ADAS: AN INTRODUCTION TO S32 DESIGN STUDIO FOR VISION IDE AND VISUAL GRAPH TOOLS

MIKE DOIDGE

TOOLS APPLICATION SPECIALIST
AMP TOOLS ENGINEERING

AMF-AUT-T2813 | AUGUST 2017

SECURE CONNECTIONS
FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2017 NXP B.V.
PUBLIC

AGENDA

- S32V Vision Pipeline
- Overview of ISP/APEX2
- S32DS for Vision IDE
- ISP Graph Tool
- ISP Graph Tool Demo
- APEX2 Graph Tool
- APEX2 Graph Tool Demo
- Q&A

S32V234 The Vision Pipeline

Each engine offers the best efficiency for certain type of functions. To let the complete system work at highest efficiency, each engine needs to work in parallel in pipeline mode.

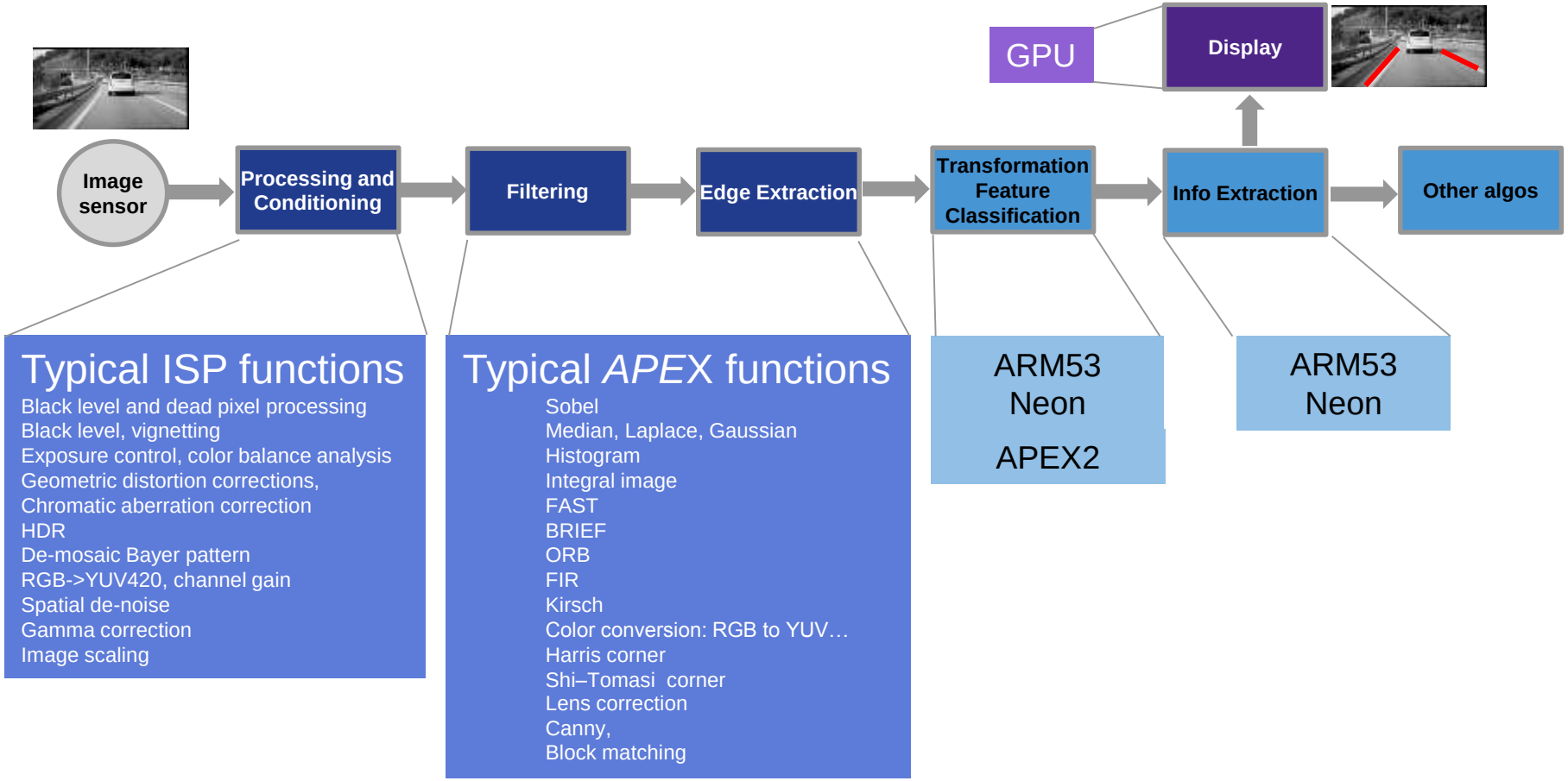
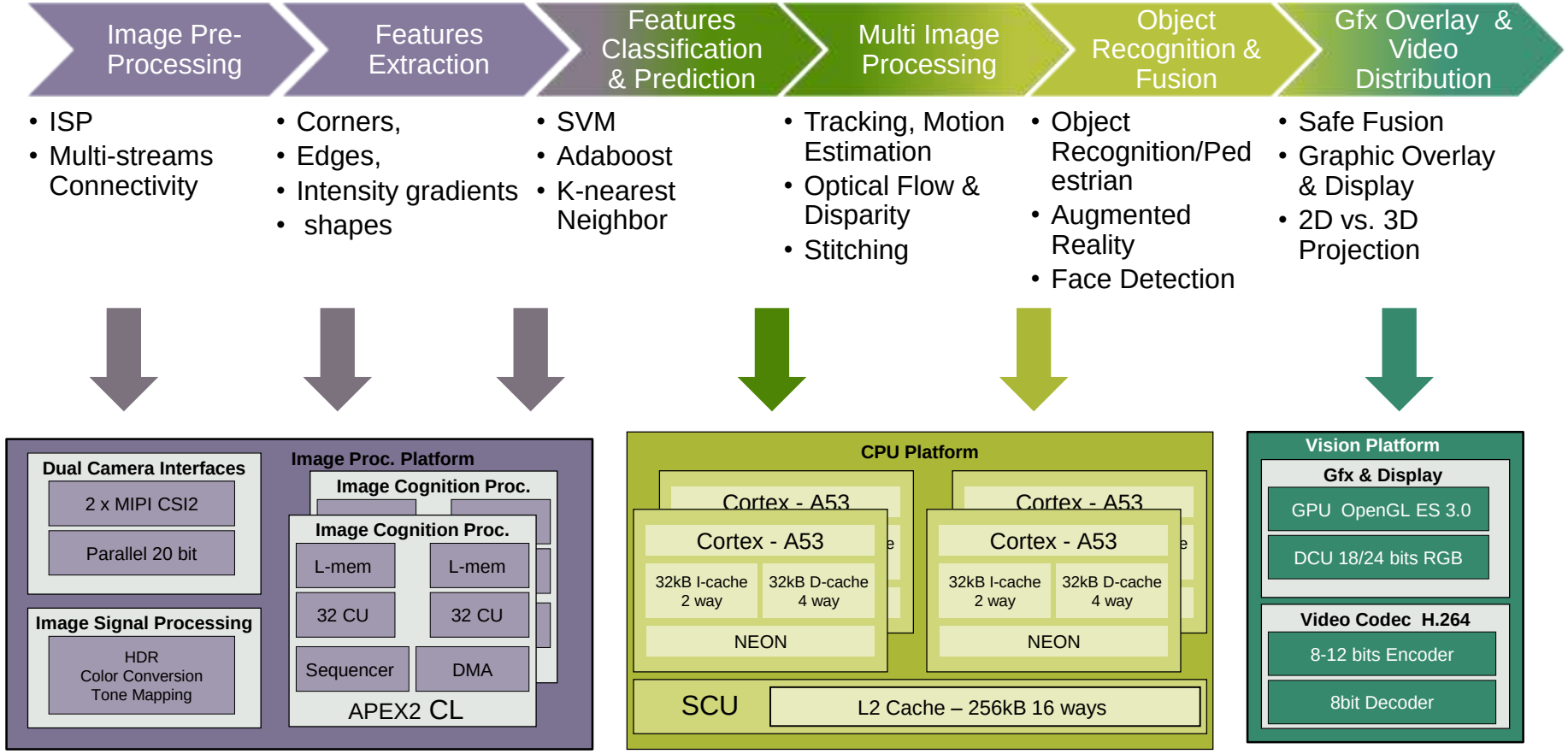


Image Processing & Data Fusion



High BW Operations
Scalable MIMD Local Memory
Soft ISP

Scalable RISC – Data Fusion
SIMD Co-Processor - Neon
Memory Hierarchy and coherency

Graphic
Video Codec
Smart Display

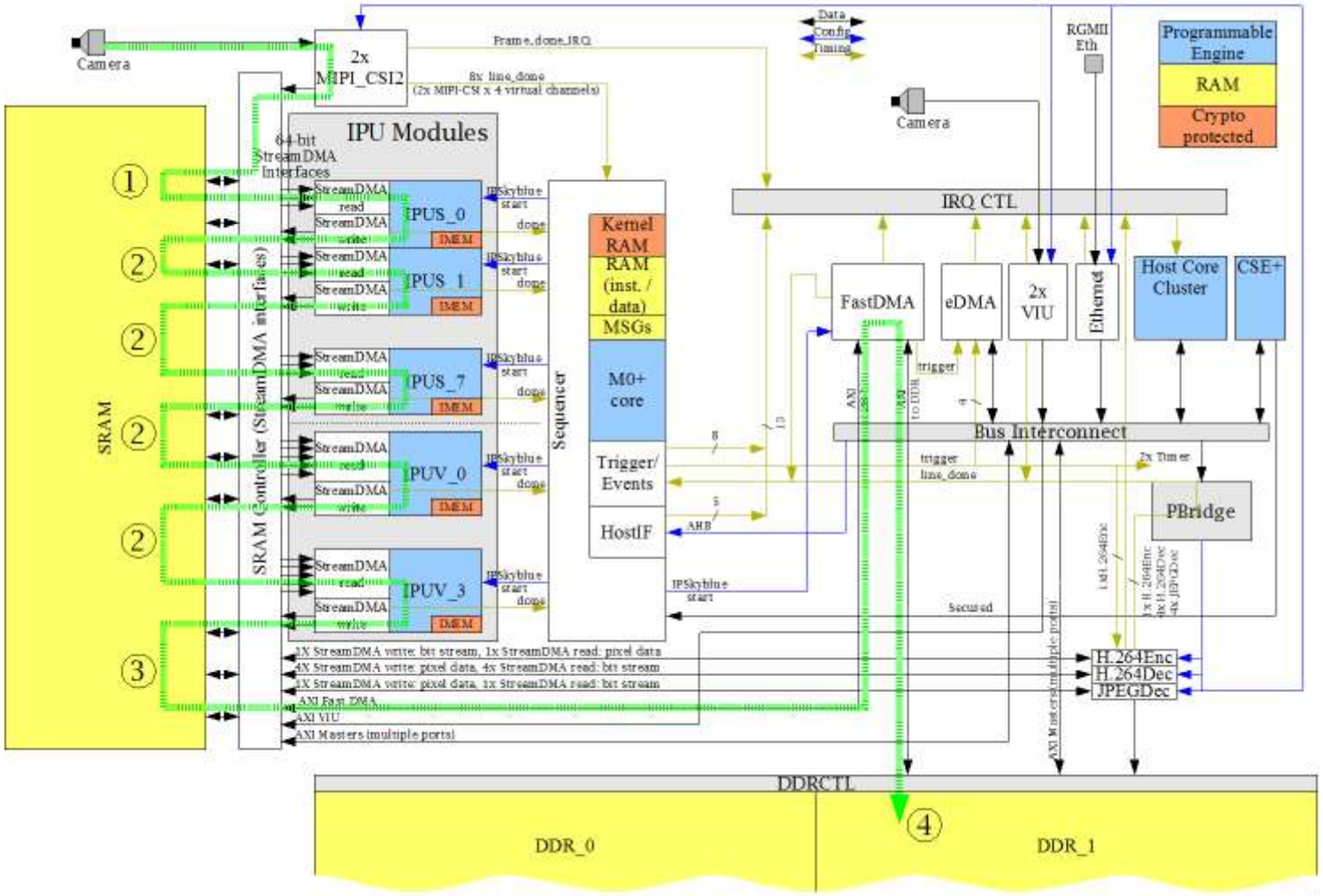
01.

