

QuickDough: A Rapid FPGA Loop Accelerator Design Framework Using Soft CGRA Overlay

Cheng Liu, Ho-Cheung Ng, **Hayden K. H. So**

Department of
Electrical and Electronic Engineering
University of Hong Kong

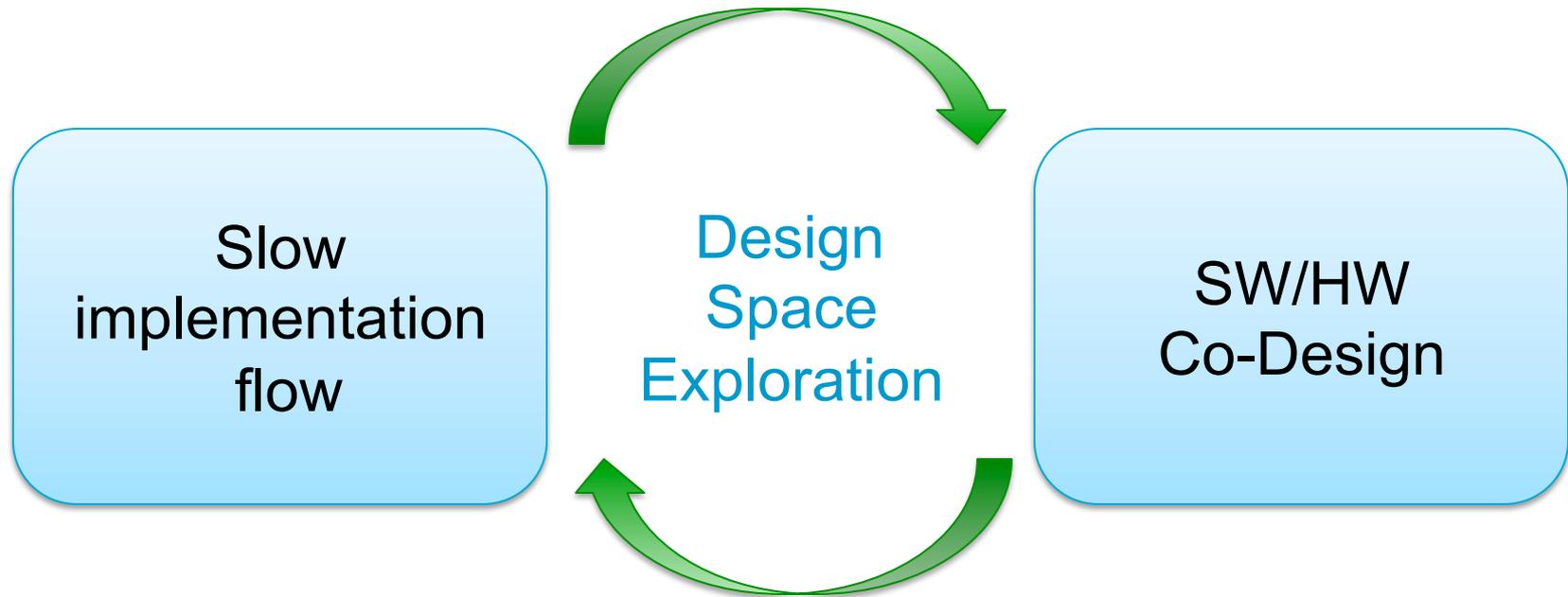
FPT 2015 – 2015/12/07



Motivation

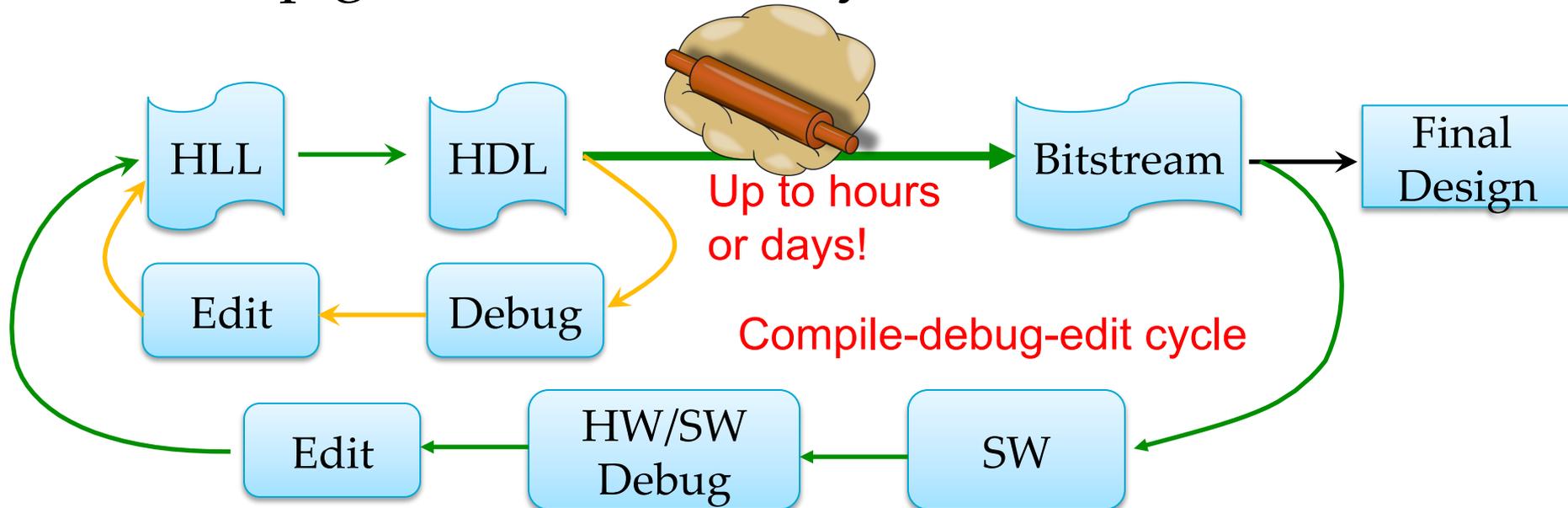
Designing FPGA Accelerators is Hard

Design productivity much lower than sw development



Need for Fast Design Iterations

- Multiple levels of **edit-compile-debug** iterations to develop good accelerator systems



