Week 16

Performance

Data Size: 64x768 768x64

	bc_matmul	matmul	Change
4 lanes	92.28%	65.67%	1.4x
8 lanes	85.63%	34.22%	2.5x
16 lanes	74.85%	17.03%	4.4x

Algorithm Update

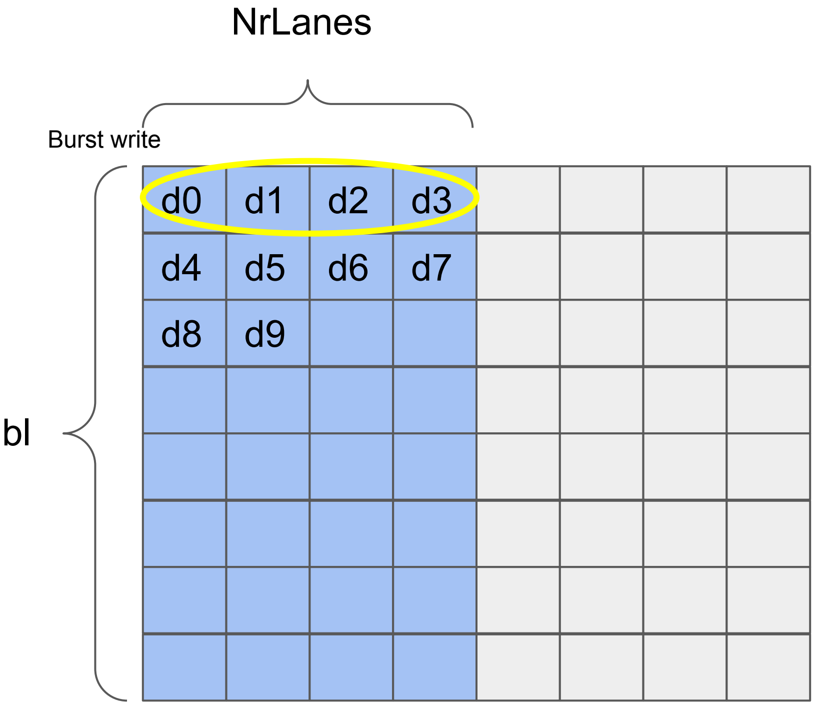
- Avoid strided memory operations
  - Store A^T instead of A

Algorithm Update

- Avoid strided memory operations
    - Store A^T instead of A
    - Load A with unit-stride load
        - utilization 90.66% → 92.28%

Algorithm Update

- Avoid strided memory operations
  - Store A^T instead of A
  - Load A with unit-stride load
    - utilization 90.66% → 92.28%
  - Use burst write to store result (TODO)
    - Memory bandwidth = 32 x NrLanes
    - length = NrLanes
    - Address alignment?



Algorithm Update

- Avoid strided memory operations
  - Store A^T instead of A
  - Load A with unit-stride load
    - utilization 90.66% → 92.28%
  - Use burst write to store result (TODO)
    - Memory bandwidth = 32 x NrLanes
    - length = NrLanes
    - Address alignment?
  - Performance of Softmax & LayerNorm ⬆
    - strided → unit-strided
    - 1.7%
  - Performance of ReLU & Dropout ⬇
    - can still use unit-strided
    - 0.3%

