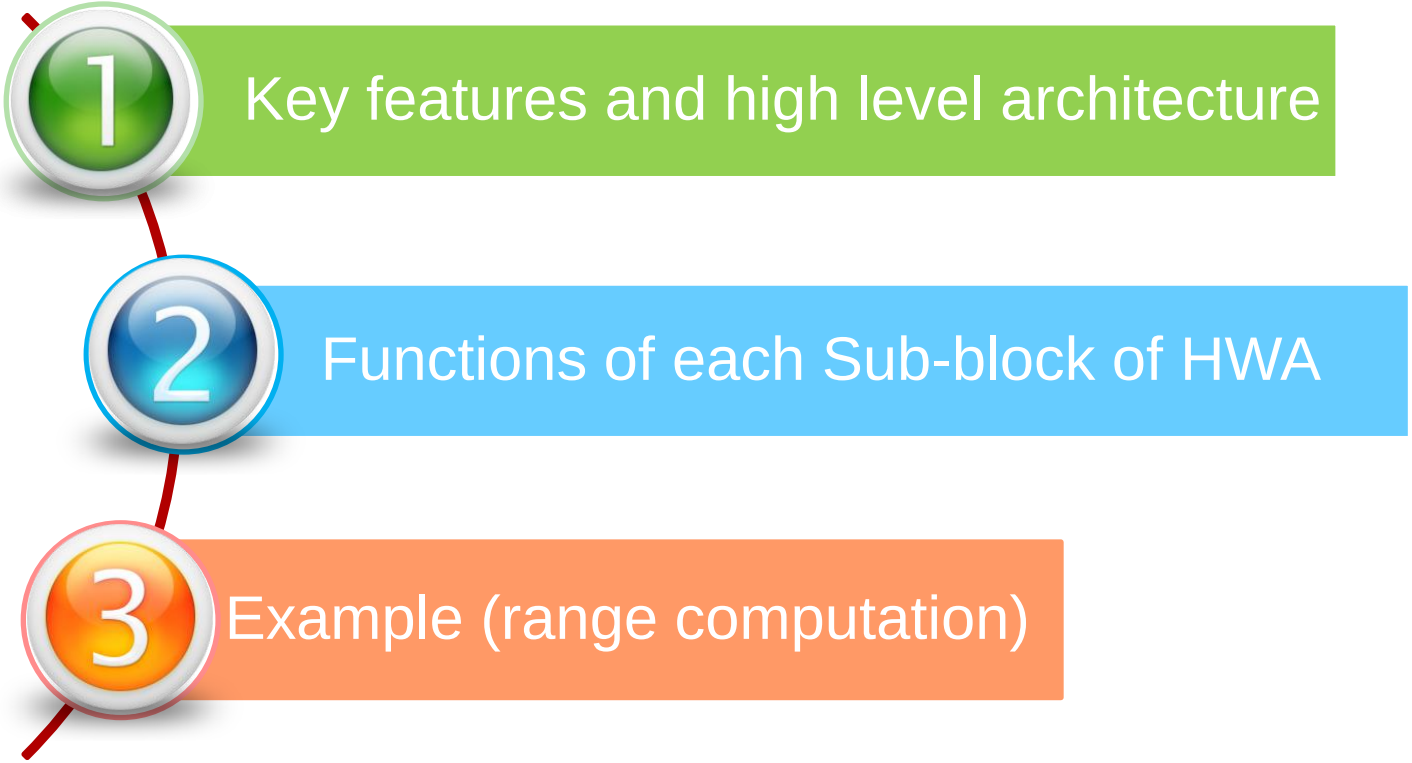
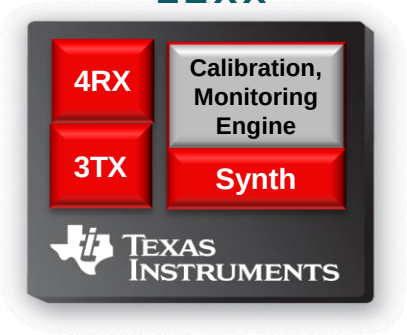
Introduction to Hardware Accelerator of mmWave Sensor 1443

