LET Coarse-Grained Resources Be Shared: Mapping Entire Neural Networks on FPGAs

Tzung-Han Juang (McGill University) Christof Schlaak (University of Edinburgh) Christophe Dubach (McGill University)

Motivation

Commercial tools such as Intel OpenCL FPGA (Field Programmable Gate Array) SDK poorly support course-grained resource sharing with functions. FPGA resource usage increases with the number of function calls.

```
void matMul(int* A, int* B, int* C, int sz) {
  for(i=0, j=0; i<sz, j<sz; i++, j++) {
    #pragma unroll
    for(int k=0; k<size; k++)
      C[i][j] = A[i][k] * B[j][k]; }}

kernel main(int* A, int* B, int* C, int sz) {
  matMul(A, B, C, sz);
  matMul(A, B, C, sz);
  matMul(A, B, C, sz);}
```

# of calls	FPGA Resources Logic(%)	RAM(%)	DSP(%)
1	23	19	34
2	32	36	68
3	Out of DSPs		

Background

We extend Shir [1, 2], a functional accelerator generation framework, to transform high-level sharing-related primitives into hardware.

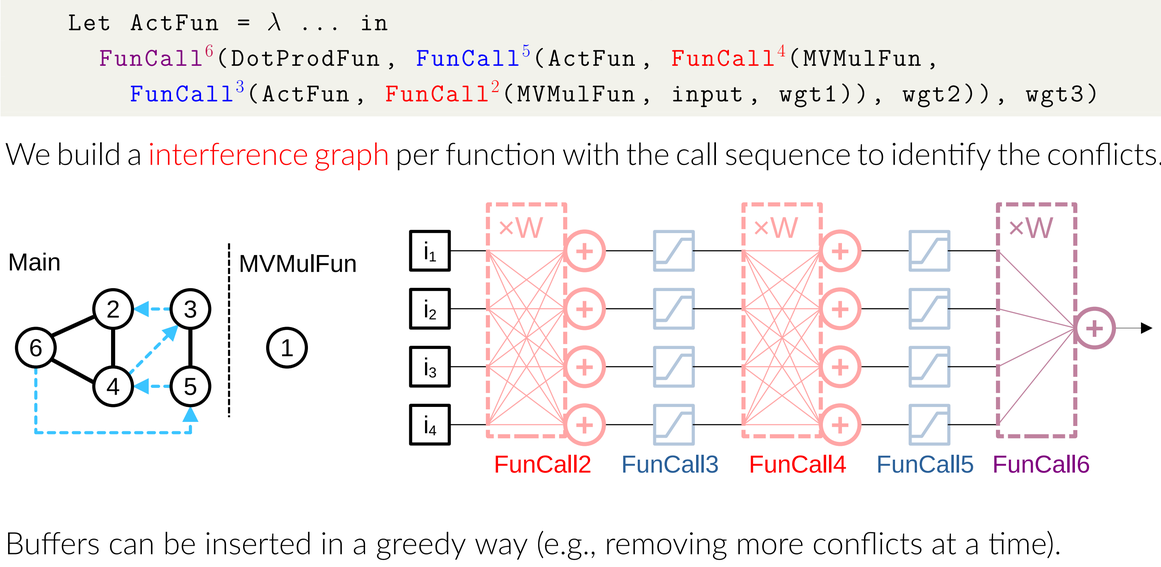
```
Let +1Fun = λ i -> Map(+1, i)) in
  B = FunCall(+1Fun, A)    // Call 0
  Out = FunCall(+1Fun, B) // Call 1
```

- `Let` defines a value under a scope.
- `λ` defines an anonymous function.
- `FunCall` calls a lambda function.

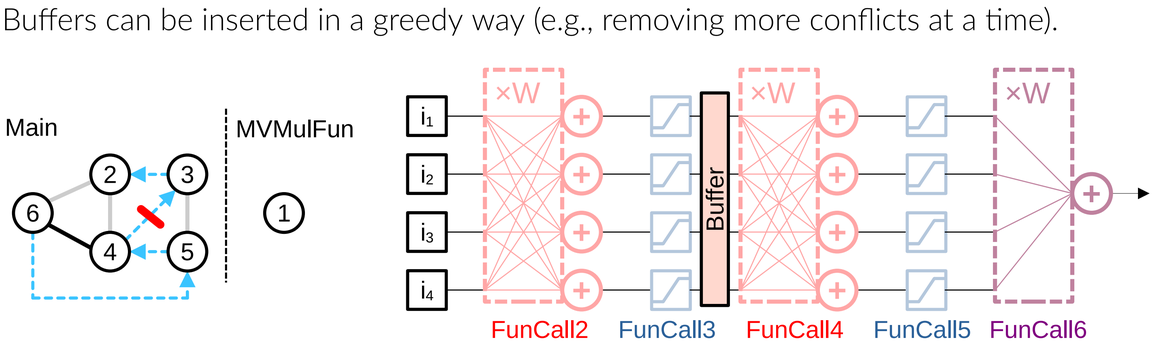
Note that a function consumes its input in a streaming way.