QuickDough Overview

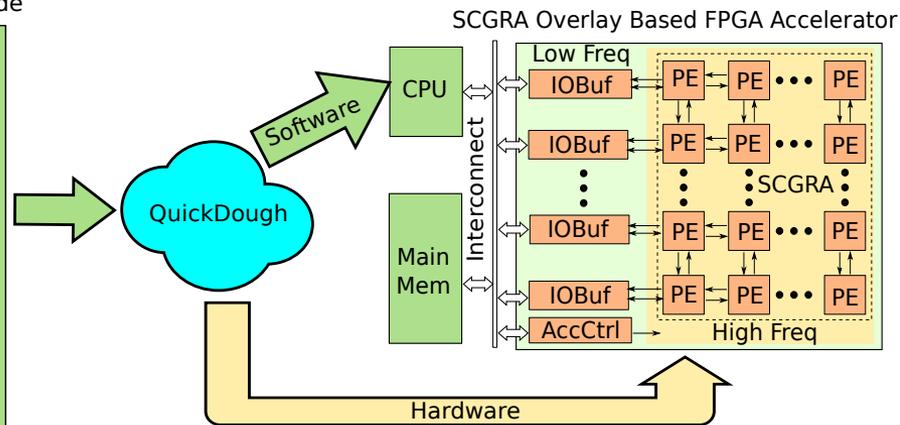
- A **high productivity compilation framework** for high-level applications on CPU-FPGA computers.
 - Compiles **software** and **FPGA accelerator** design within the same framework
 - Overall **SW-like compile speed**

Goals:

- Fast design iterations
- Good performance

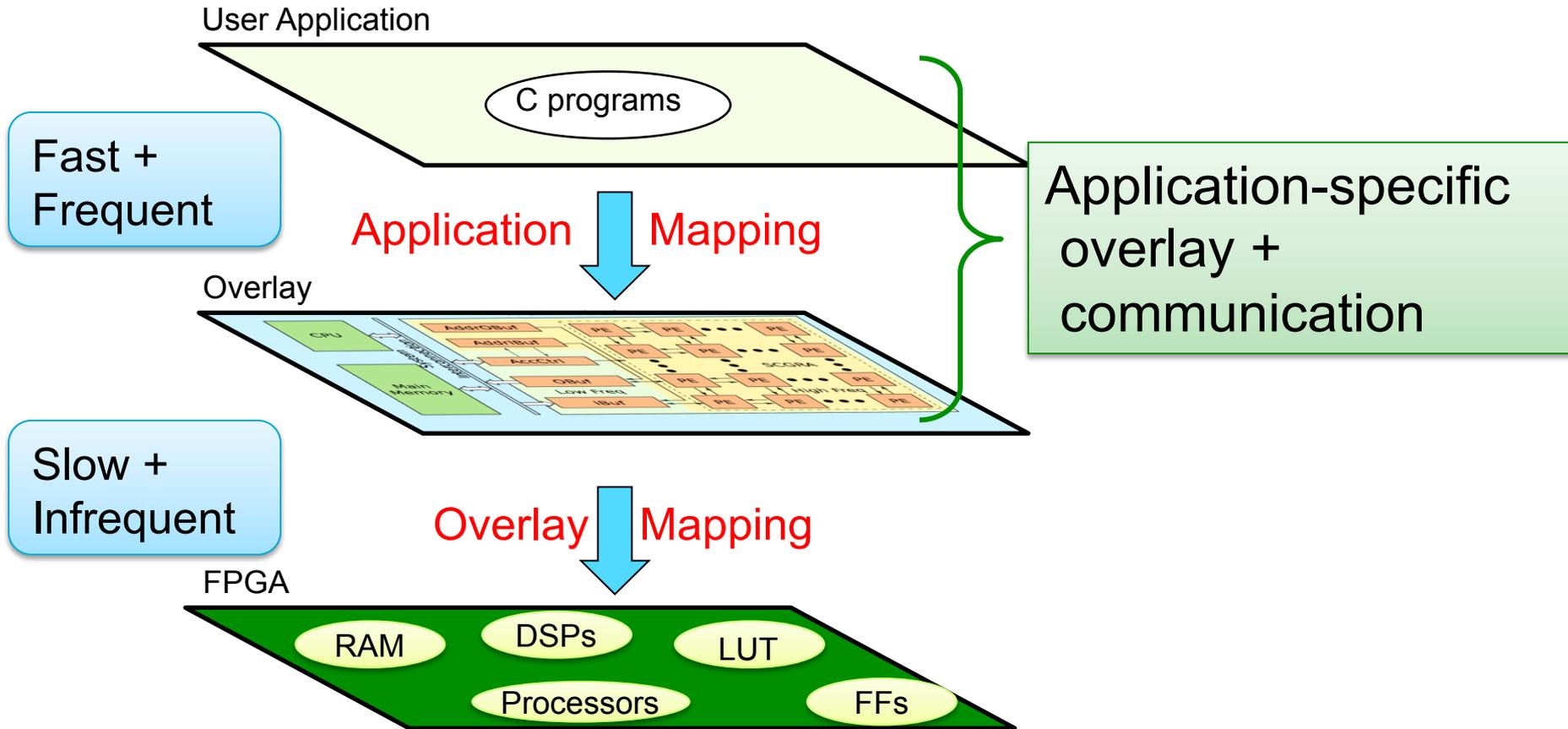
High Level Source Code

```
#define N 10000
...
//Compute kernel
for(i=0; i<N; i++){
  c[i] = a[i] x b[i]
}
...
//Compute kernel
for(i=0; i<N; i++){
  f[i] = d[i] + e[i]
}
...
```