Agenda


1
Key features and high level architecture
2
Functions of each Sub-block of HWA
3
Example (range computation)

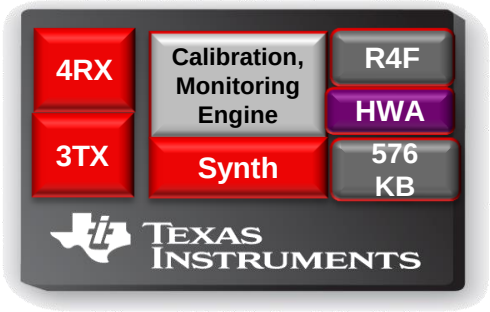
Introduction to Hardware Accelerator

12xx



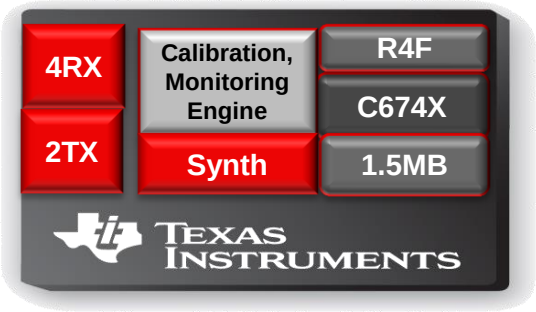
- **mmWave Sensor**

14xx



- **mmWave Sensor + Processing**

16xx



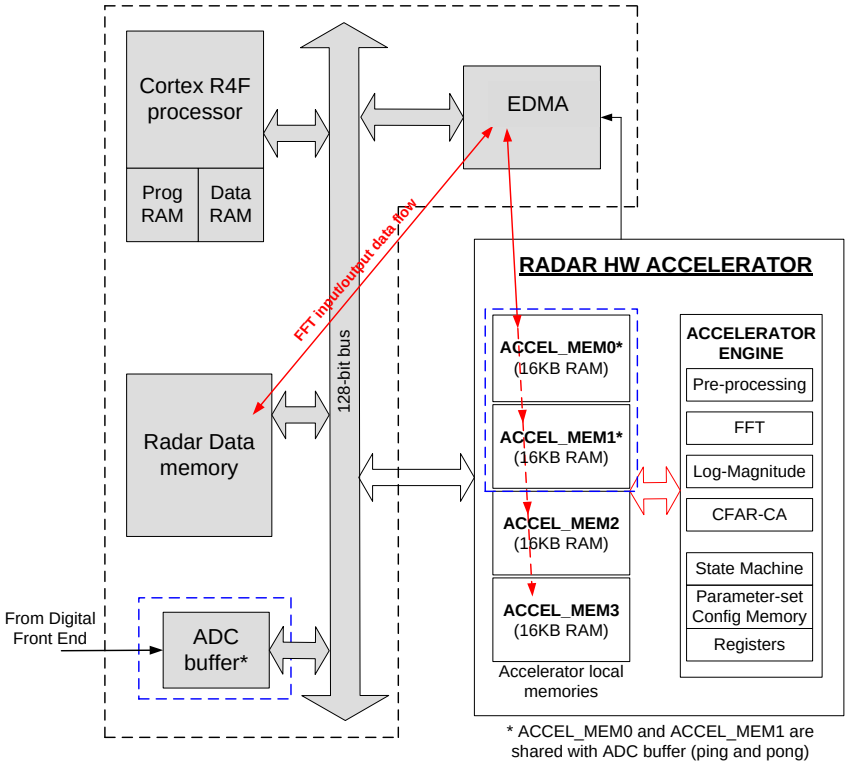
- **mmWave Sensor + Processing with DSP**

- A hardware IP introduced in mmwave sensor SOC 1443
- Off-loading the burden of signal processing from the main processor
- To obtain radar image across the range, velocity and angle dimensions

Key features

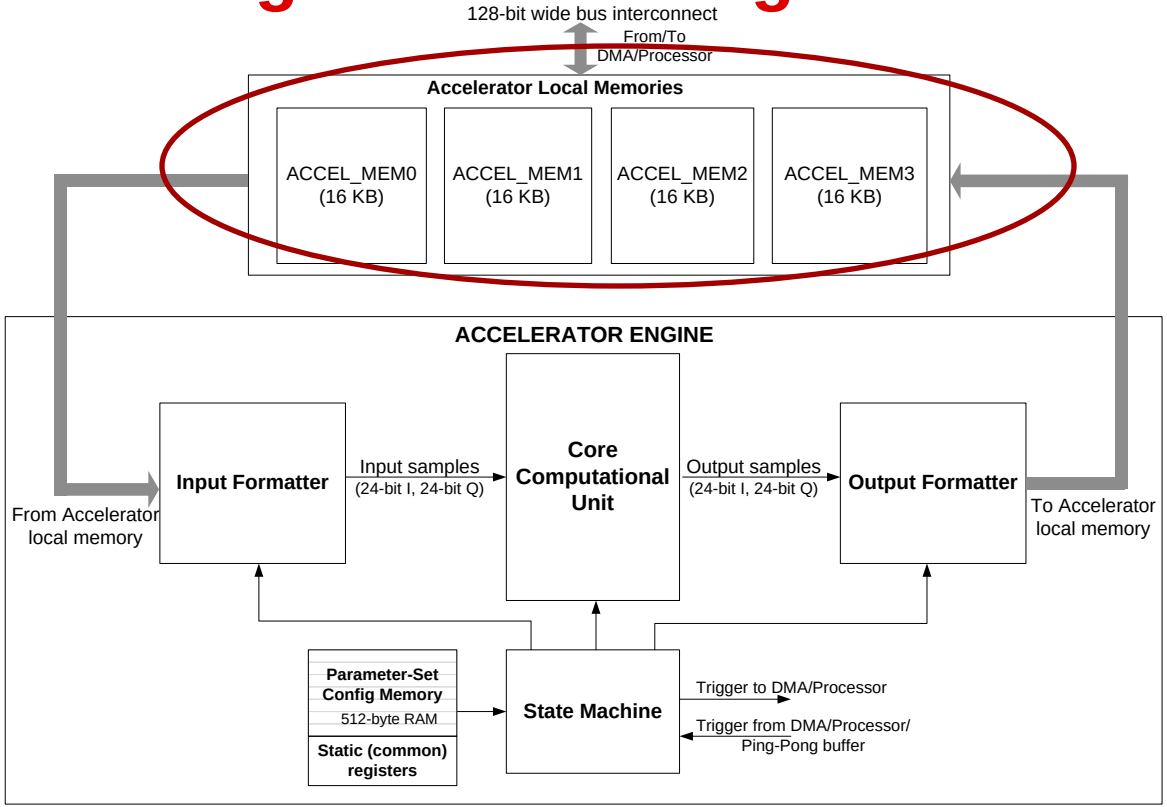
- Fast FFT with programmable FFT size up to 1024-pt
- Internal FFT bit-width of 24 bits (for each I and Q) for good SQNR performance
- Built-in capability for simple pre-FFT processing (windowing, BPM removal, interference zeroing out and complex multiplication)
- Magnitude (absolute value) and Log-Magnitude computation capability
- Flexible data flow and data sample arrangement; transpose access
- Chaining and looping mechanism along with interface to the EDMA allow the accelerator to operate with autonomously with minimal intervention from the processor
- CFAR-CA detector support, including variants CFAR-CASO and CFAR-CAGO
- FFT stitching – get equivalent 2048 or 4096-point fft

