tags: MSOC

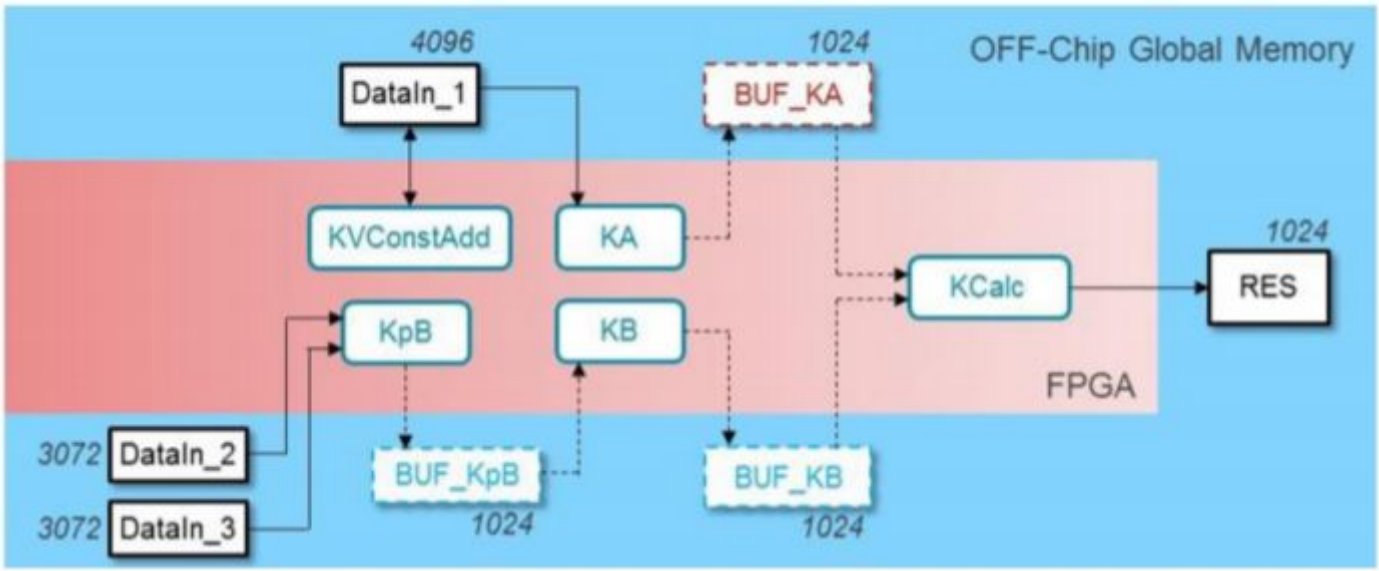
MSOC Lab3 Report

Objective

After completing this lab, we will be able to learn:

- Build a hardware acceleration application on vitis
- Understand the command in vitis platform
- Compare 4 different hardware design examples