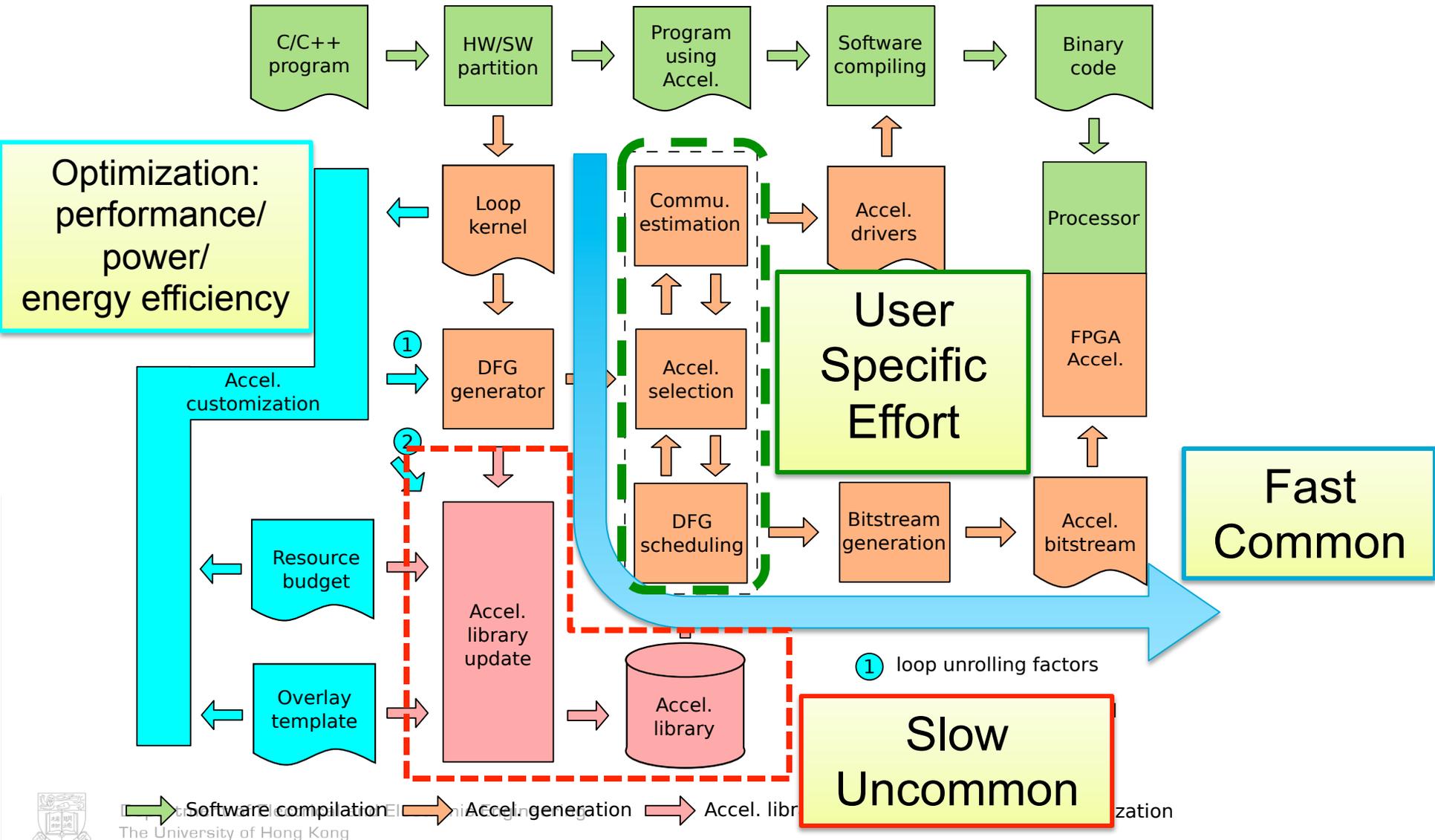


Main Idea: FPGA Overlay

A 2-step HW-SW compilation approach

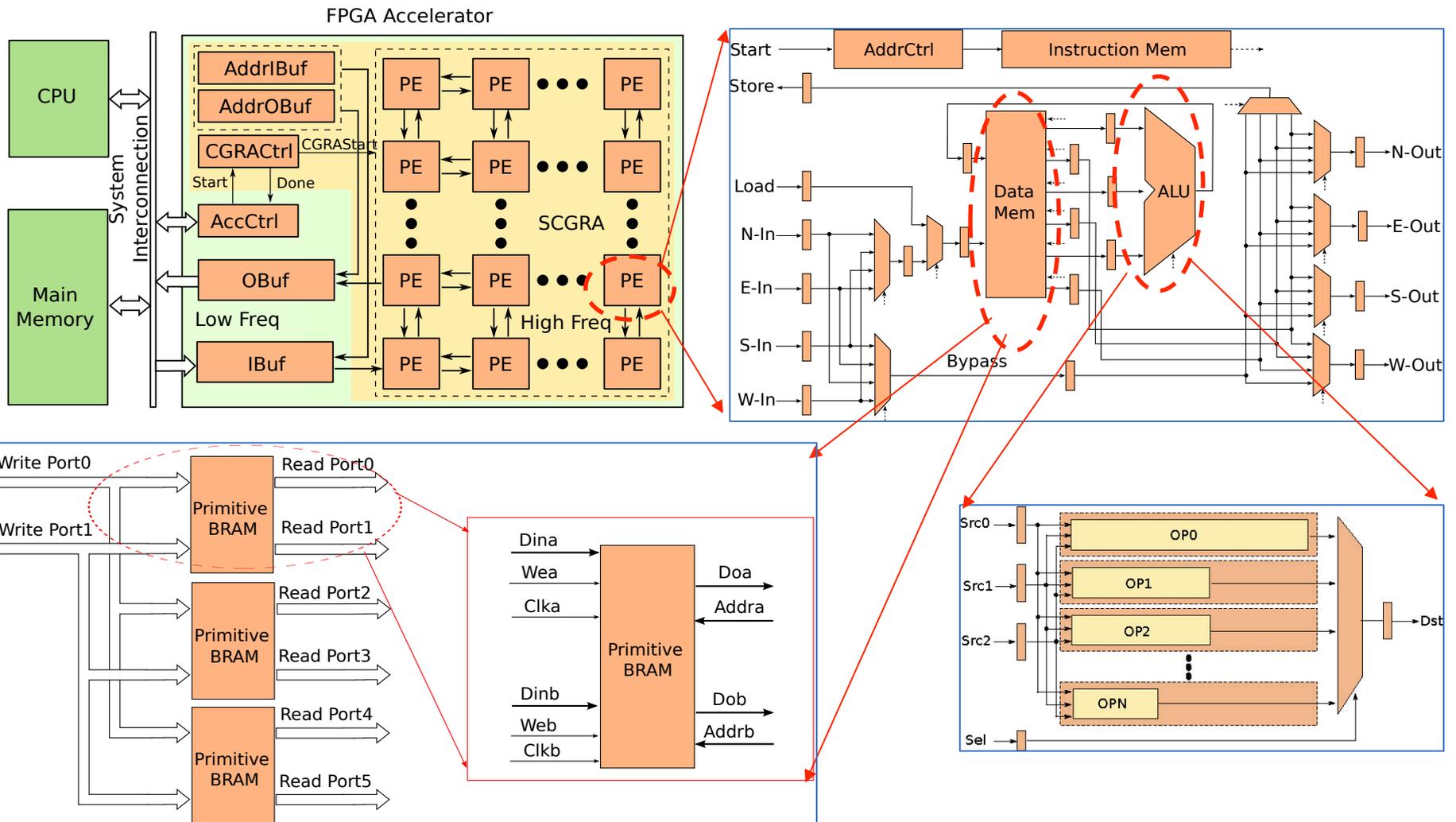


QuickDough Design Flow



→ Software compilation → Accel. generation → Accel. lib. update → Accel. library

QuickDough Soft CGRA Overlay



Highly pipelined, good scalability, simple and easy to extend



Loop Execution on the Accelerator

SW/HW Data transferred in groups