High-level Architecture



- It's connected to a 128-bit bus in the main processor system
- Typical data flow is for EDMA module to bring in samples into local memories of HWA and reads the output samples from the local memories of the HWA
- HWA and main processor operates on a single clock domain (200MHz) process one complex sample per clock cycle

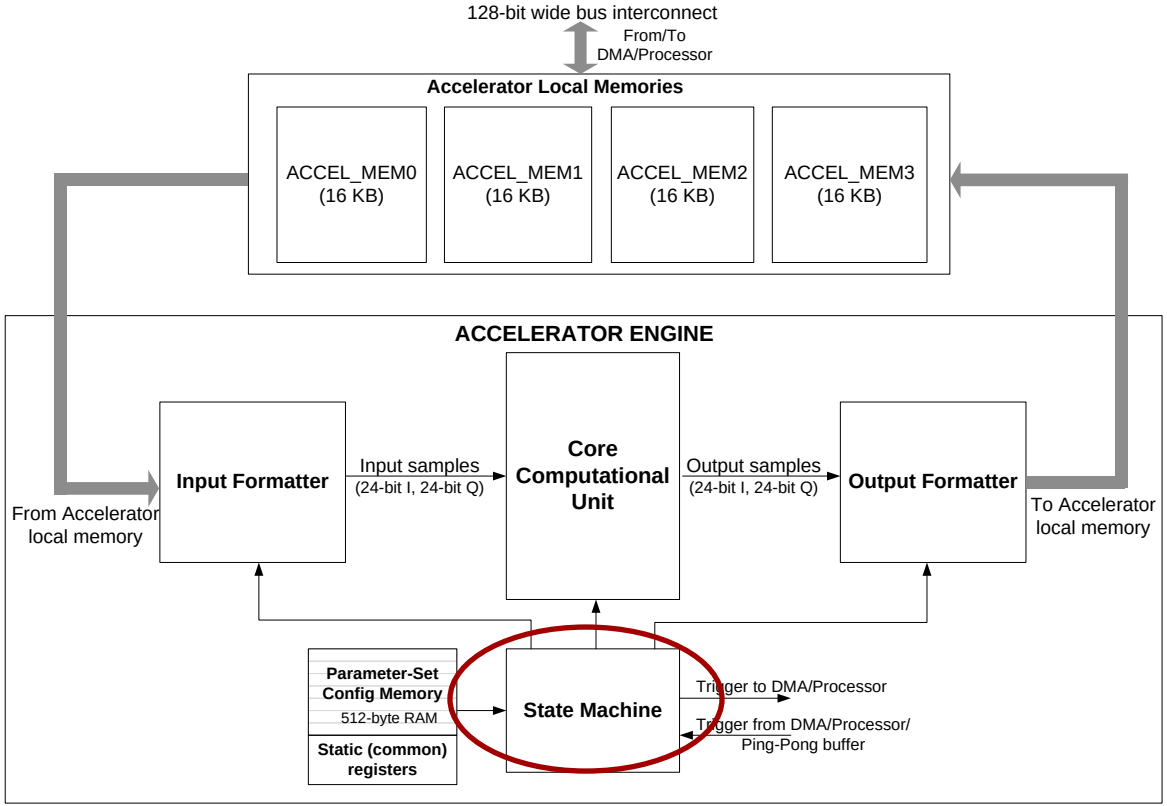
Accelerator Engine Block Diagram



Accelerator Local Memories

- Four memories, each of 16KB size (1024 words of 128 bits each)
  - ACCEL_MEM0, ACCEL_MEM1, ACCEL_MEM2, ACCEL_MEM3
- Two of the memories (ACCEL_MEM0 and ACCEL_MEM1) are directly shared with the ping and pong ADC buffers
  - No need for a EDMA transfer for 1D FFT processing,
  - When 1D FFT is complete, these memories can be used for 2D FFT processing by bringing in data via EDMA transfer
- Four memories enable “ping-pong” mechanism, for both the input and the output
- Any of the four local memories can be the “source memory” and any can be the “destination memory”, but not be both.
- Continuous address enables as large as 32KB memory when not in ping-pong mode

