



ECE527 Fall 2025 Guest Lecture

StreamTensor: Make Tensors Stream in Dataflow Accelerators for LLMs

Hanchen Ye¹, Deming Chen^{1,2}

¹University of Illinois at Urbana-Champaign

²Inspirit IoT, Inc.

Outline



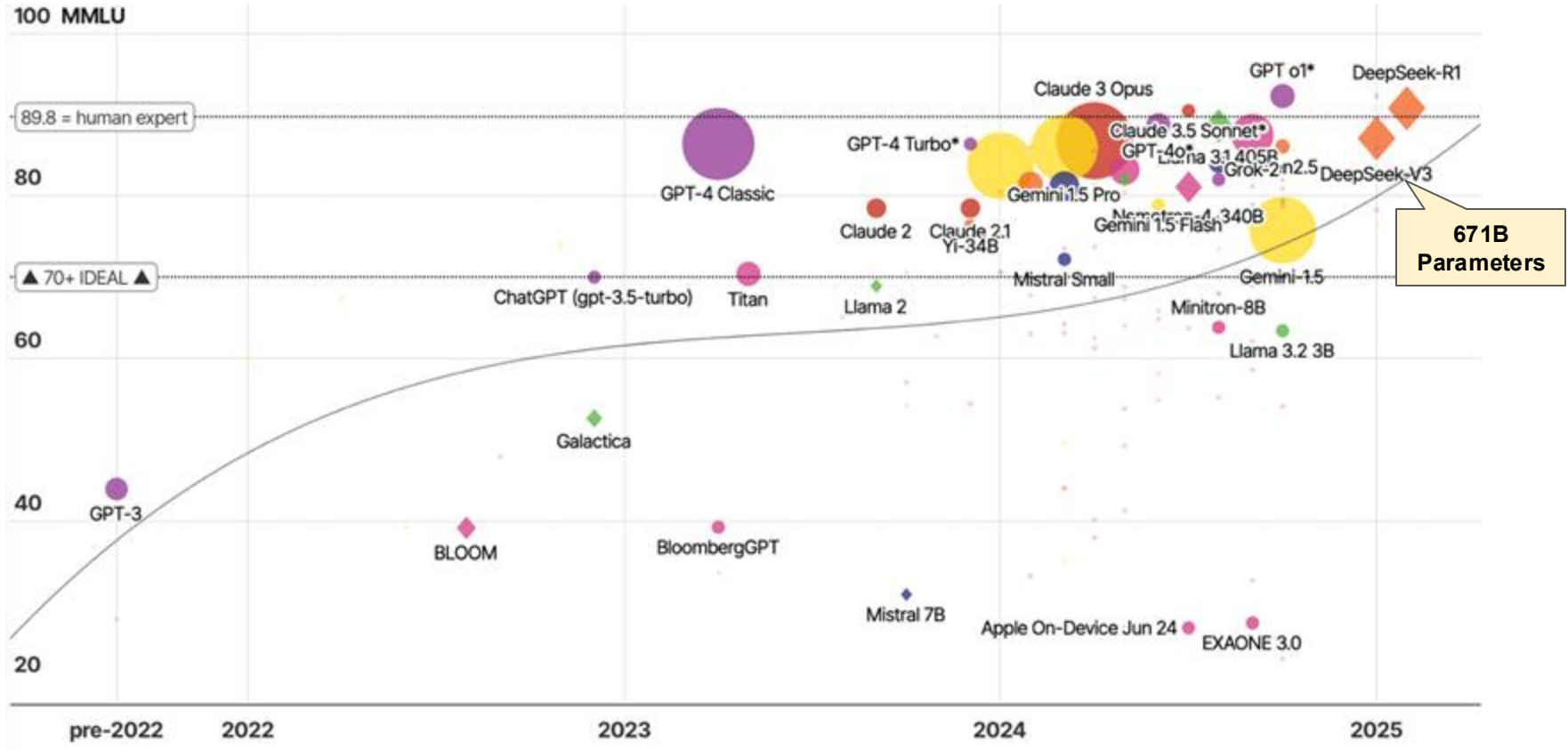
- Motivation
- StreamTensor Typing System
- StreamTensor Compilation Pipeline
- StreamTensor Design Spaces
- StreamTensor Results

StreamTensor Outline



- Motivation
- StreamTensor Typing System
- StreamTensor Compilation Pipeline
- StreamTensor Design Spaces
- StreamTensor Results

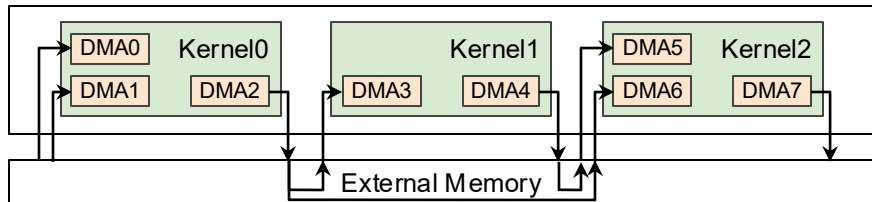
Parameter Size of LLMs Grows Fast



Source: Information is Beautiful

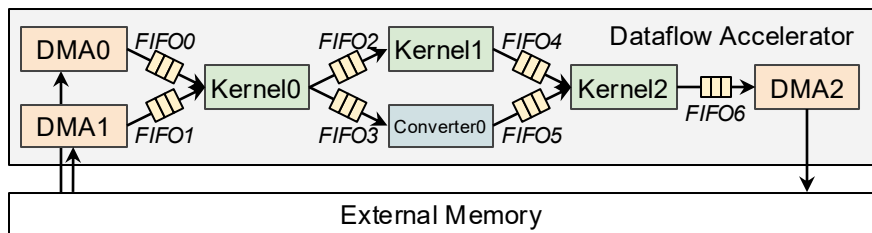
Parameters (Bn) ◇ open access MMLU = benchmark for measuring LLM capabilities

Motivation – Fight with the Memory Wall



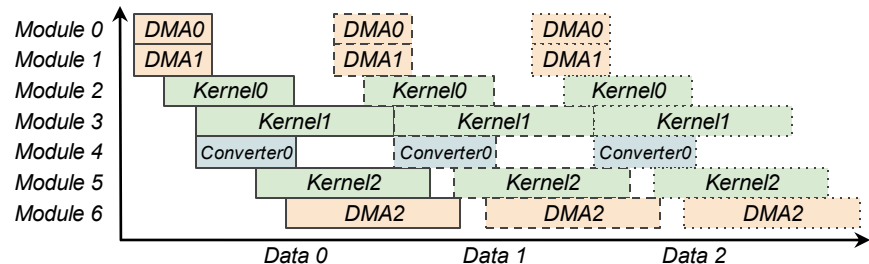
(a) A buffer-based dataflow accelerator

Stream-based Kernel Fusion



(b) A stream-based dataflow accelerator

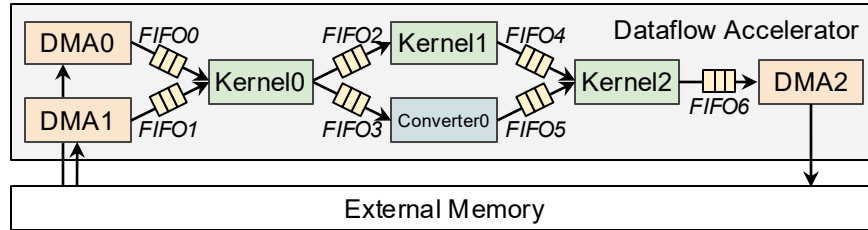
- Intermediate results are communicated through **on-chip ping-pong buffers**
- If on-chip memory resources are not enough, external memory is used
- Lead to frequent external memory access



(c) Schedule of the stream-based dataflow accelerator

- Reduce on-chip buffer utilization** ✓
- Reduce external memory access** ✓
- Improve overall latency & throughput** ✓

Pitfalls of Stream-based Dataflow Accelerator

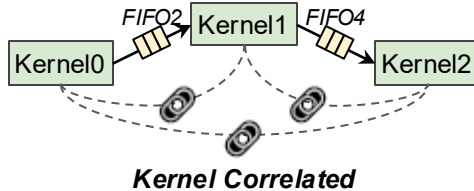


Pitfall 1 External Memory Access

- Best external memory layout matching stream pattern?
- Widen external memory interfaces to maximize bandwidth?

Pitfall 2

Inter-kernel Correlation

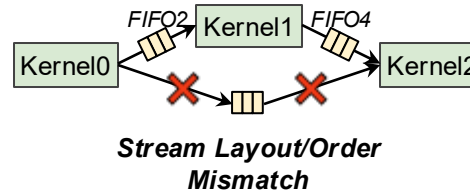


Kernel Correlated

- Kernel tiling & parallelization?
- Local buffer partition?
- Kernel loop permutation?

Pitfall 3

Kernel Fusion

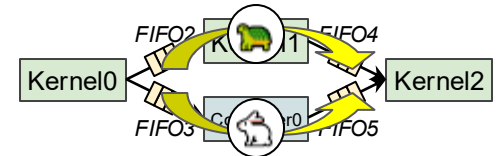


Stream Layout/Order Mismatch

- Possible to stream?
- Minimal stream buffer size?
- Global fusion strategy?

Pitfall 4

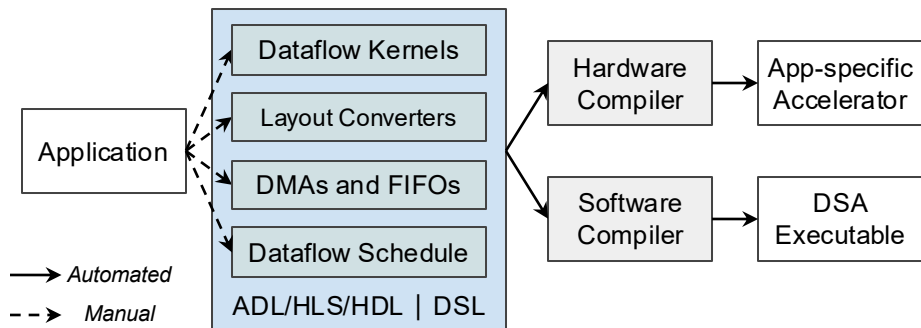
FIFO Sizing



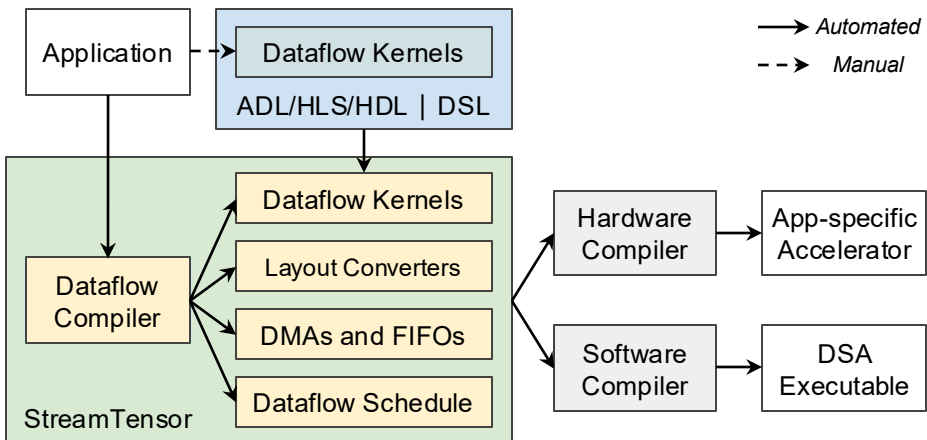
Stream Deadlock

- Latency mismatch on different paths between two kernels
- Minimal FIFO size to avoid deadlock or kernel stall?

Dataflow Accelerator Design Paradigm



↓ Paradigm Shift



- Manual dataflow kernels, layout converters, DMAs, and FIFOs design
- Difficult to comprehend the optimal solutions of the pitfalls
- Low design productivity

- Automate the generation of layout converters, DMAs, and FIFOs
- Resolve pitfalls through systematic design space exploration
- Support auto-tuned or hand-written kernel integration

StreamTensor Outline

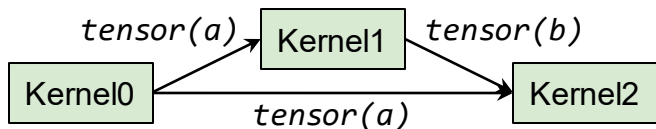


- Motivation
- StreamTensor Typing System
- StreamTensor Compilation Pipeline
- StreamTensor Design Spaces
- StreamTensor Results

Motivation of Iterative Tensor Type

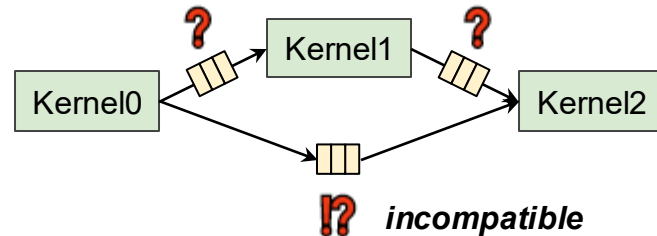


Tensor-level Graph

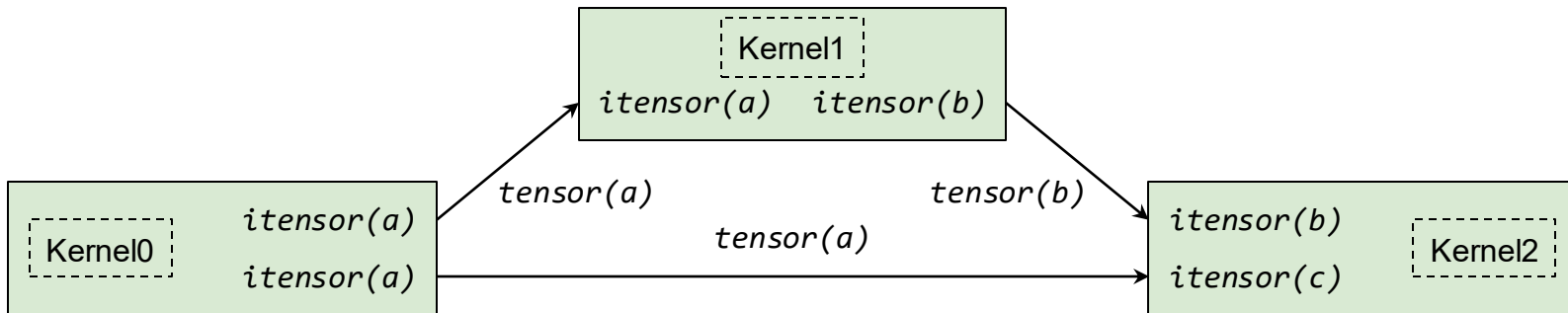


Enable Direct Streaming

Dataflow Accelerator



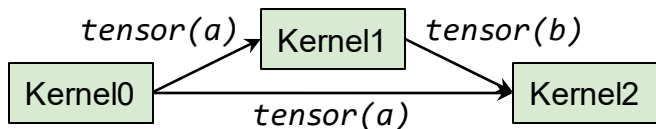
Iterative Tensor-level Graph



Motivation of Iterative Tensor Type (Cont.)

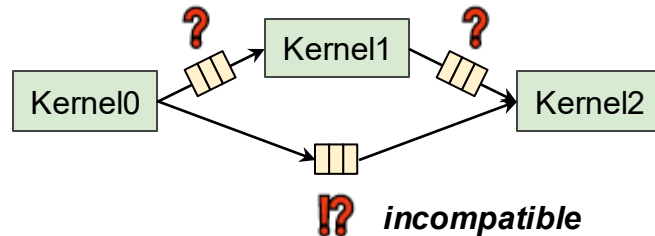


Tensor-level Graph



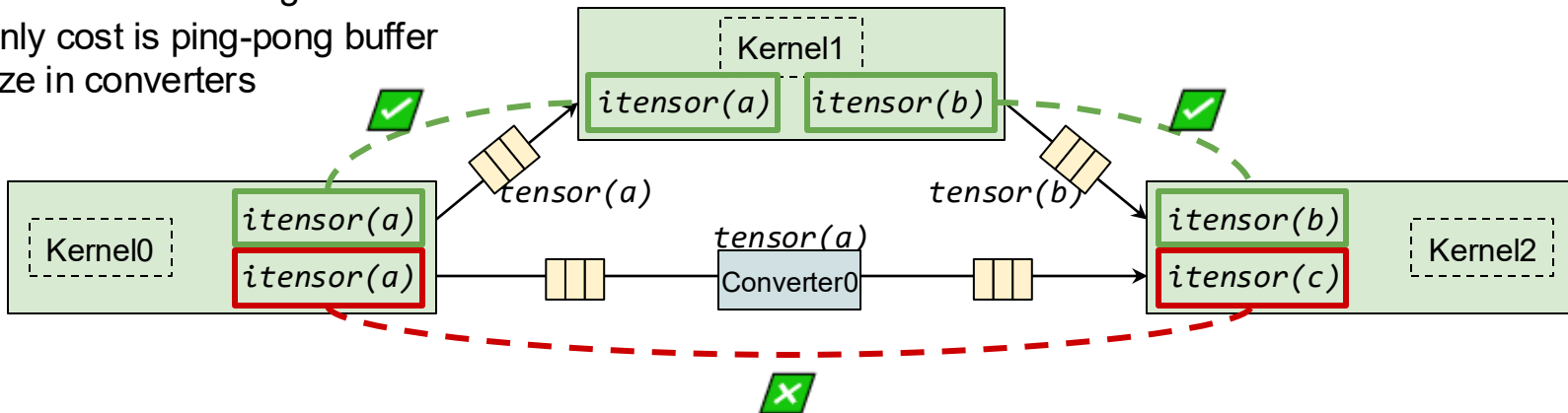
Enable Direct Streaming

Dataflow Accelerator

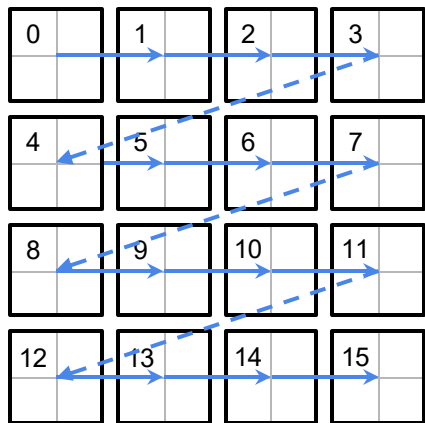


- By design, all kernels can be fused in a streaming manner
- Only cost is ping-pong buffer size in converters

Iterative Tensor-level Graph



Iterative Tensor Type



(a) itensor<2x2xf32,
iter_space: [4,4]*[2,2],
iter_map: (d0,d1)->(d0,d1)>

2x2xf32 = Element Shape & Type
iter_space = Iteration Space
 = [Tripcounts]*[Steps]
iter_map = Iteration Affine Map

Iteration Space
[4,4]*[2,2]

[0,0]

[0,2]

[0,4]

[0,6]

[2,0]

[2,2]

...

Data Space

[0,0]

[0,2]


[0,4]

[0,6]

[2,0]

[2,2]

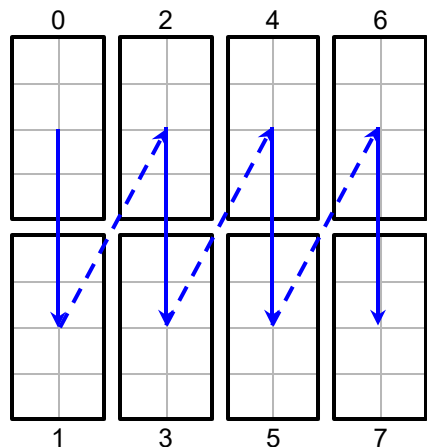
...


Affine Mapping
(d0,d1) ->(d0,d1)

Element Shape
2x2

Same itensor type means same stream layout/order

Iterative Tensor Type (Cont.)



(b) `itensor<4x2xf32,`
`iter_space: [4,2]*[2,4],`
`iter_map: (d0,d1)->(d1,d0)>`

`4x2xf32` = Element Shape & Type
`iter_space` = Iteration Space
 = [Tripcounts]*[Steps]
`iter_map` = Iteration Affine Map

Iteration Space
 $[4, 2] * [2, 4]$

$[0, 0]$

$[0, 4]$


$[2, 0]$

$[2, 4]$

$[4, 0]$

$[4, 4]$

...