ISP

ISP Overview

- Pre-processing for Vision algorithms
- Functions process the image lines sequentially and data is moved in and out of memory before/after each function executes
- Functions are executed in parallel
- Multiple functions assigned to separate engines (or IPUs – image processing units)
- Multiple engines in MIMD(multiple instruction, multiple data) pipeline working on a sliding input window
- The input is filled and the output is drained using a DMA (StreamDMA engines)
- Sequencer (M0+ core) controls the action

ISP SoC Integration and Data Flow



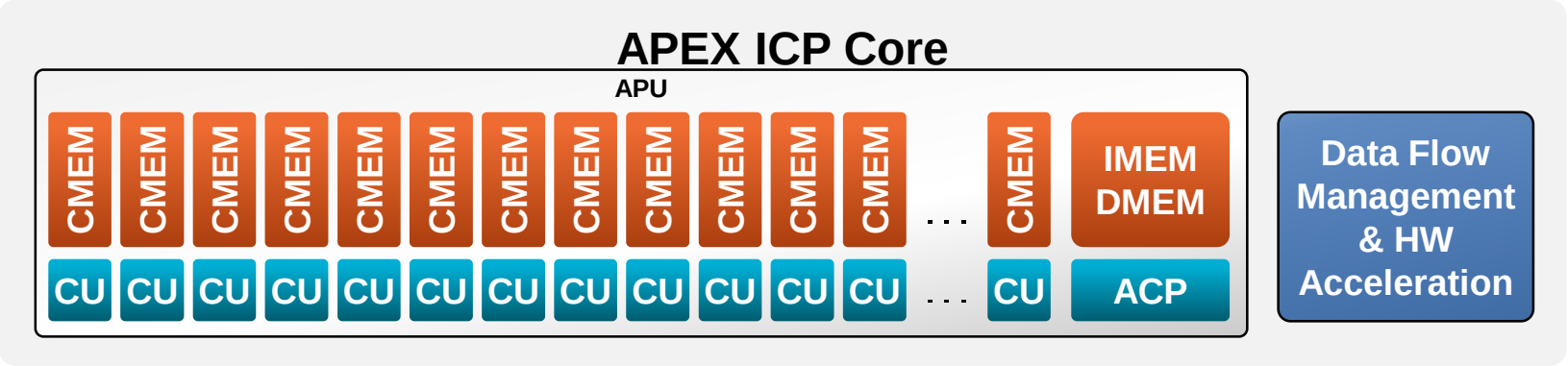
ISP Function List

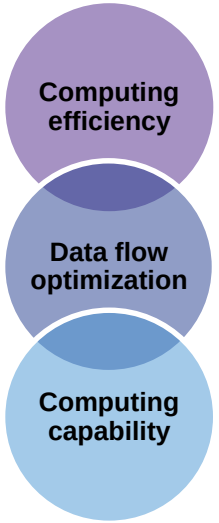
Function	Key Operation
Black Level	Averaging per channel
Dead Pixel processing	Median
Input decompression	Look-Up-Table
Black level, exposure measurement, color balancing	Histogram, channel gain
Vignetting	LUT
HDR	Weighted sum from individual exposures
De-mosaic Bayer	Interpolate missing values
RGB->YUV444, color calibration	Matrix multiplication, factors based on color calibration
Spatial de-noise	FIR and edge aware threshold
Gamma correction, Tone Mapping	LUT
YUV444 planar to YUV422 interleaved	Re-arranging samples
Optional: Image Scaling	Anti-alias (FIR), bi-linear sub-sampling

02.

APEX

Image Cognition Processor (ICP) Architecture



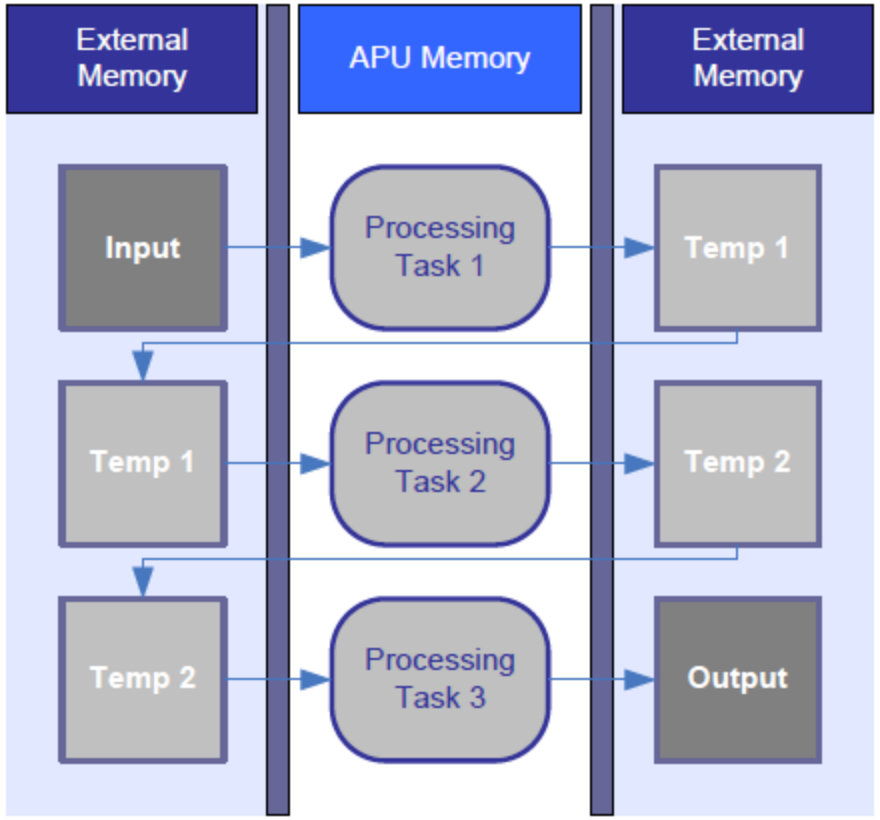


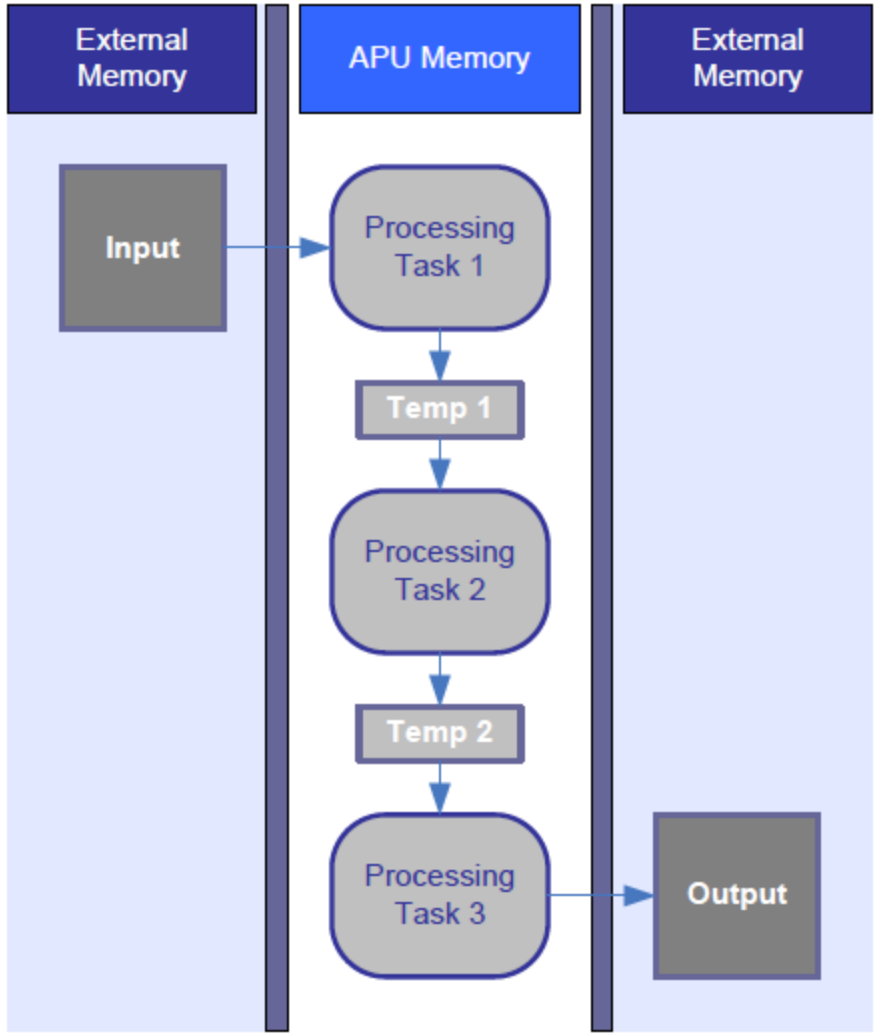
- APU is a massively parallel processor
  - Single Instruction Multiple Data (SIMD)
  - Computational Units (CU) + local dedicated memory (CMEM)
  - Array Control Processor (ACP)
- ACP manages execution and data movement
  - Dispatches instructions to the different CUs
  - Performs address decoding
  - Execute scalar operations
- Data Flow Management
  - Data transfers via DMA
  - HW acceleration

Programming APEX2 core

- C/C++ high level programming language
  - Compiler handles the appropriate generation of scalar/vectors unified instructions and manages registers and memory references
  - The scalar processing unit is programmed liked any other MCU
  - The vector processing unit handles vector data with equivalent scalar instructions
  - Dedicated vector instructions also exists such as vif, vany, vall, vget, vput, as well as optimized intrinsics for some operations
- NO ASSEMBLER required
  - Compiler has internal optimized inline assembler instructions as well as in-code compiler annotations such as for loop optimization
- Optimized Vision Library
  - APEX-CV provides high level algorithms and low level kernels for the APEX core to speed up vision application development

Minimizing data movement





APEX2 Typical Functions

- Sobel X
- Median
- Histogram
- Integral Image
- FAST
- BRIEF
- ARB
- FIR
- Kirsh
- Laplace filter
- Gaussian
- RGB to YUV
- YUV 2 RGB
- YUV 420 to RGB565
- Harris Corner
- Shi–Tomasi Corner
- Lens Correction

03.

S32DS for Vision

S32 Design Studio for Vision (S32V)

S32 Design Studio Development Environment for S32V234 by Core

Assembler
ISP Graph Tool
Compiler
Debugger
(Lauterbach)
Source Editor
New Project Wizard
Software Examples

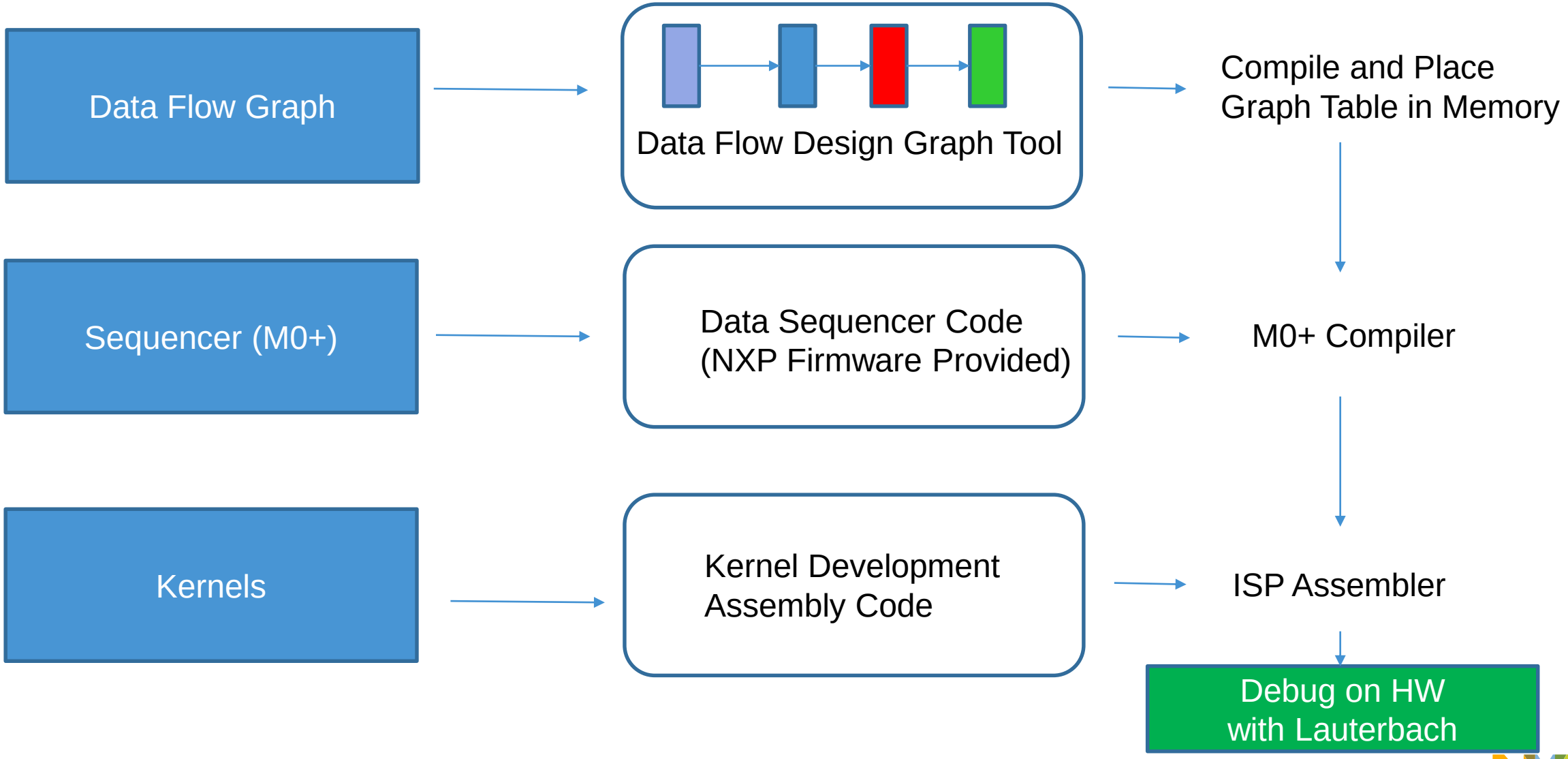
NXP APEX C/C++ Compiler
APEX Graph Tool
Debugger (Lauterbach)
Editor
New Project Wizard
Vision SDK
ACF Framework
APEX Emulation of Hardware
Example Projects

C/C++ Compiler
Debugger
(Low Cost Debugger,
Lauterbach)
Source Editor
New Project Wizard