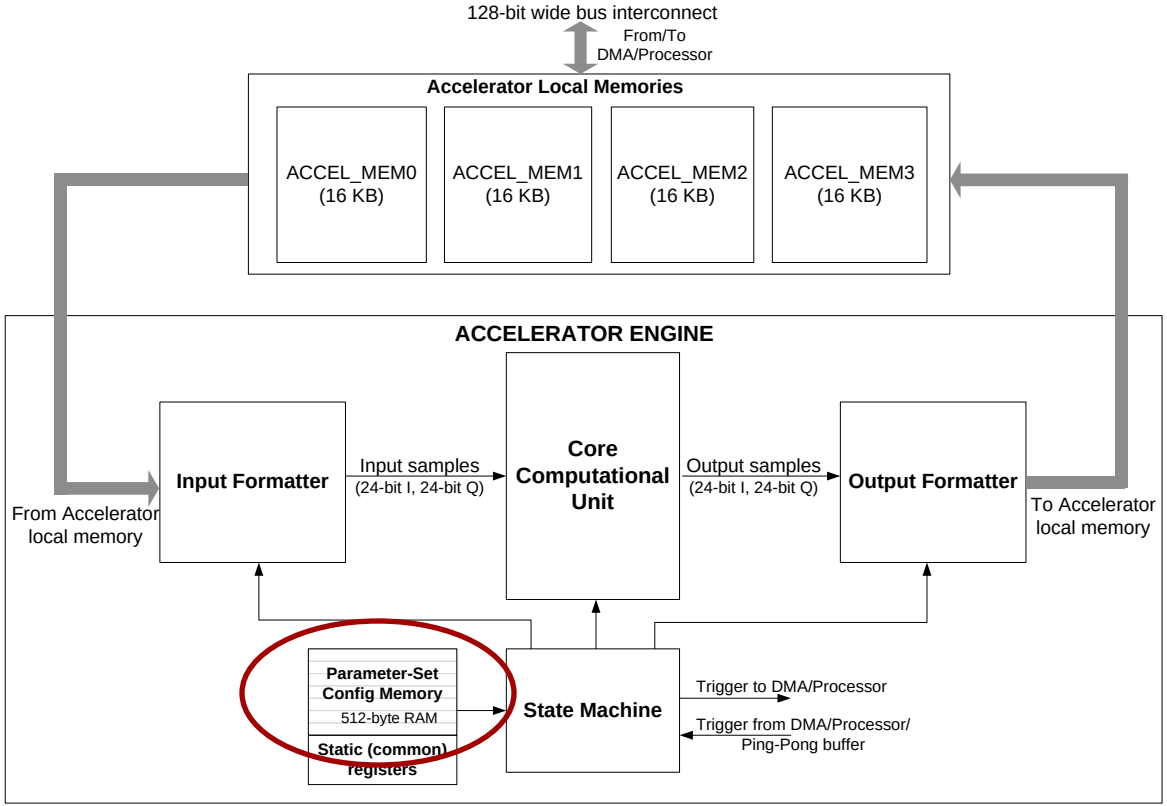
Accelerator Engine Block Diagram



State Machine

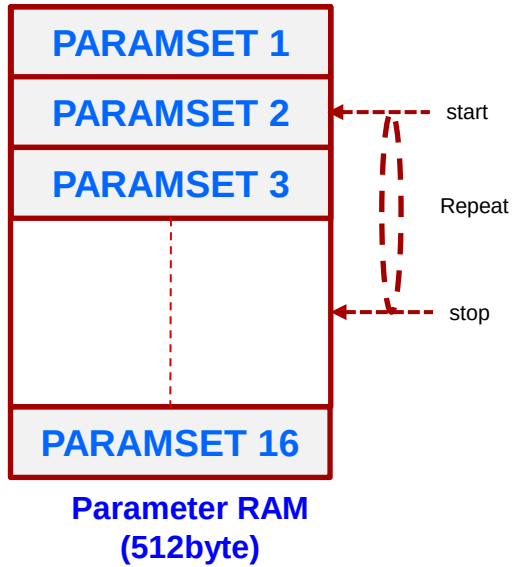
- Operation:
  - enabling/disabling of the HWA
  - Sequencing an entire set of operations and loop through operations
- Trigger Mechanisms:
  - 4 incoming trigger
    - Immediate trigger
    - Wait for processor-based software trigger
    - Wait for ADC buffer ping-to-pong or pong-to-ping switch
    - Wait for EDMA based trigger
  - 2 outgoing trigger
    - To main processor
    - To EDMA