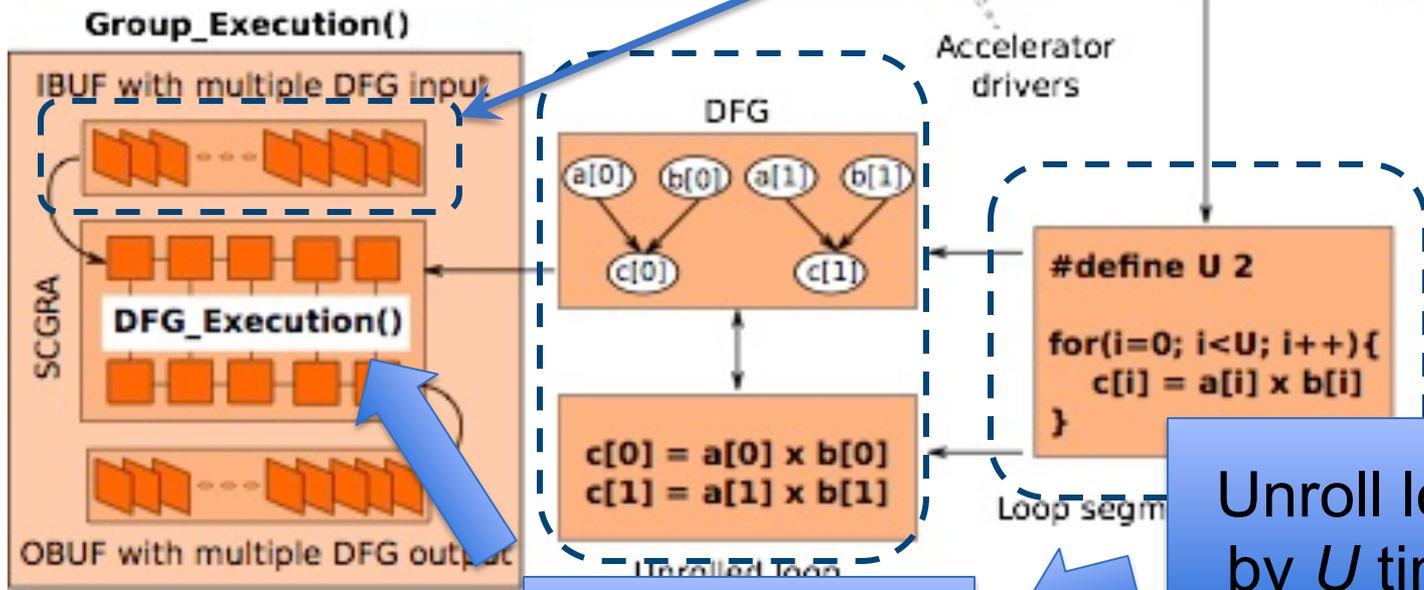
On-chip buffer holds G loop groups data

```

// ...
// Compute kernel
for(i=0; i<N; i++){
  c[i] = a[i] x b[i]
}
// ...

// Compute kernel
// Group Size: G
#define N 10000
#define G 10
for(i=0; i<N/G; i++){
  To_FPGA(a[G], b[G])
  Group_Execution()
  To_Main_Mem(c[G])
}

// Unrolling factor: 2
#define G 10
#define U 2
for(i=0; i<G/U; i++){
  DFG_Execution()
}
    
```