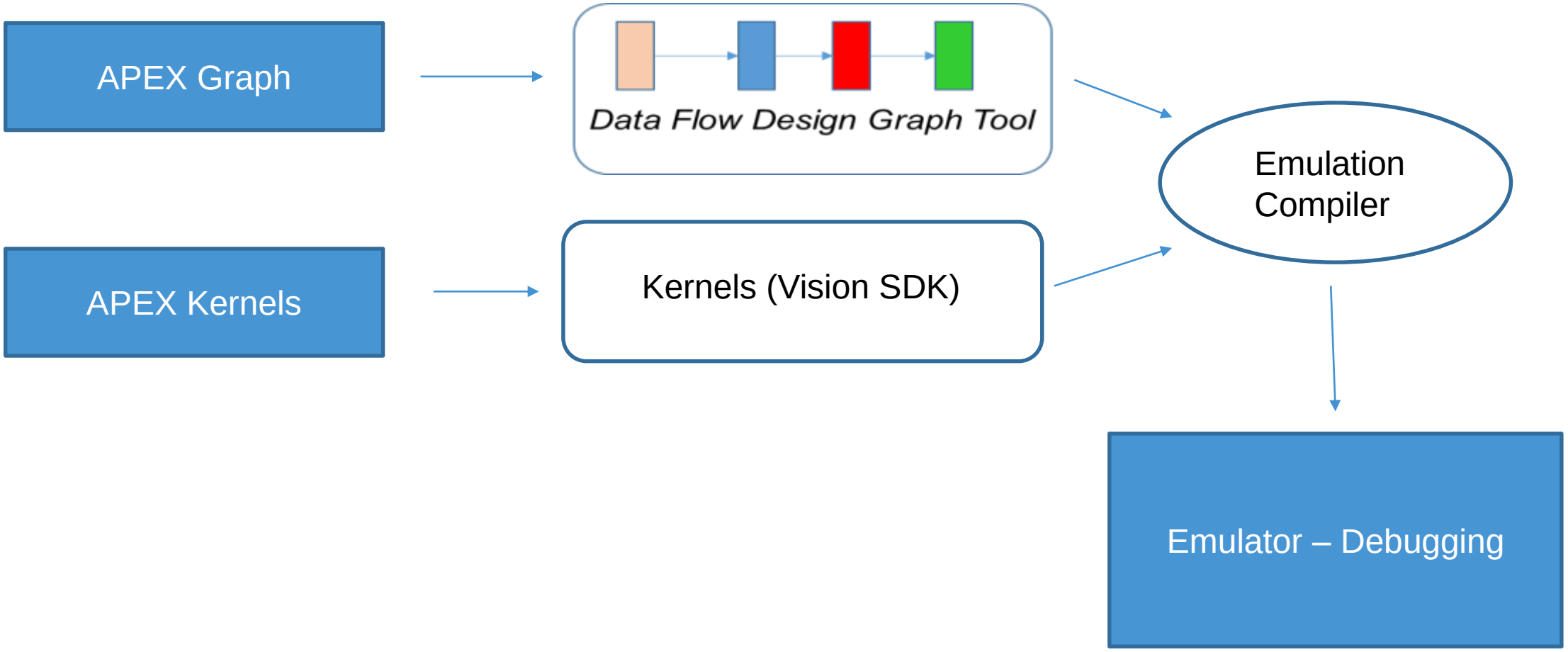
C/C++ Compiler
Multi-Core Debug Support
Task Aware Debug Support
(Low Cost Debugger,
Lauterbach)
Source Editor
New Project Wizard

Support for both IDE and Command line development on Windows and Linux platforms.

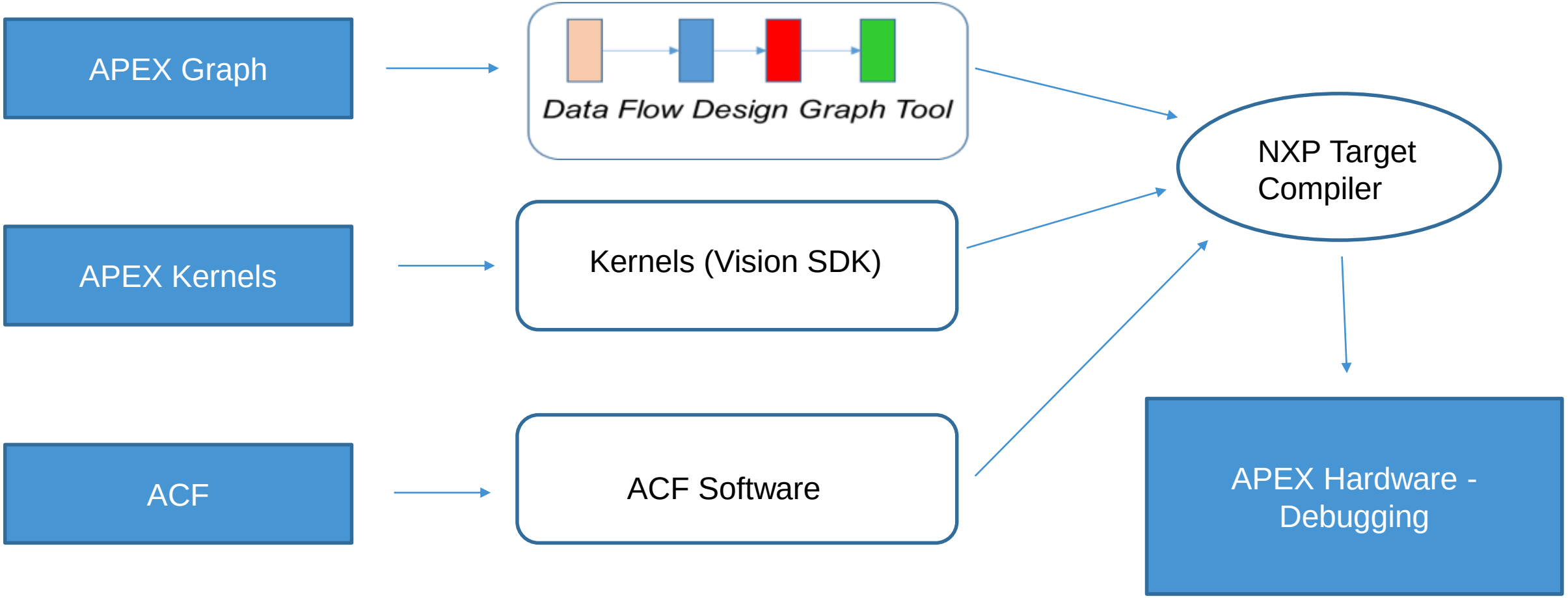
ISP Software Development Workflow



APEX Software Development Workflow to Emulator

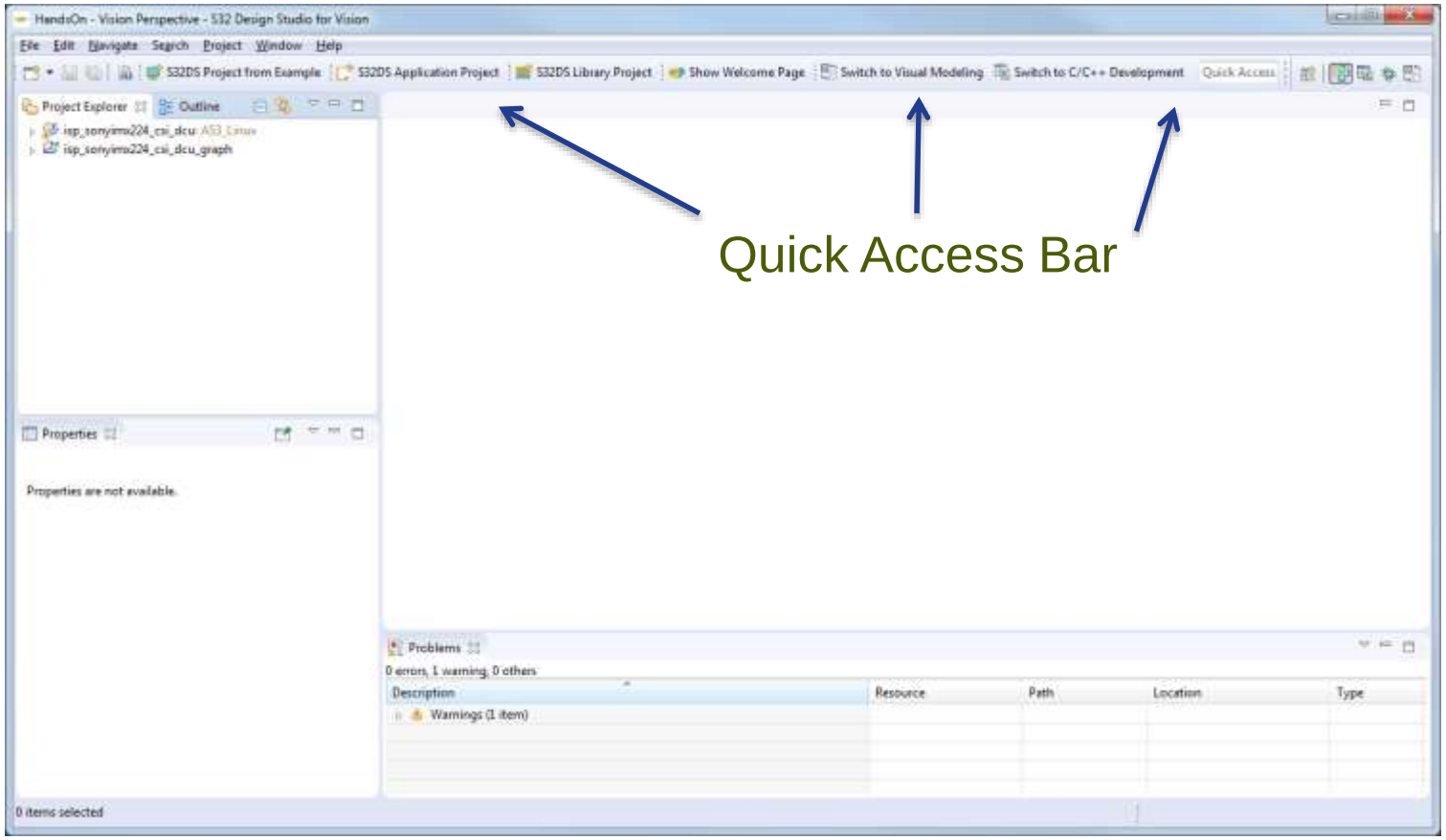


APEX Software Development Workflow to APEX



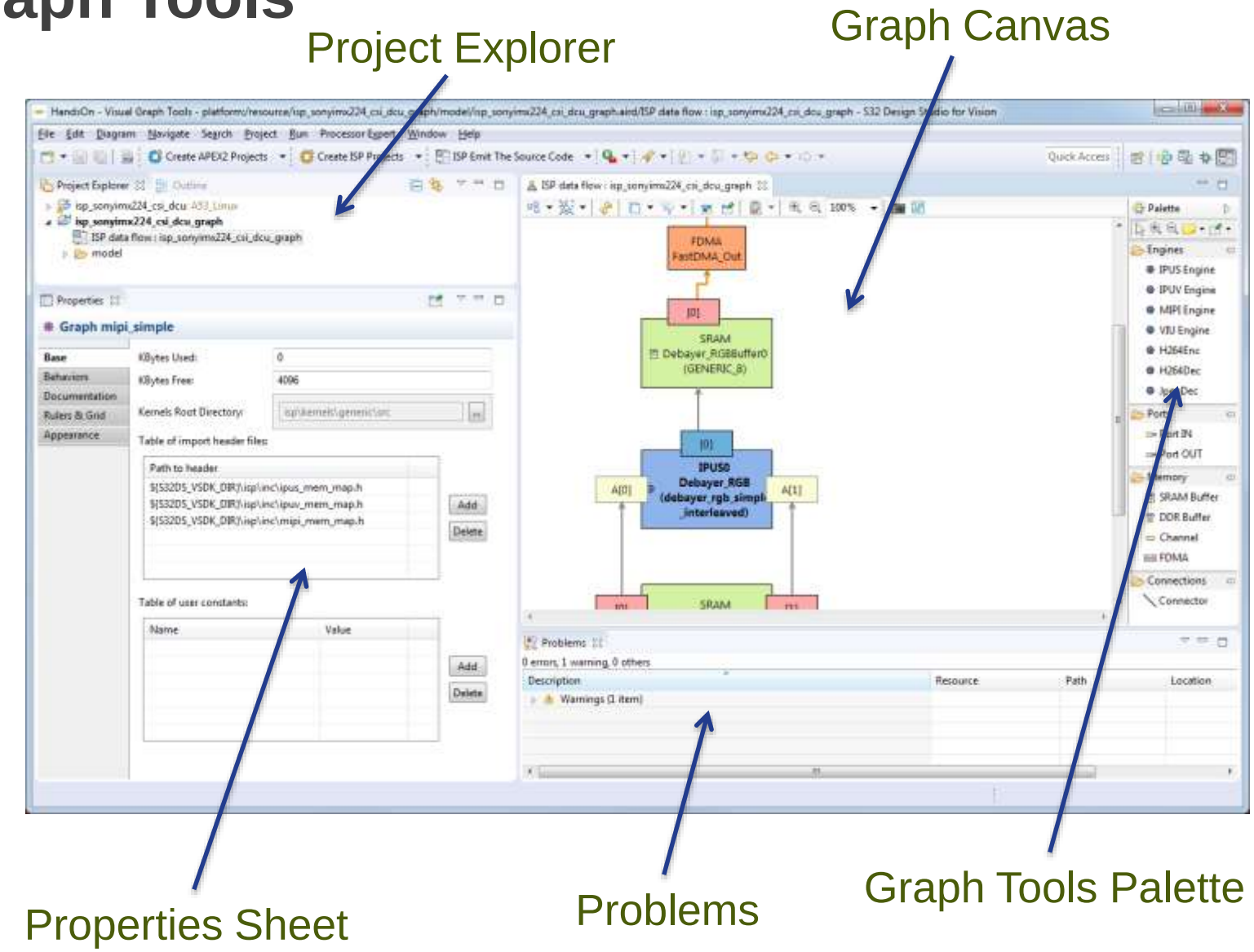
Perspectives – Vision Perspective

- Typically the first perspective you see
- Quick access bar provides buttons to launch wizards and other perspectives:
  - S32DS Project from Example
  - S32DS Application Project
  - S32DS Library Project
  - Switch to Visual Modeling
  - Switch to C/C++ Development
- Can be used for some limited development work