Accelerator Engine Block Diagram

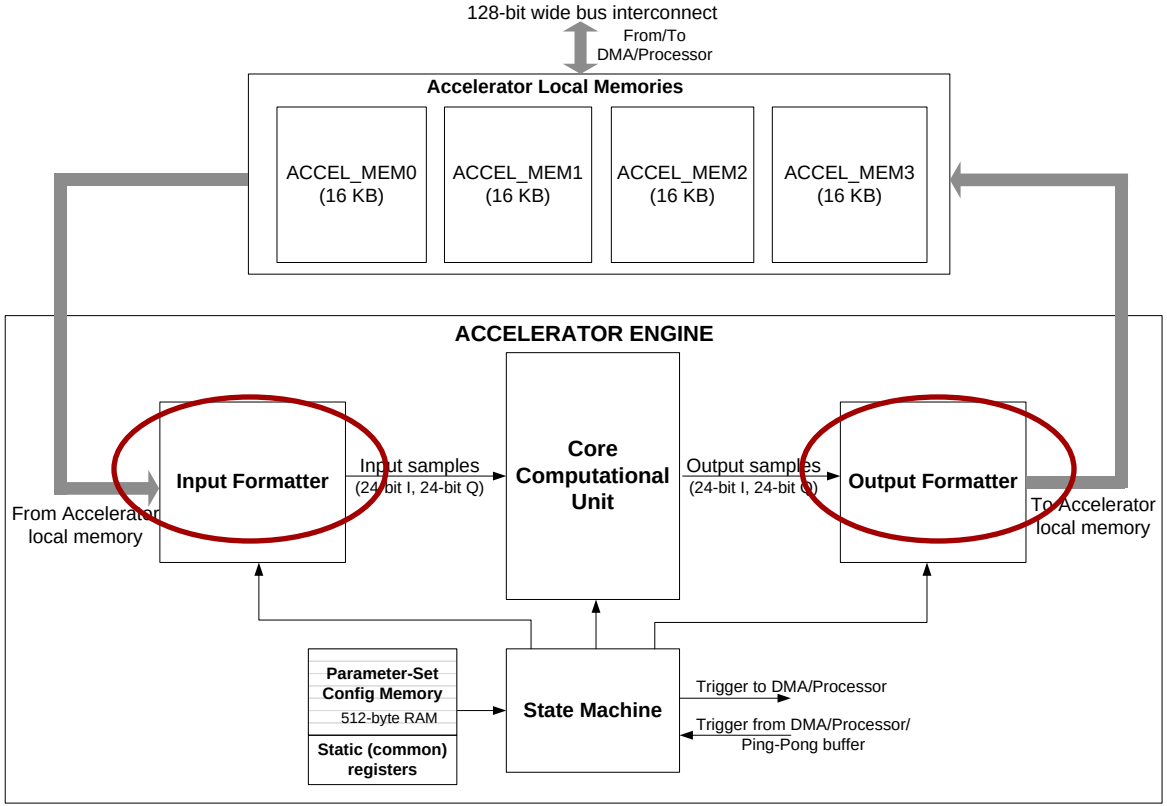


Parameter-set Configuration Memory

- Structure:
  - 16 parameter sets
  - Each set consists of eight 32-bit registers
- State machine can be programmed to loop through a specific subset of parameter sets
- Functions:
  - Choosing and configuring each of the blocks in the computational engine
  - Configuring the input/output data format
  - Configuring the 2D-memory indexing
  - Configuring the trigger for the parameter set

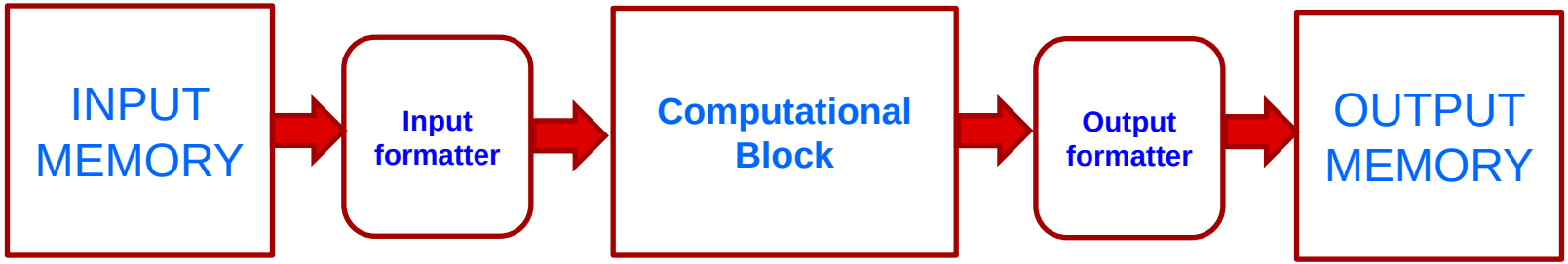


Accelerator Engine Block Diagram



Input/Output Formatter

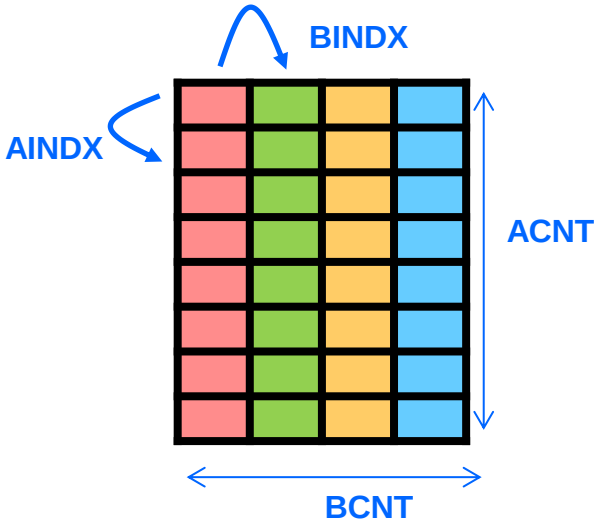
- Accept various data formats: real/complex , 16/32-bit , signed/unsigned
- Supports data scaling/formatting at input/output
- Input formatter supports zero-padding