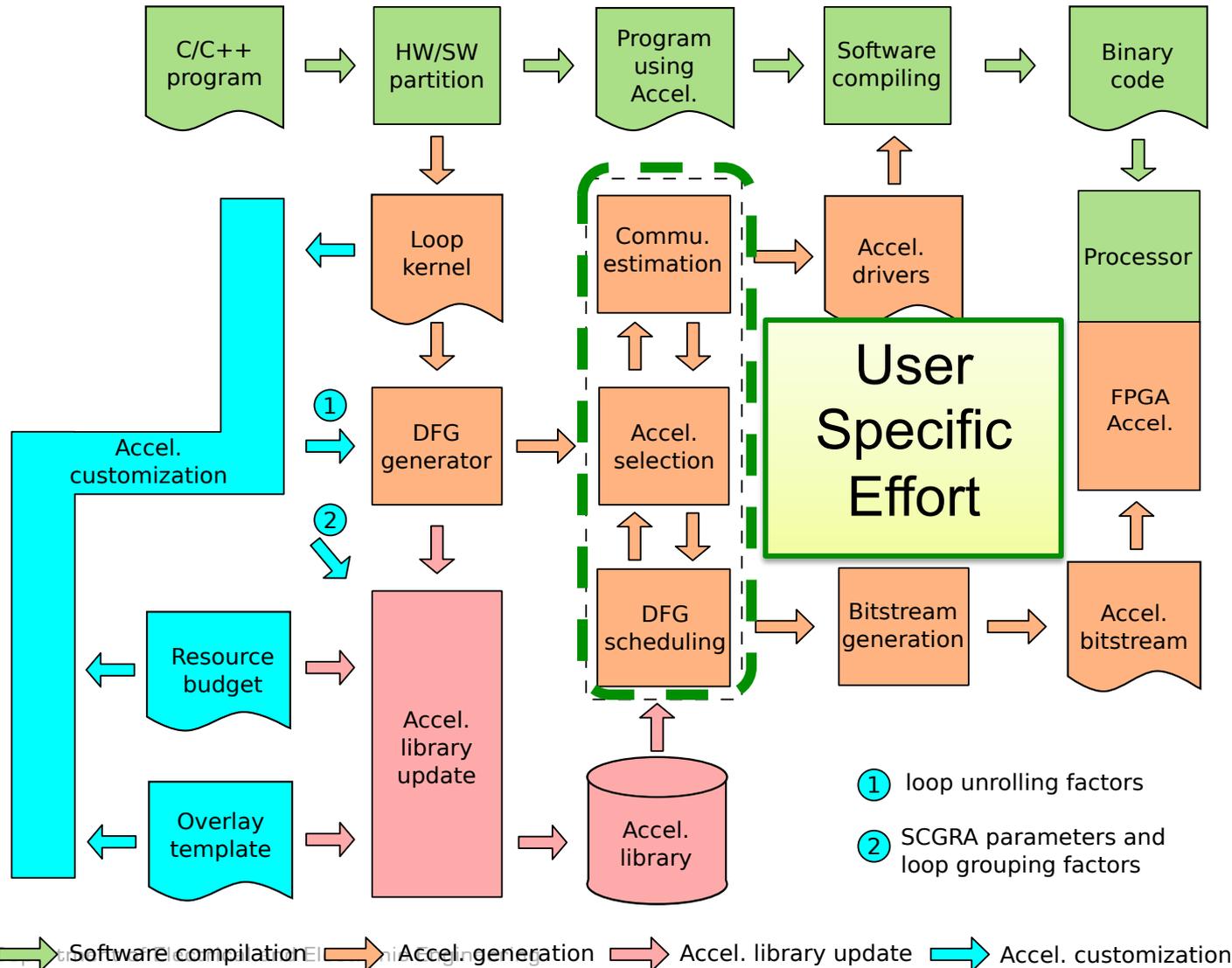


DFG from unrolled loop

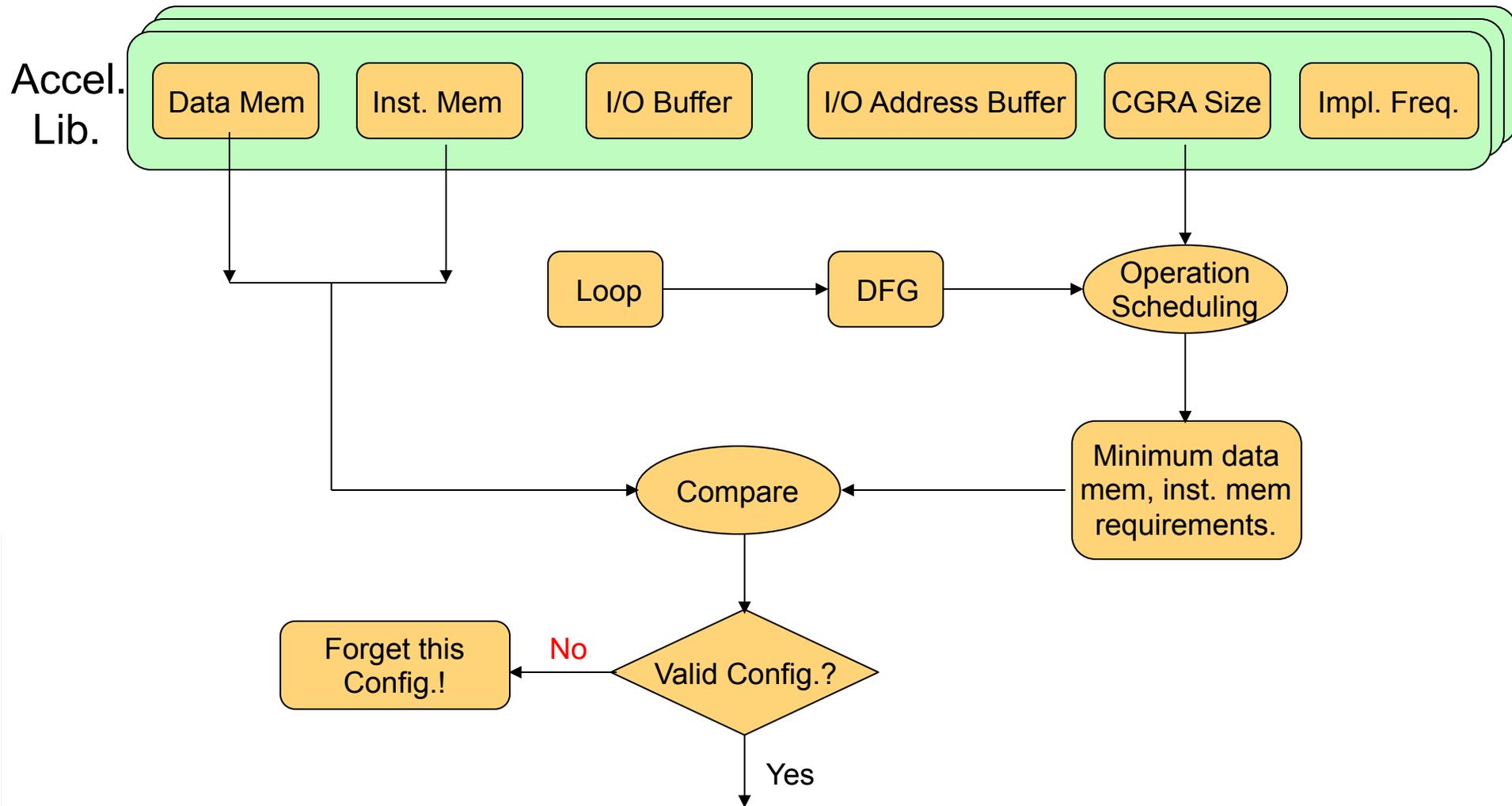
Unroll loop by U times



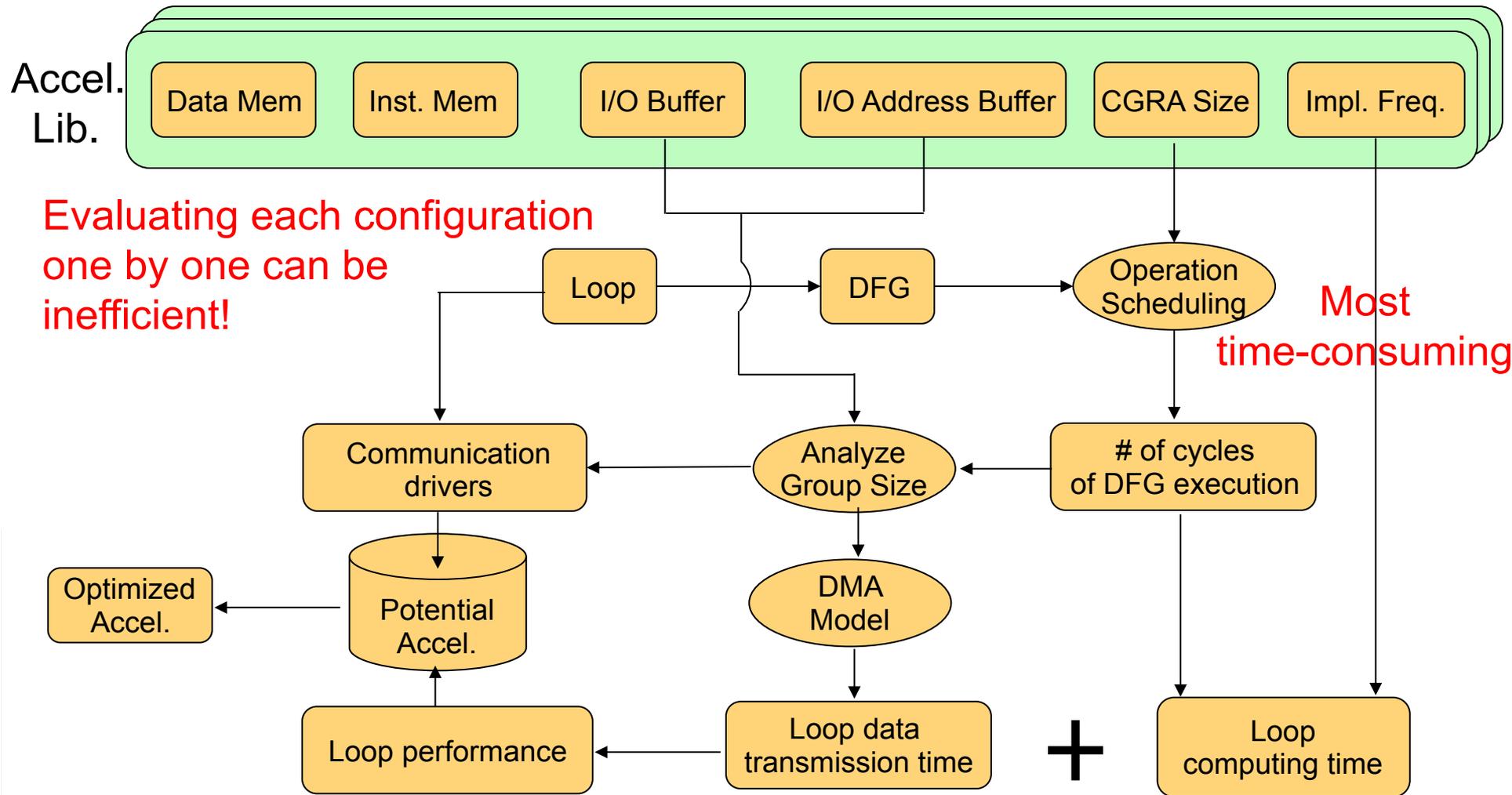
QuickDough Design Flow



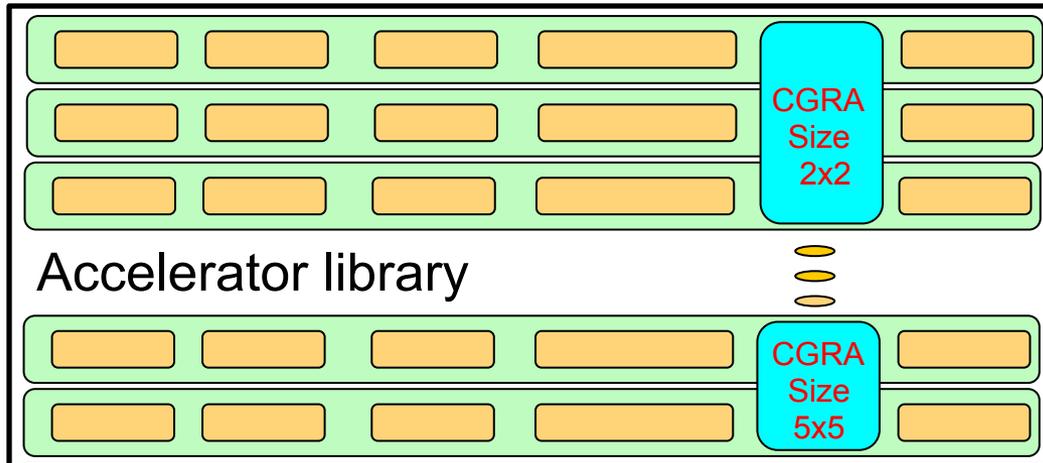