Function Call Conflicts

Conflict: Two function calls are data-dependant and access the same function.



Conflict Removal: A function is not accessed at the same time.



Handling Conflicts

However, an expression can contain multiple functions and a function call can be inside another function. Conflicts can be hidden behind indirect calls.

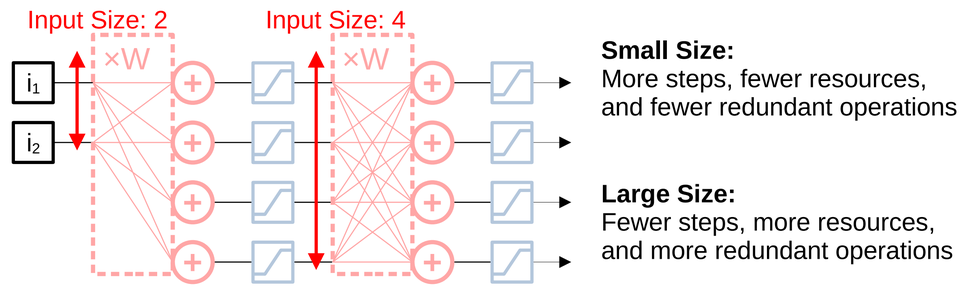
```
Let DotProdFun = λ  ... in
  Let MVMulFun = λ ... FunCall¹(DotProdFun, ...) ... in
    Let ActFun = λ ... in
      FunCall⁶(DotProdFun, FunCall⁵(ActFun, FunCall⁴(MVMulFun,
        FunCall³(ActFun, FunCall²(MVMulFun, input, wgt1)), wgt2)), wgt3)
```

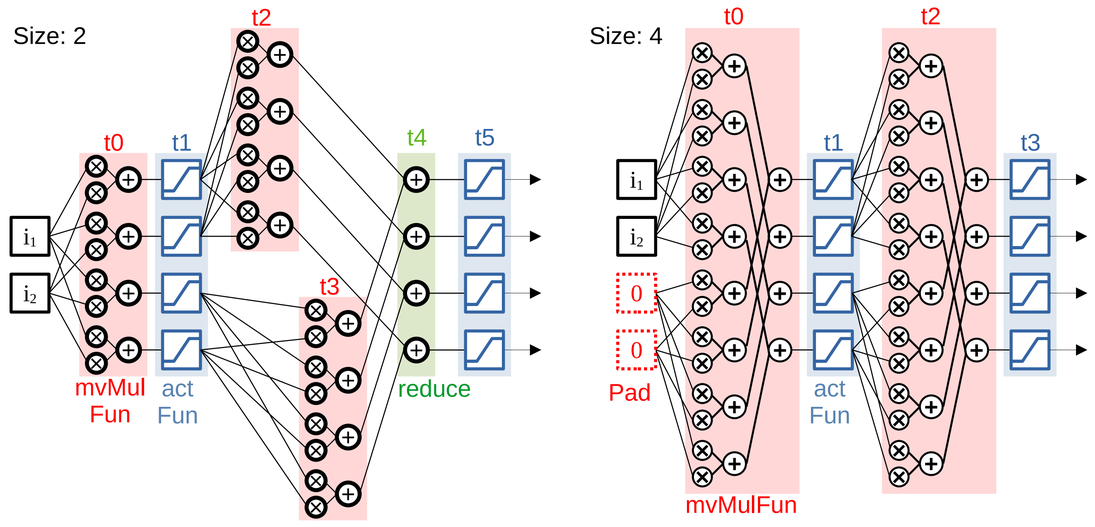
We build a interference graph per function with the call sequence to identify the conflicts.

Buffers can be inserted in a greedy way (e.g., removing more conflicts at a time).