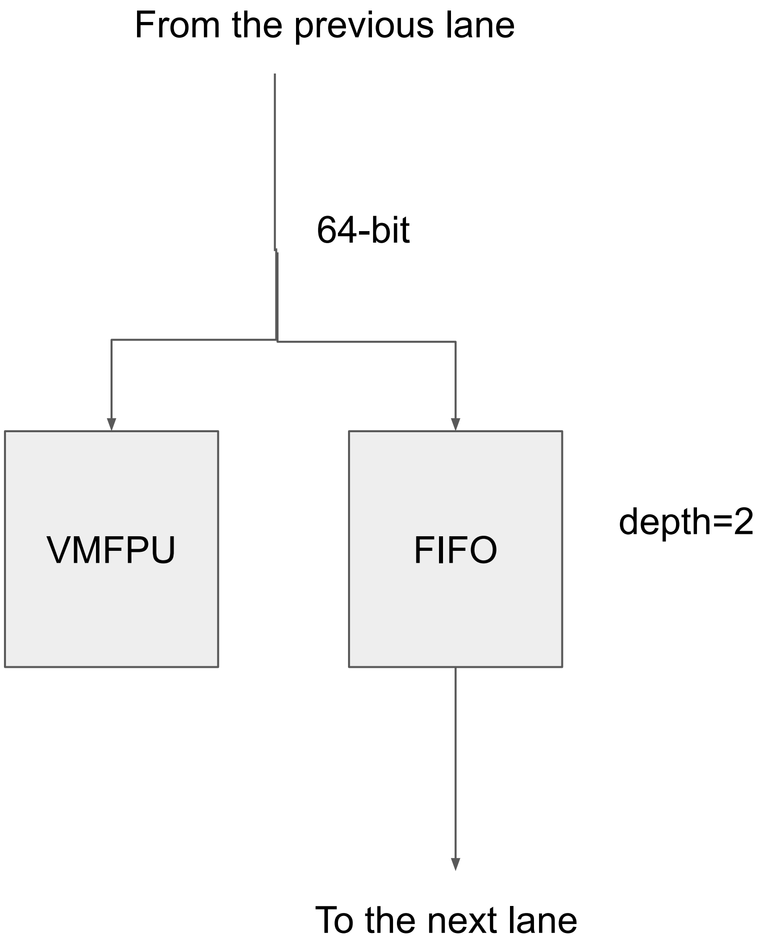
Algorithm Update

- Avoid strided memory operations
    - Store A^T instead of A
    - Load A with unit-stride load
        - utilization 90.66% → 92.28%
    - Use burst write to store result (TODO)
        - Memory bandwidth = 32 x NrLanes
        - length = NrLanes
        - Address alignment?
    - Performance of Softmax & LayerNorm ↑
        - strided → unit-strided
        - 1.7%
    - Performance of ReLU & Dropout ↓
        - can still use unit-strided
        - 0.3%

- Use different registers to load matrix A
    - false data dependency

Hardware Update

- Broadcast data
    - FIFO not full & VMFPU ready → ACK
    - If the next lane is not ready, the current lane can still execute.
    - Cut the InOut path.
        - ready_o = vmfpu_ready & ready_next



Analysis of Decreasing Utilization

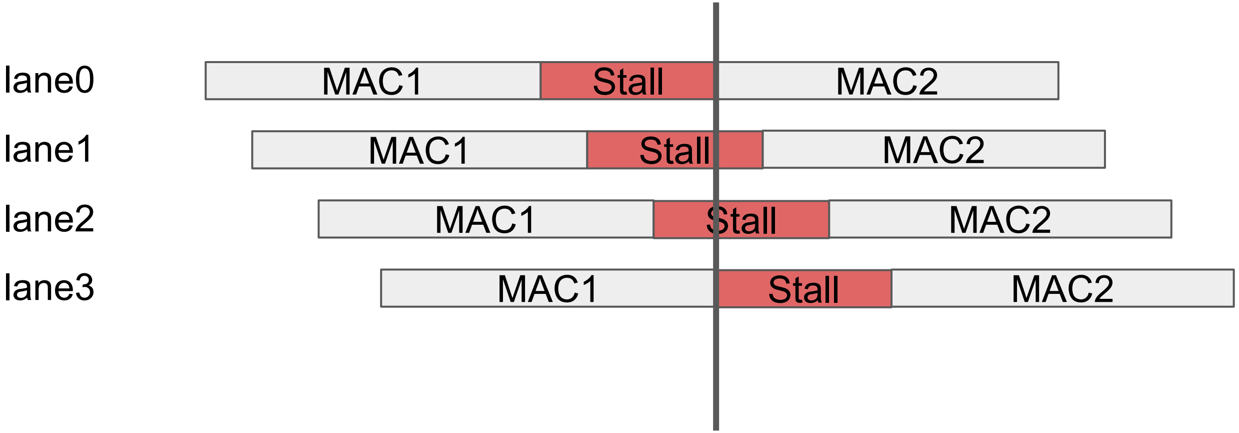
- NrLanes ↑ Utilization ↓
- MAC1 writes to vd, MAC2 reads vd (RAW)
    - MAC2 depends on MAC1
    - MAC2 can only receive operands if MAC1 is done

	bc_matmul
4 lanes	92.28%
8 lanes	85.63%
16 lanes	74.85%

Analysis of Decreasing Utilization

- NrLanes ↑ Utilization ↓
- MAC1 writes to vd, MAC2 reads vd (RAW)
  - MAC2 depends on MAC1
  - MAC2 can only receive operands if MAC1 is done

	bc_matmul
4 lanes	92.28%
8 lanes	85.63%
16 lanes	74.85%



lane0
MAC1
Stall
MAC2
lane1
MAC1
Stall
MAC2
lane2
MAC1
Stall
MAC2
lane3
MAC1
Stall
MAC2