex FFT_output_sw[N], FFT_output_hw[N];
float complex FFT_rev_sw[N];
//// Software 8-point FFT
XTime_SetTime(0); // Setting Timer to value 0
XTime_GetTime(&PS_start_time); // Get Start Time
bitreverse(FFT_input, FFT_rev_sw);
FFT_stages(FFT_rev_sw, FFT_output_sw);
XTime_GetTime(&PS_end_time); // Get End Time
//// Hardware 16-point FFT
int status;
XAxiDma AxiDMA;
status = DMA_Init(&AxiDMA, XPAR_AXI_DMA_0_DEVICE_ID);
if(status)
return 1;
XTime_SetTime(0); // Setting Timer to value 0
XTime_GetTime(&PL_start_time); // Get Start Time
// Simple DMA Transfers
status = XAxiDma_SimpleTransfer(&AxiDMA, (UINTPTR)FFT_output_hw, (sizeof(float complex)*N), XAXIDMA_DEVICE_TO_DMA);
status = XAxiDma_SimpleTransfer(&AxiDMA, (UINTPTR)FFT_input, (sizeof(float complex)*N), XAXIDMA_DMA_TO_DEVICE);
// POLLING-Check whether the DMA-to-Device and Device-to-DMA transfers are complete
while(XAxiDma_Busy(&AxiDMA, XAXIDMA_DMA_TO_DEVICE));
while(XAxiDma_Busy (&AxiDMA, XAXIDMA_DEVICE_TO_DMA));
XTime_GetTime(&PL_end_time); // Get End Time
//////// Verifying Hardware result with Software
for(int i=0;i<N;i++){
printf("\n\rPS Output- %f+%fI, PL Output- %f+%fI", crealf(FFT_output_sw[i]), cimagf(FFT_output_sw[i]), crealf(FFT_output_hw[i]), cimagf(FFT_output_hw[i]));
float diff1= abs(crealf(FFT_output_sw[i])-crealf(FFT_output_hw[i]));
float diff2= abs(cimagf(FFT_output_sw[i])-cimagf(FFT_output_hw[i]));
if(diff1>=0.0001 && diff2>=0.0001){
printf("\n\nData Mismatch found at index %d ",i);
break;
}
else{
Writable
Smart Insert
91 : 11
Search
ENG
IN
16:07
18-11-2023


Lab10_HW.sdk - Debug - Lab10_fft/src/helloworld.c - Xilinx SDK
File Edit Source Refactor Navigate Search Project Run Xilinx Window Help
Quick Access
system.hdf system.mss dma_init.h helloworld.c asm_vectors.S _exit.c
bitreverse(FFT_input, FFT_rev_sw);
FFT_stages(FFT_rev_sw, FFT_output_sw);
XTime_GetTime(&PS_end_time); // Get End Time
//// Hardware 16-point FFT
int status;
XAxiDma AxiDMA;
status = DMA_Init(&AxiDMA, XPAR_AXI_DMA_0_DEVICE_ID);
if(status)
return 1;
XTime_SetTime(0); // Setting Timer to value 0
XTime_GetTime(&PL_start_time); // Get Start Time
// Simple DMA Transfers
status = XAxiDma_SimpleTransfer(&AxiDMA, (UINTPTR)FFT_output_hw, (sizeof(float complex)*N), XAXIDMA_DEVICE_TO_DMA);
status = XAxiDma_SimpleTransfer(&AxiDMA, (UINTPTR)FFT_input, (sizeof(float complex)*N), XAXIDMA_DMA_TO_DEVICE);
// POLLING-Check whether the DMA-to-Device and Device-to-DMA transfers are complete
while(XAxiDma_Busy(&AxiDMA, XAXIDMA_DMA_TO_DEVICE));
while(XAxiDma_Busy (&AxiDMA, XAXIDMA_DEVICE_TO_DMA));
XTime_GetTime(&PL_end_time); // Get End Time
//////// Verifying Hardware result with Software
for(int i=0;i<N;i++){
printf("\n\rPS Output- %f+%fI, PL Output- %f+%fI", crealf(FFT_output_sw[i]), cimagf(FFT_output_sw[i]), crealf(FFT_output_hw[i]), cimagf(FFT_output_hw[i]));
float diff1= abs(crealf(FFT_output_sw[i])-crealf(FFT_output_hw[i]));
float diff2= abs(cimagf(FFT_output_sw[i])-cimagf(FFT_output_hw[i]));
if(diff1>=0.0001 && diff2>=0.0001){
printf("\n\nData Mismatch found at index %d ",i);
break;
}
else{
printf("DMA Transfer Successful!");
}
}
printf("\n\r- Execution Time Comparison--");
float time=0;
time= (float)1.0*(PS_end_time-PS_start_time)/(COUNTS_PER_SECOND/1000000);
printf("\n\rExecution time for PS in micro-seconds: %f", time);
time=0;
time= (float)1.0 * (PL_end_time-PL_start_time)/(COUNTS_PER_SECOND/1000000);
printf("\n\rExecution time for PL in micro-seconds: %f", time);
return 0;
}
Writable
Smart Insert
91 : 11
Search
ENG
IN
16:08
18-11-2023