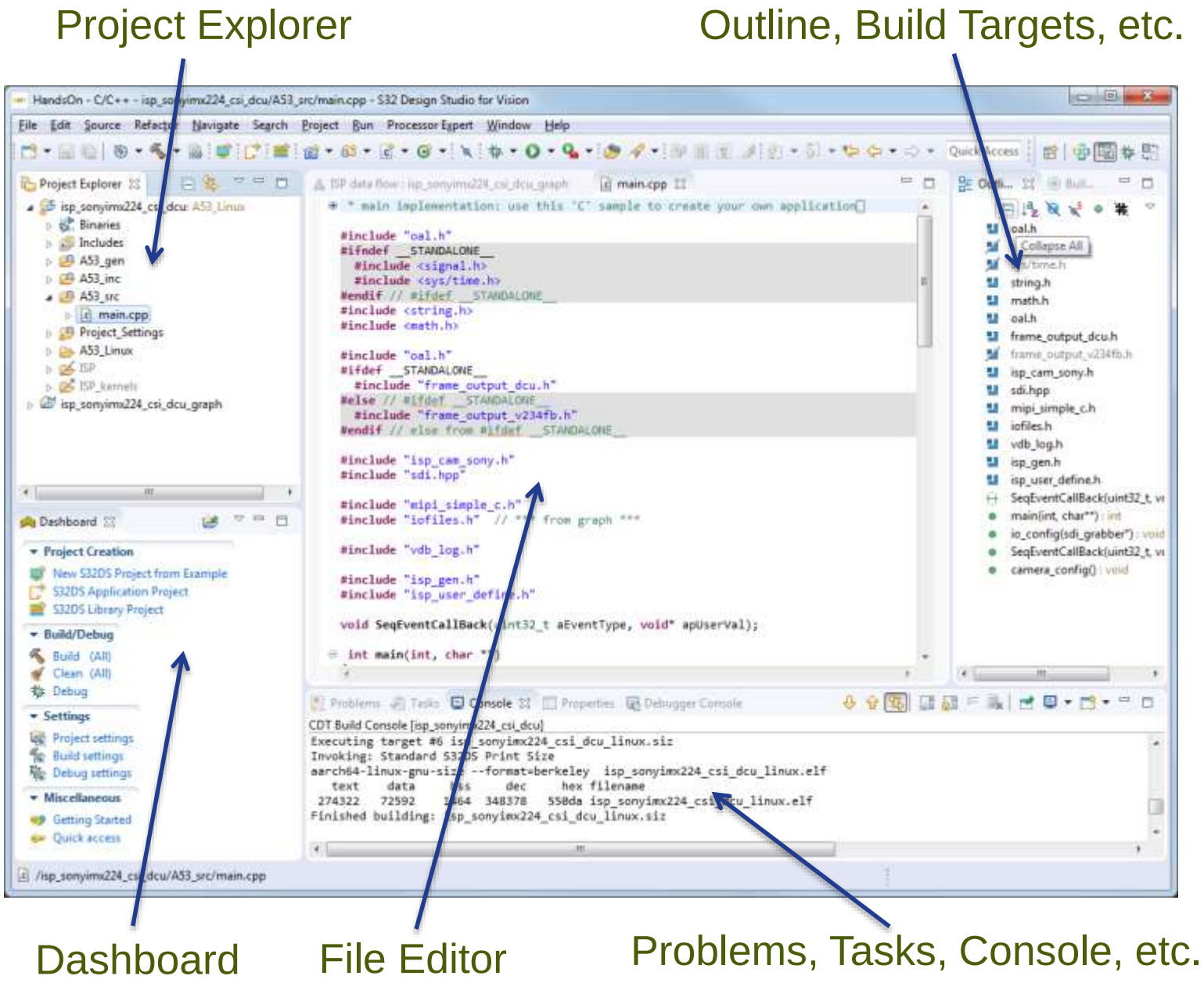
Perspectives – Visual Graph Tools

- The primary perspective for visual graph development
- Launched from quick access bar on Vision Perspective OR when new Visual Graph project is created
- In addition to the Project Explorer and Problems views, it consists of:
  - Graph Canvas
  - Graph Tools Palette
  - Properties Sheet
  - Project Explorer
  - Problems



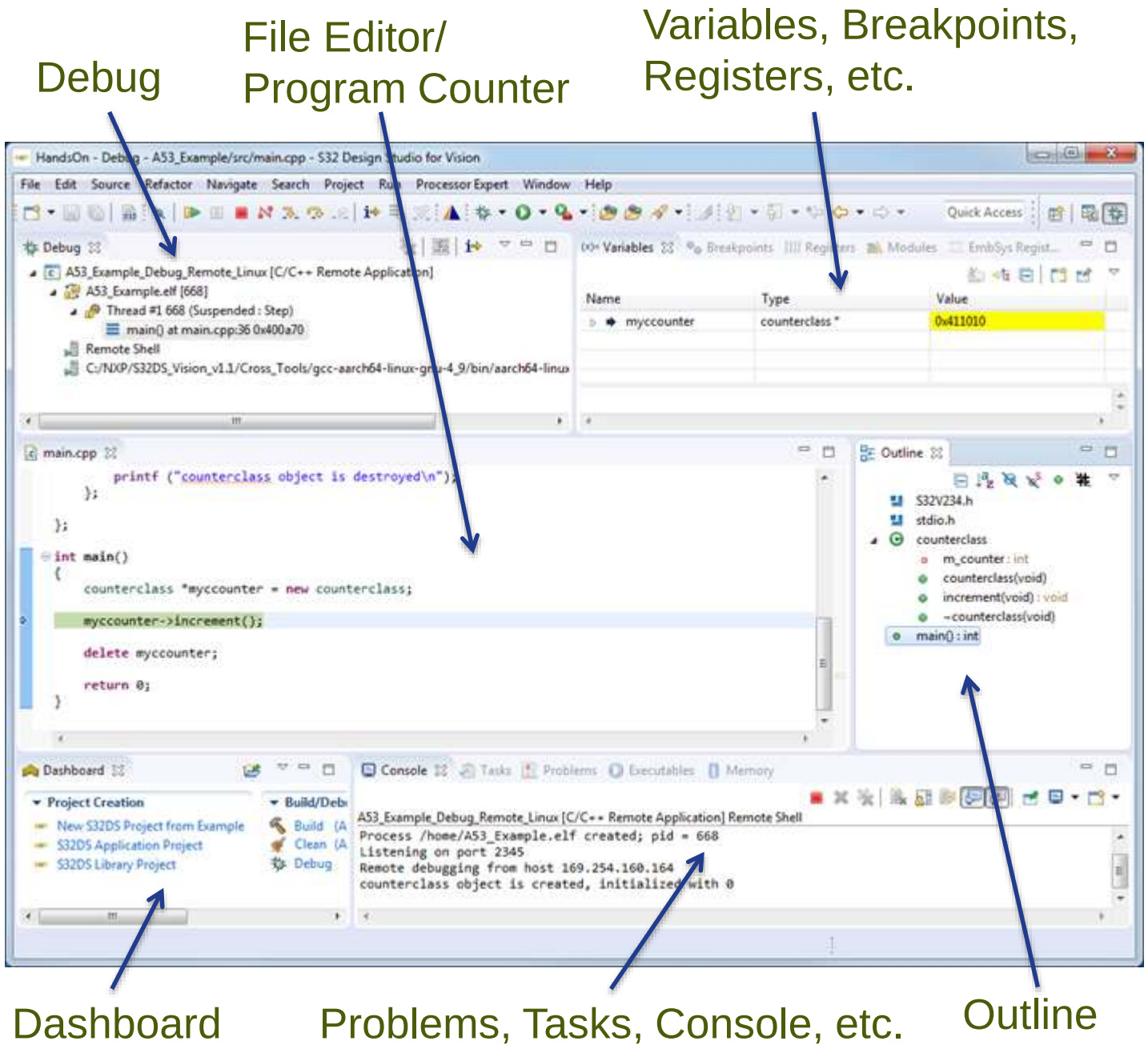
Perspectives – C/C++

- Standard S32 Design Studio Perspective
- Launched from quick access bar on Vision Perspective
- Consists of:
  - Project Explorer
  - Dashboard
  - File Editor
  - Outline, Build Targets, etc.
  - Problems, Tasks, Console, etc.
- The primary perspective for c-code development



Perspectives – Debugger

- Standard S32 Design Studio Perspective
- Launched from quick access bar on Vision Perspective
- Consists of:
  - Debug
  - File Editor/Program Counter
  - Dashboard
  - Variables, Breakpoints, Registers, etc.
  - Outline
  - Problems, Tasks, Console, etc.
- The perspective for C/ASM debugging

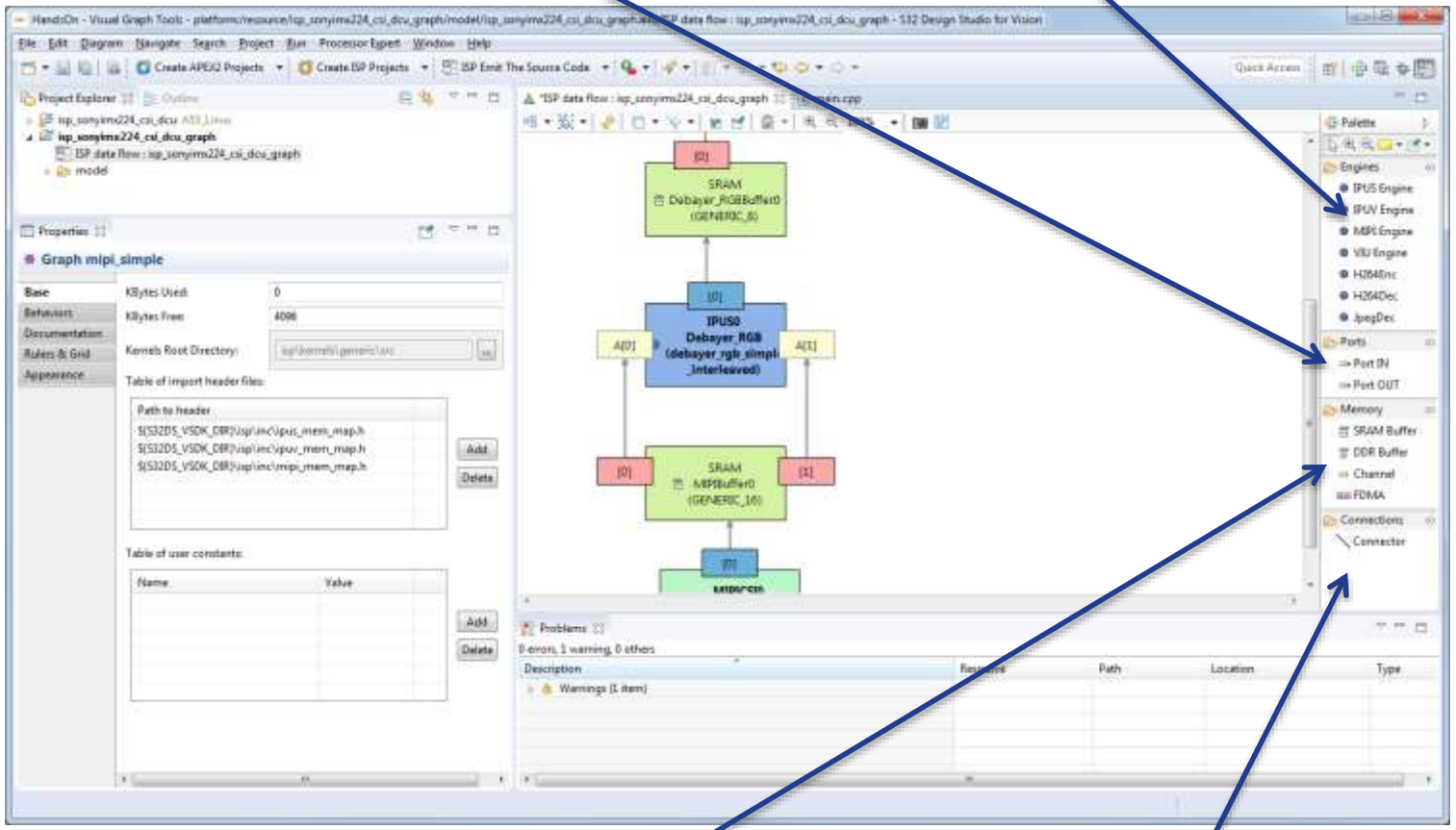


04.