Accelerator Selection(1)



Accelerator Selection(2)



Accelerator Selection(3)

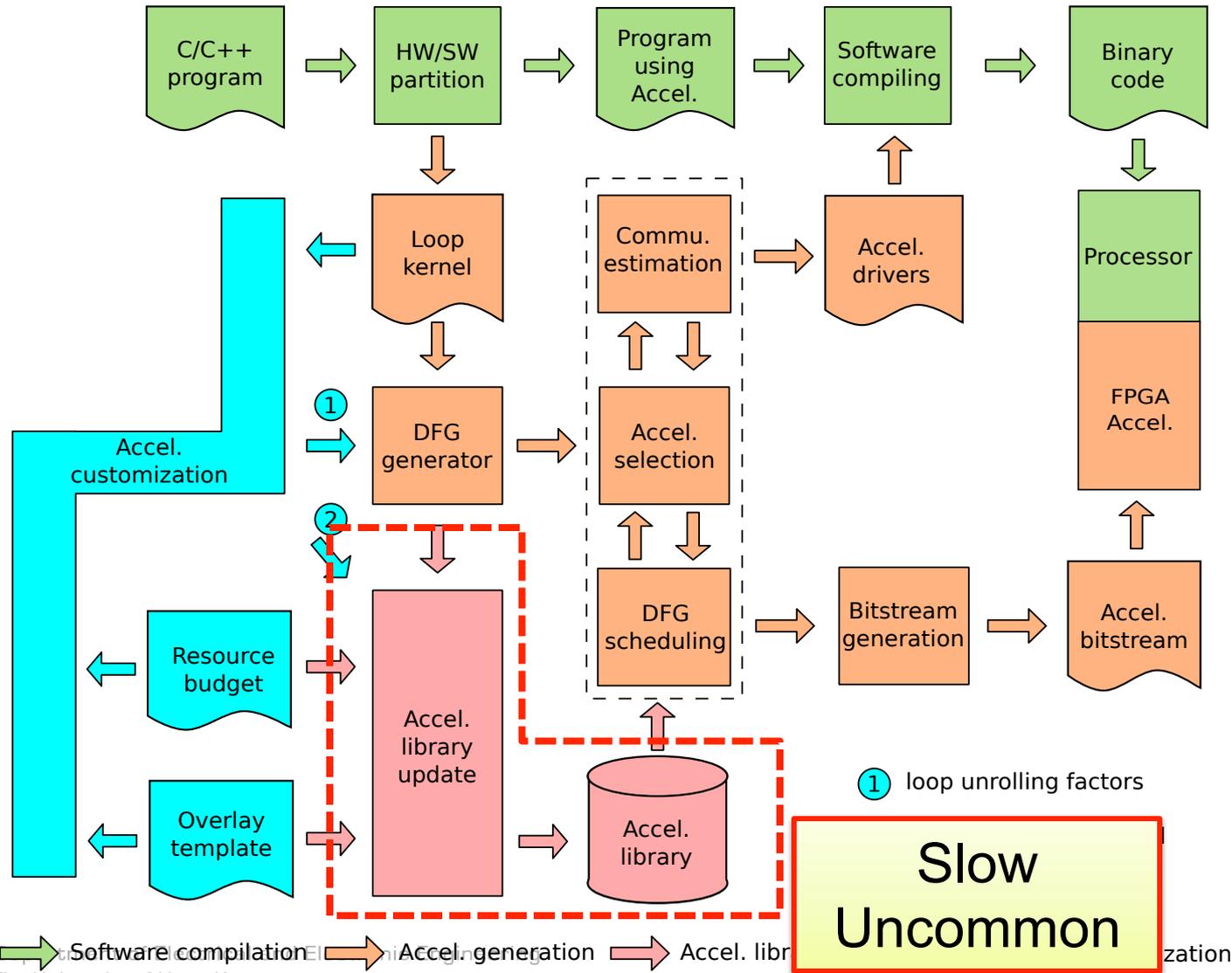


A single operation scheduling can be reused to evaluate a group of accelerator configurations.