- BPM removal is accomplished in the input formatter block

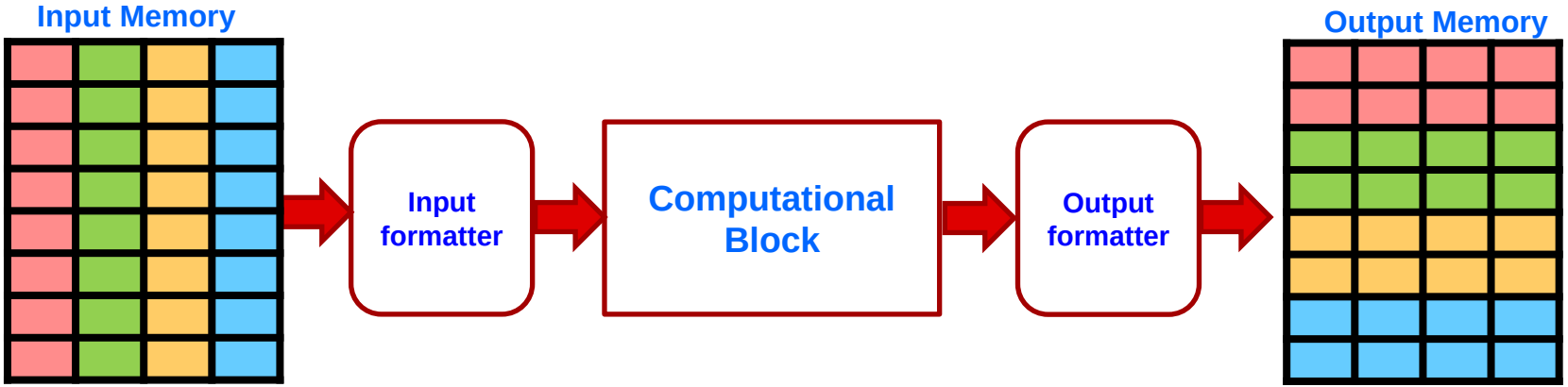
Input/Output Formatter: 2D Memory indexing

- Flexible data streaming is made possible
  - ACNT : Number of samples (sample count)
  - AINDX : Separation between consecutive samples
  - BCNT : Number of iterations
  - BINDX : Offset per iteration
- Data streaming is separately configurable at the source and destination:
  - SRC_ACNT, SRC_AINDX, SRC_BINDX, DST_ACNT, DST_AINDX, DST_BINDX
- BCNT is the same for both source and destination.



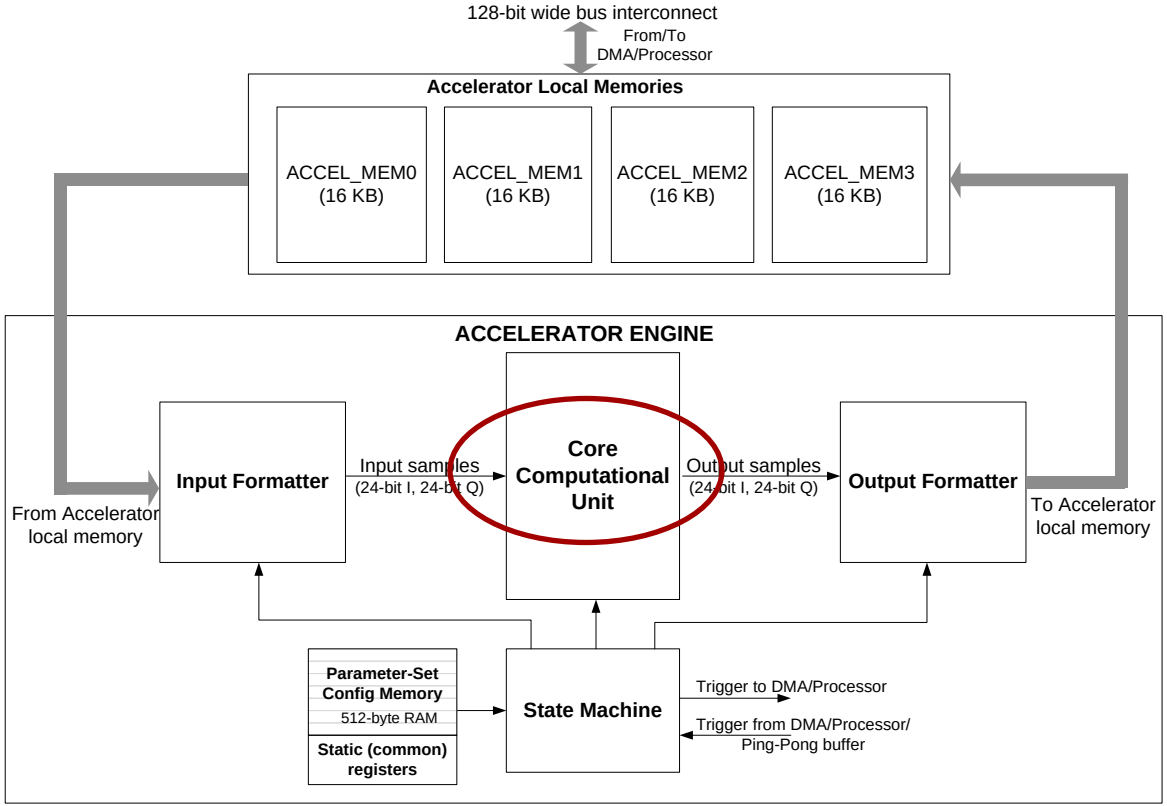
Input/Output Formatter: 2D Memory indexing