ISP Graph Tool

ISP Graph User Interface

- New Project Wizard to create starter project
  - including all libraries and SW required
- High Level Graph tools for algorithm formation
  - drag and drop creation of algorithm graphs
  - Graph == Mathematical with Visual Interface
- Vision SDK of Kernels for Vision processing
- Assembler for building executable at a button press
- Debugging Algorithm on ISP Hardware
- Example projects of Vision Processing on ISP

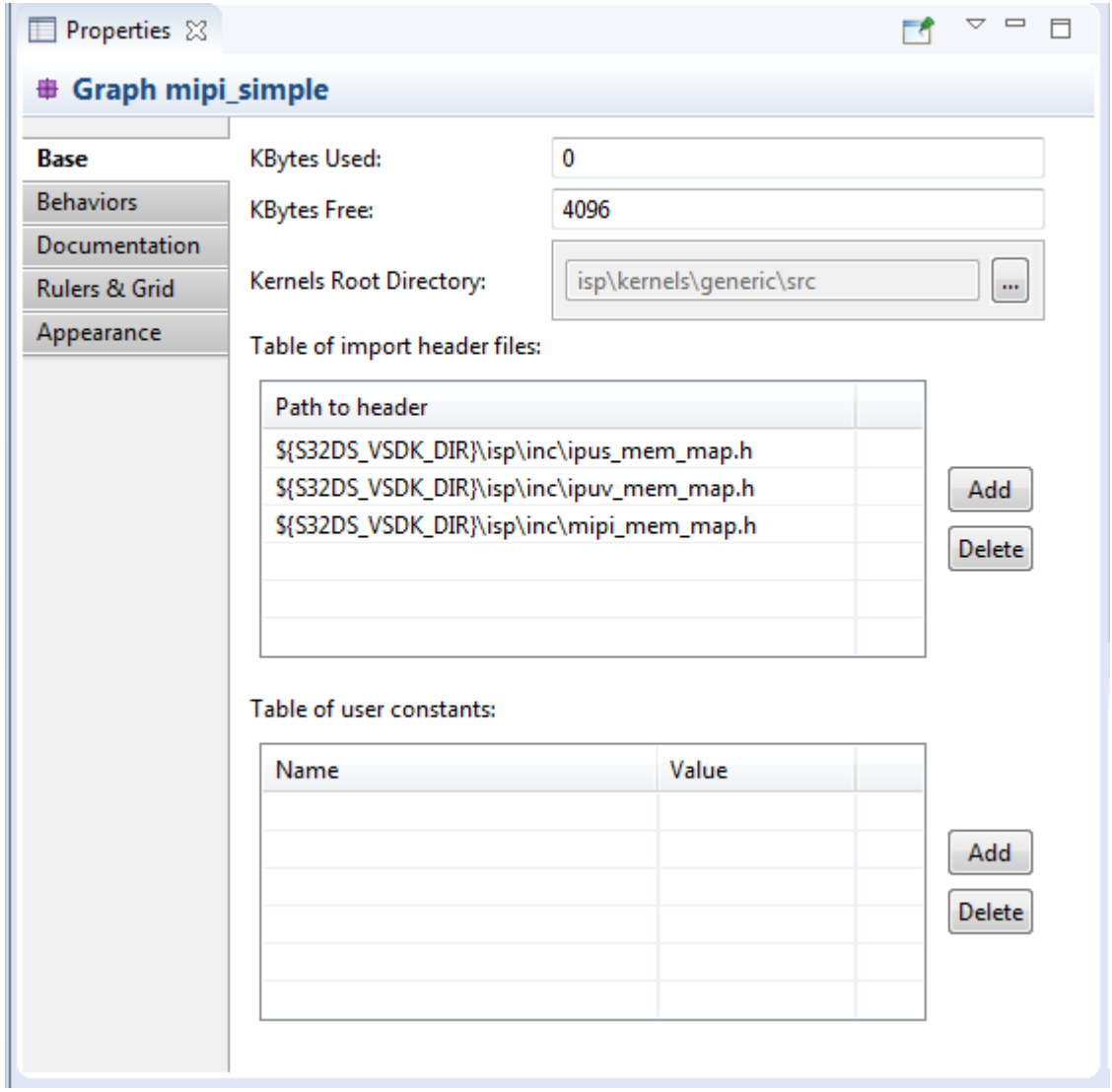


Graph

The most basic element of a graph is the graph itself. You can select it by clicking into the empty white canvas.

Properties:

- Kernels root directory.
  - Relative path to the directory which contains the ISP kernels.
- Kilobytes free.
  - Read-only property.
  - Shows how much space in the SRAM is free.
- Kilobytes used.
  - Read-only property.
  - Shows how much space in the SRAM is already taken by buffers.

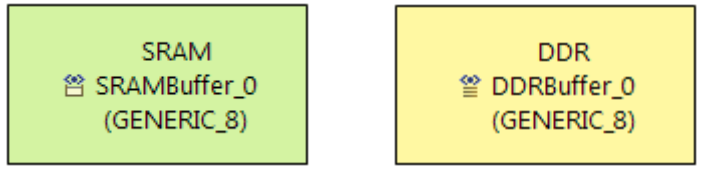


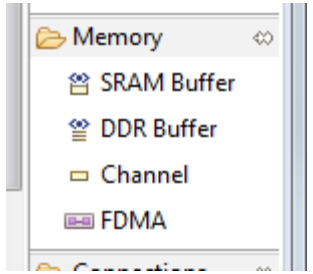
Buffers

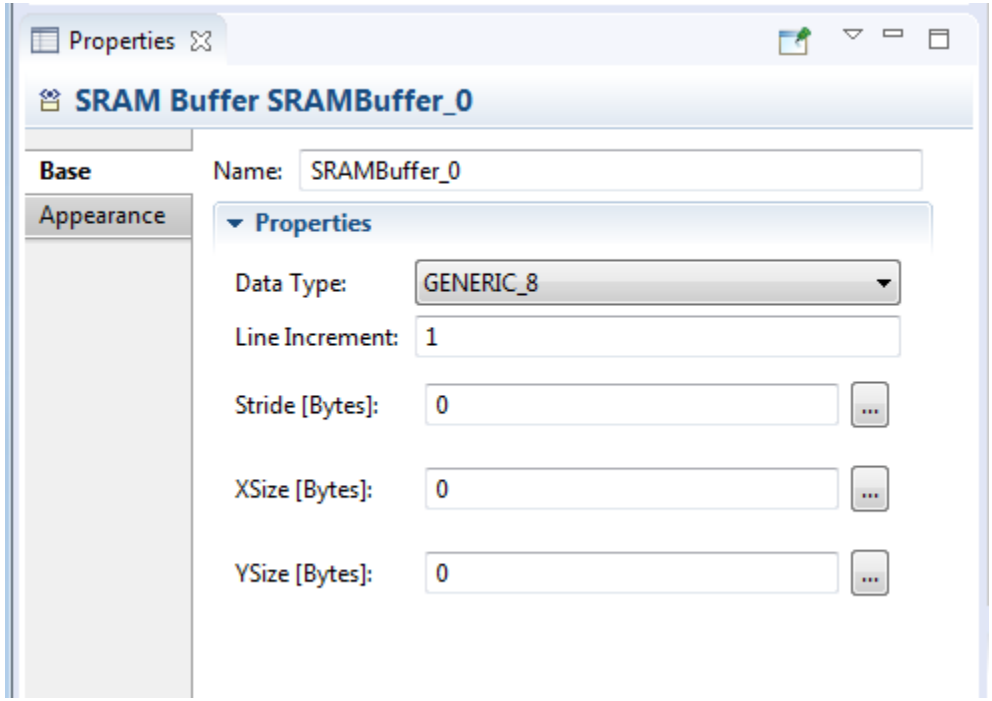
Data cannot be transferred directly from an engine to another engine - you have to use buffers.

Two types:

1. ISP Buffer (in SRAM)
2. ISD DDR Buffer (in DDR)
   - Can only be accessed via the FastDMA transfer.



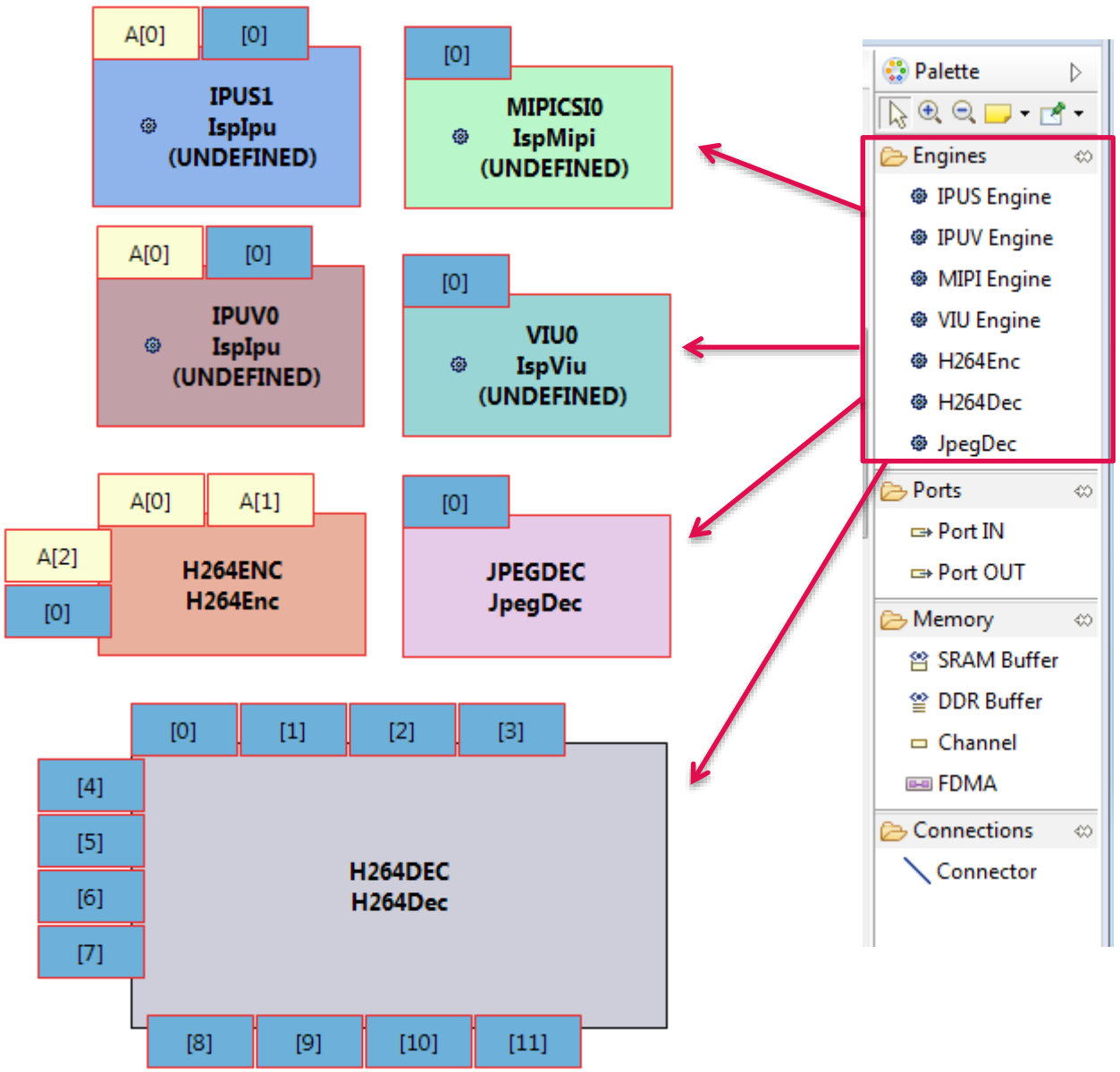




Engines

Several types:

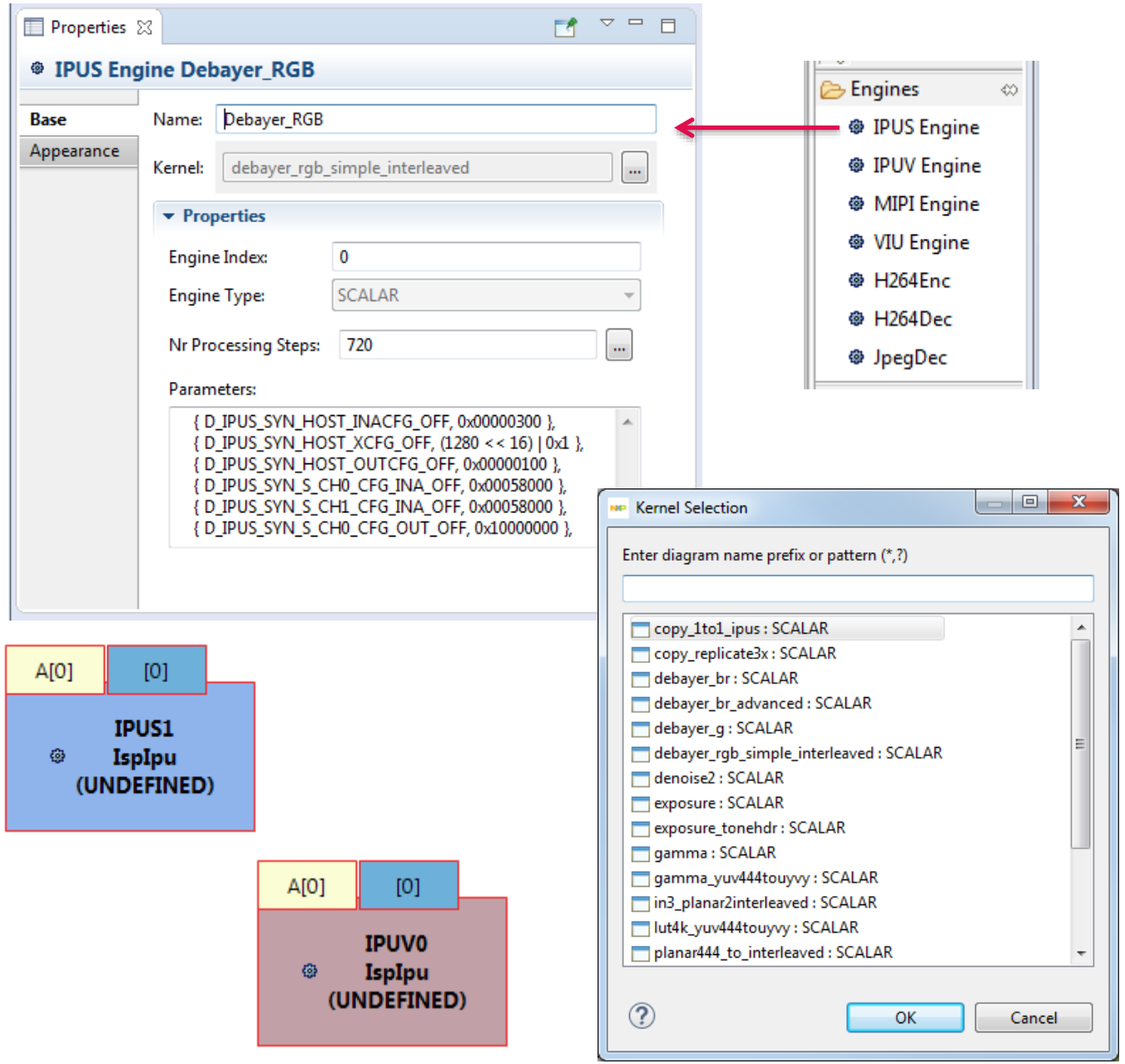
1. IPU (Image Processing Unit)
   a) Vector
   b) Scalar
2. MIPI (Mobile Industry Processor Interface)
   - Handles the camera input over the MIPI-CSI interface.
3. VIU (Video Interface Unit)
   - Handles the camera input over the VIU interface.
4. Other
   - Encoders & Decoders.