Dealing with Different Shapes

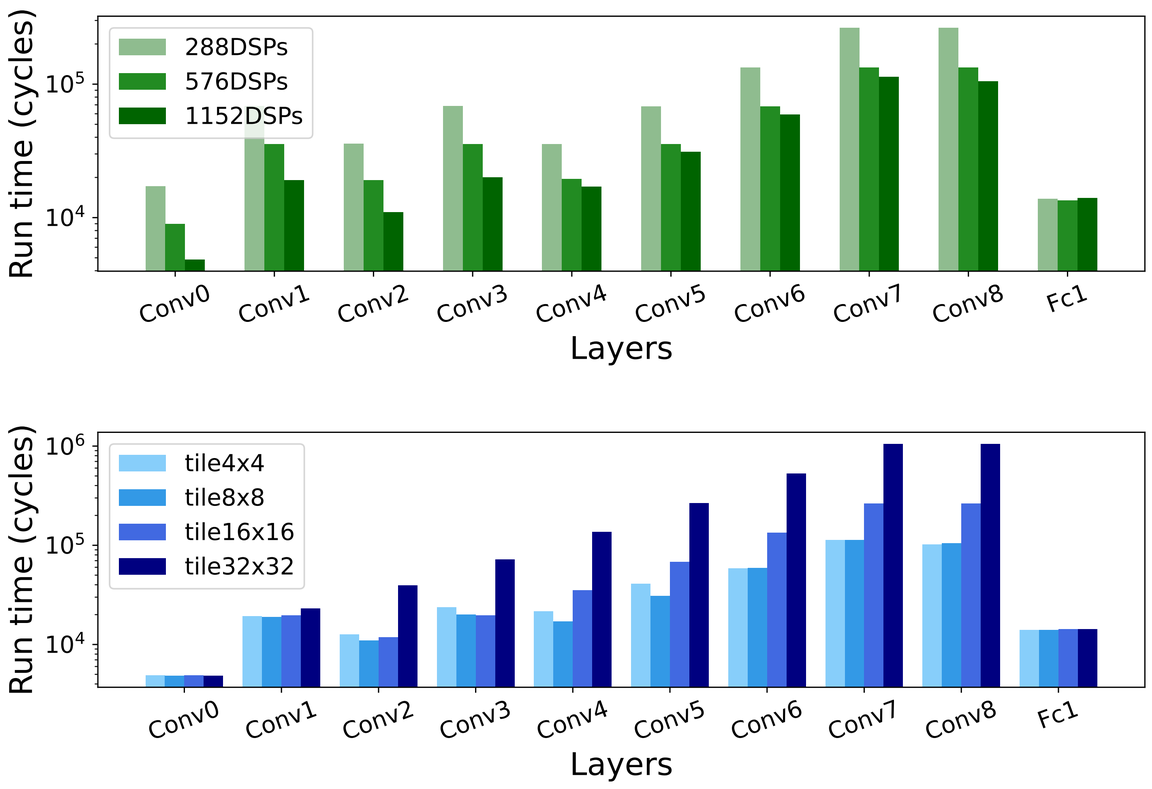
A trade-off is required if we want to share a function with different sizes.





Evaluation

The experiments include data transfer between the host and Arria 10 FPGA via PCIE.



Comparison with state-of-the-art	FPGA	OPs/Cycle	DSPs.
VGG-CIFAR (32 img. size, int8)			
ScaleHLS [3]	VU9P	653	878/2280
This paper	Arria 10	2028	1152/1518
VGG Convolutions (224 img. size, int16)			
OpenCL Winograd [4]	Arria 10	1202	544/1518
OpenCL Direct Conv [5]	Arria 10	1678	543/1518
This paper	Arria 10	2225	576/1518
Tiny Yolo v2 full network (416 img. size, int8)			
OpenCL YOLO [6]	Arria 10	2632	884/1518
This paper	Arria 10	3042	1152/1518

References

[1] C. Schlaak, T.-H. Juang, and C. Dubach, "Memory-aware functional ir for higher-level synthesis of accelerators," ACM Transactions on Architecture and Code Optimization, jan 2022.

[2] C. Schlaak, T.-H. Juang, and C. Dubach, "Optimizing data reshaping operations in functional irs for high-level synthesis," in ACM International Conference on Languages, Compilers, and Tools for Embedded Systems, LCTES, 2022.

[3] H. Ye, H. Jun, H. Jeong, S. Neuendorffer, and D. Chen, "Scalehls: A scalable high-level synthesis framework with multi-level transformations and optimizations: Invited," in Proceedings of the 59th ACM/IEEE Design Automation Conference, DAC22, Association for Computing Machinery, 2022.

[4] Z. Bai, H. Fan, L. Liu, L. Liu, and D. Wang, "An opencl-based fpga accelerator with the winograd's minimal filtering algorithm for convolution neuron networks," in 2019 IEEE 5th International Conference on Computer and Communications, 2019.

[5] H. Li, Acceleration of Deep Learning on FPGA. PhD thesis, University of Windsor (Canada), 2017.

[6] K. Xu, X. Wang, X. Liu, C. Cao, H. Li, H. Peng, and D. Wang, "A dedicated hardware accelerator for real-time acceleration of yolov2," Journal of Real-Time Image Processing, vol. 18, 2021.