System diagram



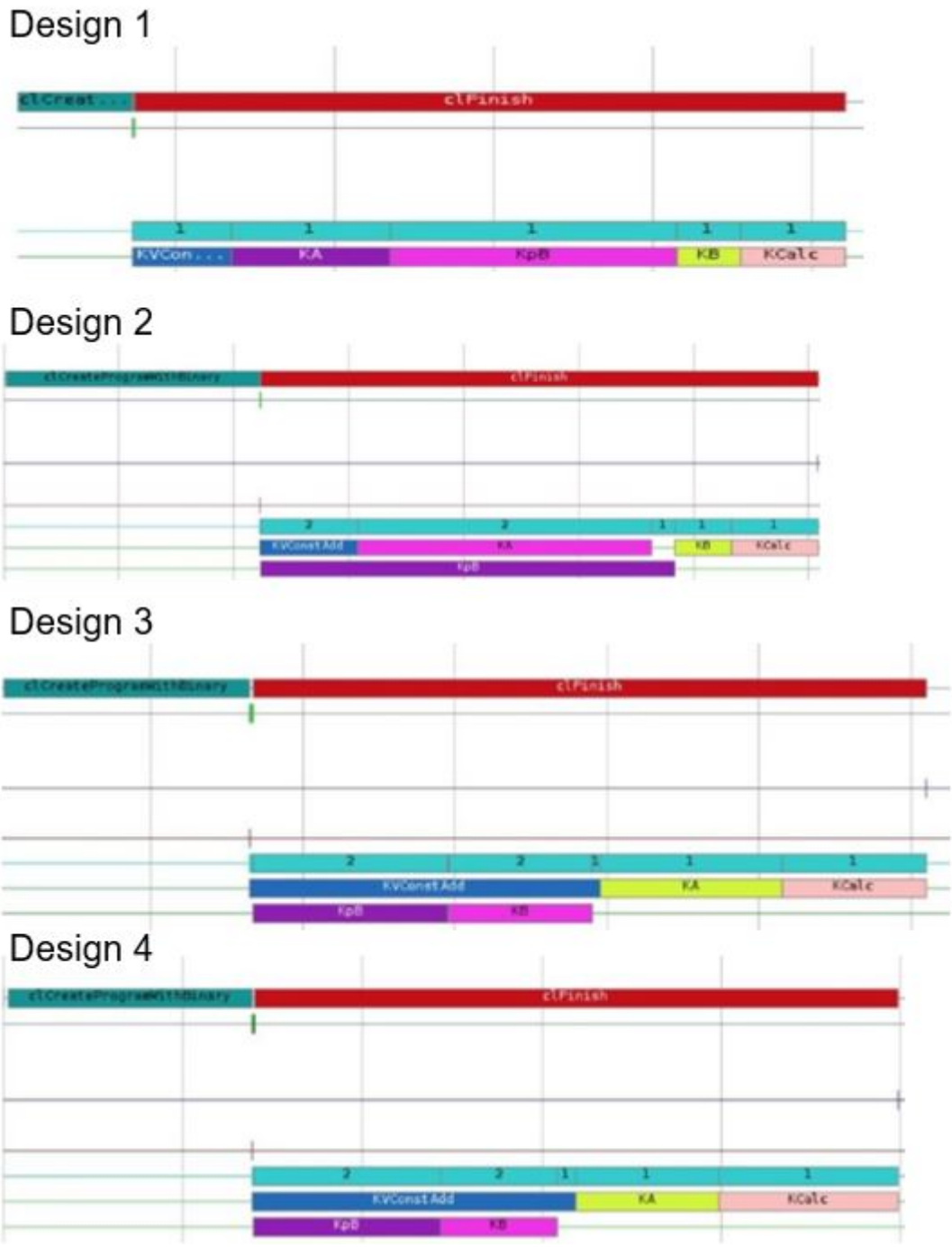
Optimizations

In this lab, we compare 4 kinds of designs. The following table shows the different of each files.

	1	2	3	4
impl.	Naive	Parallel	Dataflow	Array partition
host.cpp	0	1	1	1
K_KpB.cpp	0	0	1	1
K_KA.cpp	0	0	0	1
K_KB.cpp	0	0	0	1

- host.cpp: Version 1 Binds Multiple buffer in ‘Mem_Pointers’ and call ‘clEnqueueMigrateMemObjects’ once, while version 0 calls the function for every buffer.
- K_KpB.cpp: Version 1 adds ‘memcpy’ function in I/O and includes additional pragma ‘#pragma HLS dataflow’ and ‘max_read_burst_length=256 max_write_burst_length=256’ for m_axi interface.
- K_KA/K_KB.cpp: Version 1 includes additional pragma ‘#pragma HLS array_partition variable=TMP_BUF block factor=3 dim=1’ compared with version 0.

Application timeline & Observation



We can have some observation from the above figure:

- KpB in Design 2 starts simultaneouly with KConstAdd, thus having lower latency. This is cause by the event enque process in ‘host.cpp’. Host program in Design 1 allows KpB to

execute only after KA is done; however, KpB has no dependency with KA, which waste a lot of time.

- KpB in Design 3 takes much less time compared with Design 2, resulting lower overall latency. This is cause by the databurst transmission specified in the “KpB.cpp” kernel.
- KA and KB takes less cycles, resulting in lower overall latenccy. The ‘#pragma HLS array_partition variable=TMP_BUF block factor=3 dim=1’ pragma makes the input bandwidth become 3x due to partition the array into three RAMs.

Profile Summary & Observation

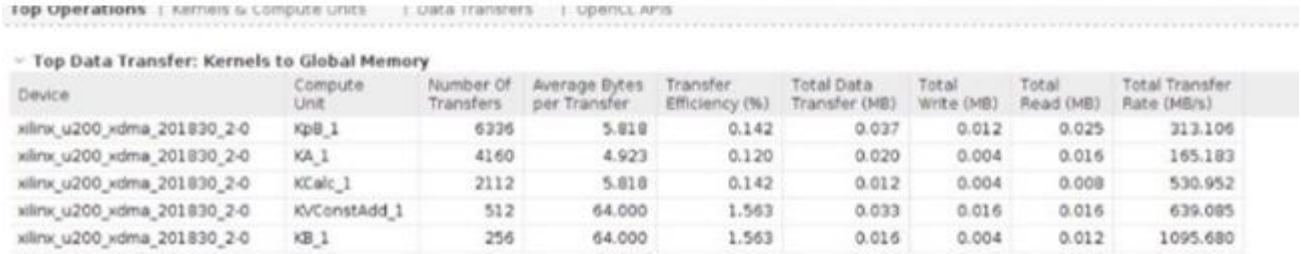
Design 1

Top Operations | Kernels & Compute Units | Data Transfers | OpenCL APIs

Top Data Transfer: Kernels to Global Memory

Device	Compute Unit	Number Of Transfers	Average Bytes per Transfer	Transfer Efficiency (%)	Total Data Transfer (MB)	Total Write (MB)	Total Read (MB)	Total Transfer Rate (MB/s)
xilinx_u200_xdma_201830_2-0	KpB_1	6336	5.818	0.142	0.037	0.012	0.025	531.846
xilinx_u200_xdma_201830_2-0	KA_1	4160	4.923	0.120	0.020	0.004	0.016	246.975
xilinx_u200_xdma_201830_2-0	KCalc_1	2112	5.818	0.142	0.012	0.004	0.008	530.952
xilinx_u200_xdma_201830_2-0	KVConstAdd_1	512	64.000	1.563	0.033	0.016	0.016	699.473
xilinx_u200_xdma_201830_2-0	KB_1	256	64.000	1.563	0.016	0.004	0.012	1095.680