Affine Mapping
 $(d_0, d_1) \rightarrow (d_1, d_0)$

Element Shape
 4×2

Data Space

$[0, 0]$

$[4, 0]$

$[0, 2]$

$[4, 2]$

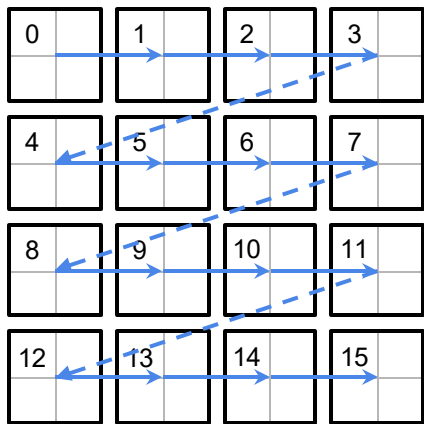
$[0, 4]$

$[4, 4]$

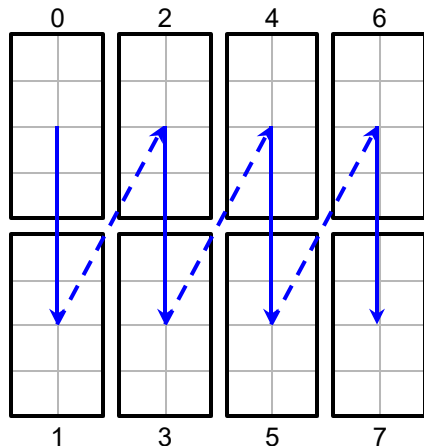
...

Transposed stream layout/order

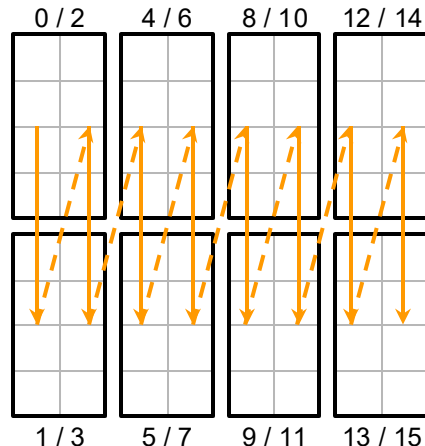
Iterative Tensor Type (Cont.)



(a) $\text{itensor}\langle 2 \times 2 \times f32, \text{iter_space: } [4, 4] * [2, 2], \text{iter_map: } (d_0, d_1) \rightarrow (d_0, d_1) \rangle$

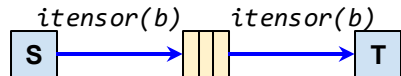


(b) $\text{itensor}\langle 4 \times 2 \times f32, \text{iter_space: } [4, 2] * [2, 4], \text{iter_map: } (d_0, d_1) \rightarrow (d_1, d_0) \rangle$

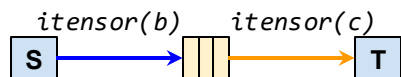


(c) $\text{itensor}\langle 4 \times 2 \times f32, \text{iter_space: } [4, 2, 2] * [2, 1, 4], \text{iter_map: } (d_0, d_1, d_2) \rightarrow (d_2, d_0) \rangle$

tensor $\langle 8 \times 8 \times f32 \rangle$

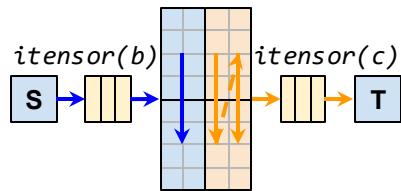


Case1 ✓



Case2 ✗

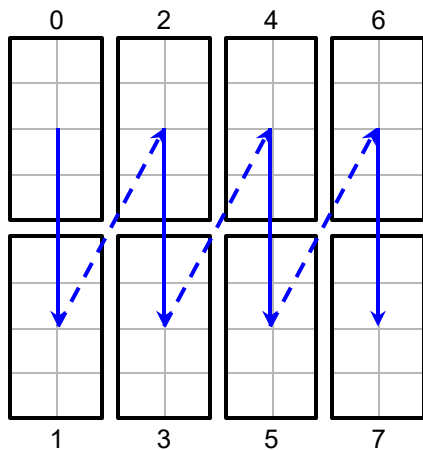
Insert Buffer



Case2 w/ ping-pong buffer ✓

The minimal buffer size is inferred from itensor types, which are used as “cost” during kernel fusion

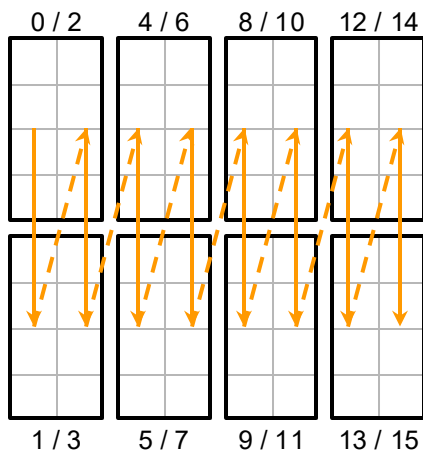
Infer Buffer Shape from Iterative Tensor Types



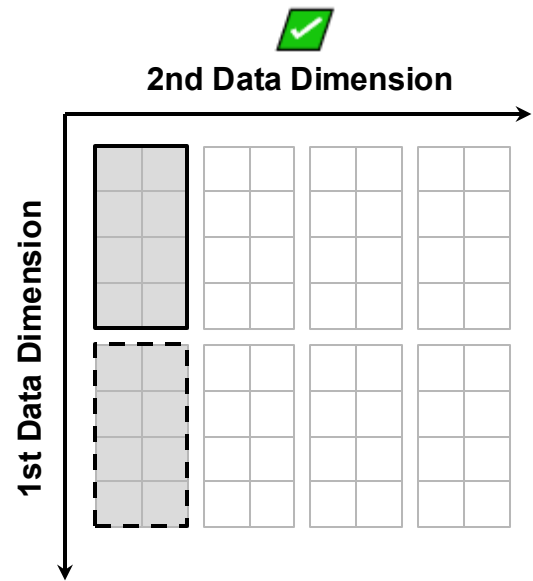
(b) itensor<4x2xf32,
iter_space: [4,2]*[2,4],
iter_map: (d0,d1)->(d1,d0)>

Element Shape

Iteration Dimension



(c) itensor<4x2xf32,
iter_space: [4,2,2]*[2,1,4],
iter_map: (d0,d1,d2)->(d2,d0)>



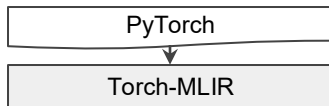
*Traverse each data dimension
from little-endian*

StreamTensor Outline

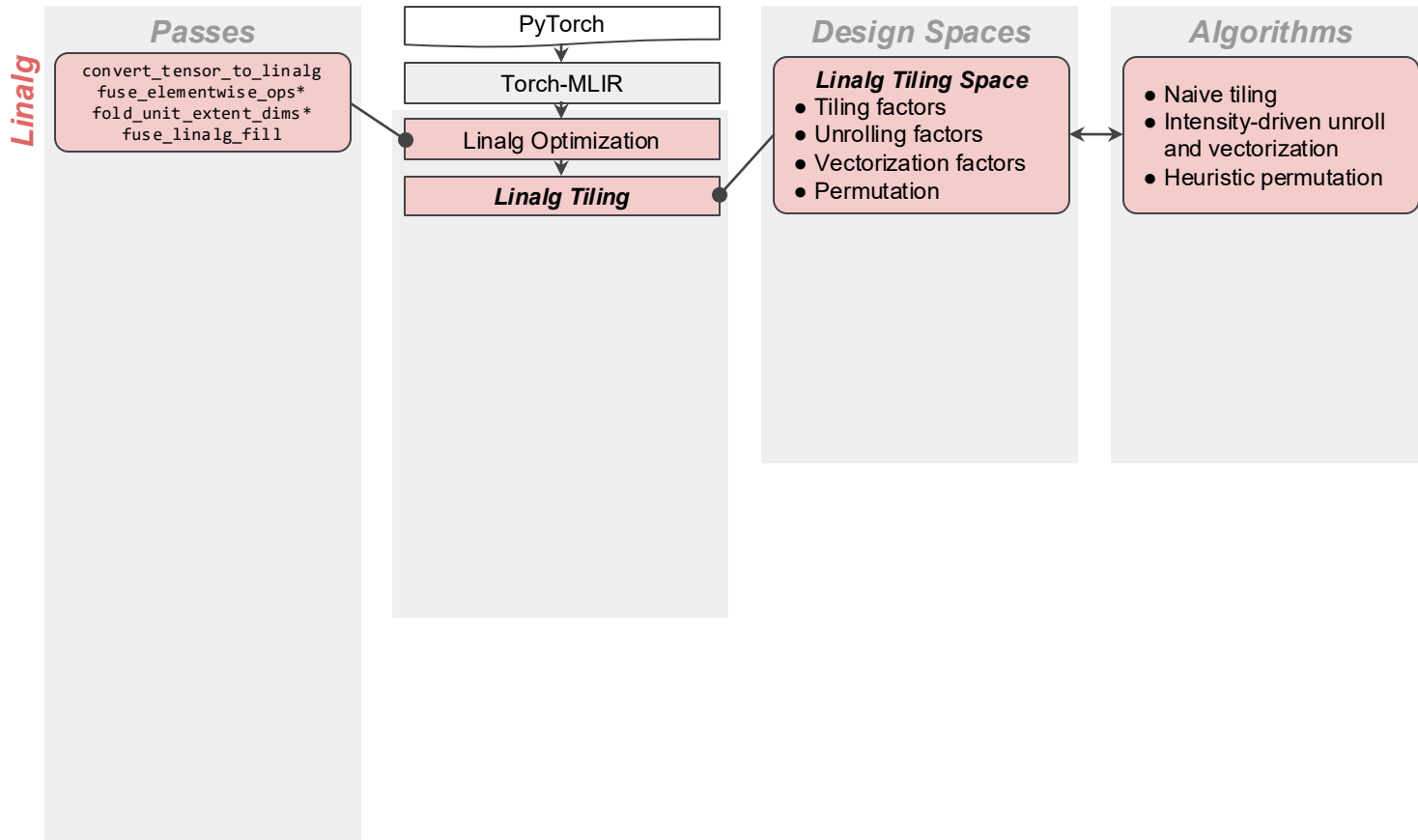


- Motivation
- StreamTensor Typing System
- **StreamTensor Compilation Pipeline**
- StreamTensor Design Spaces
- StreamTensor Results

