Reading-Paper-And-Its-Code

OPU: An FPGA-based Overlay Processor for Convolutional Neural Networks

in folder src: its main and tests code

Part3 driver.cc Time:2022-10-14

Device作为最高层抽象，模拟了整个OPU overlay，包括其中的component、instruction、flow和layer等；
除了compiler部分，当导入已编译的数据后，通过driver.cc中的main函数初始化即可开始OPU的模拟。

- main function Part

```
//initialize the class Device
Device* dev = new Device();

//初始化flow: IF,COMPUTE,STORE 加载数据
//剩下的均与指令相关
dev->RunLoad(load);
dev->RunCompute(compute);
dev->RunStore(store);
dev->FetchInsn();
dev->Run();
```

无非数据加载，执行指令几项。

重新阅读Device类

- 整理变量以及变量类型

变量名称	变量类型
Special purpose registers	
*reg_*	OPURegFile
Scratchpad memory	
*ins_ram_*	Ins_ram_t -> SRAM<32, 2<<15, 1>
*fm_ram_a_*	Fm_ram_t -> SRAM<512, 4096, 1>
*fm_ram_b_*	Fm_ram_t -> SRAM<512, 4096, 1>
*wgt_ram_a_*	Wgt_ram_t -> SRAM<512, 64, 16>
*wgt_ram_b_*	Wgt_ram_t -> SRAM<512, 64, 16>
*bias_ram_a_*	Bias_ram_t -> SRAM<1024, 1, 1>

变量名称	**变量类型**
*bias_ram_b_*	Bias_ram_t -> SRAM<1024, 1, 1>
IPA	
*ipa_*	IPA_t -> IPA<16, 32, PRECISION>
Partial sum buffer	
*tmp_buf_*	Tmp_buf_t -> SRAM<642PRECISION, 2<<11, 1>
DDR	
dram	DRAM -> VirtualMemory
dw	
*wgt_ram_dw_*	Wgt_ram_dw_t -> SRAM<1024, 64, 16>
*ipa_dw_*	IPA_DW_t -> IPA<16, 64, PRECISION>
Double buffering	
*fm_ram_vec_*	std::vector<Fm_ram_t*>
*wgt_ram_vec_*	std::vector<Wgt_ram_t*>
*bias_ram_vec_*	std::vector<Bias_ram_t*>
*ins_ram_vec_*	std::vector<Ins_ram_t*>
fm_ram_id	size_t
wgt_ram_id	size_t
bias_ram_id	size_t
ins_ram_id	size_t
Control flow	
Global variables	
Function wrappers	
Sub-function wrappers	
Utility function	
Debug	

instruction.h中的类其成员函数均为赋值，并无操作

for each layer:
dev->FetchInsn() // fetch one instruction block 获取一整个指令块
dev->Run(#phases) // run until no new events pop out by default