IPU Engine

Properties:

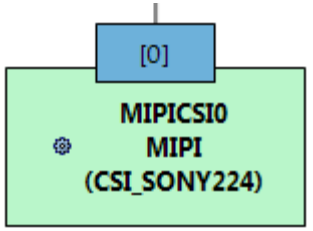
- Name.
- Kernel
  - Pop-up menu containing the list of all kernels found in the kernel directory.
  - The chosen processing kernel has to match the engine index – not all IPU Engines are equally capable!
- Index in the engines array.
- Type
  - Vector/Scalar.
- Number of processing steps.
- Parameters
  - in the format produced by the *kernel designer*.

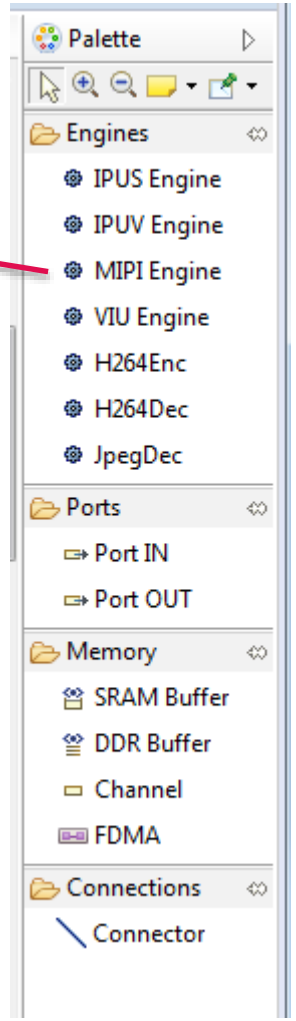


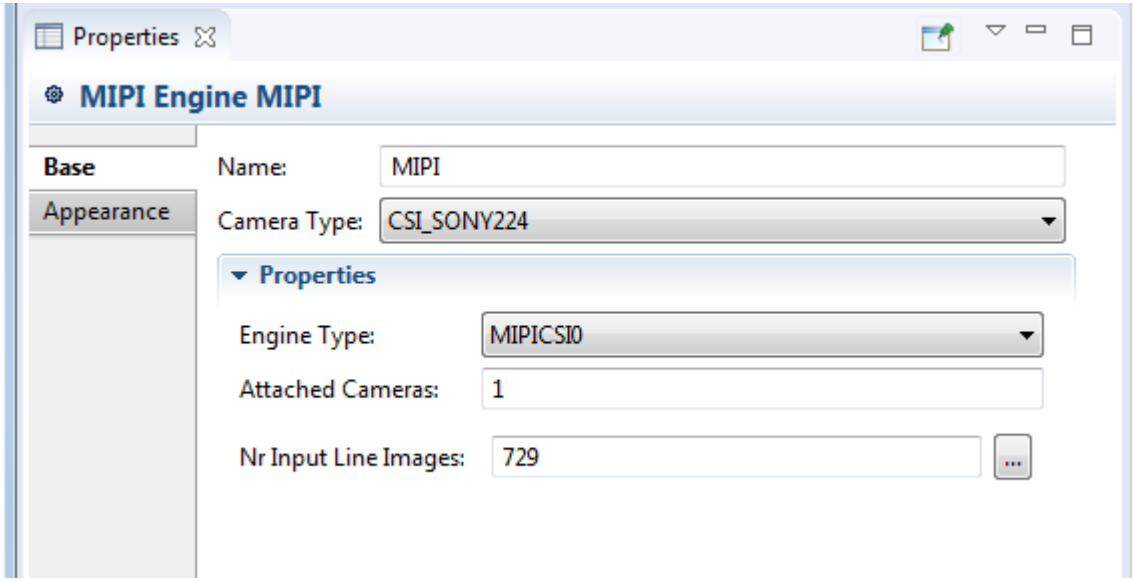
MIPI Engine

Properties:

- Name.
- Camera type.
- Engine Type (MIPICSI0 or MIPICSI1).
- Number of attached cameras.
- Number of input image lines.



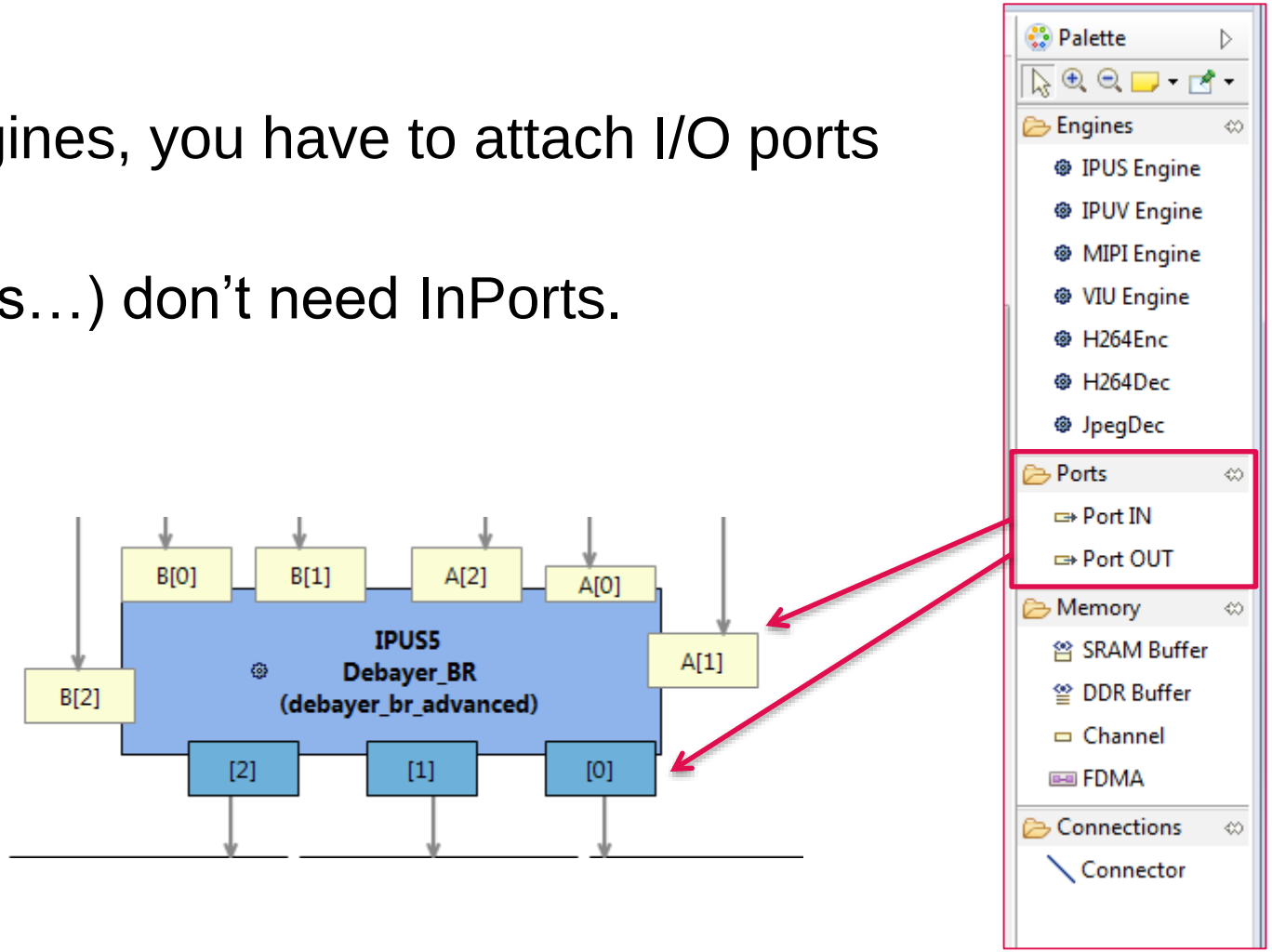




Engine I/O ports

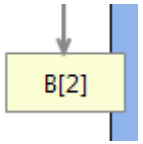
To get the data in and out of the engines, you have to attach I/O ports to them.

Image Sources (MIPI, VIU, Decoders…) don’t need InPorts.



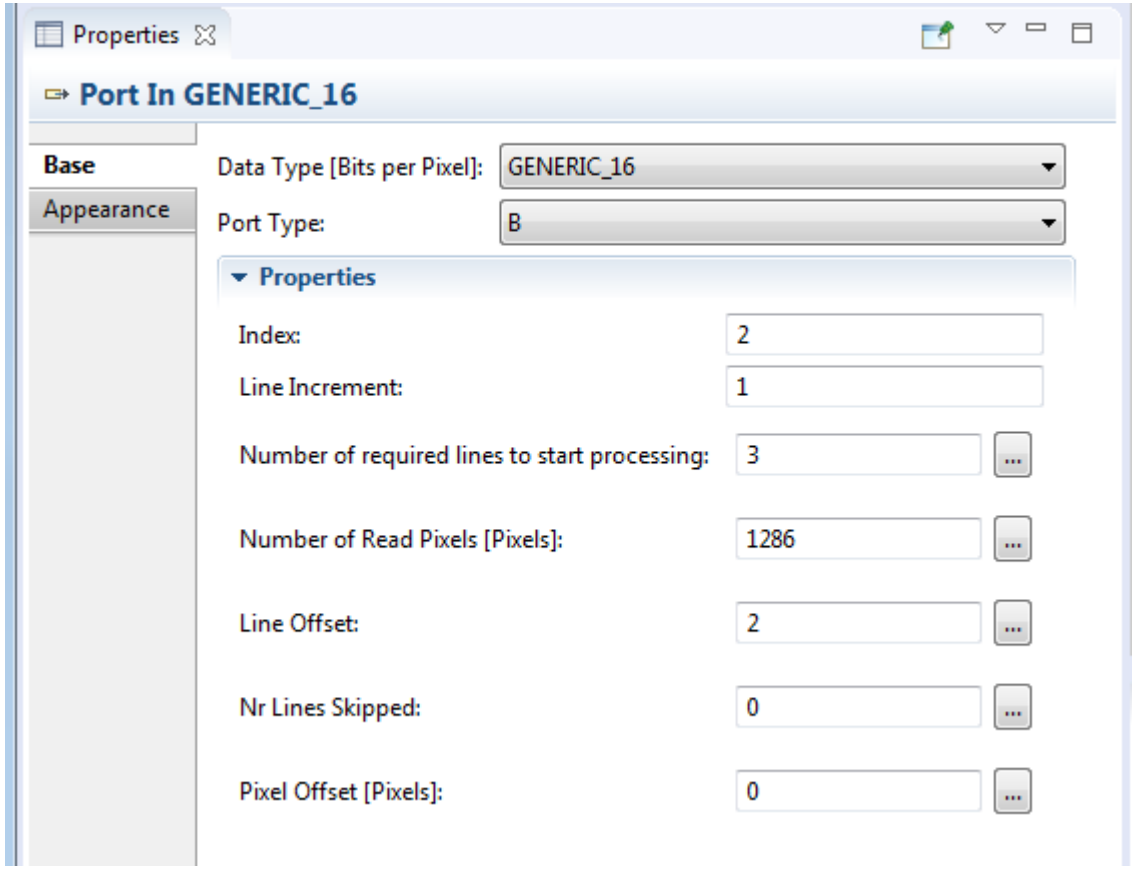
Engine input ports

Define how the engine reads the data from the buffer.



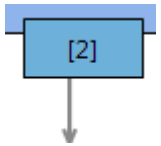
Properties:

- Data type – has to match the data type of the source buffer.
- Port type
  - A
  - B
  - Alpha.
- Index – has to be unique among other input ports of the engine.
- Line increment – if you want to override the line increment of the buffer. If you do not want to override it, leave it 0.
- Number of required lines to start processing.
- Number of Read Pixels – number of pixels read from a line.
- Line Offset.
- Number of Lines Skipped.
- Pixel Offset.