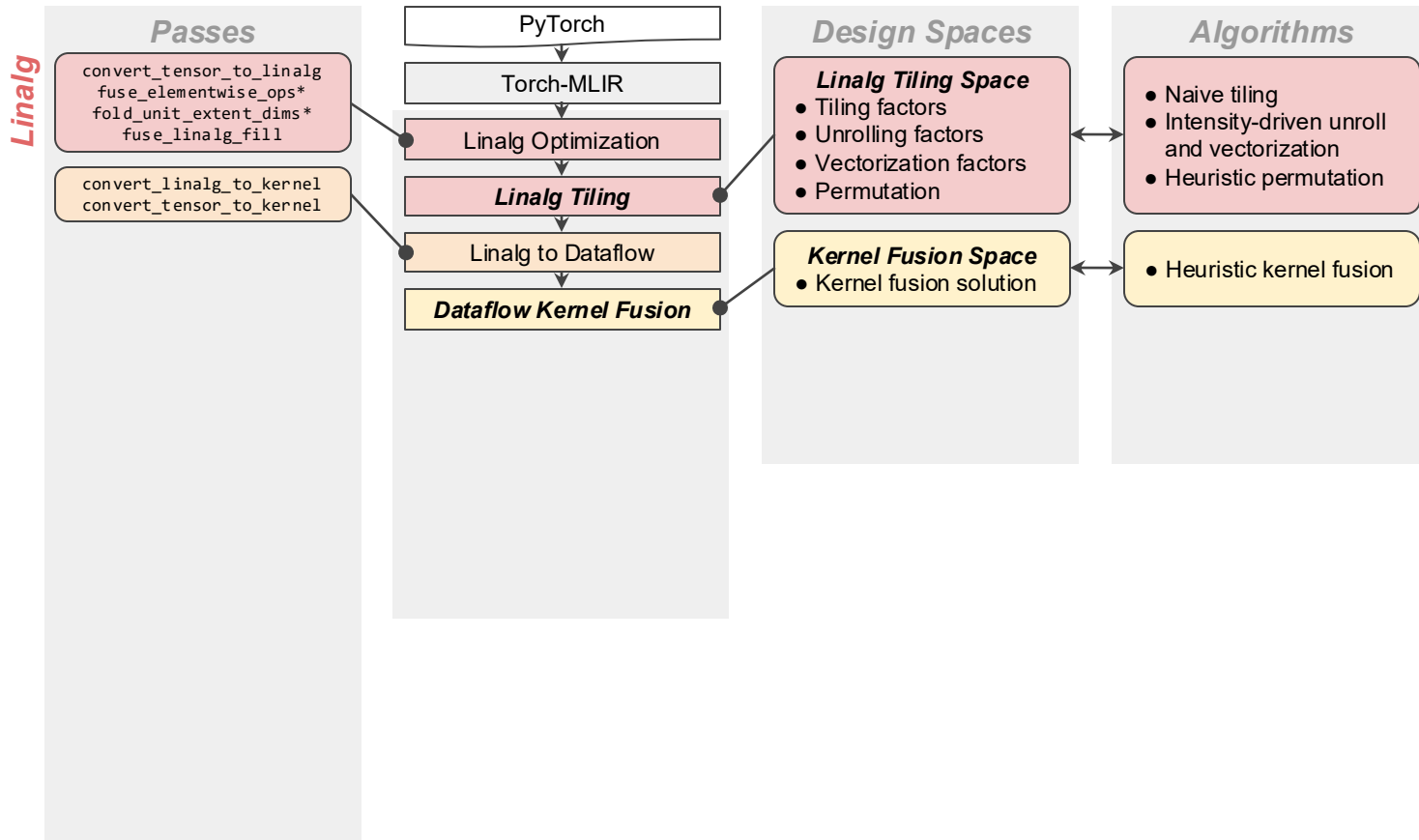
StreamTensor Framework Overview



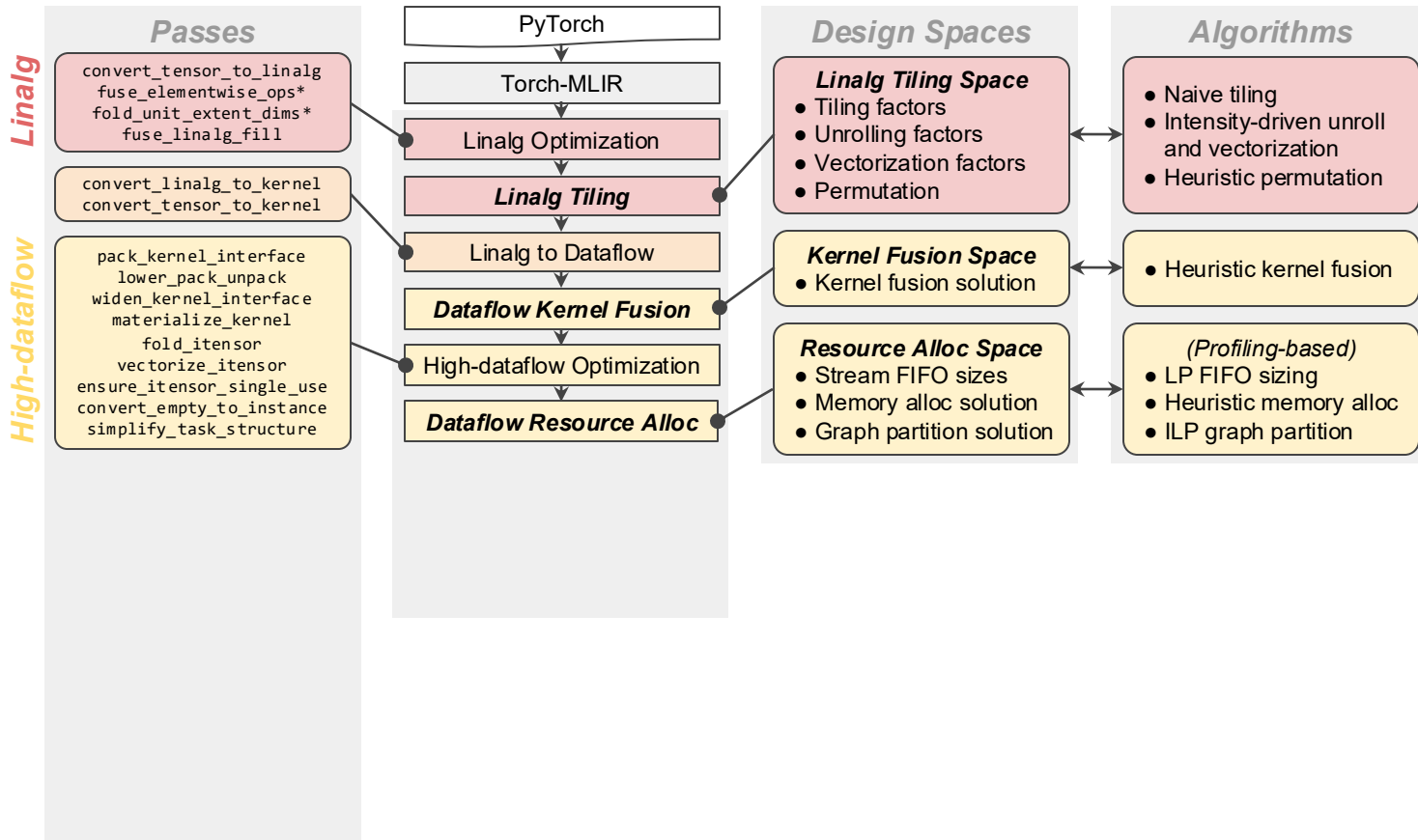
StreamTensor Framework Overview



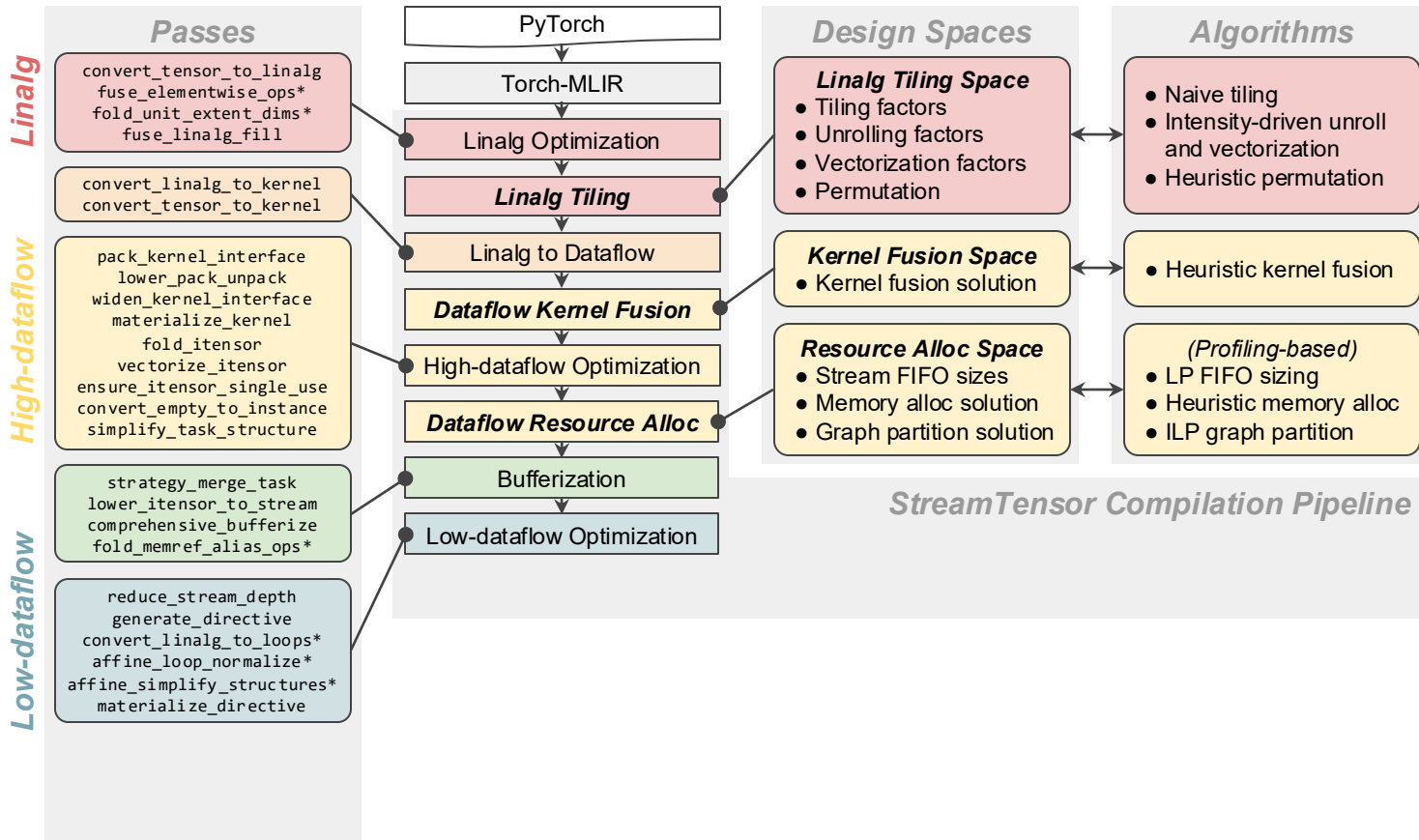
StreamTensor Framework Overview



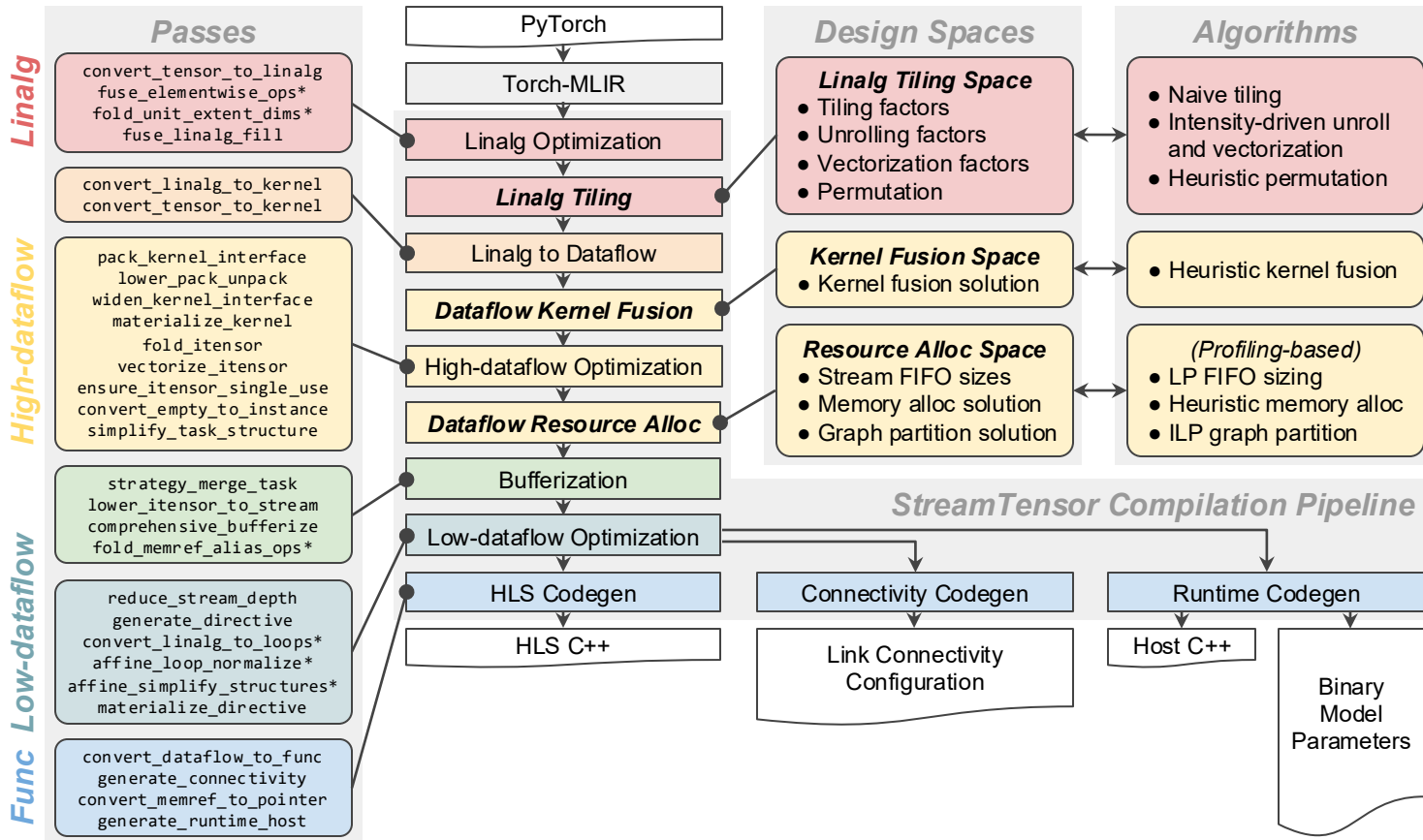
StreamTensor Framework Overview



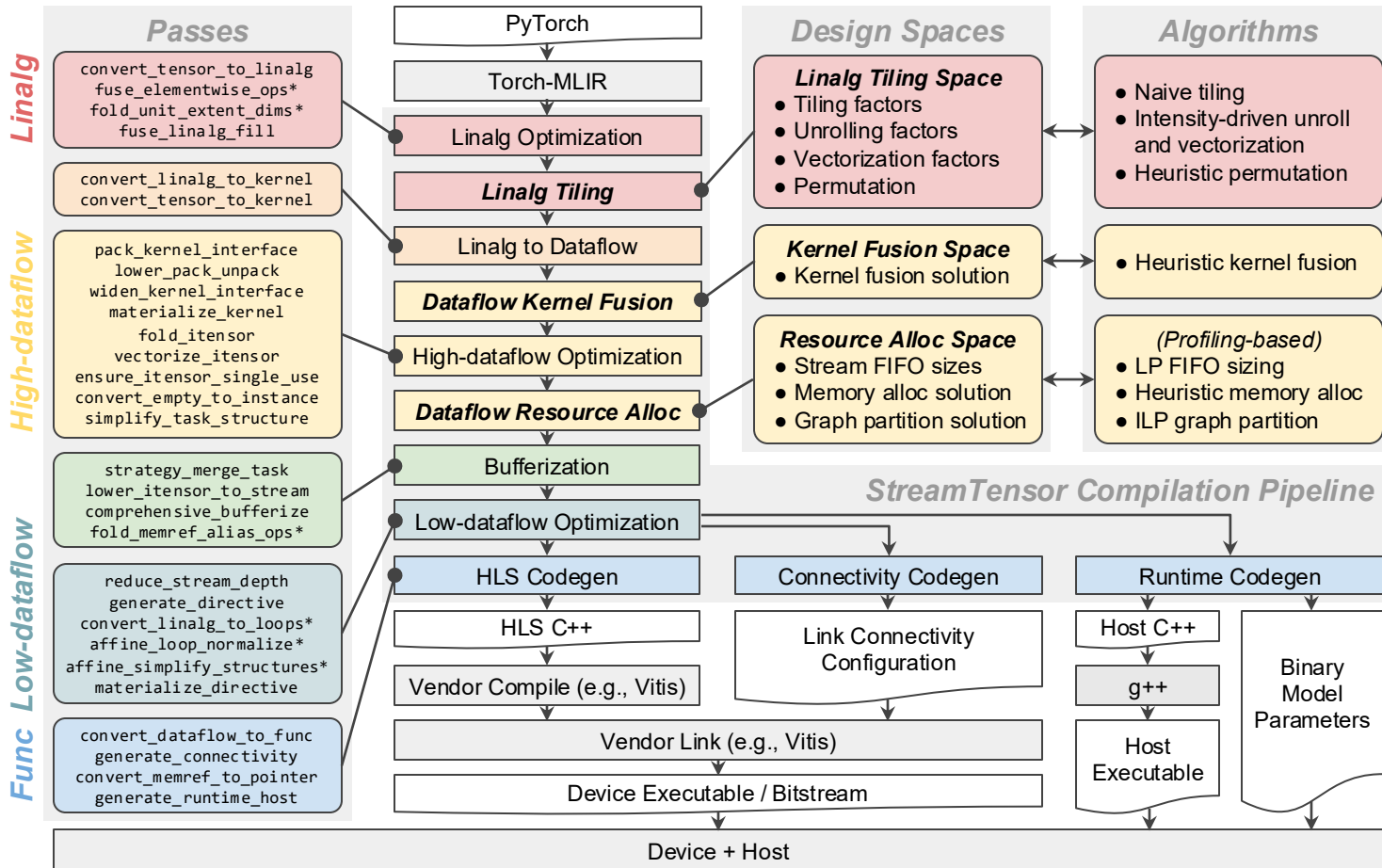
StreamTensor Framework Overview



StreamTensor Framework Overview



StreamTensor Framework Overview

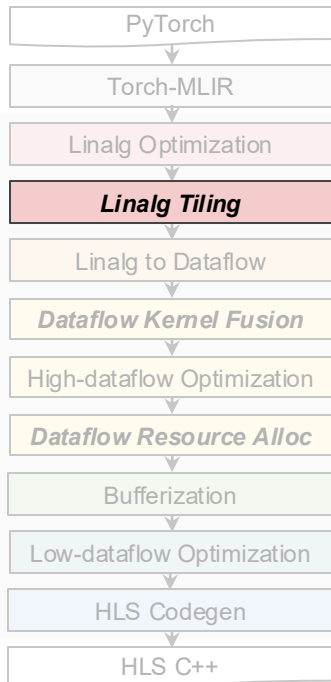


StreamTensor Outline



- Motivation
- StreamTensor Typing System
- StreamTensor Compilation Pipeline
- **StreamTensor Design Spaces**
- StreamTensor Results

Linalg Tiling Space



```

1 float A[32][16];
2 NODE0_I: for (int i=0; i<32; i++)
3   NODE0_K: for (int k=0; k<16; k++)
4     A[i][k] = ...; // Load array A.
5
6 float B[16][16];
7 NODE1_K: for (int k=0; k<16; k++)
8   NODE1_J: for (int j=0; j<16; j++)
9     B[k][j] = ...; // Load array B.
10
11 float C[16][16];
12 NODE2_I: for (int i=0; i<16; i++)
13   NODE2_J: for (int j=0; j<16; j++)
14     NODE2_K: for (int k=0; k<16; k++)
15       C[i][j] = A[i*2][k] * B[k][j];
  
```

HIDA IA+CA Algorithm

Node	Intensity	Parallel Factor		Loop Unroll Factors			
		w/o IA	w/ IA	IA+CA	IA	CA	Naive
Node0	512	32	4	[4, 1]	[2, 2]	[8, 4]	[4, 8]
Node1	256	32	2	[1, 2]	[1, 2]	[4, 8]	[4, 8]
Node2	4,096	32	32	[4, 8, 1]	[4, 8, 1]	[4, 8, 1]	[4, 8, 1]

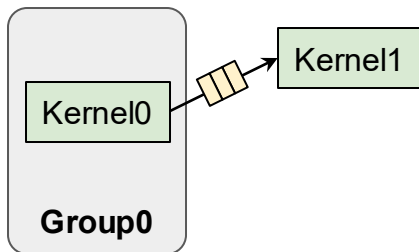
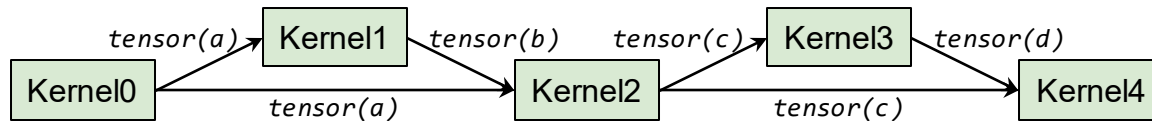
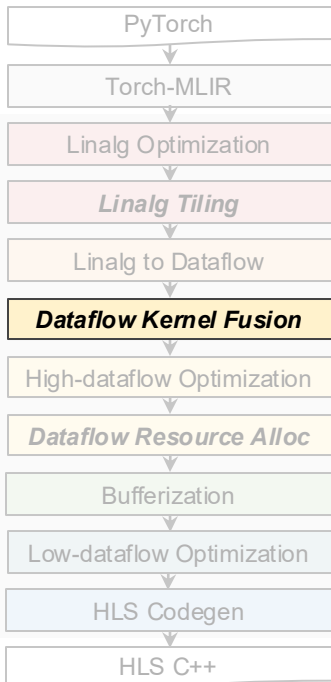
*Intensity-aware (IA)
Connectedness-aware (CA)
HIDA DSE*

*Naive
ScaleHLS
DSE*

Array	Array Partition Factors				Bank Number				
	IA+CA	IA	CA	Naive	IA+CA	IA	CA	Naive	
A	[8, 1]	[8, 2]	[8, 4]	[8, 8]	8	16	32	64	8x
B	[1, 8]	[2, 8]	[4, 8]	[8, 8]	8	16	32	64	8x
C	[4, 8]	[4, 8]	[4, 8]	[4, 8]	32	32	32	32	1x

The design space is exposed at Python level, open for auto-tuning or other optimization algorithm

Kernel Fusion Space

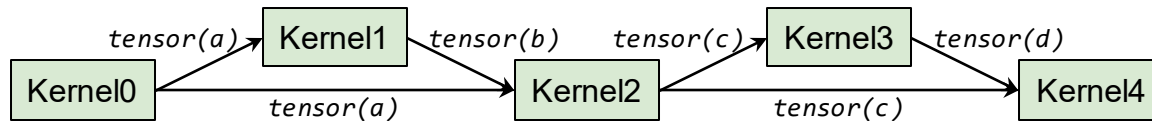
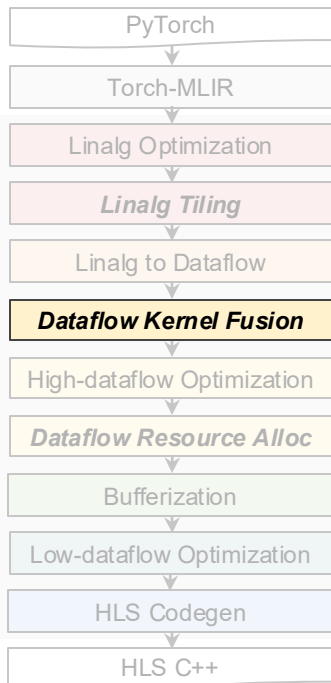


*Topologica
l Order
Traversal*

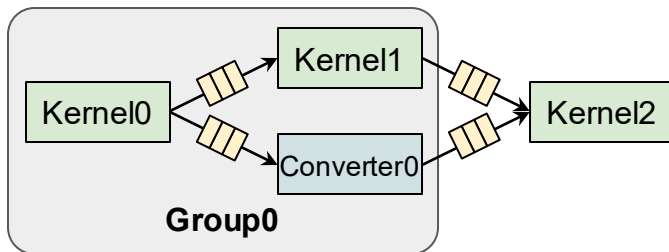
*Group0 On-
chip
Memory
Available*

	Group0	Group1
Kernels	Kernel0	
Cost	Local _{Kernel0}	

Kernel Fusion Space (Cont.)



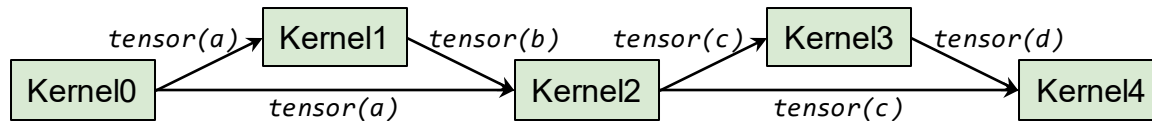
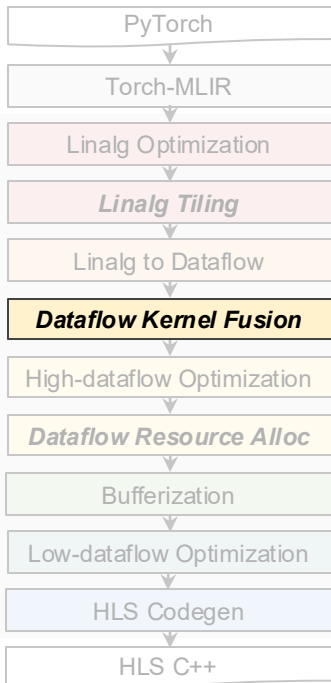
*Topologica
I Order
Traversal*



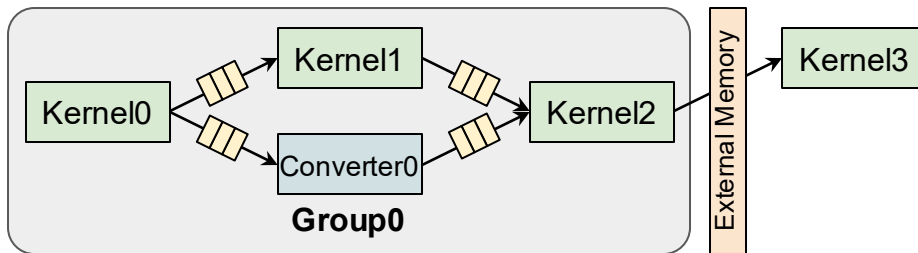
*Group0 On-
chip
Memory
Available*

	Group0		Group1
Kernels	Kernel0	Kernel1	
Cost	Local _{Kernel0}	Local _{Kernel1}	

Kernel Fusion Space (Cont.)



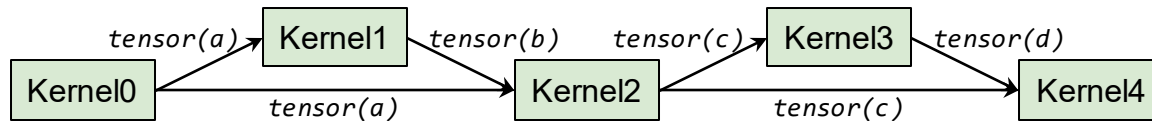
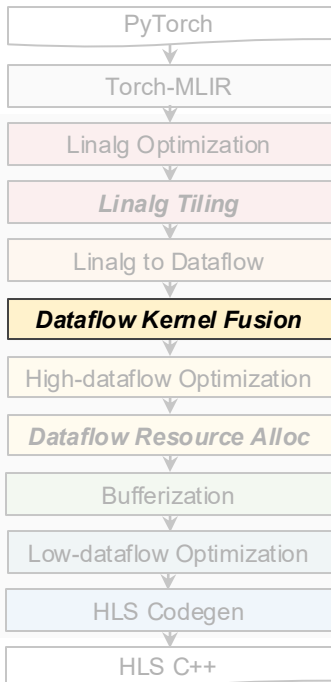
*Topologica
I Order
Traversal*



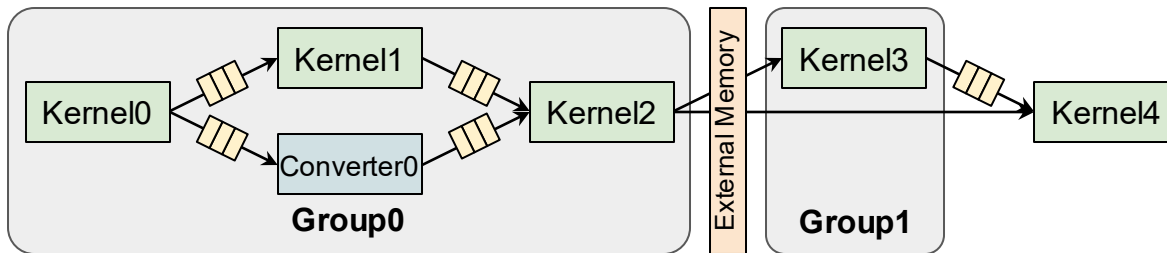
*Group0
On-chip
Memory
Full*

	Group0		Group1
Kernels	Kernel0 Kernel2	Kernel1	
Cost	Local _{Kernel0} Converter0	Local _{Kernel1} Local _{Kernel2}	

Kernel Fusion Space (Cont.)



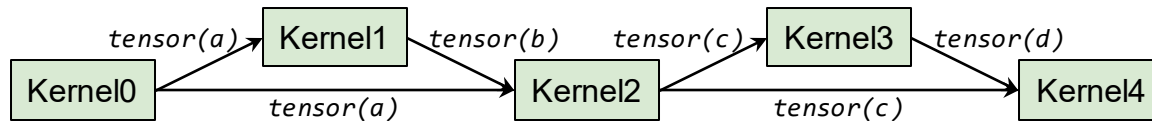
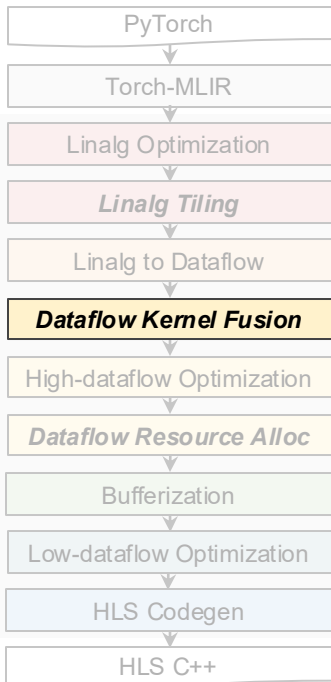
*Topologica
l Order
Traversal*



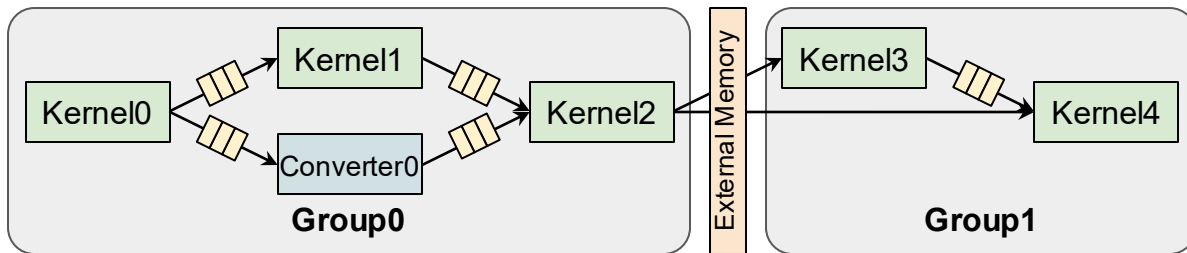
*Select the
Nearest
Predecessor*

	Group0		Group1
Kernels	Kernel0 Kernel2	Kernel1	Kernel3
Cost	Local _{Kernel0} Converter0	Local _{Kernel1} Local _{Kernel2}	Local _{Kernel3}

Kernel Fusion Space (Cont.)