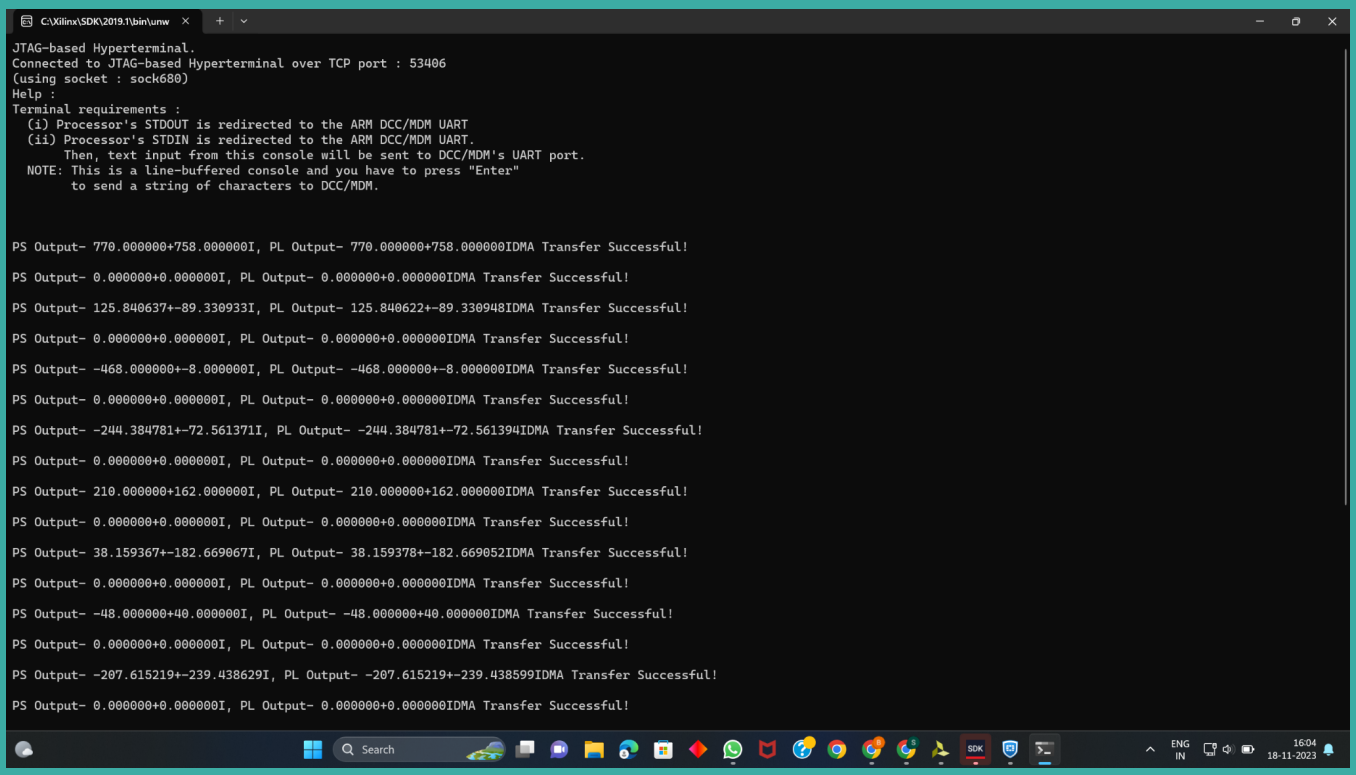
BLOCK DESIGN

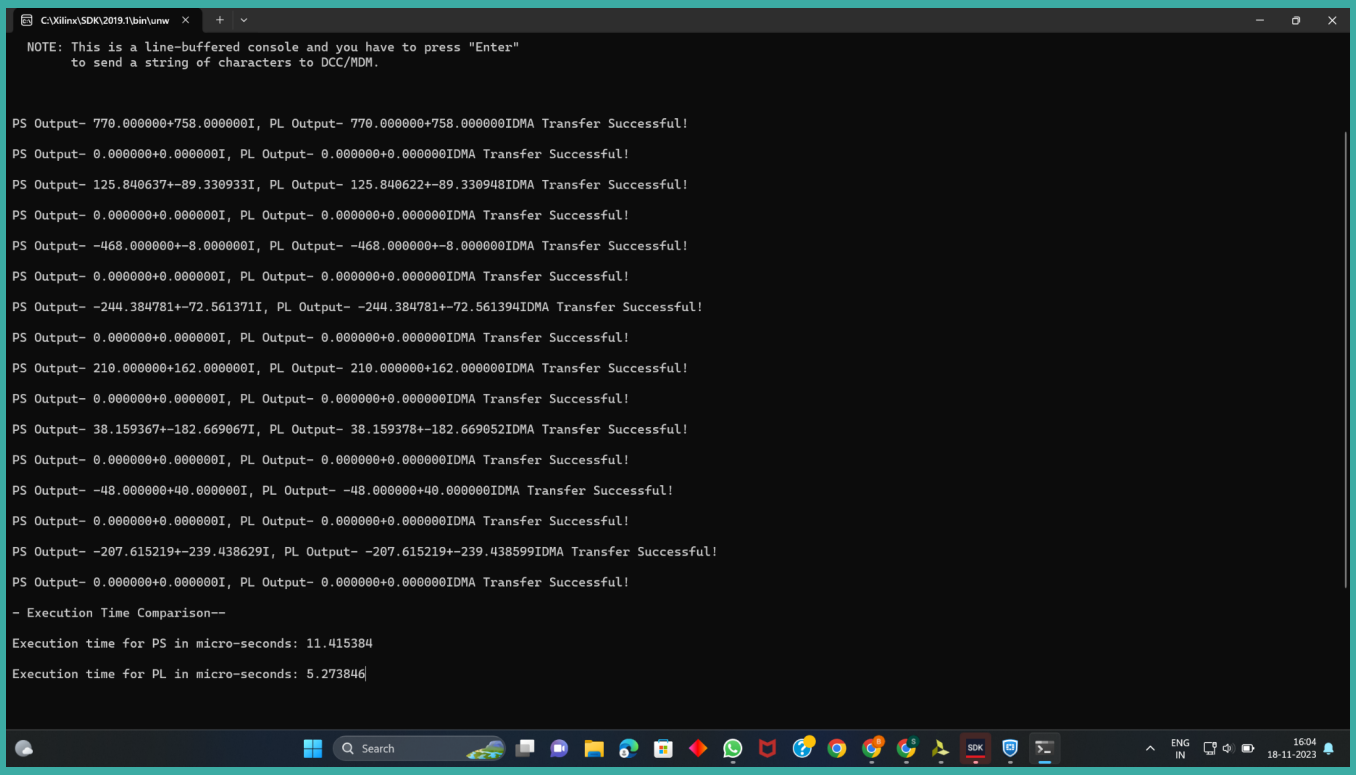


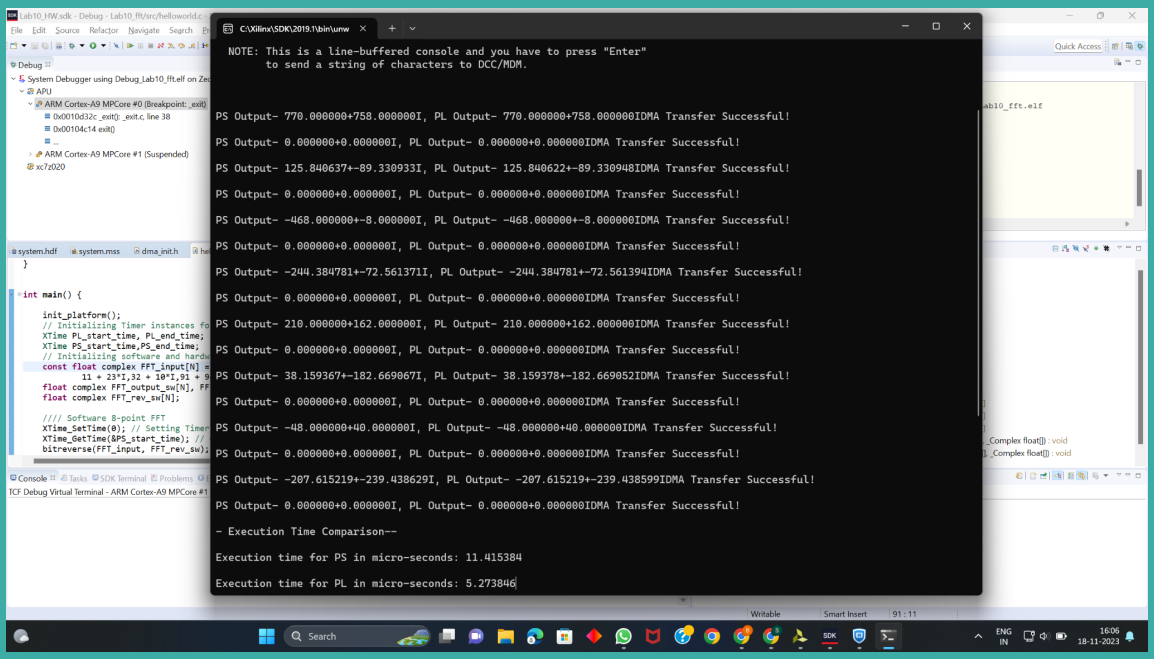


jtag terminal output

16 Point FFT:







(ZOOMED OUTPUT)

```
PS Output- 770.000000+758.000000I, PL Output- 770.000000+758.000000IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- 125.840637+-89.330933I, PL Output- 125.840622+-89.330948IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- -468.000000+-8.000000I, PL Output- -468.000000+-8.000000IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- -244.384781+-72.561371I, PL Output- -244.384781+-72.561394IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- 