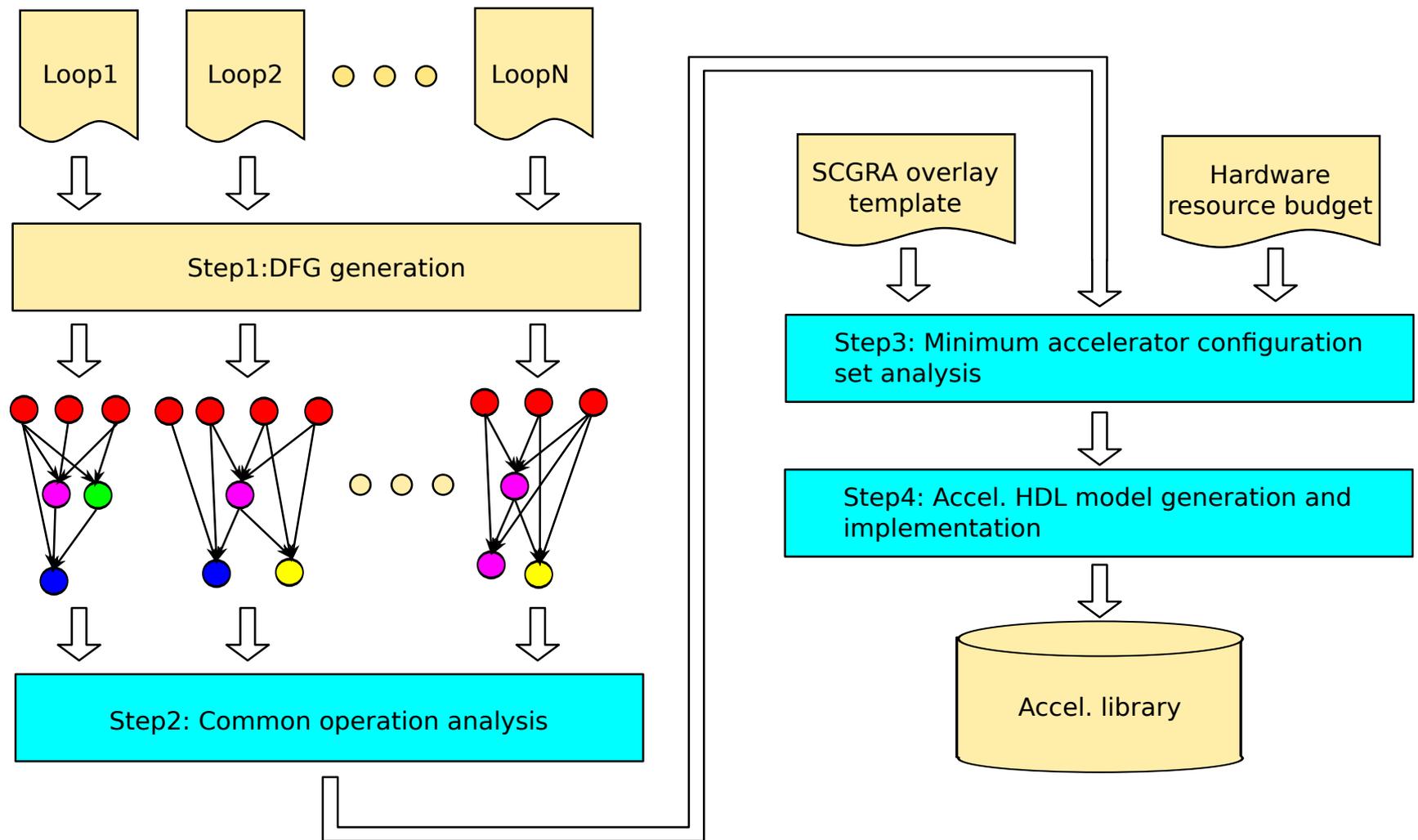
■ Selection Strategy

- -O0: accelerators with smallest CGRA size
- -O1: target three groups of accelerators with different SCGRA size (small, medium, large)
- -O2: full search

QuickDough Design Flow



Accelerator Library Pre-build





Evaluation

- Two Goals:

Compilation
Time

Resulting
System
Performance



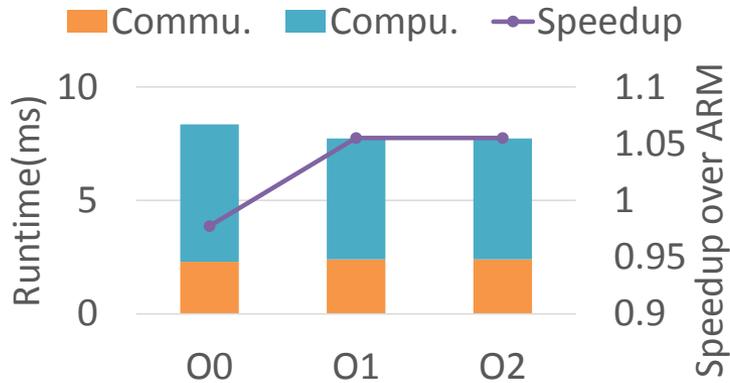
Benchmark

Benchmark	App Parameters	Loop kernel
MM (Matrix multiplication)	matrix size: 100100	100
FIR	# of input: 10000 # of Taps +1: 50	0
SE (Sobel edge detector)	# of vertical pixels:128 # of horizontal pixels: 128	
KM (Kmean)	# of nodes: 5000 # of centroids: 4 # of node dimension: 2	

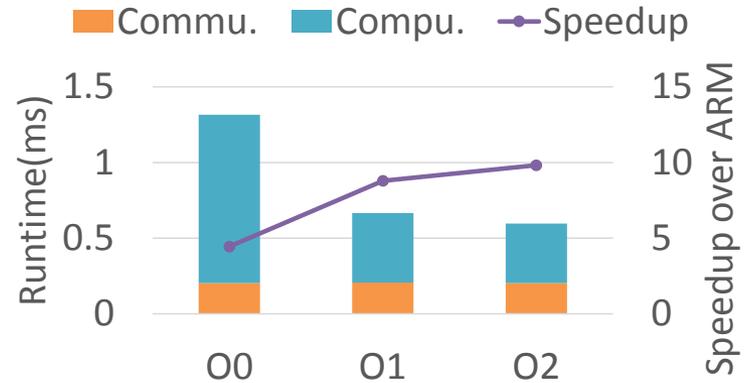
Benchmark	Unrolling	DFG Size
MM	1100	
FIR	5050	200
SE	163	
KM	2	