*Topologica
I Order
Traversal*



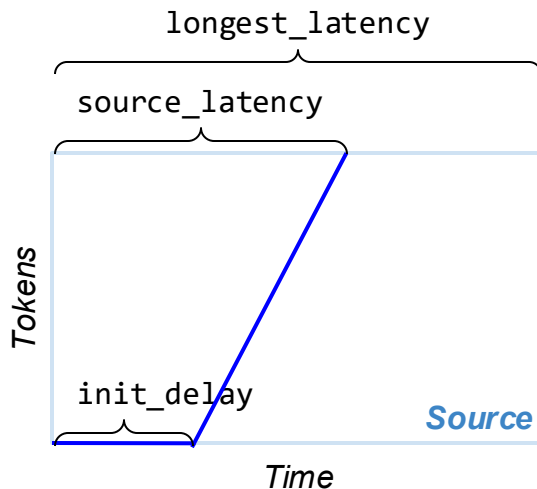
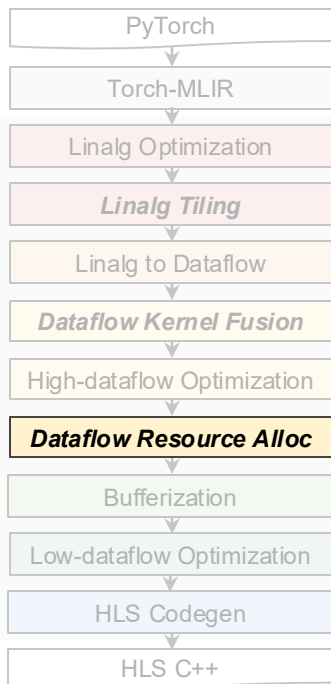
*Intra-group:
Streaming*

*Inter-group:
External
Memory*

	Group0		Group1
Kernels	Kernel0 Kernel2	Kernel1 Local _{Kernel2}	Kernel3 Kernel4
Cost	Local _{Kernel0} Converter0	Local _{Kernel1} Local _{Kernel2}	Local _{Kernel3} Local _{Kernel4}

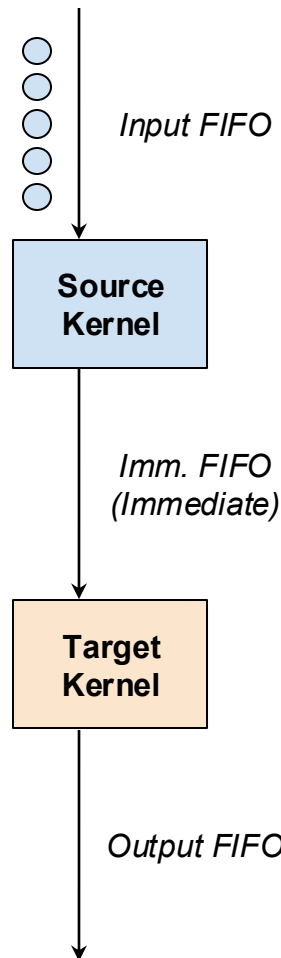
*Each Group
I
One Kernel
I
One Bitstream
(if on FPGA)*

Resource Allocation Space - FIFO Sizing

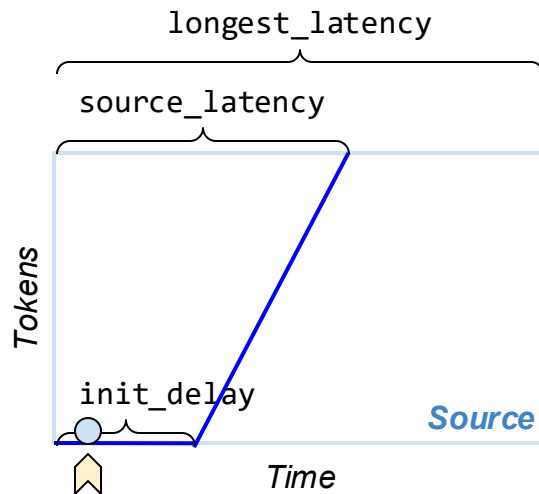
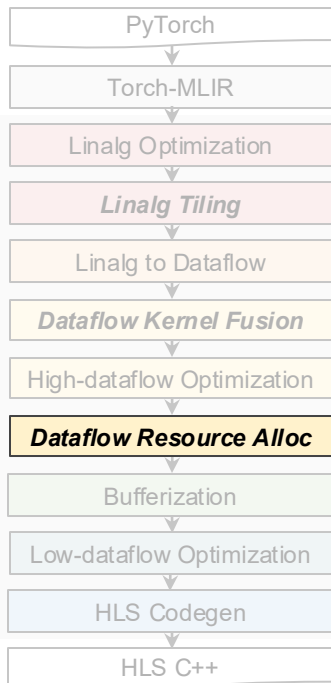


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

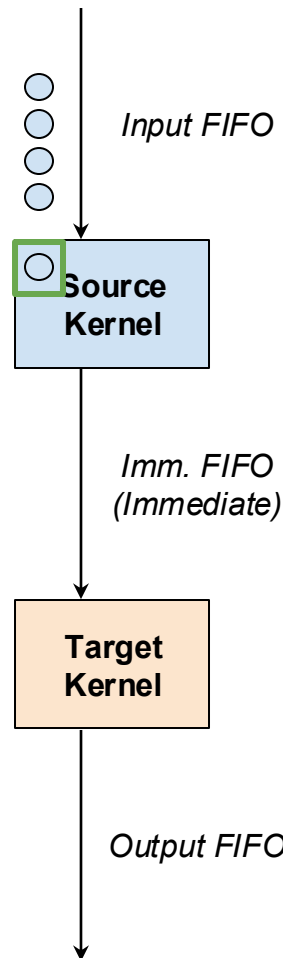


Token Behavior Modeling

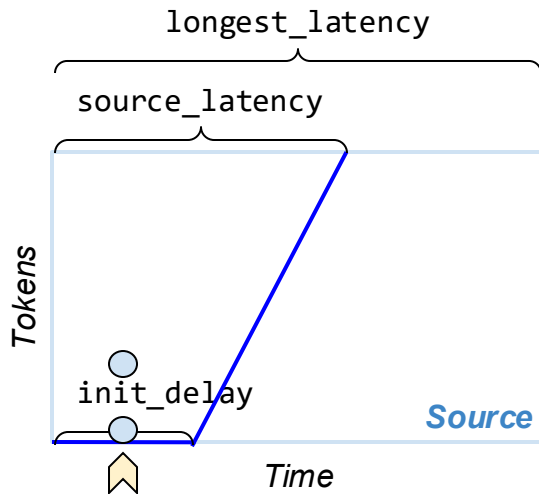
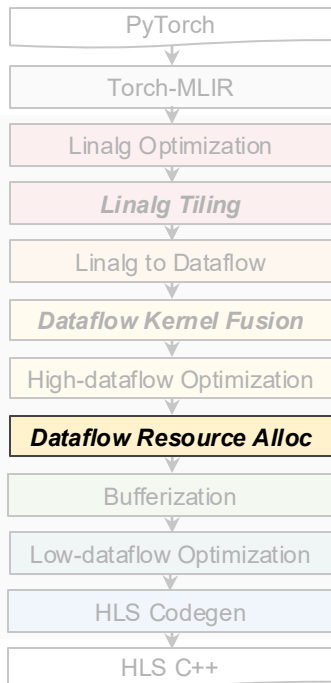


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

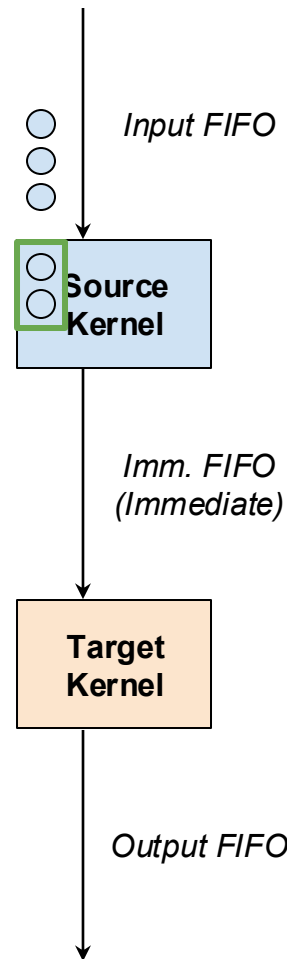


Token Behavior Modeling (Cont.)

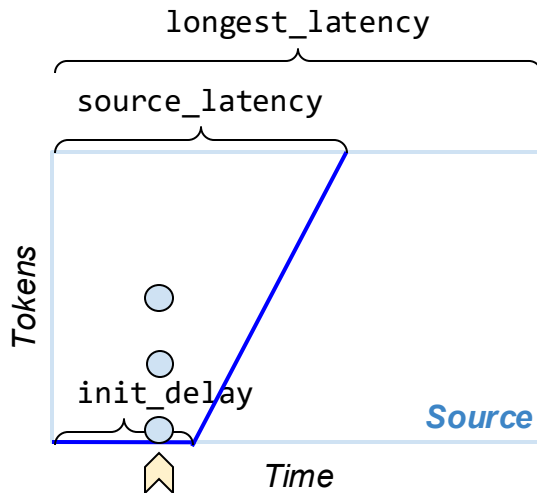
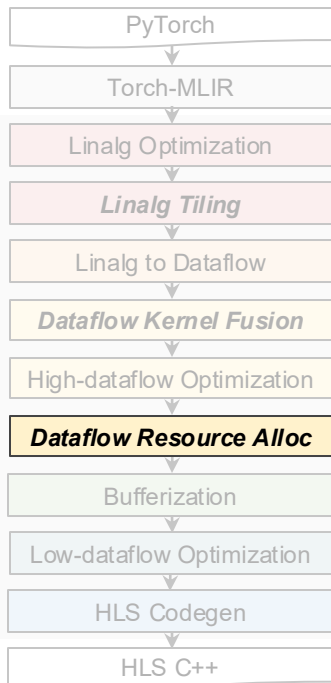


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

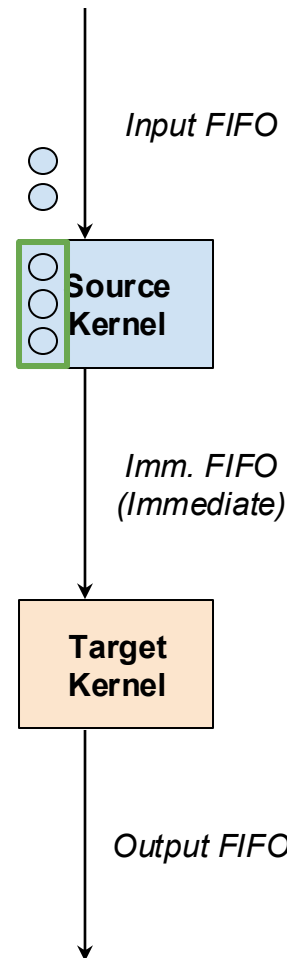


Token Behavior Modeling (Cont.)

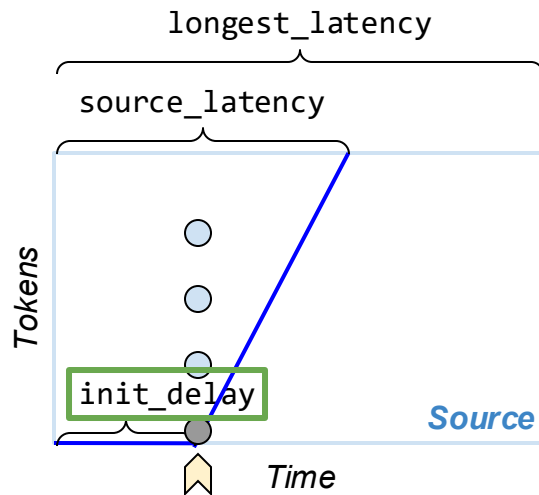
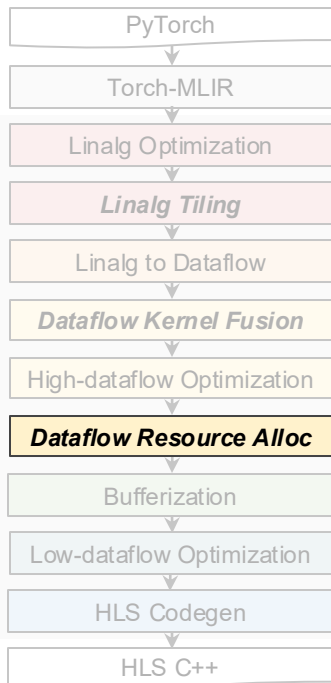


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

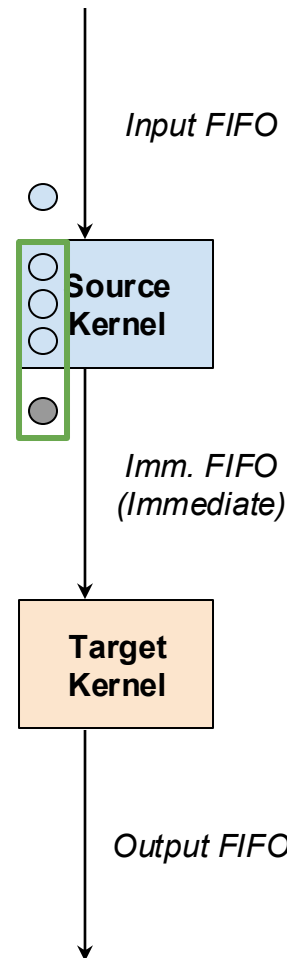


Token Behavior Modeling (Cont.)

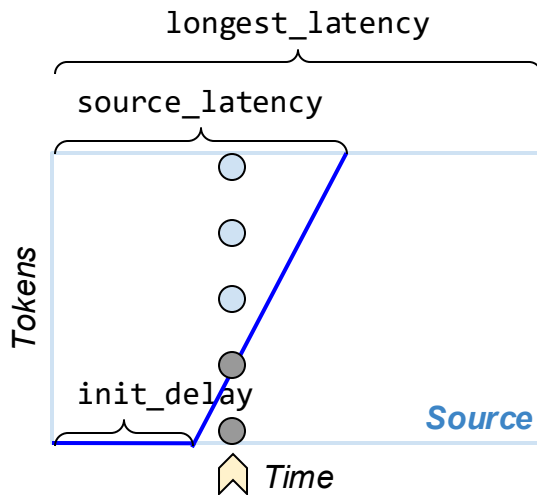
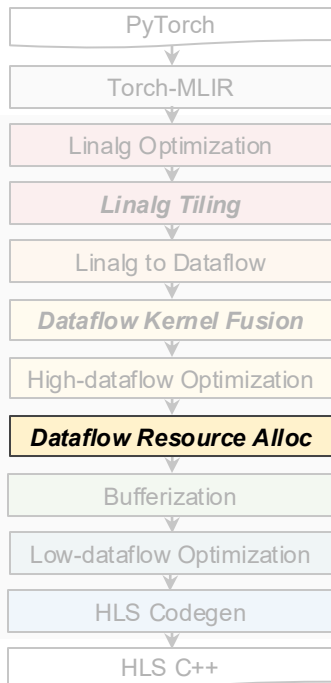


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

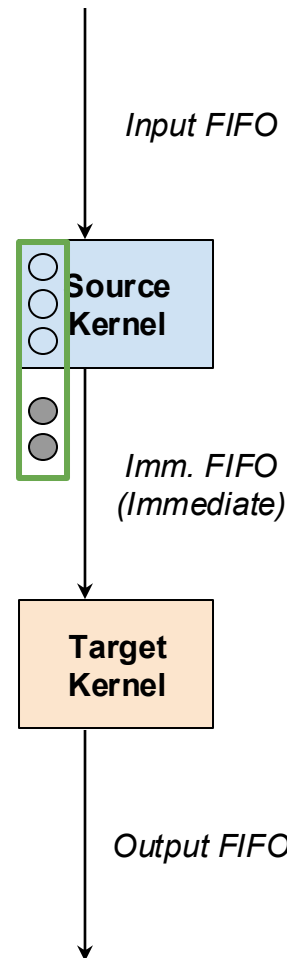


Token Behavior Modeling (Cont.)

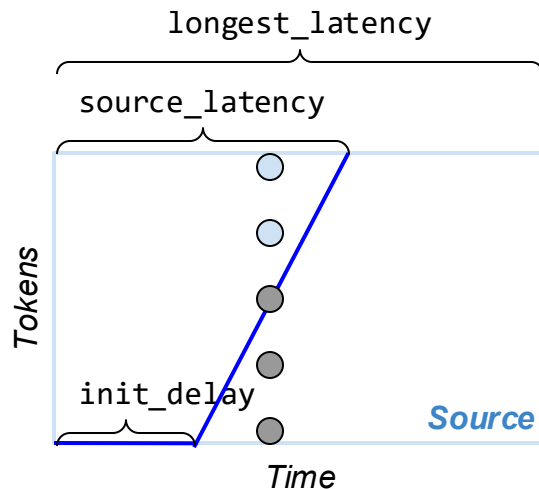
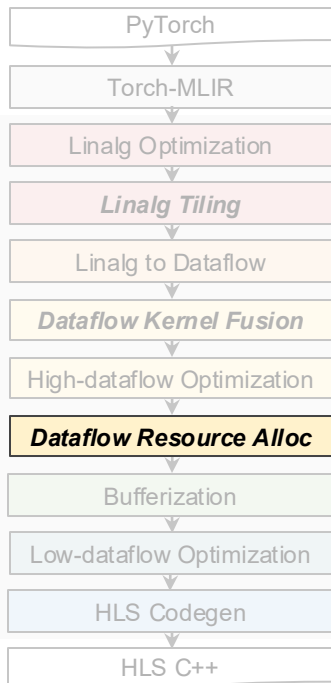


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

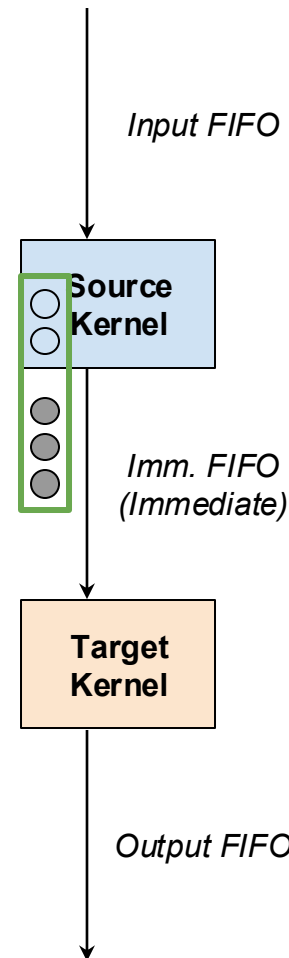


Token Behavior Modeling (Cont.)

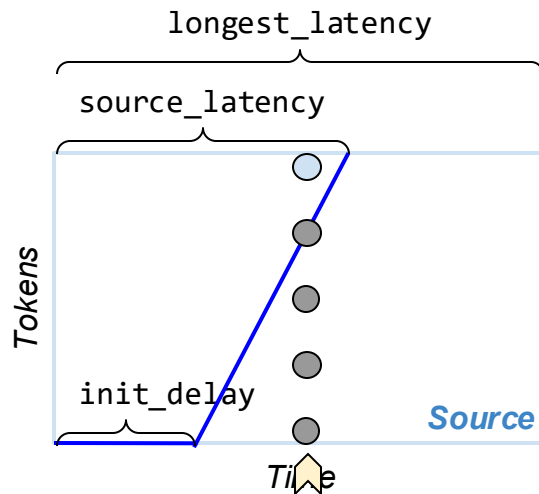
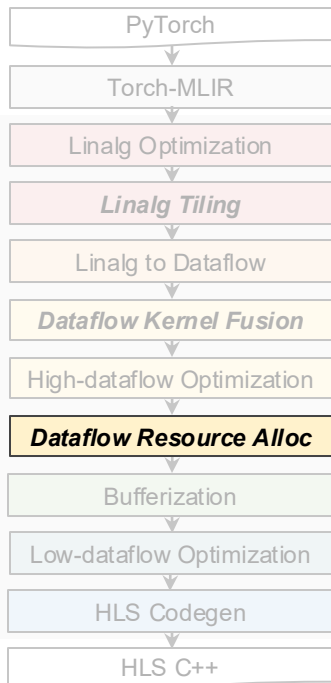


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

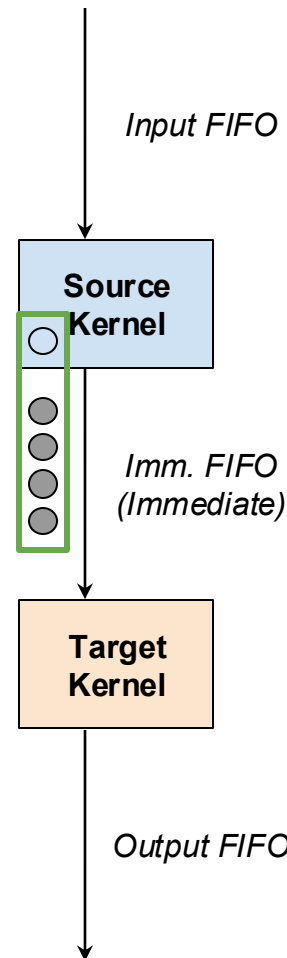


Token Behavior Modeling (Cont.)