Design 2

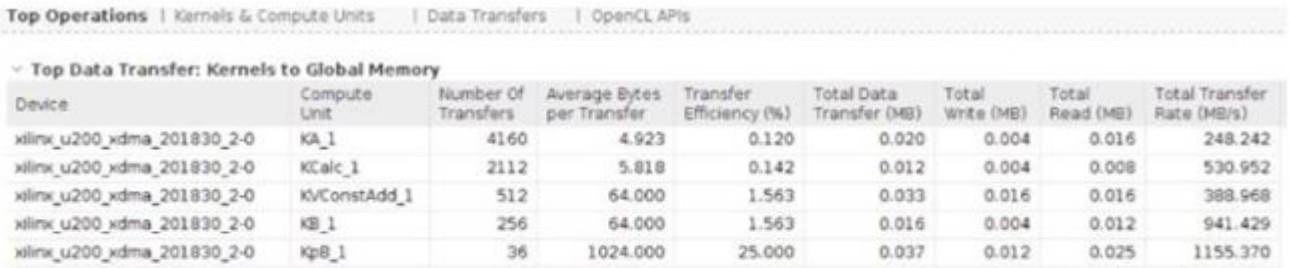


Top Operations | Kernels & Compute Units | Data Transfers | OpenCL APIs

Top Data Transfer: Kernels to Global Memory

Device	Compute Unit	Number Of Transfers	Average Bytes per Transfer	Transfer Efficiency (%)	Total Data Transfer (MB)	Total Write (MB)	Total Read (MB)	Total Transfer Rate (MB/s)
xilinx_u200_xdma_201830_2-0	KpB_1	6336	5.818	0.142	0.037	0.012	0.025	313.106
xilinx_u200_xdma_201830_2-0	KA_1	4160	4.923	0.120	0.020	0.004	0.016	165.183
xilinx_u200_xdma_201830_2-0	KCalc_1	2112	5.818	0.142	0.012	0.004	0.008	530.952
xilinx_u200_xdma_201830_2-0	KVConstAdd_1	512	64.000	1.563	0.033	0.016	0.016	639.085
xilinx_u200_xdma_201830_2-0	KB_1	256	64.000	1.563	0.016	0.004	0.012	1095.680

Design 3



Top Operations | Kernels & Compute Units | Data Transfers | OpenCL APIs

Top Data Transfer: Kernels to Global Memory

Device	Compute Unit	Number Of Transfers	Average Bytes per Transfer	Transfer Efficiency (%)	Total Data Transfer (MB)	Total Write (MB)	Total Read (MB)	Total Transfer Rate (MB/s)
xilinx_u200_xdma_201830_2-0	KA_1	4160	4.923	0.120	0.020	0.004	0.016	248.242
xilinx_u200_xdma_201830_2-0	KCalc_1	2112	5.818	0.142	0.012	0.004	0.008	530.952
xilinx_u200_xdma_201830_2-0	KVConstAdd_1	512	64.000	1.563	0.033	0.016	0.016	388.968
xilinx_u200_xdma_201830_2-0	KB_1	256	64.000	1.563	0.016	0.004	0.012	941.429
xilinx_u200_xdma_201830_2-0	KpB_1	36	1024.000	25.000	0.037	0.012	0.025	1155.370

Design 4

Top Operations | Kernels & Compute Units | Data Transfers | OpenCL APIs

Top Data Transfer: Kernels to Global Memory

Device	Compute Unit	Number Of Transfers	Average Bytes per Transfer	Transfer Efficiency (%)	Total Data Transfer (MB)	Total Write (MB)	Total Read (MB)	Total Transfer Rate (MB/s)
xilinx_u200_xdma_201830_2-0	KCalc_1	2112	5.818	0.142	0.012	0.004	0.008	531.258
xilinx_u200_xdma_201830_2-0	KVConstAdd_1	512	64.000	1.563	0.033	0.016	0.016	388.707
xilinx_u200_xdma_201830_2-0	KA_1	320	64.000	1.563	0.020	0.004	0.016	1182.680
xilinx_u200_xdma_201830_2-0	KB_1	256	64.000	1.563	0.016	0.004	0.012	982.254
xilinx_u200_xdma_201830_2-0	KpB_1	36	1024.000	25.000	0.037	0.012	0.025	1155.370

- The data transfer rate of KpB Design 3 is much higher than Design 2 due to lower latency cause by data burst.

- The data transfer rate of KA in Design 4 is much higher than Design 3 due to lower latency cause by array partition.

Learnt

- The vitis command in host.cpp
- Advantages of databurst and array partition