210.000000+162.000000I, PL Output- 210.000000+162.000000IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- 38.159367+-182.669067I, PL Output- 38.159378+-182.669052IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- -48.000000+40.000000I, PL Output- -48.000000+40.000000IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

PS Output- -207.615219+-239.438629I, PL Output- -207.615219+-239.438599IDMA Transfer Successful!

PS Output- 0.000000+0.000000I, PL Output- 0.000000+0.000000IDMA Transfer Successful!

- Execution Time Comparison--

Execution time for PS in micro-seconds: 11.415384

Execution time for PL in micro-seconds: 5.273846
```

Expected output from Matlab

```
>> x = [11 + 23i,32 + 10i,91 + 94i,15 + 69i,47 + 96i,44 + 12i,96 + 17i,49 + 58i,11 + 23i,32 + 10i,91 + 94i,15 + 69i,47 + 96i,44 + 12i,96 + 17i,49 + 58i];
>> fft(x)

ans =

   1.0e+02 *

  Columns 1 through 9

   7.7000 + 7.5800i   0.0000 + 0.0000i   1.2584 - 0.8933i   0.0000 + 0.0000i  -4.6800 - 0.0800i   0.0000 + 0.0000i  -2.4438 - 0.7256i   0.0000 + 0.0000i   2.1000 + 1.6200i

  Columns 10 through 16

   0.0000 + 0.0000i   0.3816 - 1.8267i   0.0000 + 0.0000i  -0.4800 + 0.4000i   0.0000 + 0.0000i  -2.0762 - 2.3944i   0.0000 + 0.0000i
```

Thank You