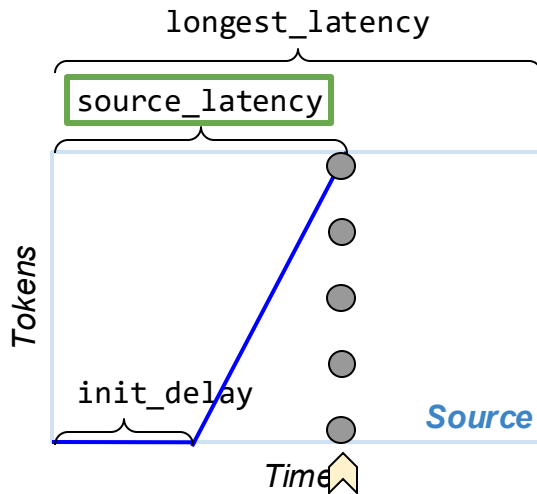
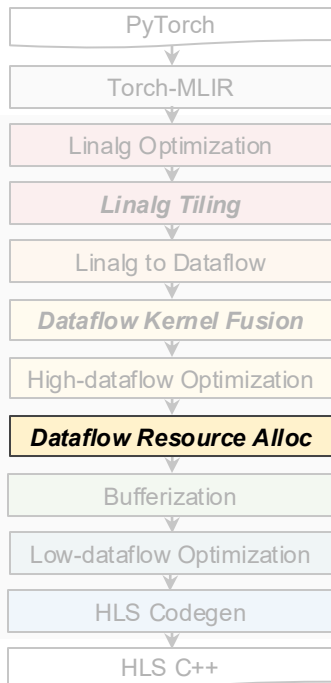


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

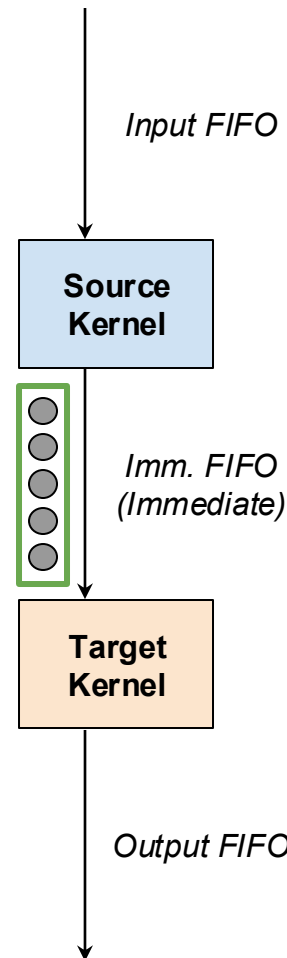


Token Behavior Modeling (Cont.)

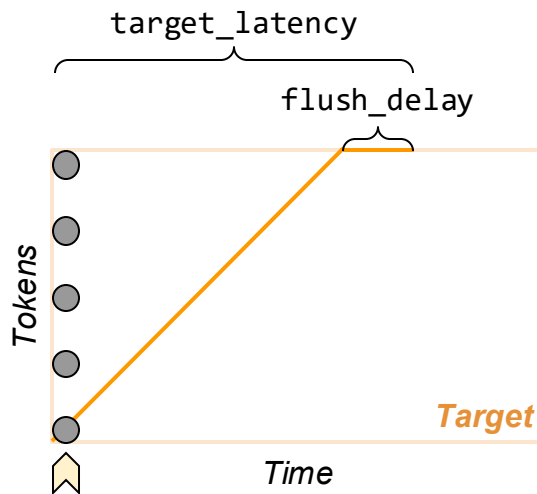
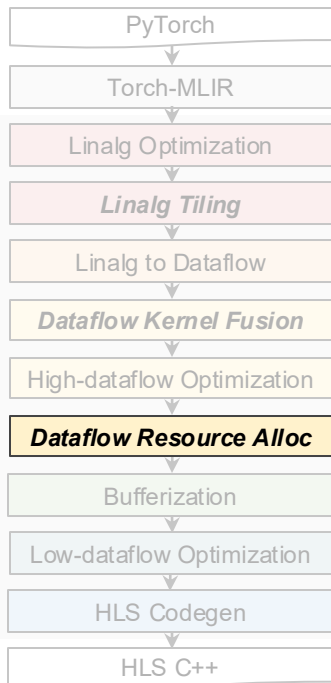


Source Token Production Curve (Push Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

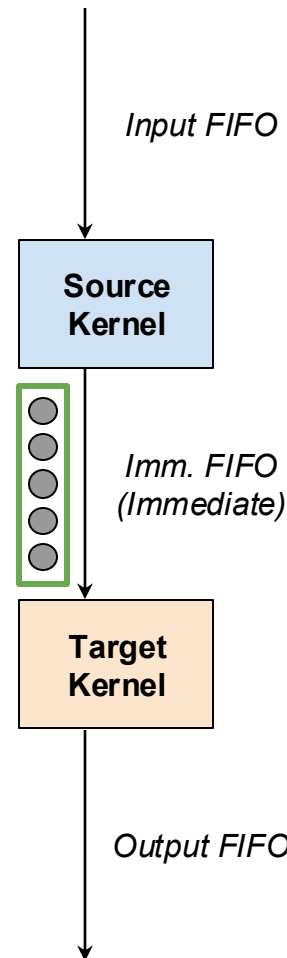


Token Behavior Modeling (Cont.)

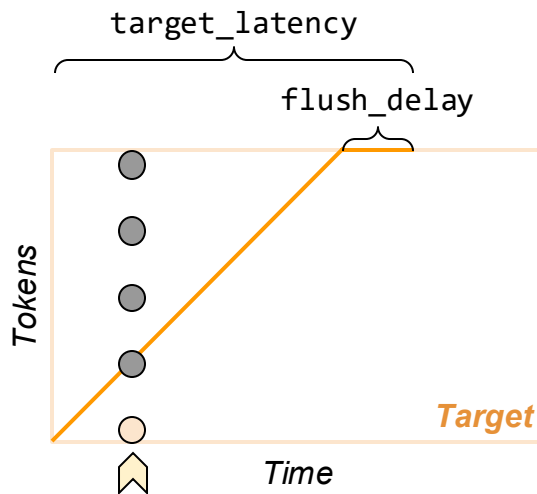
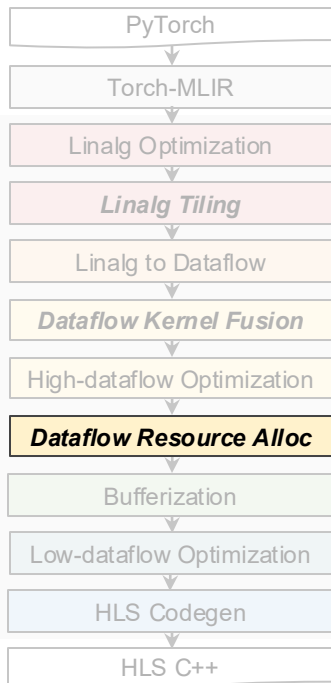


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

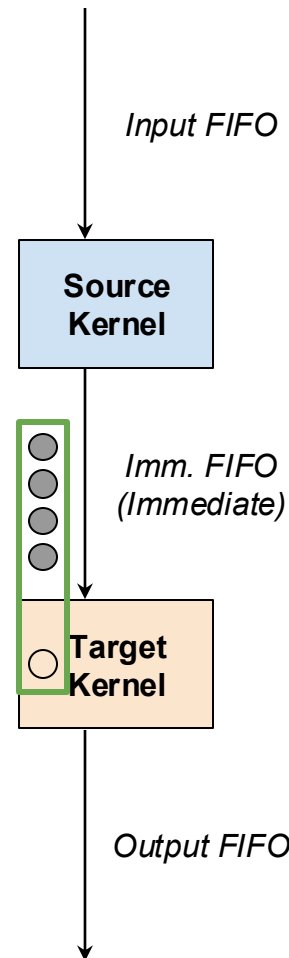


Token Behavior Modeling (Cont.)

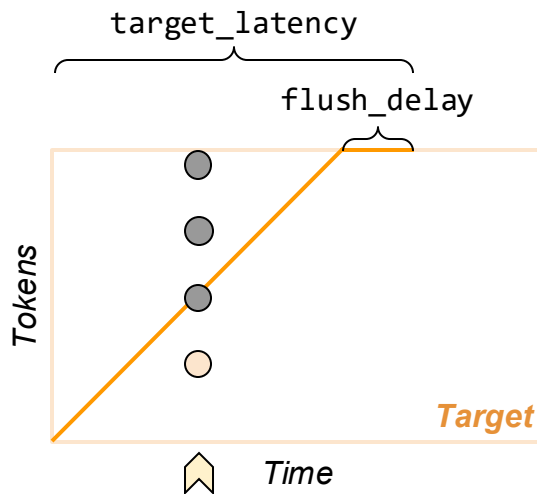
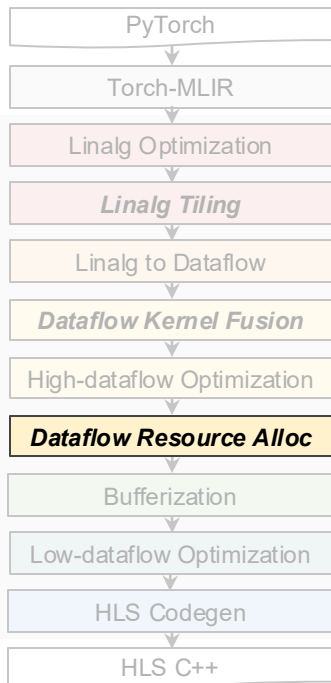


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

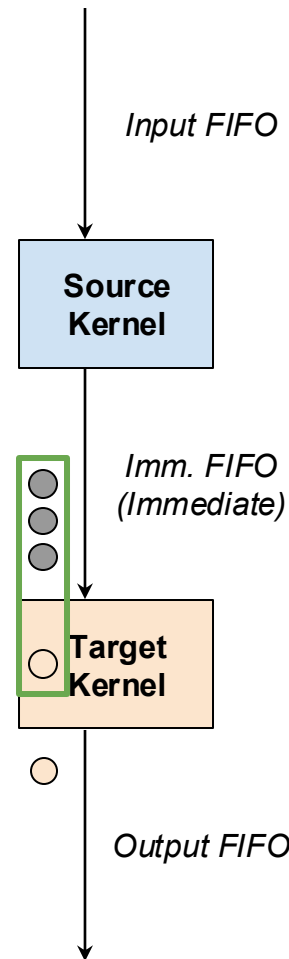


Token Behavior Modeling (Cont.)

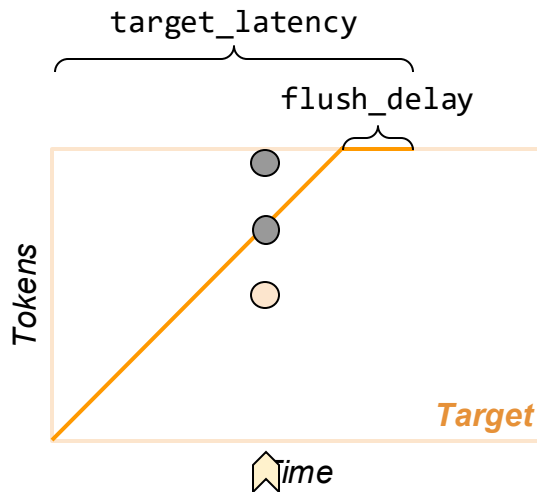
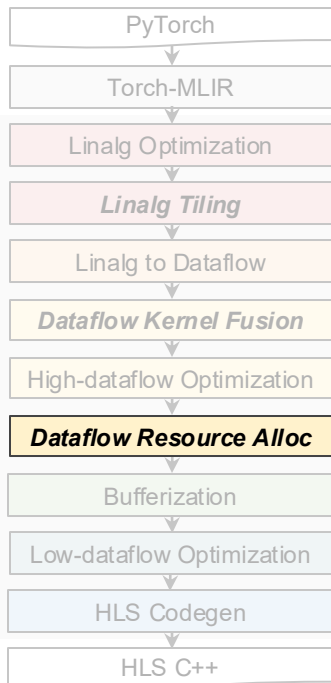


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

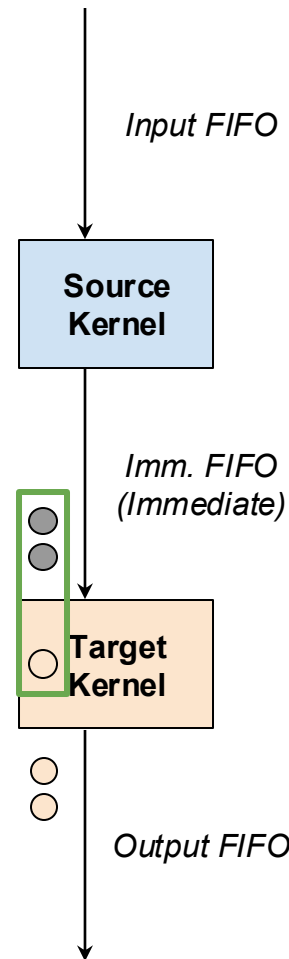


Token Behavior Modeling (Cont.)

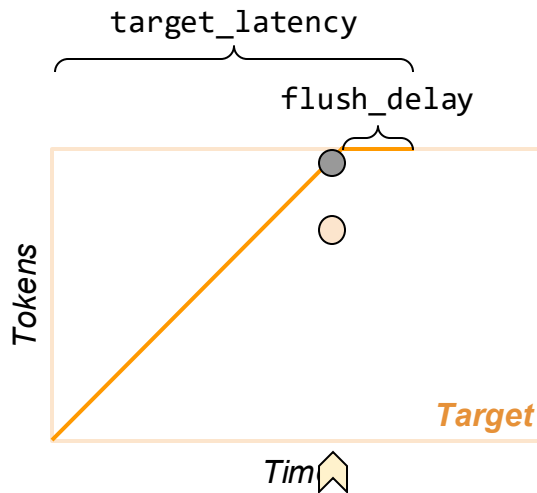
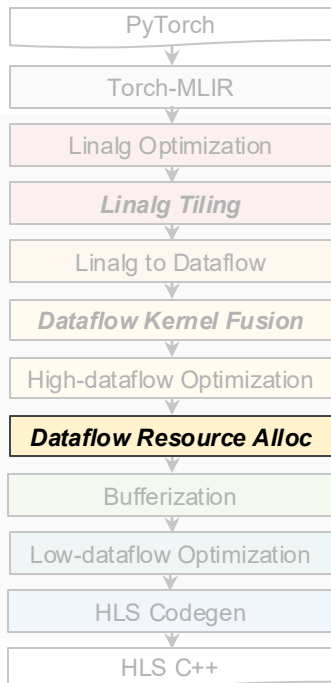


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

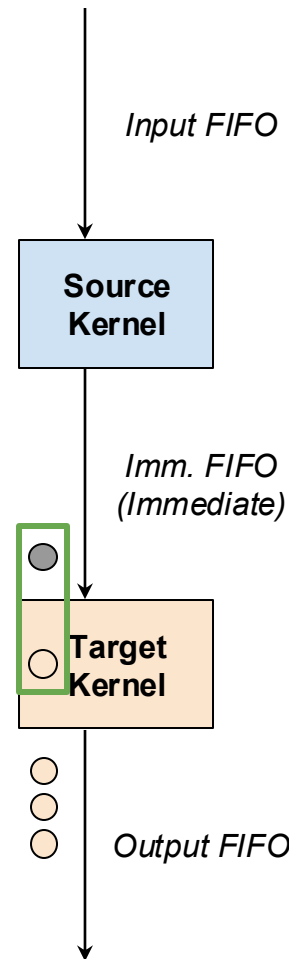


Token Behavior Modeling (Cont.)

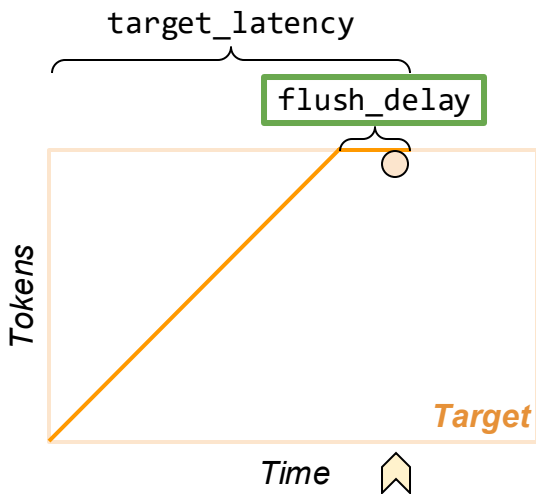
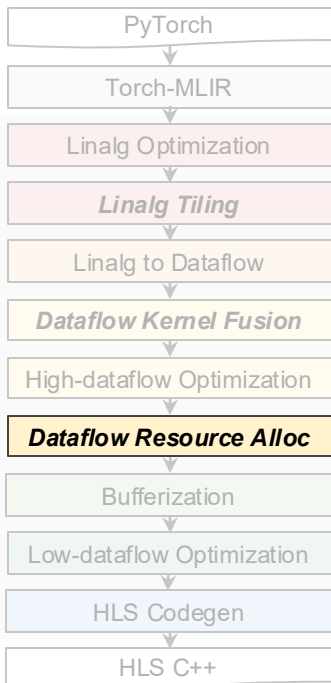


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

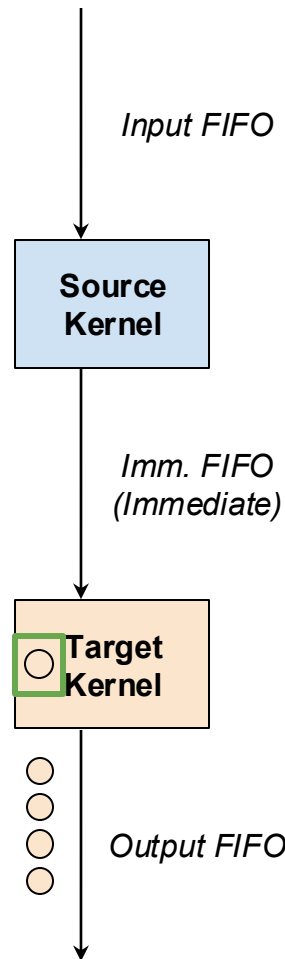


Token Behavior Modeling (Cont.)

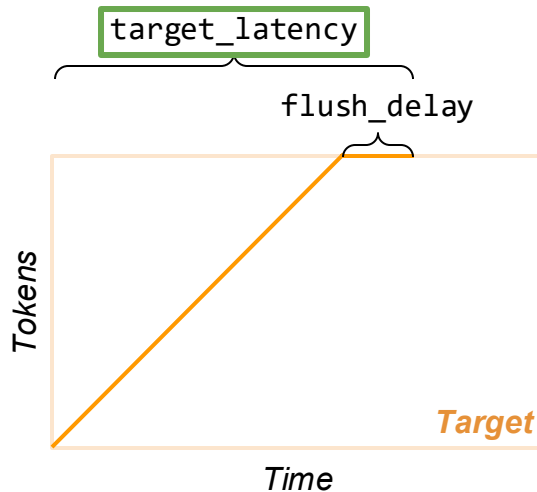
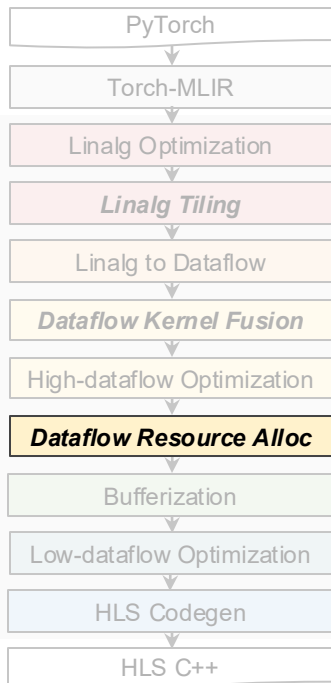


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

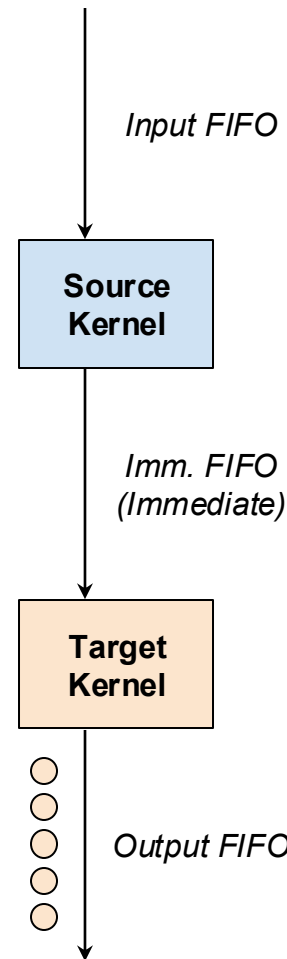


Token Behavior Modeling (Cont.)