- Input/Output data need not to be placed contiguously (see input memory)
- Easy to repeat the same computation across multiple data vectors



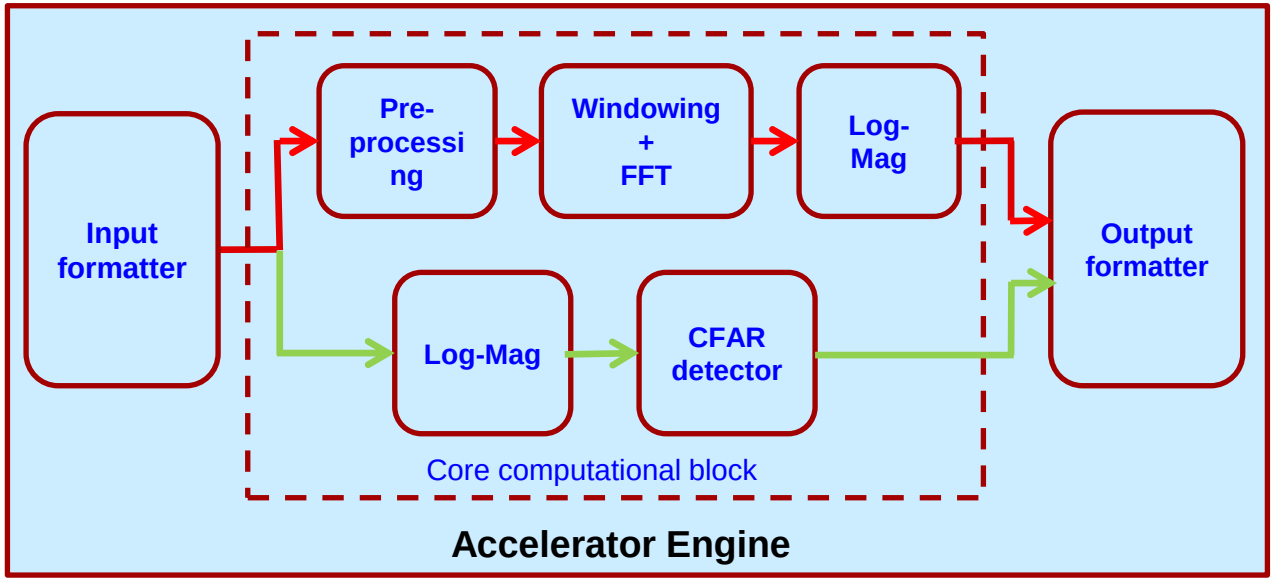
In this example, the same computation (say an FFT) , can be performed across 4 data streams.
This can be achieved with a single programming of the accelerator.

Accelerator Engine Block Diagram

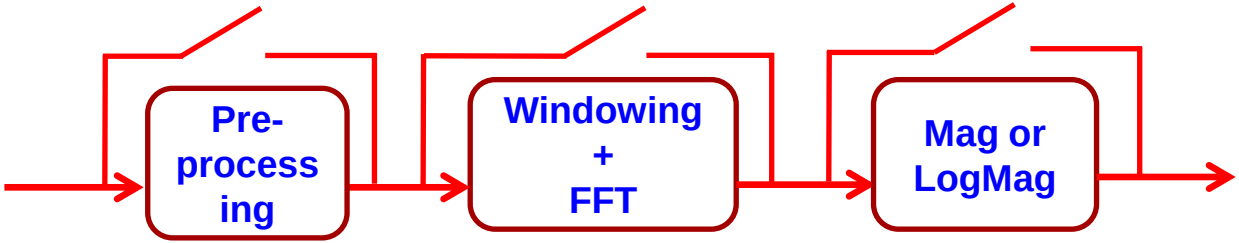


Core Accelerator Engine

- FFT Engine path and CFAR Engine path
- Only one of these two paths can be operational at any given instant