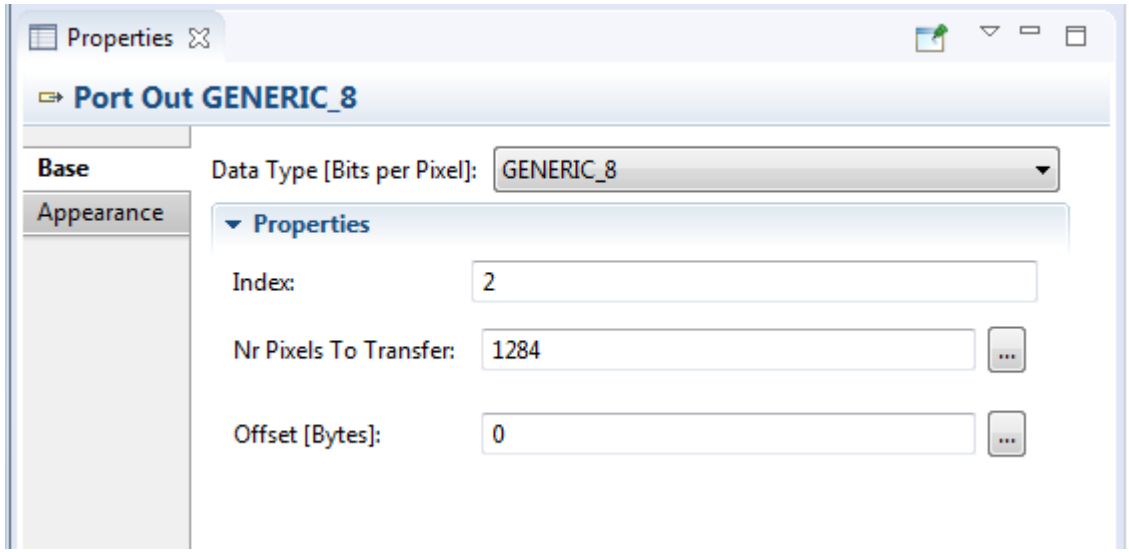
Engine output ports

Define how the engine writes the data to the buffer.



Properties:

- Data type – has to match the data type of the target buffer.
- Index – has to be unique among other output ports of the engine.
- Number of Pixels To Transfer.
- Offset – number of pixels to skip (in case of several engines writing to one buffer).



05.

ISP Graph Demo

06.

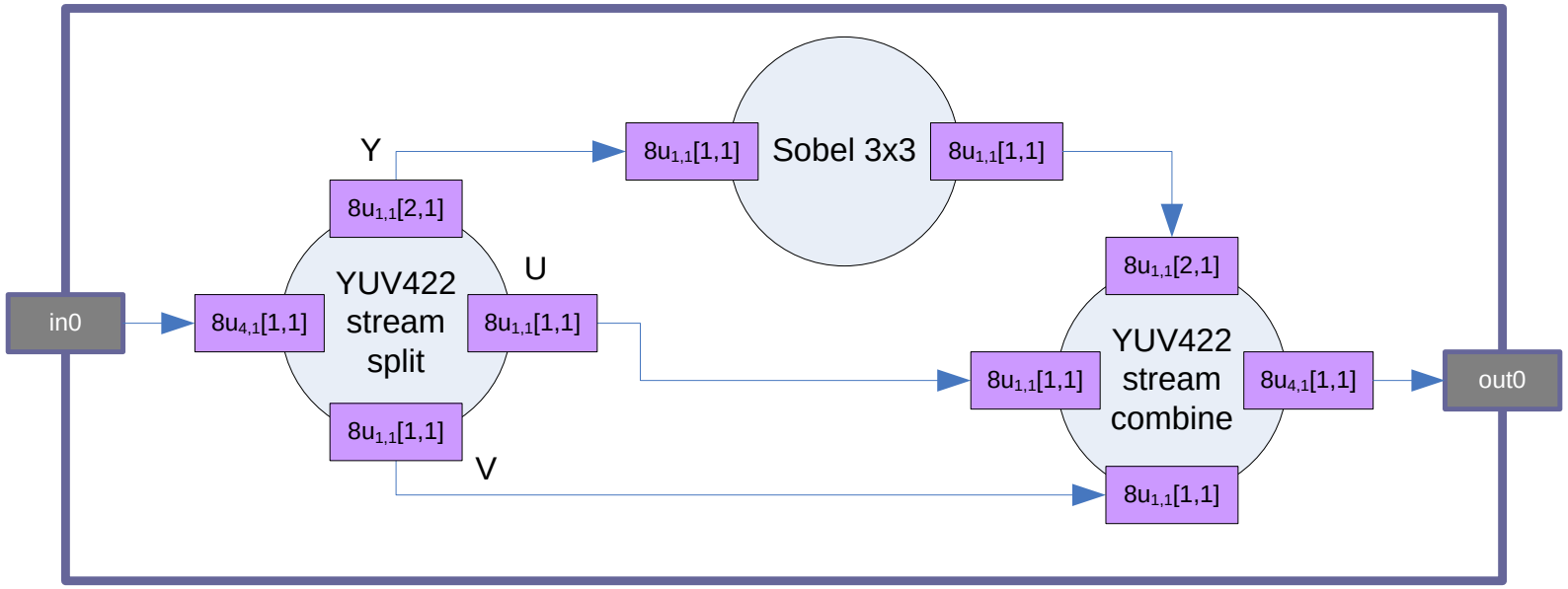
APEX Graph Tool

ACF Kernel

A kernel is a well-defined unit of processing that executes on a specific processor.

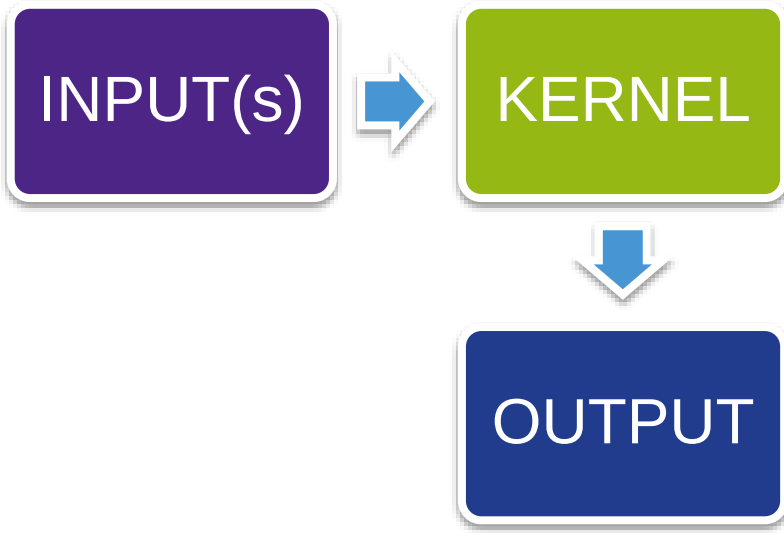
Graph Definition

- An ACF graph is basically a directed acyclic graph (DAG) comprised of kernels and the directed connections between them.

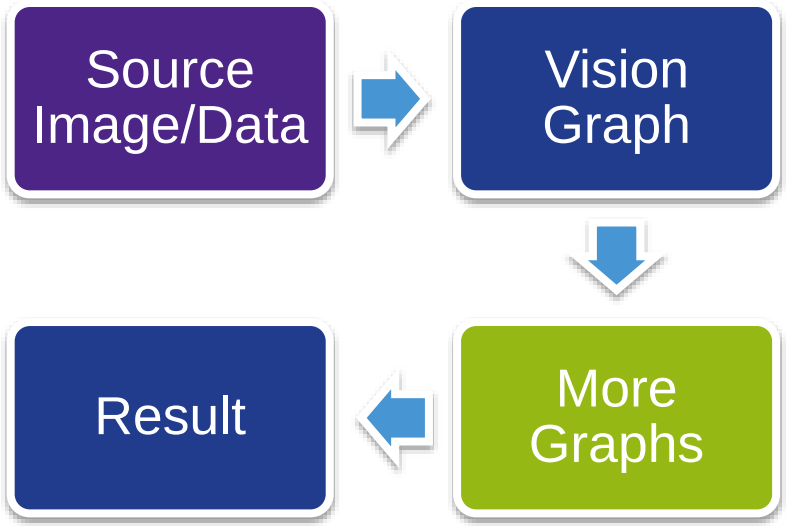


APEX2 Graph Tool Structure

Vision Graph Components



Program Flow Components

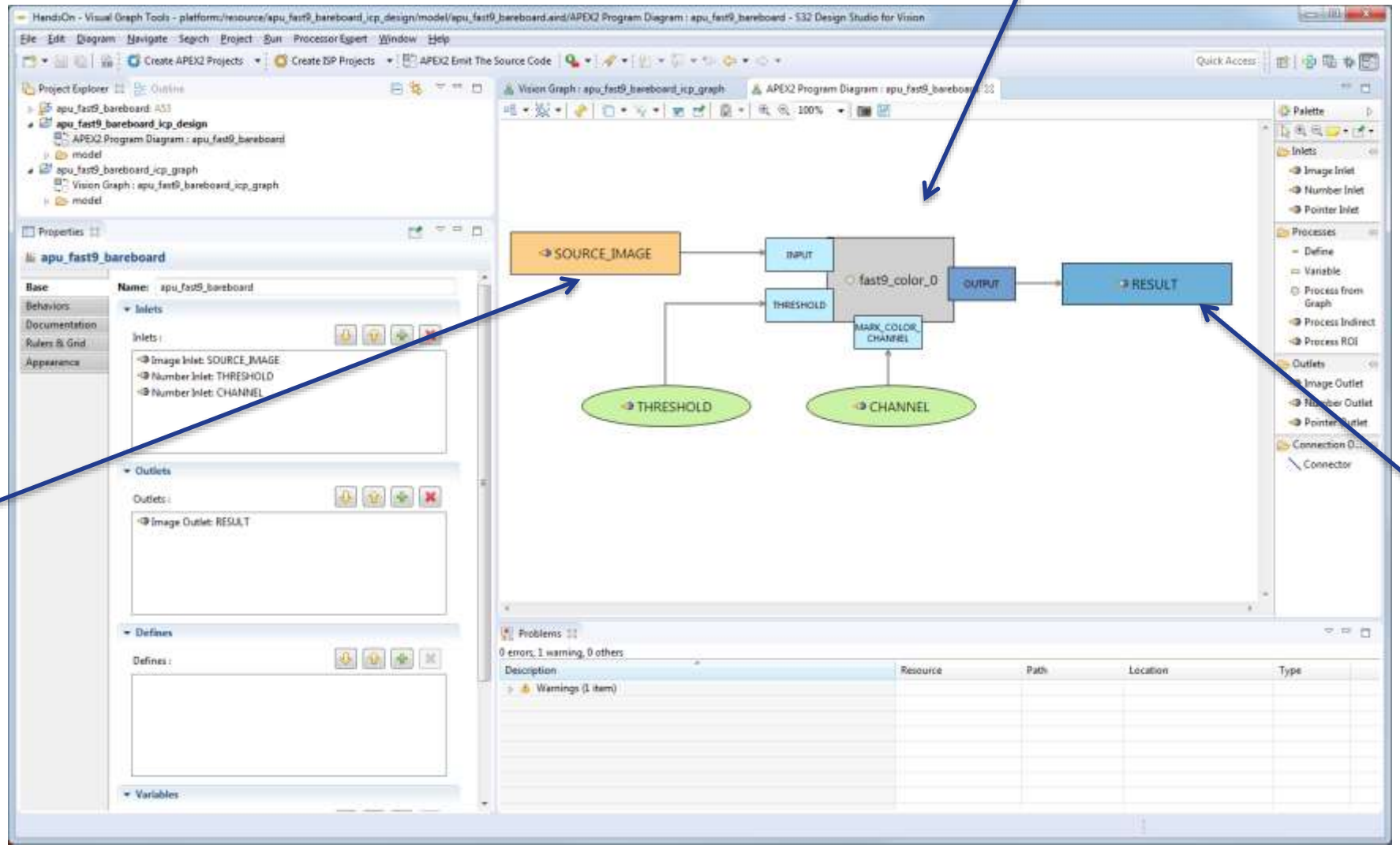


APEX2 Graph Tool – Program Diagram

Vision Graph

Input Image Data

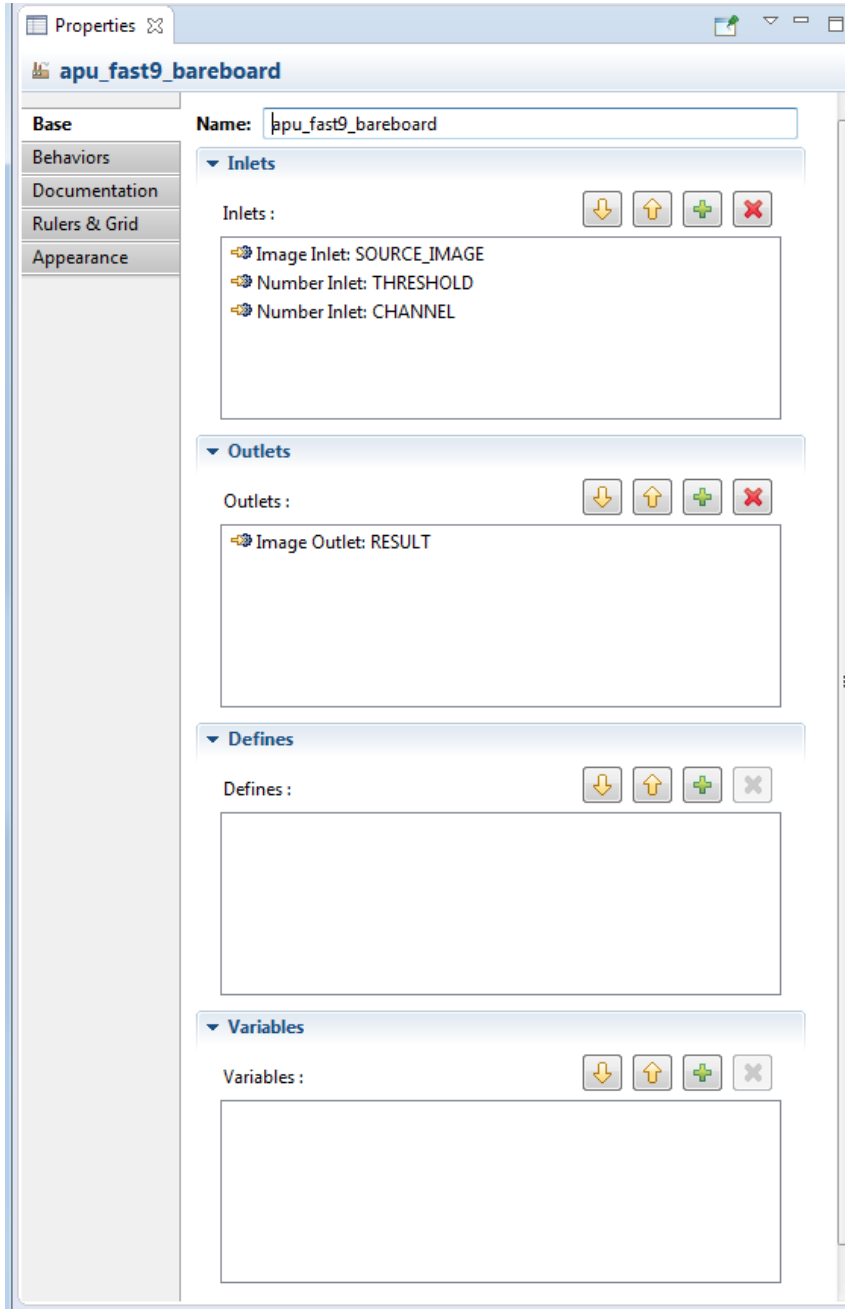
Processed Output



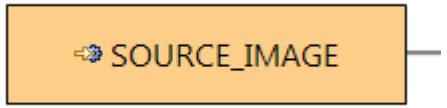
Program Diagram

The most basic element of a program diagram is the diagram itself. You can select it by clicking into the empty white canvas.