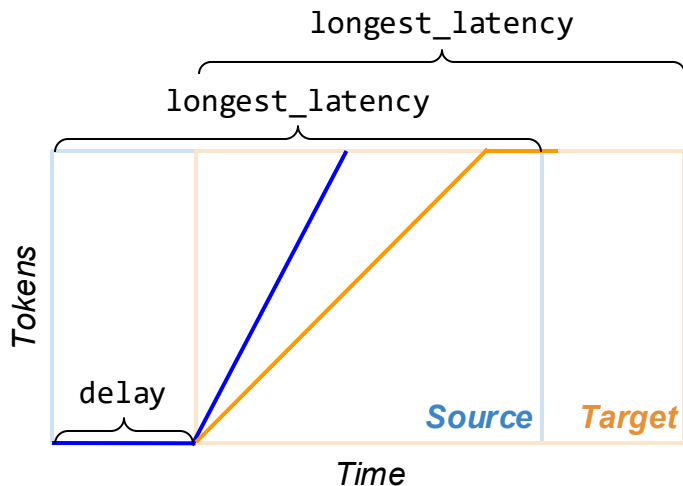
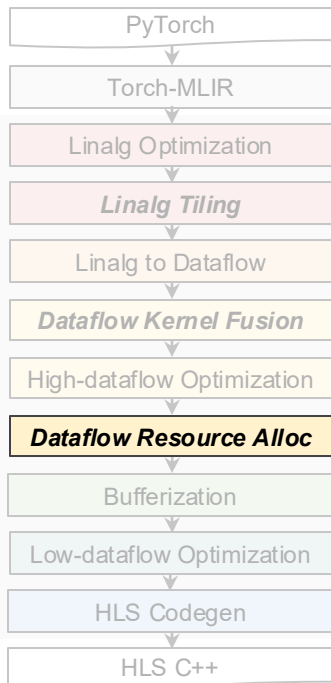


Target Token Consumption Curve (Pull Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

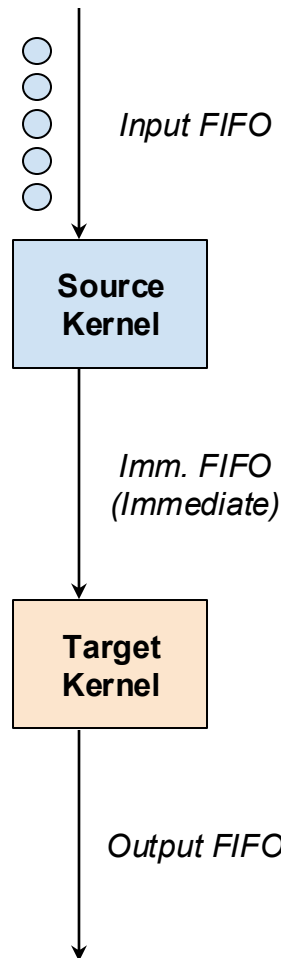


Token Behavior Modeling (Cont.)

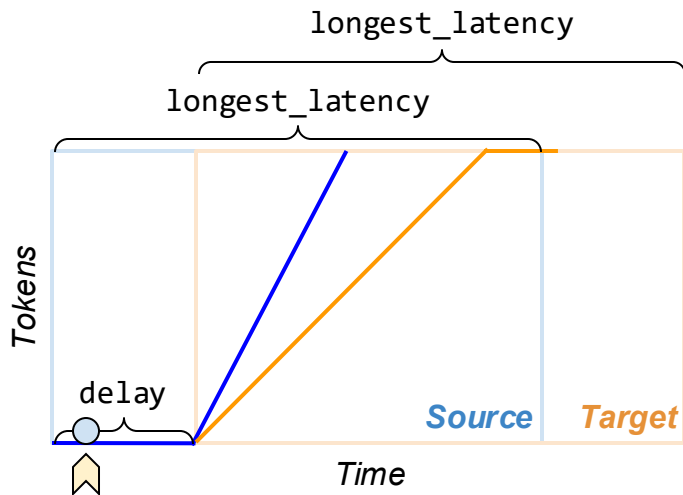
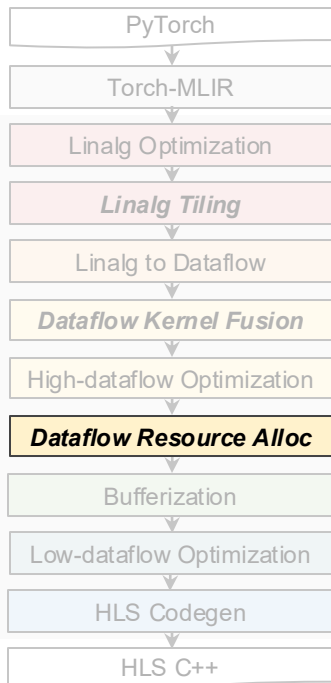


**Overall Token Status Curve
(Imm. FIFO)**

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

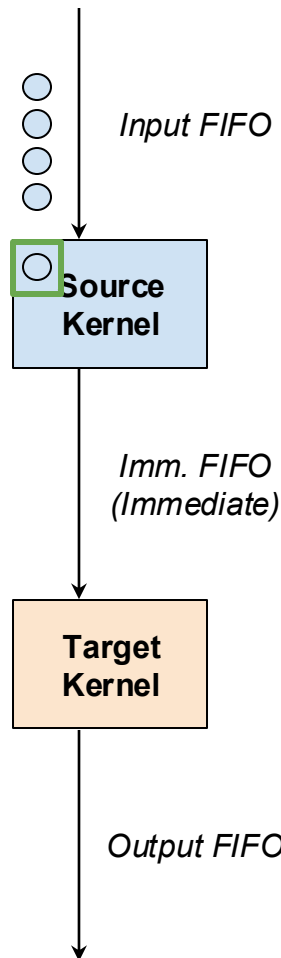


Token Behavior Modeling (Cont.)

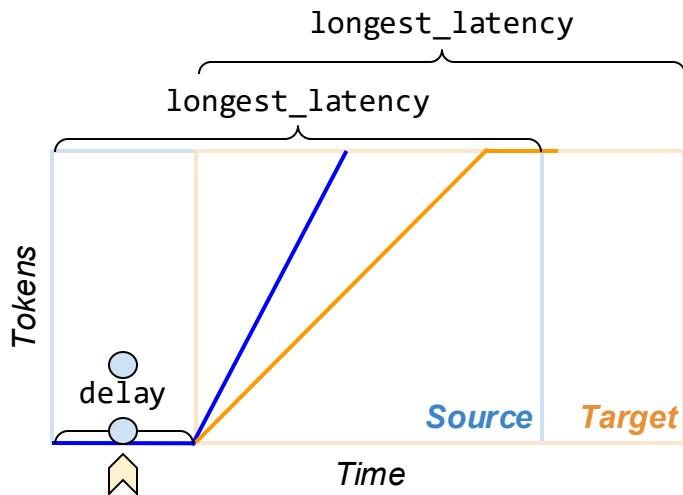
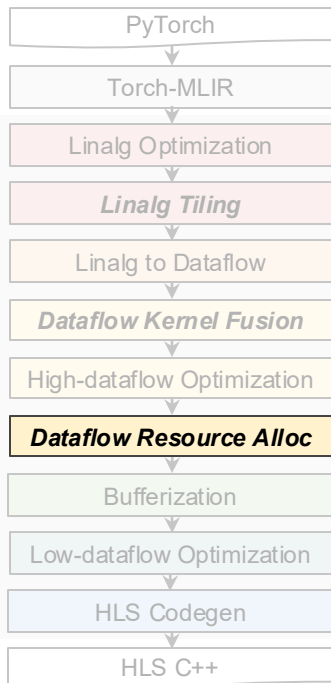


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

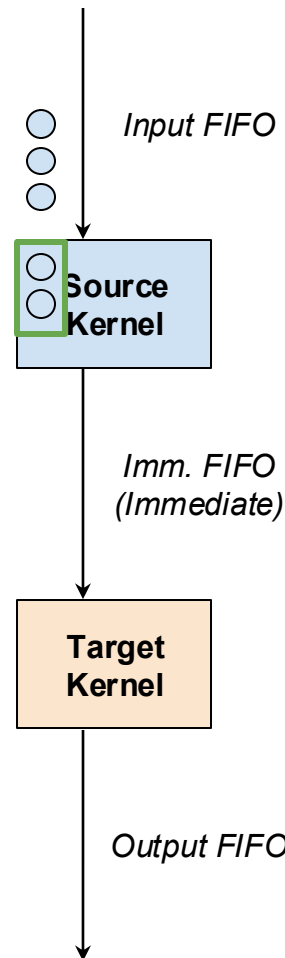


Token Behavior Modeling (Cont.)

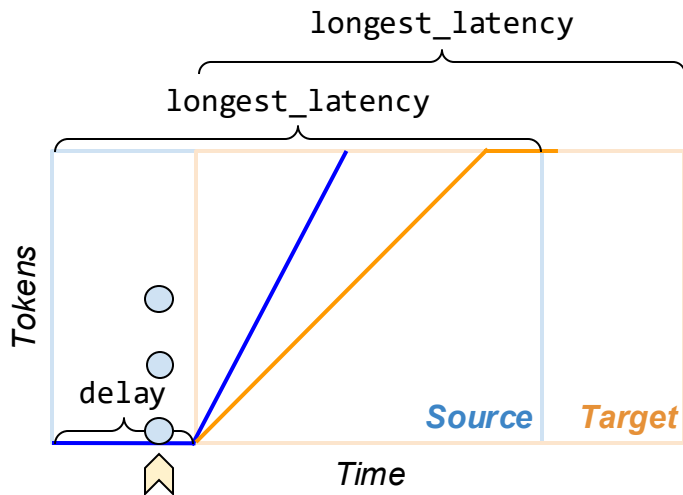
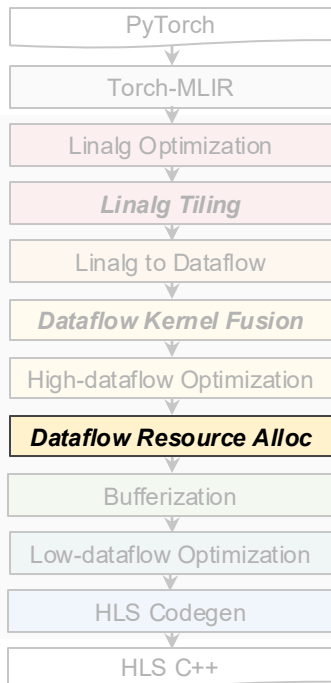


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

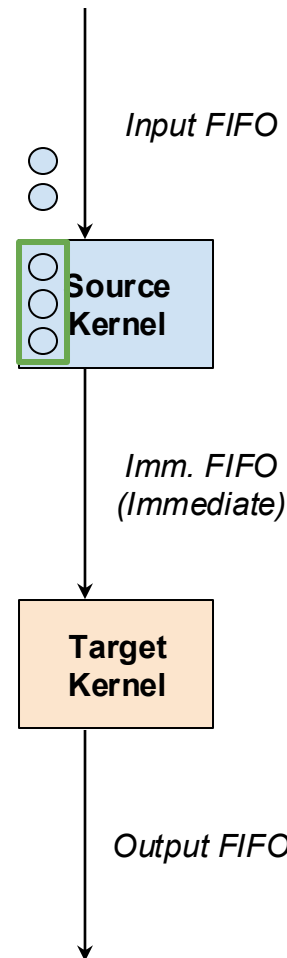


Token Behavior Modeling (Cont.)

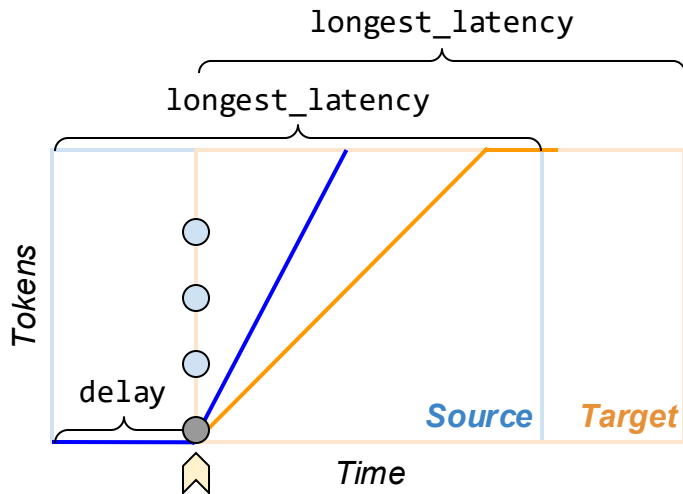
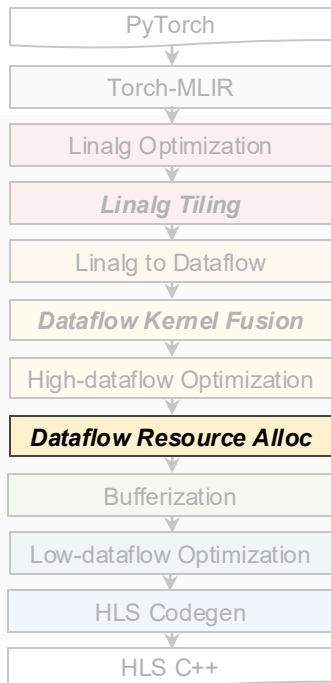


**Overall Token Status Curve
(Imm. FIFO)**

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

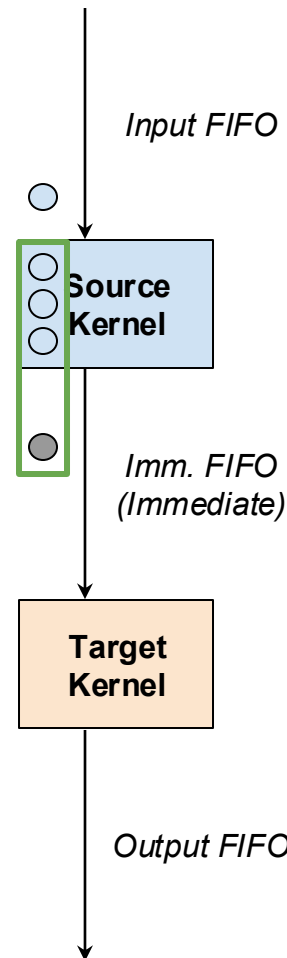


Token Behavior Modeling (Cont.)

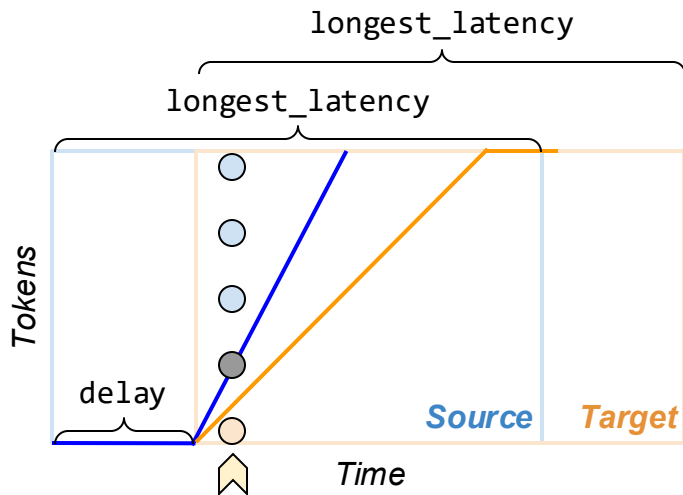
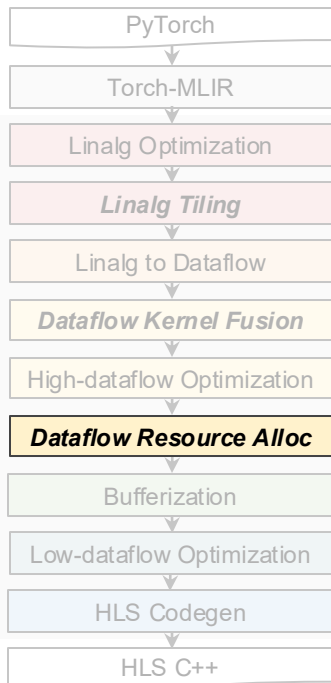


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

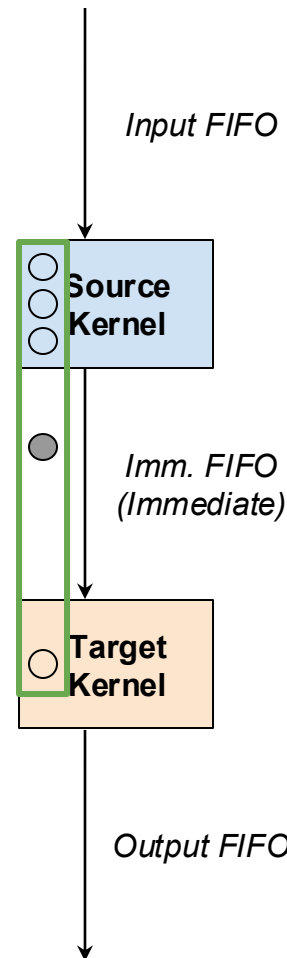


Token Behavior Modeling (Cont.)

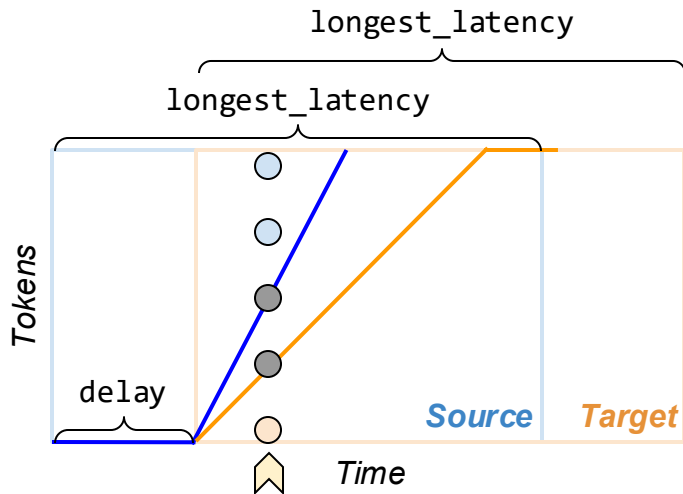
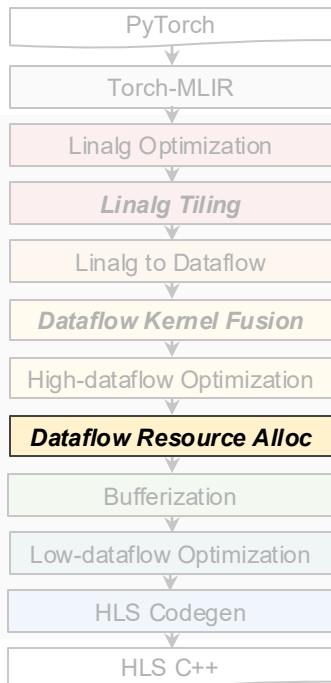


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

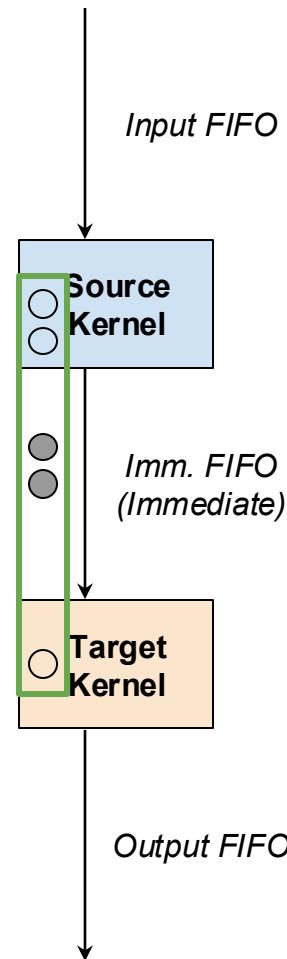


Token Behavior Modeling (Cont.)