Hardware Accelerator : FFT Path

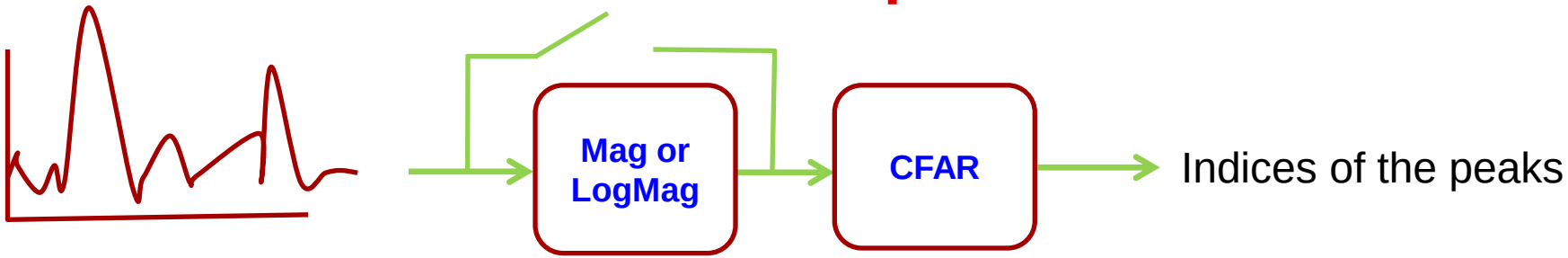


- Perform range/doppler/angle FFT’s
- In a one sample per clock streaming manner
- By-pass option controlled through parameter-set registers

Hardware Accelerator : FFT Path

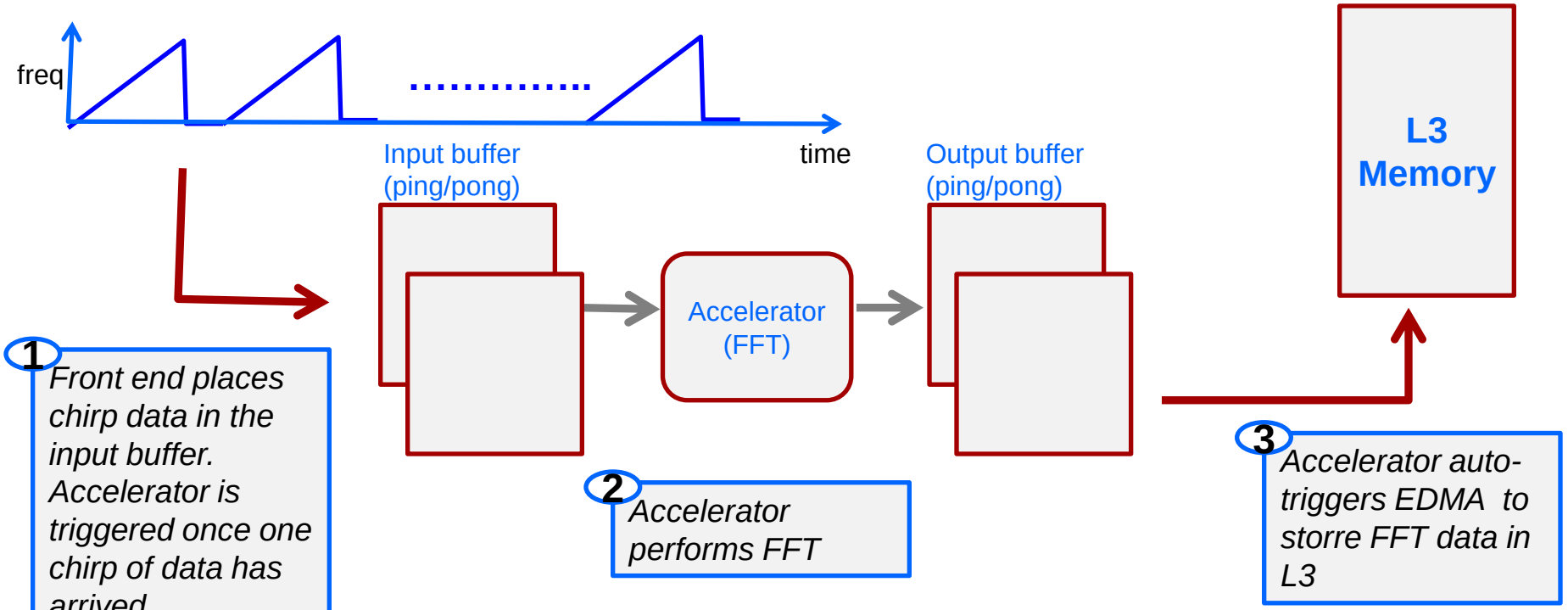
- Pre-processing:
  - Interference zeroing out (zero out samples that are larger than a specified threshold)
  - Complex Pre-multiplication, to support functions such as
    - Complex de-rotation of antenna phases
    - DFT mode: frequency shifter with auto-increment mode
  - FFT stitching mode
- Windowing+FFT:
  - Dedicated Window RAM for window coefficients
  - FFT sizes supported power of 2 until 1024
- Post-processing – Magnitude/Log-Magnitude:

Hardware Accelerator : CFAR path



- Cell averaging (CA) is used for threshold calculation
- This path implements a CFAR detector on a real input vector
  - Option of taking Mag/LogMag prior to CFAR (in case input is complex)
- The output is a list of indices corresponding to detected peaks
- Various popular flavors of the CFAR algorithm are supported (CFAR-CA, CFAR-CAGO, CFAR-CASO)

Example: Range computation (1D FFT)



Operations proceed in a ping/pong fashion : When the front end is filling up the ping buffer, the accelerator processes the pong buffer

Example: Range computation (1D FFT)

PARAMSET 1 CONFIG:

- Is triggered by availability of ADC data
- Picks data from ping input buffer
- Computes 4 range-FFTs
- Places FFT data in pong output buffer
- Trigger an EDMA channel (say edma_chan_1) to transfer data to L3