HARDWARE ACCELERATED SORTING

CPE-593 APPLIED DATA STRUCTURES AND ALGORITHMS – (FALL 2022)

| 23 |
| 43 |
| 34 |
| 26 |
| 63 |
| 20 |
| 27 |
| 13 |
| 2 |

4-element sorter

| 26 | 34 | 43 | 63 |
FIFO

Finite state machine

4-element sorter

| 2 | 13 | 20 | 27 |
FIFO

>?

MUX 2:1

| 2 | 13 | 20 | 26 | 27 | 34 | 43 | 63 |

Gaurav Dubey
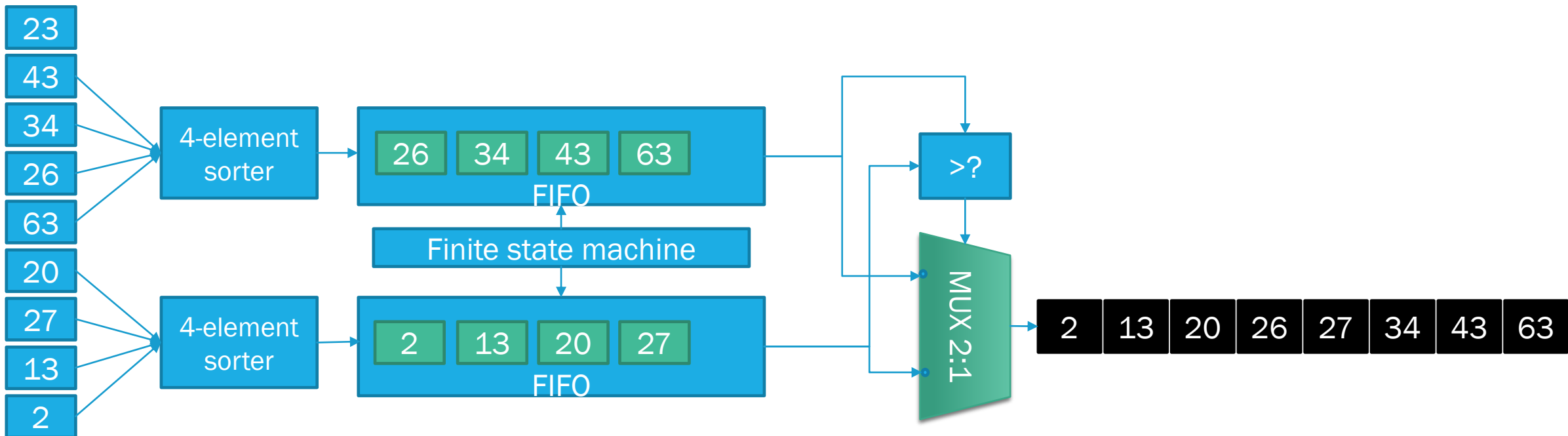
PROPOSED FEATURES (WILL DROP SOME DEPENDING ON PROGRESS)

- The digital logic will be capable of sorting 2^N elements stored in the main memory. (N>=2)

- The hardware will only require start address of the array which contains unsorted data and length of array.

- Each element of the array must be 128 bits wide, 64 bits for C++ pointer to the class on which we apply sort, 32 bits for class member variable which is used for sort and rest 32 bit as reserved.

- A programmable parameter to configure if the class member variable is signed or unsigned.

- The hardware starts the sorting when signaled and can be paused at anytime for debugging.

- The hardware signals when it is done with sorting and when it is requested to pause the sorting.

- The hardware can work on the same space allocated for input array i.e. over-write the contents of the input array in the memory and still function properly.