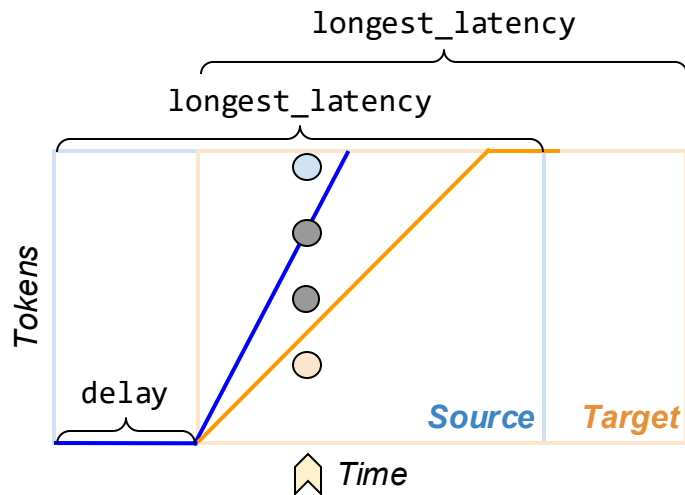
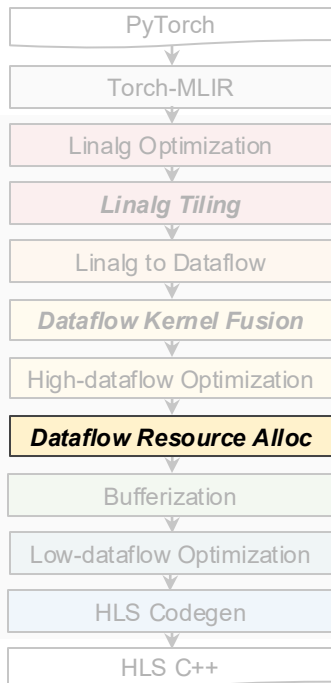


**Overall Token Status Curve
(Imm. FIFO)**

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

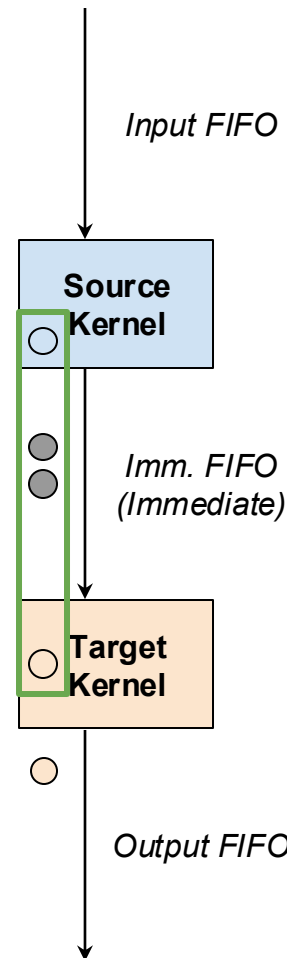


Token Behavior Modeling (Cont.)

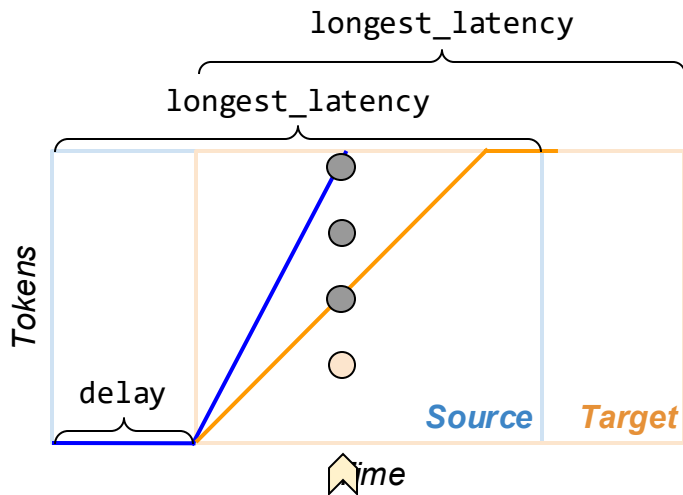
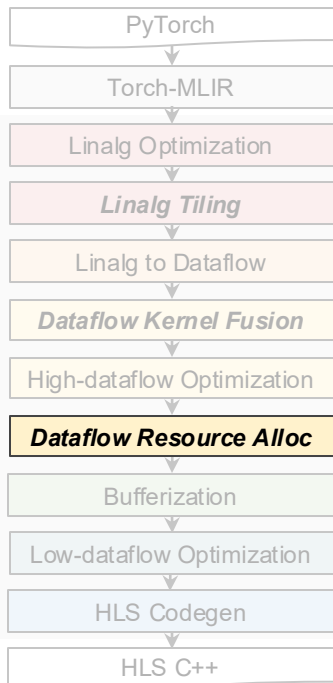


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

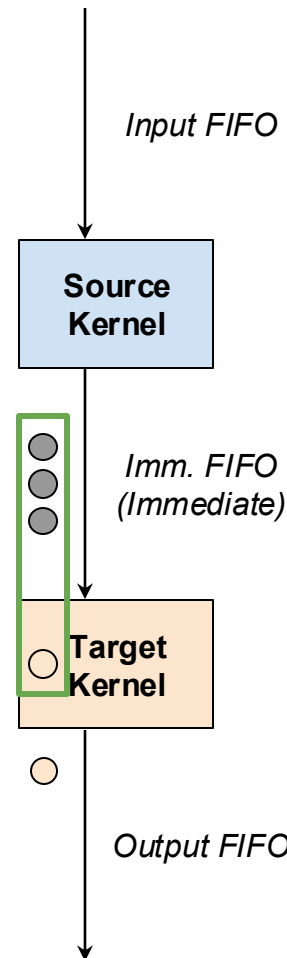


Token Behavior Modeling (Cont.)

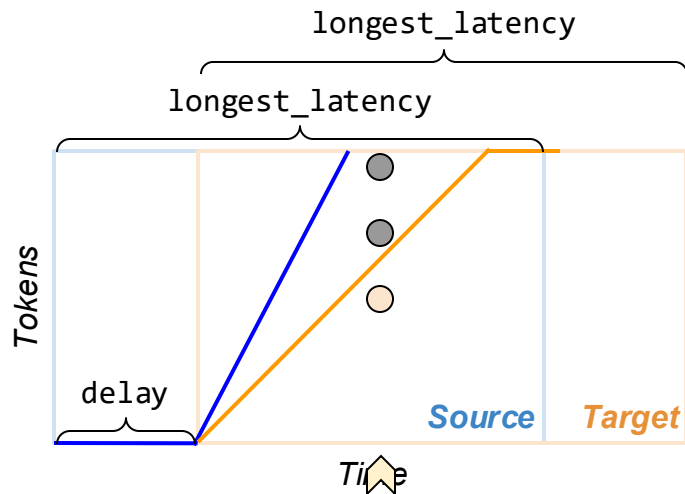
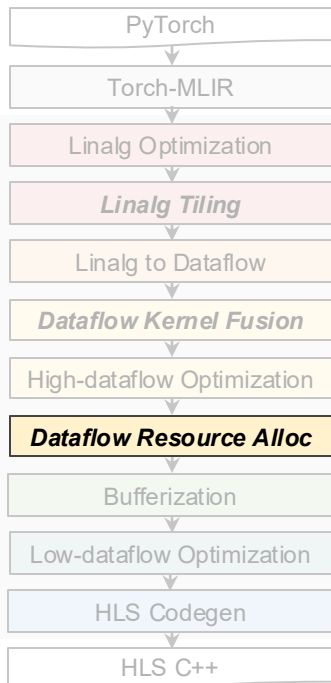


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

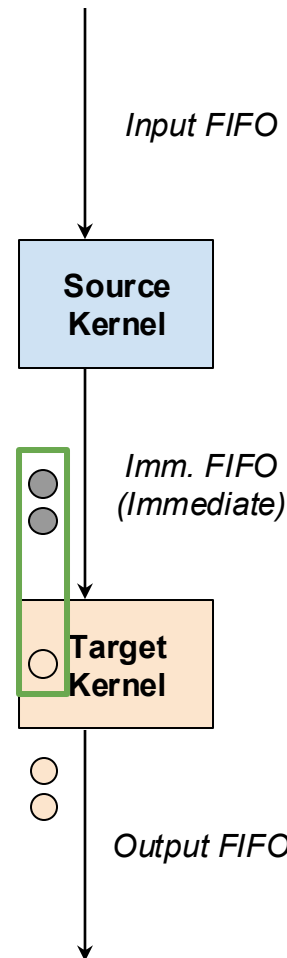


Token Behavior Modeling (Cont.)

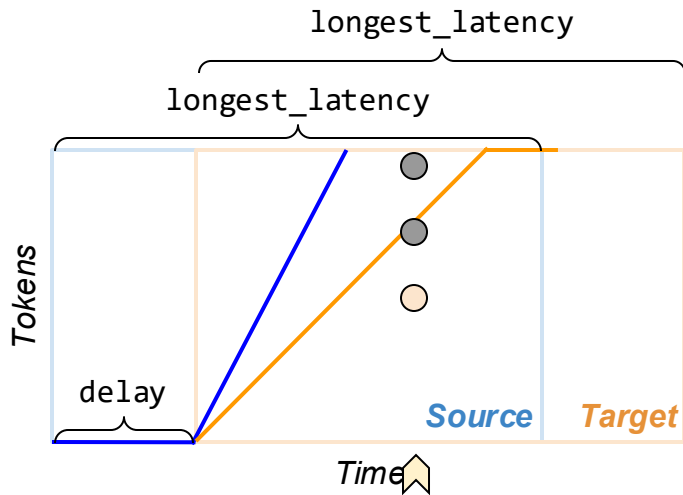
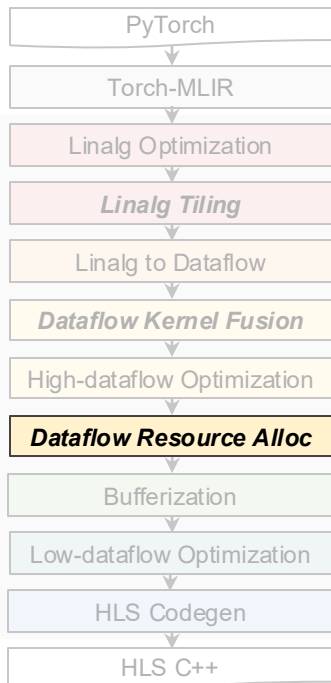


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

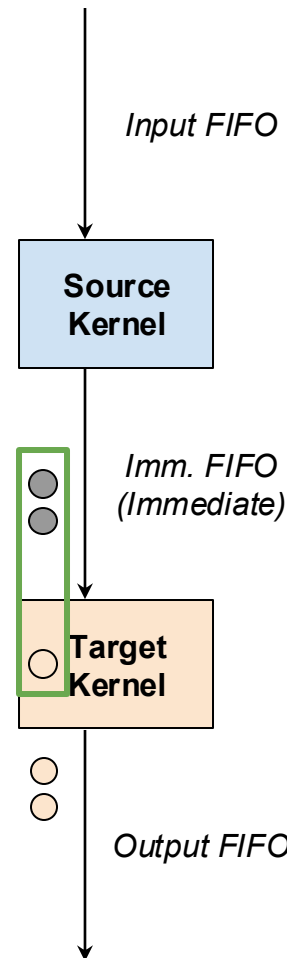


Token Behavior Modeling (Cont.)

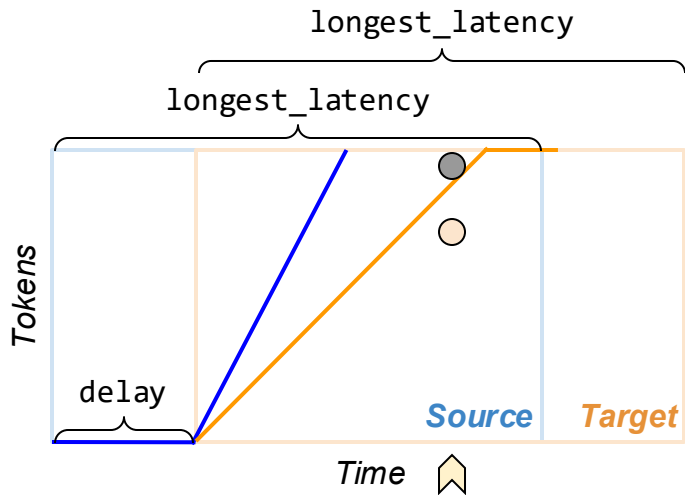
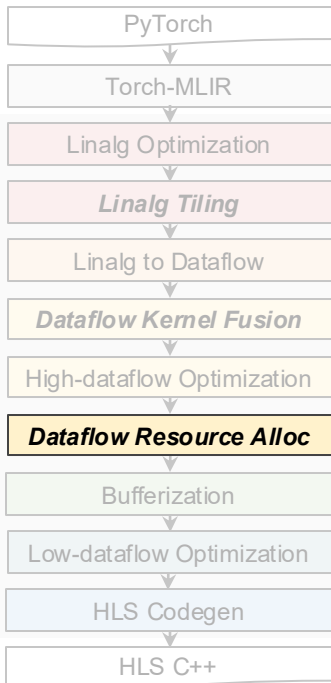


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

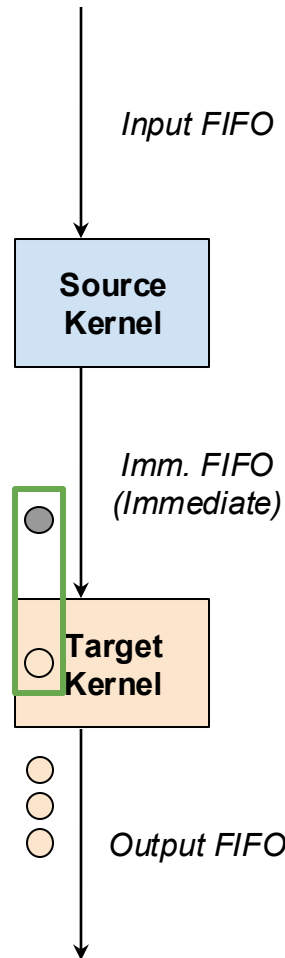


Token Behavior Modeling (Cont.)

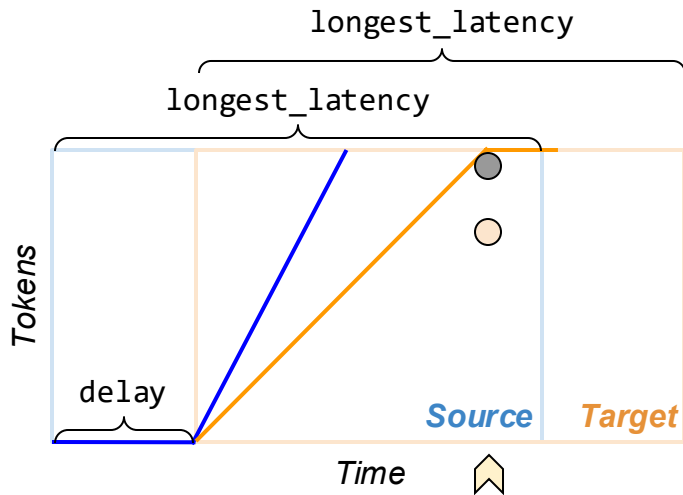
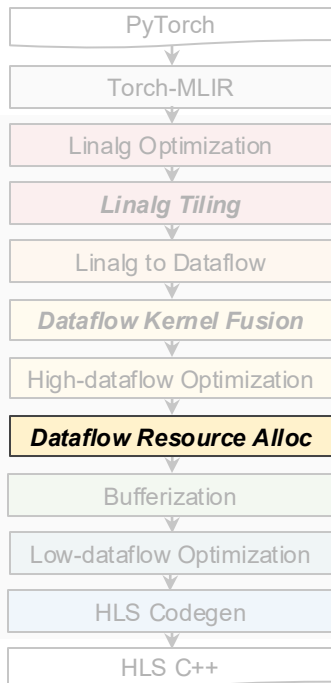


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
 longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

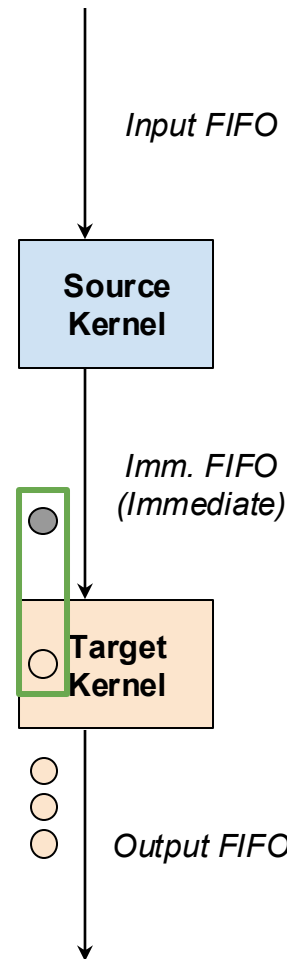


Token Behavior Modeling (Cont.)

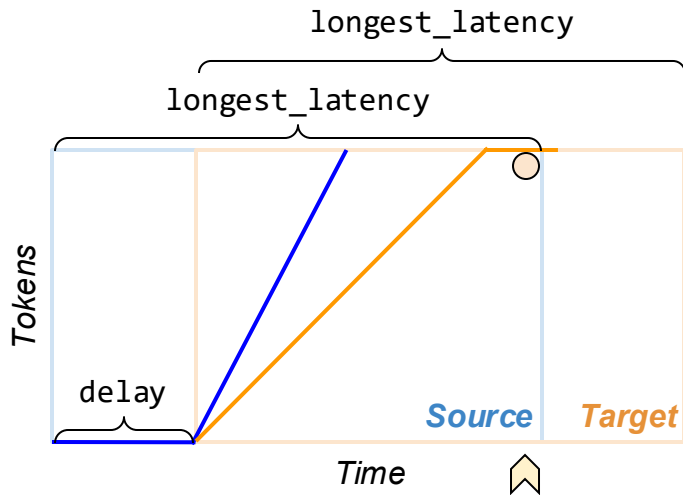
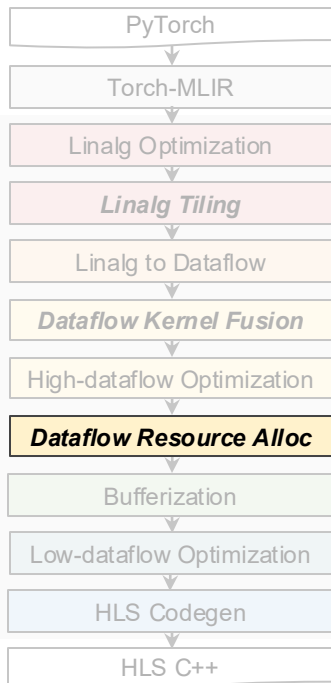


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

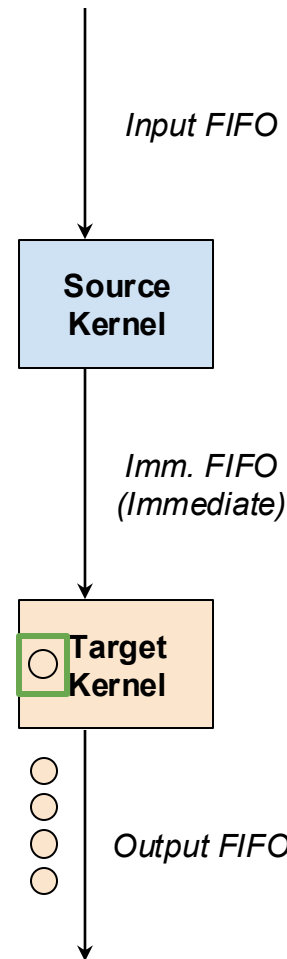


Token Behavior Modeling (Cont.)

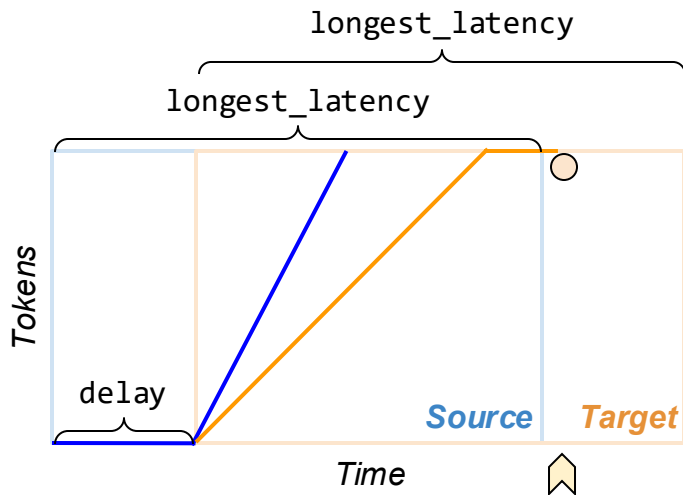
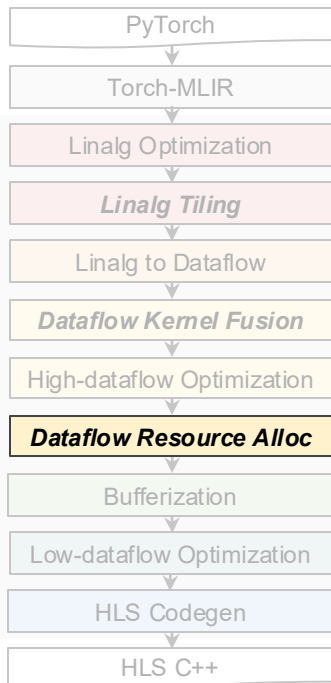


Overall Token Status Curve (Imm. FIFO)

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

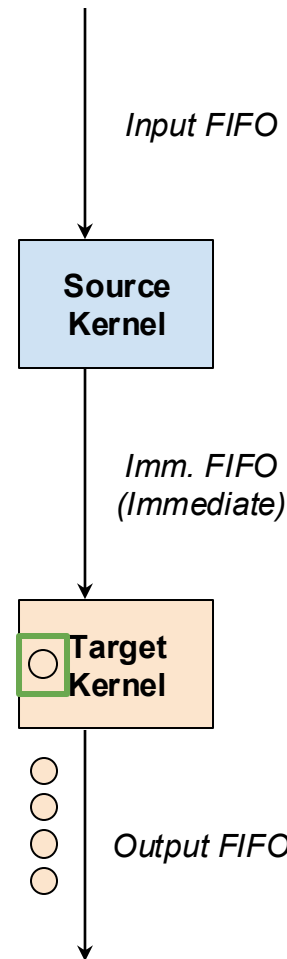


Token Behavior Modeling (Cont.)