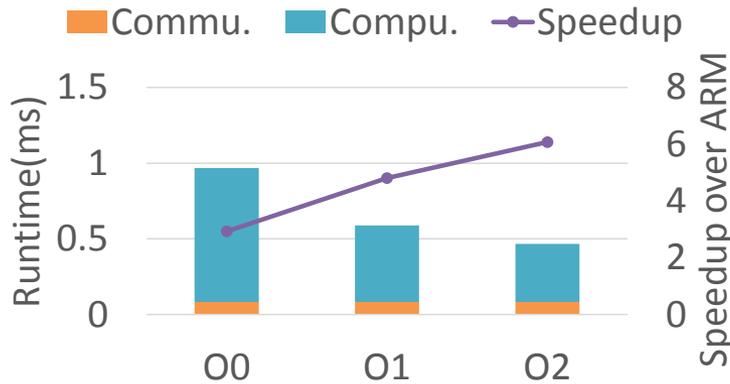
Accelerator Performance



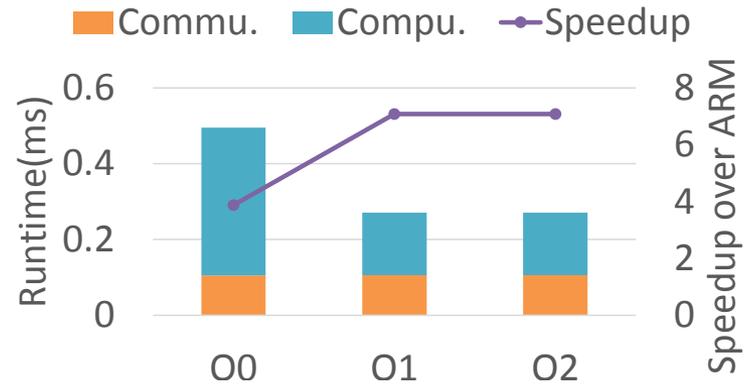
MM



FIR



SE

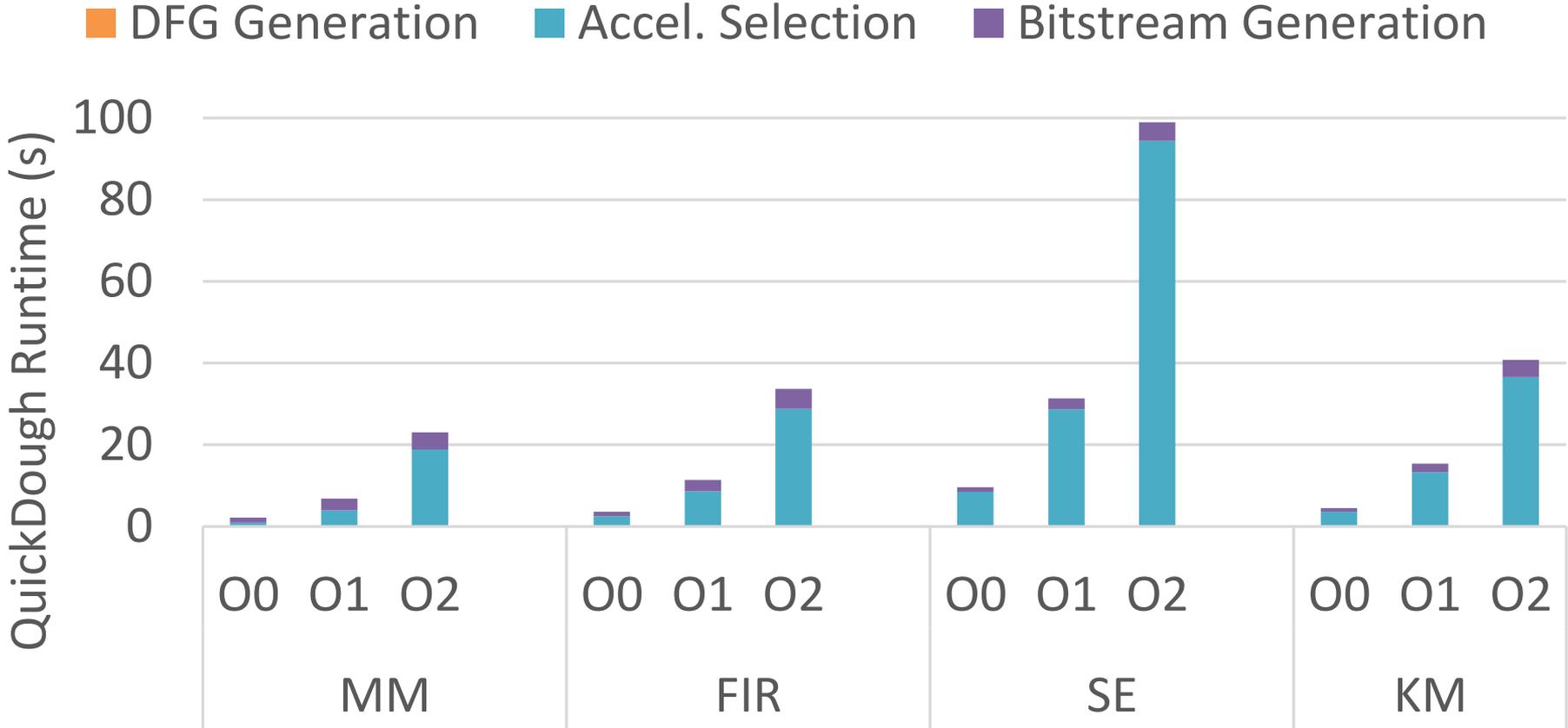


KM





Accelerator Generation Time



Resulting Accelerator Configurations

Opt. Option	Resulting Config	MM	FIR	SE	KM
O0	SCGRA size	2x2	2x2	2x2	2x2
	Inst. Mem Depth	4K	4K	4K	4K
	I/O Buffer Depth	4K	4K	4K	4K
	Grouping Factor	50x5x100	2500x50	128x64x3x3	1250x4x2
O1	SCGRA Size	3x3	3x3	3x3	5x5
	Inst. Mem Depth	2K	2K	4K	1K
	I/O Buffer Depth	2K	2K	1K	2K
	Grouping Factor	25x5x100	1250x50	64x32x3x3	1000x4x2
O2	SCGRA size	3x3	4x4	4x4	5x5
	Inst. Mem Depth	2K	1K	2K	1K
	I/O Buffer Depth	2K	8K	1K	2K
	Grouping Factor	25x5x100	5000x50	64x32x3x3	1000x4x2





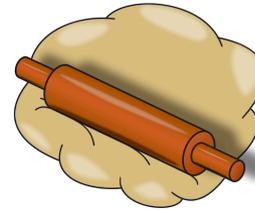
Conclusions

- QuickDough can generate HW/SW accelerator systems **rapidly** and with **good performance**.
- Overlay allows rapid compilation
- A overlay library assist rapid performance tuning
- Data I/O co-optimization with selected overlay



THANK YOU

Questions?



Happy baking...



**2ND INTERNATIONAL WORKSHOP ON OVERLAY ARCHITECTURES FOR
FPGAS (OLAF)**

February 21, 2016 - Monterey, CA, USA

[HOME](#) [NEWS](#) [SUBMISSION](#) [PROGRAM](#) [DATE & VENUE](#) [ORGANIZERS](#)

February 21, 2016
(Co-located at FPGA'16)
Monterey, CA
USA

WELCOME TO OLAF

[View](#) [Edit](#)

The second workshop on overlay architectures for FPGA (OLAF) will be co-located with FPGA'16. OLAF'16 will be set as a half day workshop to gather researchers in the community to exchange their knowledge and to explore ideas on the use of overlay architectures on FPGA devices.

OLAF was started in response to the growing interest in utilizing virtual coarse-grain architectures overlaying on top of fine-grained FPGA devices for sake of improved designer productivity, debugging

2nd International Workshop on Overlay Architecture (OLAF)

Co-located with FPGA'16

Deadline: **Jan 11, 2016**

<http://olaf.eecs.berkeley.edu>



Department of Electrical and Electronic Engineering
The University of Hong Kong