HARDWARE ARCHITECTURE
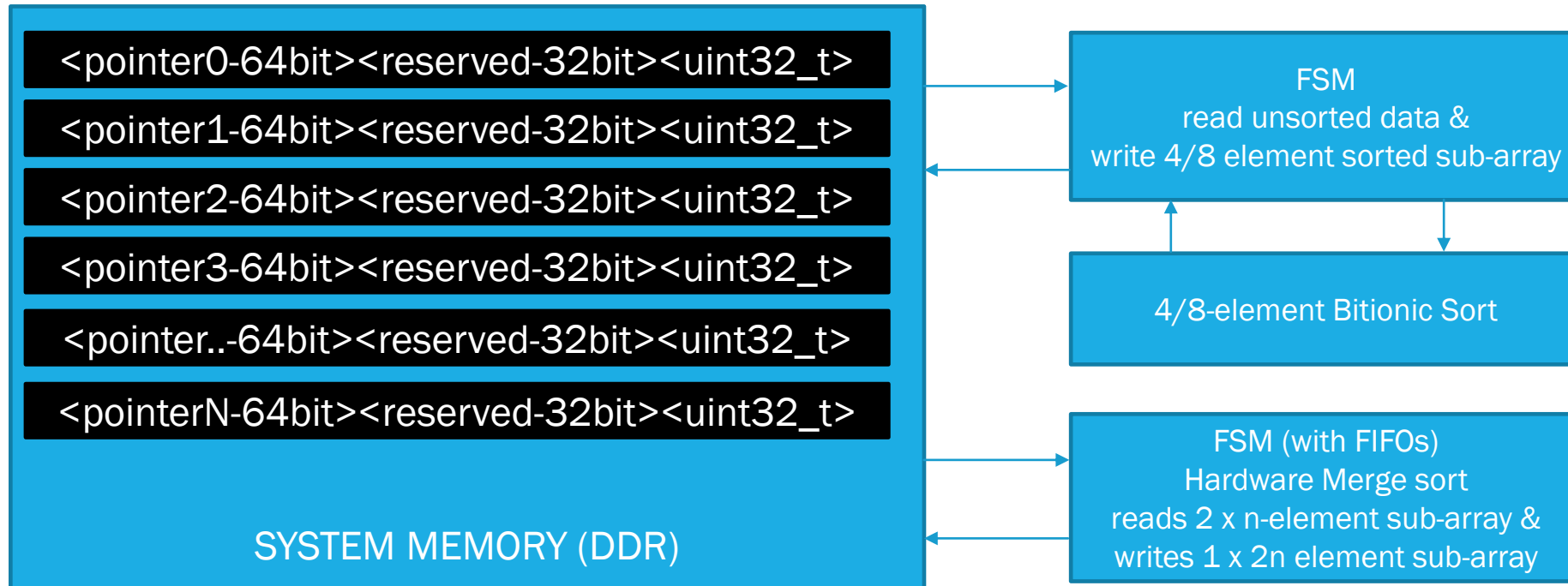
4/8 ELEMENT VECTOR PROCESSOR (BITONIC SORT)
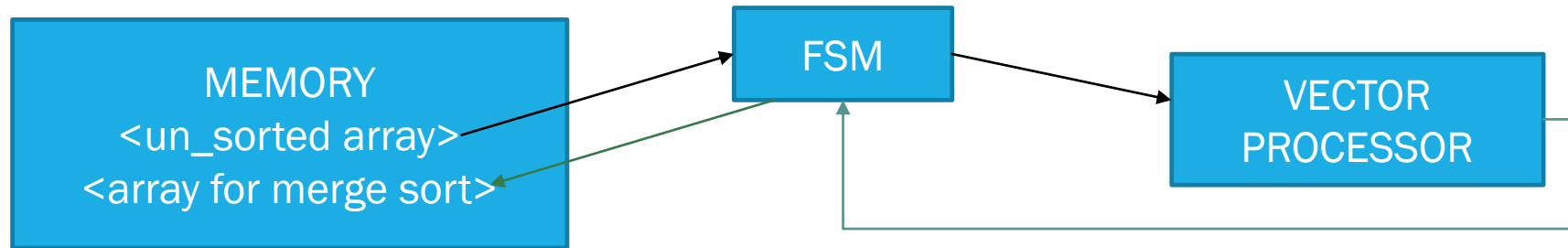
- Finite state machine (FSM) will start reading input array content until all the elements of the input array are read.

  - Once reading is complete then FSM waits until merge sorting is complete i.e. FSM will block pipeline to vector processor.

- FSM will ensure that the vector processor continuously (zero idle cycle whenever memory is available) receives 4/8 elements for bitonic sort.

- Once bitonic sort is complete for a batch of 4/8 element, the FSM stores this sorted sub-array to desired memory location for further merge sort.

HARDWARE MERGE SORT
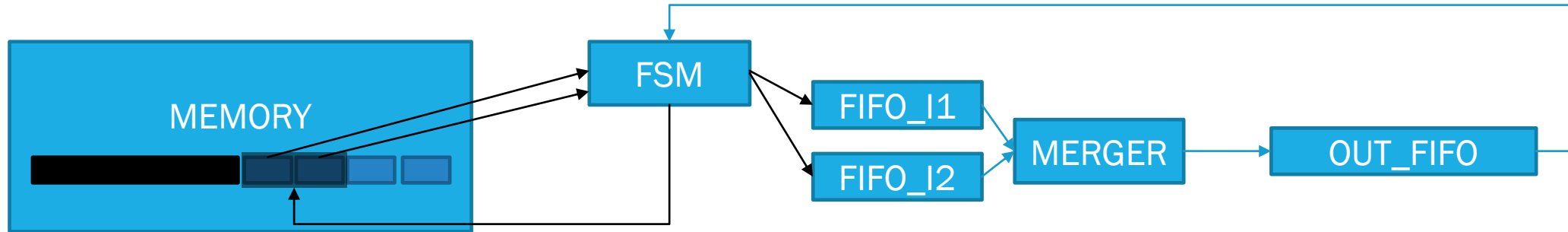
- FSM will start running merge sort algorithm as soon as vector processor is done with sorting on 2 batches 4/8 element sort.

- FSM will mimic recursive merge sort calls by filling 2^M sorted elements each in FIFO_I1 and FIFO_I2, so that merge sort algorithm can produce a sorted array of 2^{M+1} elements (buffered in OUT_FIFO).

- The merge sorted array is written back to memory so that further merge sorting can be done until the whole array is sorted.

- Finally, once sorting is complete an interrupt is generated.