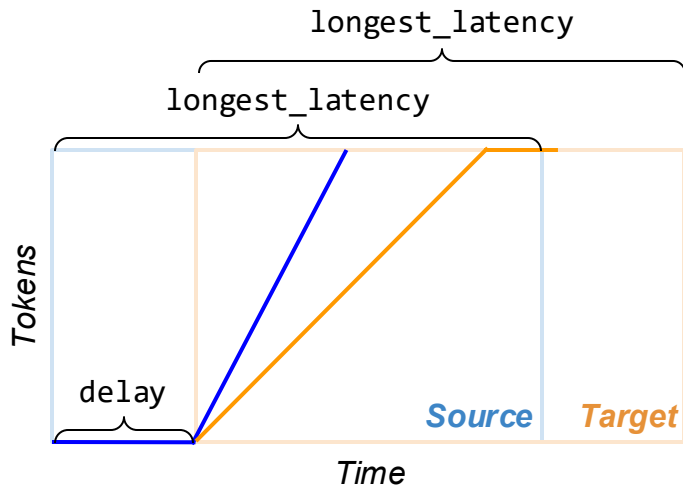
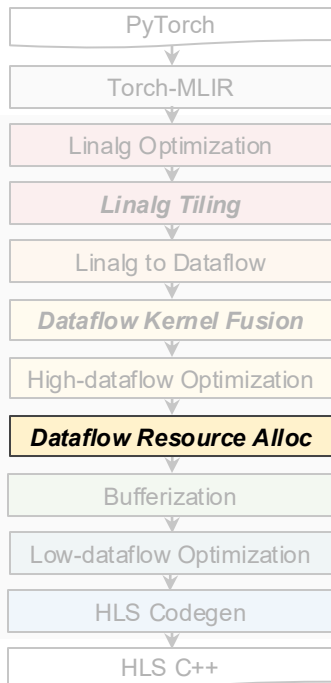


**Overall Token Status Curve
(Imm. FIFO)**

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

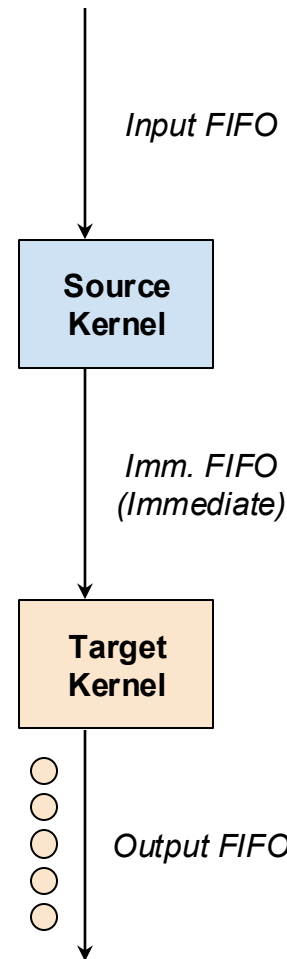


Token Behavior Modeling (Cont.)

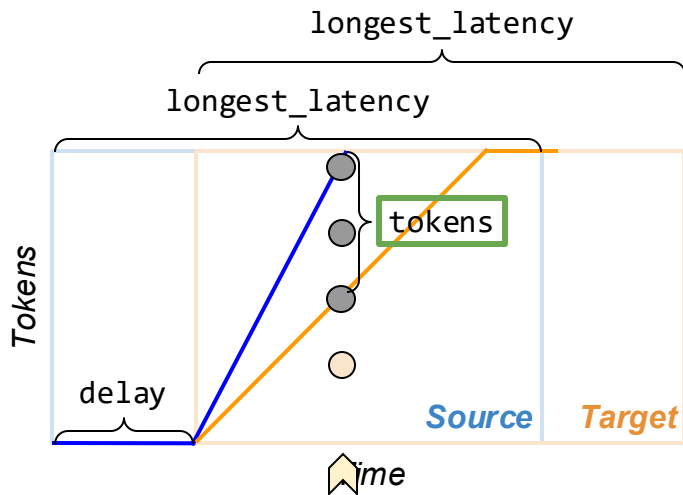
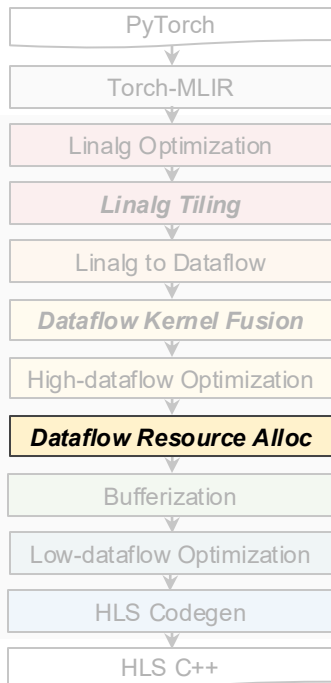


**Overall Token Status Curve
(Imm. FIFO)**

token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput

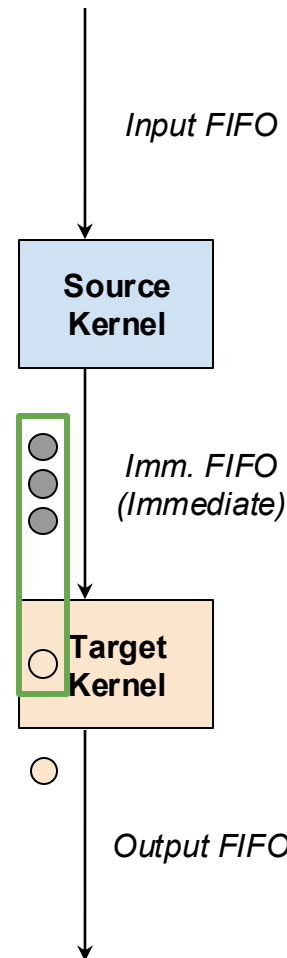


Token Behavior Modeling (Cont.)

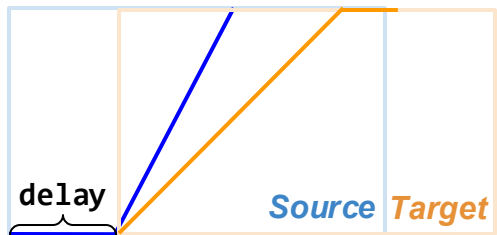
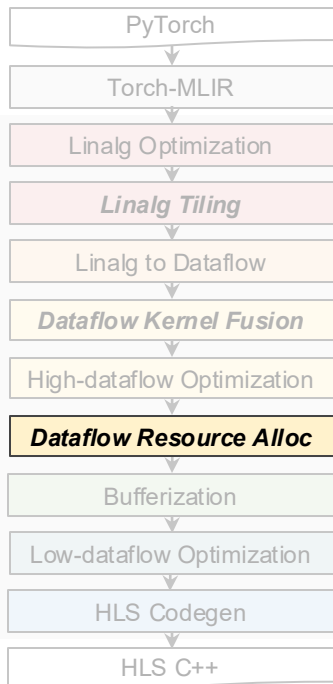


**Overall Token Status Curve
(Imm. FIFO)**

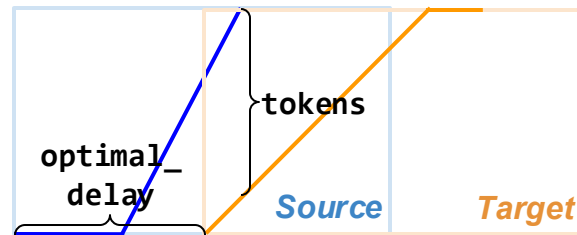
token = Atomic element passed between dataflow kernels
longest_latency = The longest latency among all the dataflow kernels in the accelerator, determining the maximum throughput



Linear Programming (LP) Formulation



- (1) Calculate token production/consumption curves based on kernel profiling results
- (2) Calculate minimum delay given a strategy



- (3) Solve optimal_delay with LP
- (4) Calculate tokens based on optimal_delay given a strategy

Transform FIFO sizing into a scheduling problem

StreamTensor Outline



- Motivation
- StreamTensor Typing System
- StreamTensor Compilation Pipeline
- StreamTensor Design Spaces
- **StreamTensor Results**

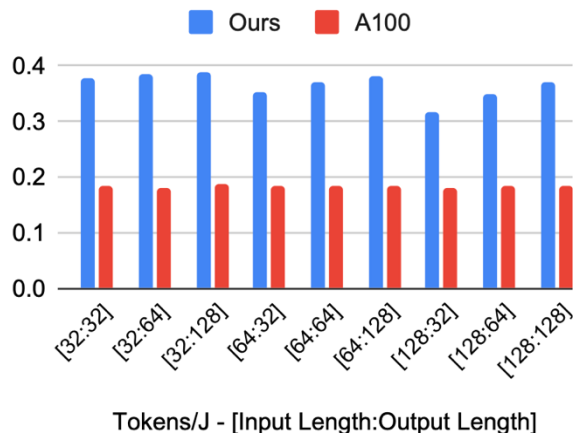
FPGA On-board Results on GPT-2



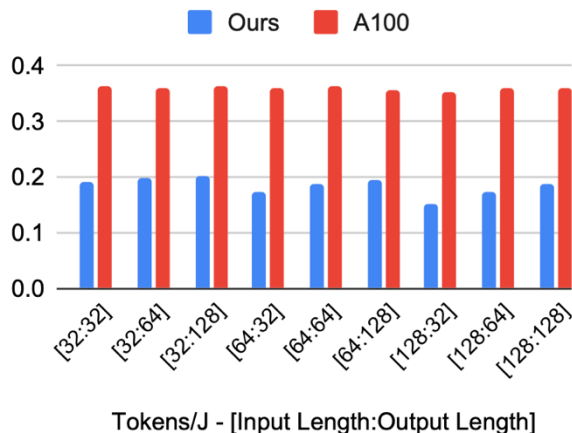
[Input Len: Output Len]	Ours			Allo [15] (Ratio of $\frac{Ours}{Allo}$)			DFX [29] (Ratio of $\frac{Ours}{DFX}$)		
	Latency (ms)	TTFT (ms)	Speed (token/s)	Latency (ms)	TTFT (ms)	Speed (token/s)	Latency (ms)	TTFT (ms)	Speed (token/s)
[32:32]	194.99	34.59	199.51	238.32 (0.82x)	81.50 (0.42x)	204.05 (0.98x)	350.00 (0.56x)	177.20 (0.20x)	185.19 (1.08x)
[64:64]	358.24	61.27	215.51	476.64 (0.75x)	162.99 (0.38x)	204.05 (1.06x)	694.70 (0.52x)	349.10 (0.18x)	185.19 (1.16x)
[128:128]	696.65	125.35	224.05	953.28 (0.73x)	325.98 (0.38x)	204.05 (1.10x)	1384.00 (0.50x)	692.80 (0.18x)	185.19 (1.21x)
[256:256]	1387.76	272.85	229.61	1906.56 (0.73x)	651.96 (0.42x)	204.05 (1.13x)	2800.00 (0.50x)	1417.60 (0.19x)	185.19 (1.24x)
Geo. Mean	-	-	-	0.76x	0.40x	1.06x	0.52x	0.19x	1.17x

[Input Len: Output Len]	Ours			A100 (Ratio of $\frac{Ours}{A100}$)			2080Ti (Ratio of $\frac{Ours}{2080Ti}$)		
	Latency (ms)	TTFT (ms)	Speed (token/s)	Latency (ms)	TTFT (ms)	Speed (token/s)	Latency (ms)	TTFT (ms)	Speed (token/s)
[32:32]	194.99	34.59	199.51	291.16 (0.67x)	8.72 (3.97x)	113.30 (1.76x)	518.46 (0.38x)	24.98 (1.38x)	64.85 (3.08x)
[64:64]	358.24	61.27	215.51	567.41 (0.63x)	8.76 (6.99x)	114.56 (1.88x)	1010.81 (0.35x)	25.23 (2.43x)	64.94 (3.32x)
[128:128]	696.65	125.35	224.05	1118.28 (0.62x)	8.65 (14.49x)	115.35 (1.94x)	3969.76 (0.18x)	25.26 (4.96x)	32.45 (6.90x)
[256:256]	1387.76	272.85	229.61	2227.79 (0.62x)	8.53 (31.99x)	115.35 (1.99x)	7914.23 (0.18x)	25.23 (10.81x)	32.45 (7.08x)
Geo. Mean	-	-	-	0.64x	10.65x	1.89x	0.25x	3.67x	4.73x

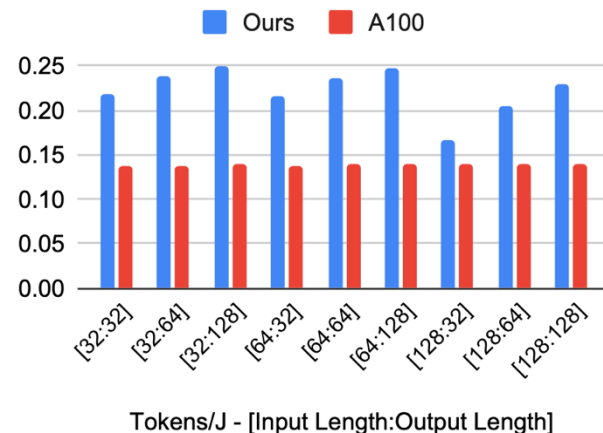
FPGA On-board Results on Emerging LLMs



(a) Qwen.



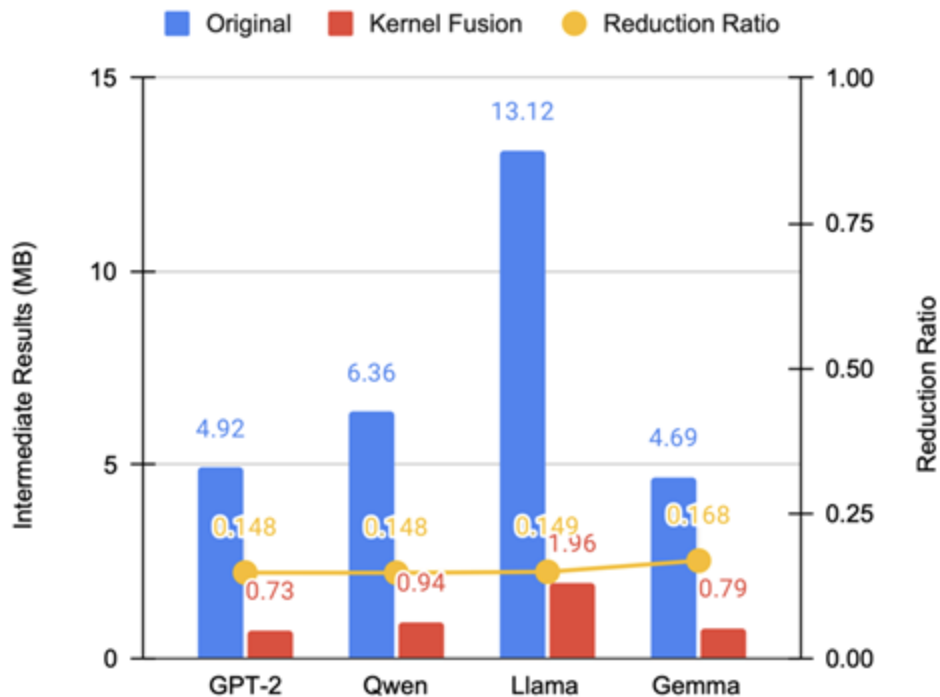
(b) Llama.



(c) Gemma.

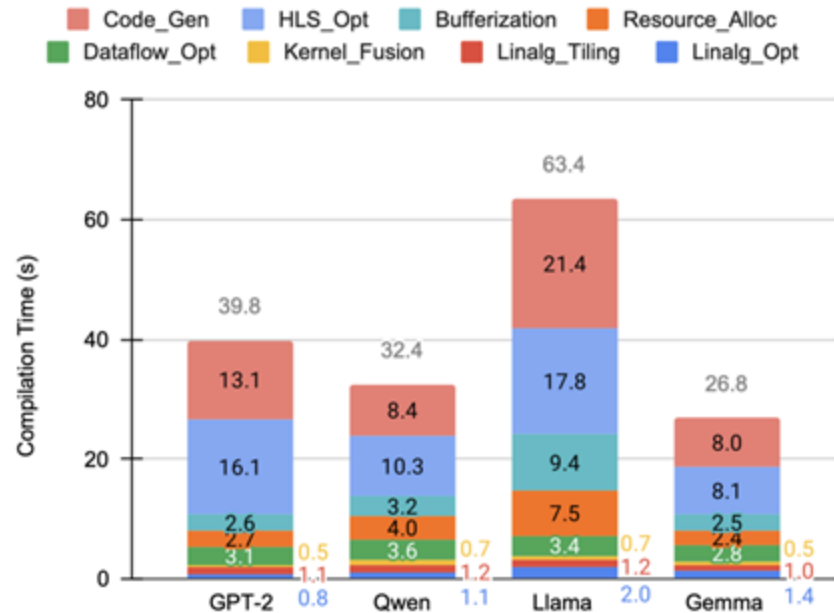
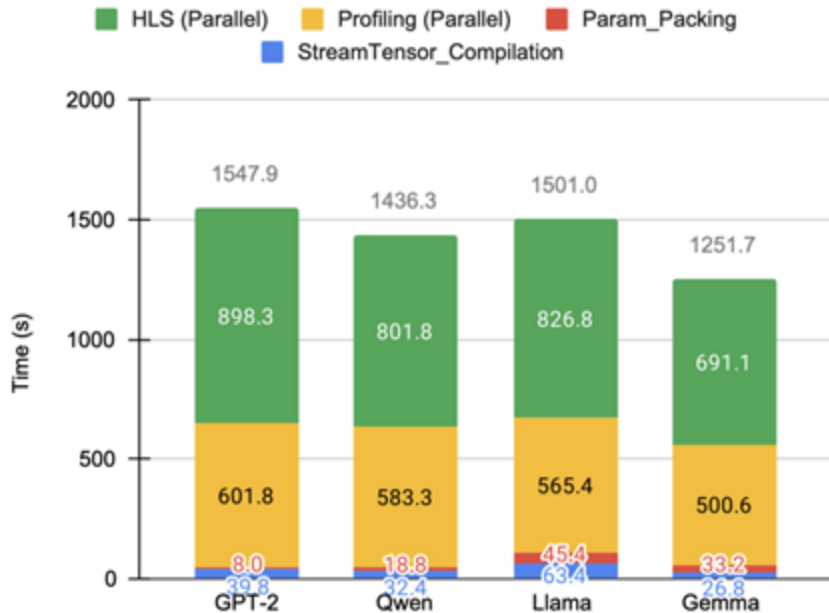
- Achieved higher energy efficiency on Qwen and Gemma.
- Energy efficiency of Llama is lower than GPU because the intermediate results of Llama is larger than other models, limiting the design space explorations.

On-chip Memory Reduction through Kernel Fusion



- Only consider intermediate results, i.e., activations, in this study.
- Parameters are always stored in external memory.
- On-chip memory are reduced to 0.15x - 0.17x through stream-based kernel fusion.
- Without kernel fusion, the on-chip memory resources are not enough to store all intermediate results

RTL Generation Time



- For RTL generation, the downstream HLS synthesis and profiling consume most execution time
- StreamTensor compilation and parameter packing only consume a tiny portion of execution time
- For StreamTensor compilation, the low-level transforms (bufferization, HLS opt., and codegen) consume much more execution than high-level transforms (Linalg tiling, kernel fusion, dataflow opt.), showing the efficiency of the proposed itensor-based dataflow optimizations.



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN



Thank You!

Hanchen Ye¹, Deming Chen^{1,2}

¹University of Illinois at Urbana-Champaign

²Inspirit IoT, Inc.