- Properties:
  - Name.
    - Name of the program diagram.
- Inlets.
  - List of the sources added to the diagram. Includes:
    - Image Inlet
    - Number Inlet
    - Pointer Inlet
- Outlets.
  - List of the sinks added to the diagram. Includes:
    - Image Outlet
    - Number Outlet
    - Pointer Outlet
- Defines.
  - List of the constants defined within the program diagram.
- Variables
  - List of the global variables defined within the program diagram.

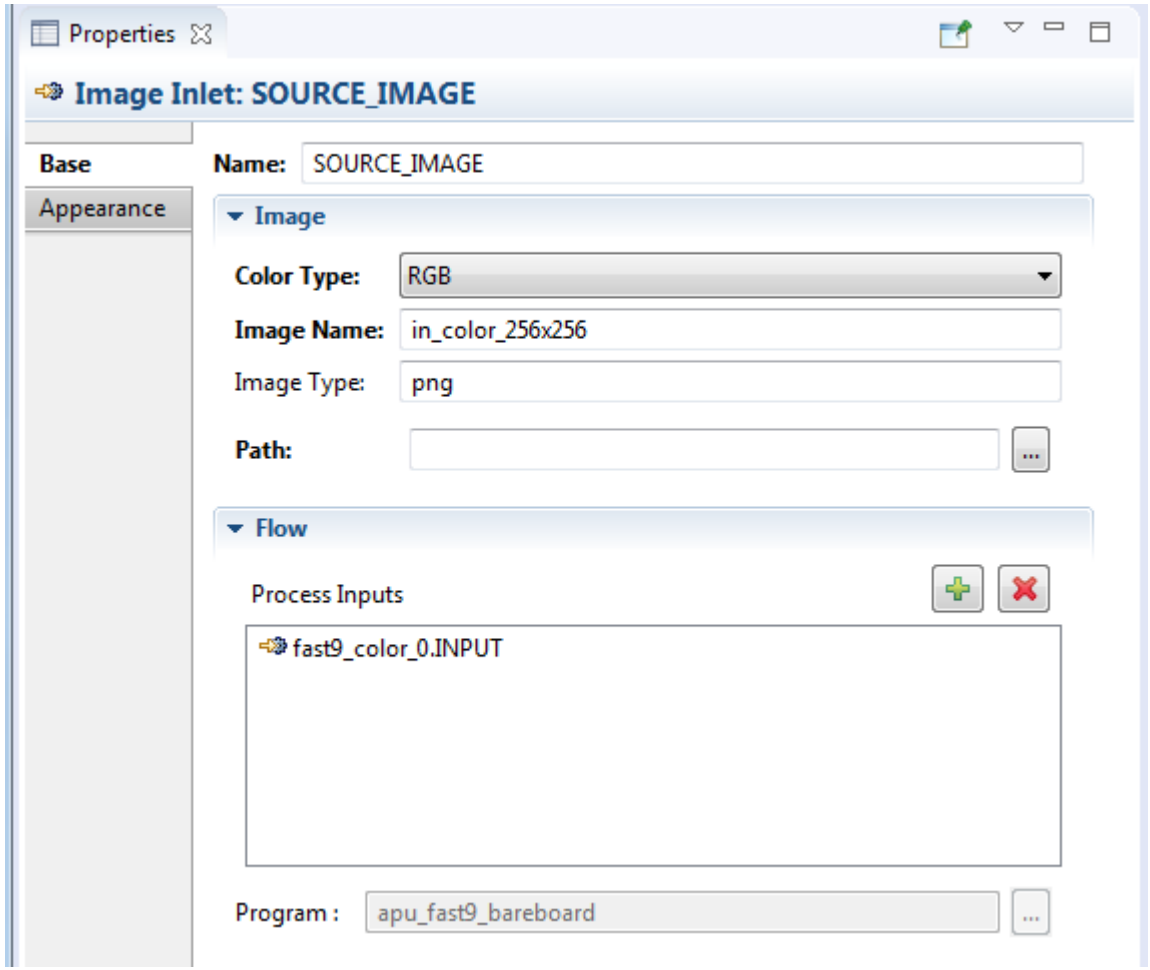


Program Diagram – Image Inlet



Properties:

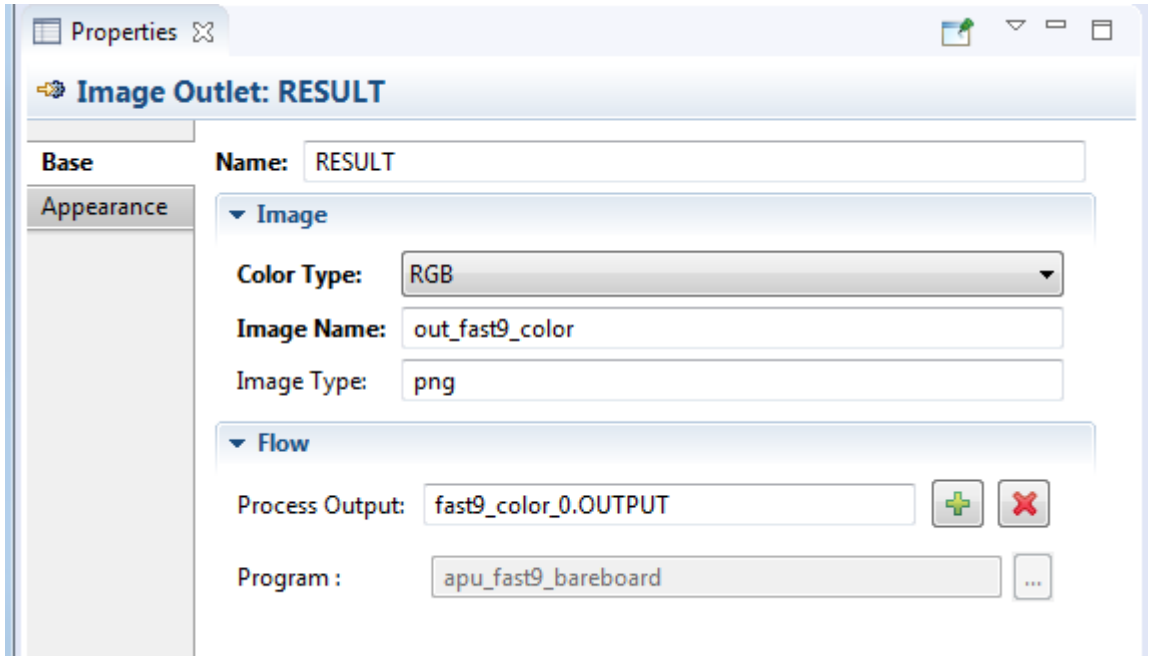
- Name.
  - Name of the image inlet.
- Color Type.
  - The color type of the image:
    - GRAYSCALE
    - RGB
- Image Name.
  - Filename of the image.
- Image Type.
  - File extension indicating the type of image file (png, jpg, gif, etc.).
- Path.
  - File system directory path to the image file
- Process Inputs.
  - List of the process input block inputs this image inlet is connected to.
- Program.
  - Name of the program diagram which contains this image inlet



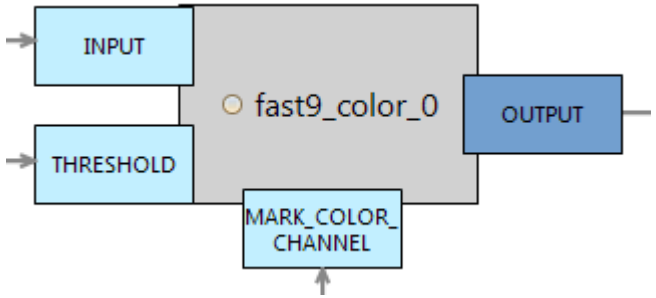
Program Diagram – Output

Properties:

- Name.
  - Name of the image outlet.
- Color Type.
  - The color type of the image:
    - GRAYSCALE
    - RGB
- Image Name.
  - Filename of the image.
- Image Type.
  - File extension indicating the type of image file (png, jpg, gif, etc.).
- Process Output.
  - List of the process input block inputs this image outlet is connected to.
- Program.
  - Name of the program diagram which contains this image outlet

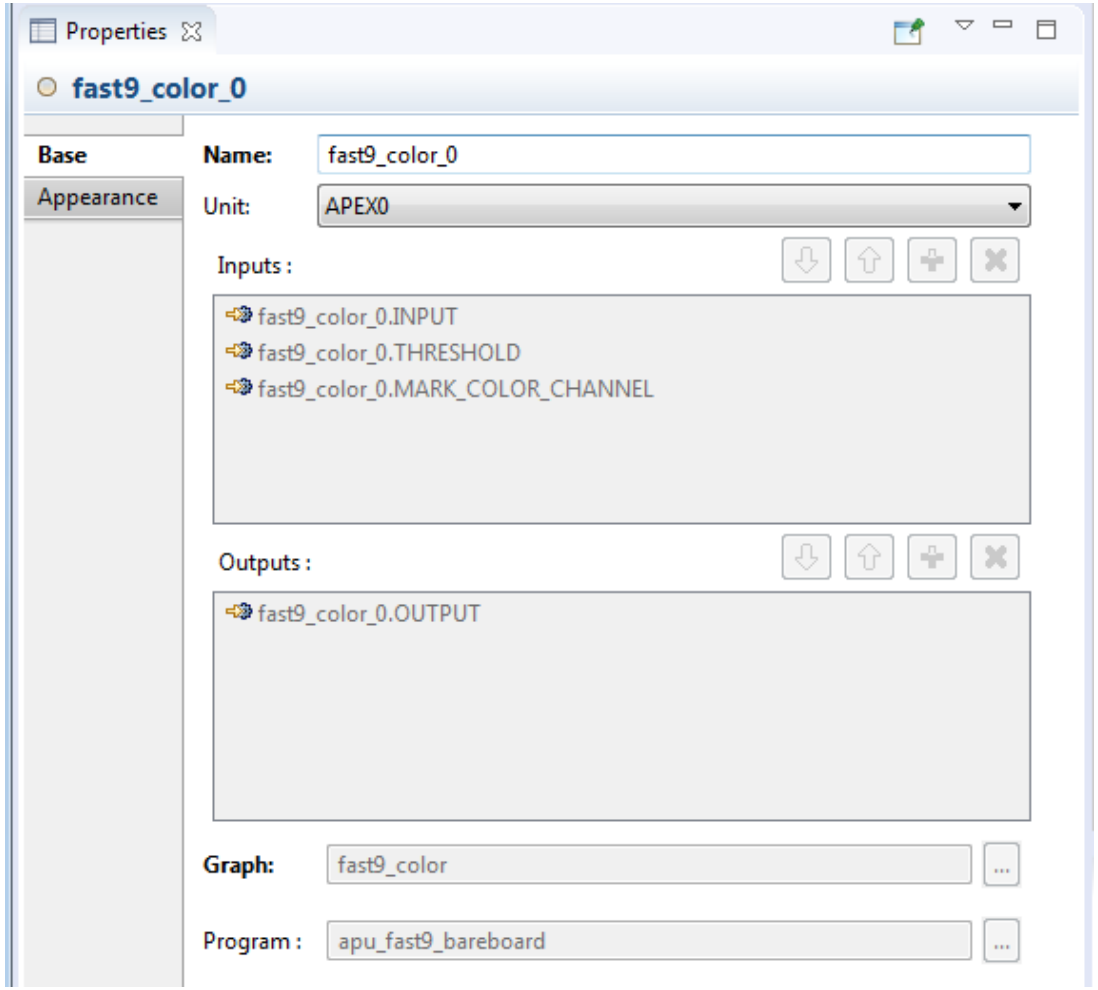


Program Diagram – Process from Graph

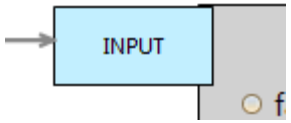


Properties:

- Name.
  - Name of this process from graph block which represents a vision graph. By default, name is same as the graph it represents.
- Unit.
  - Choose from the two APEX2 cores on which this process from graph should execute:
    - APEX0
    - APEX1
- Inputs.
  - List of the program diagram inputs connected to the inputs of this process from graph.
- Outputs.
  - List of the program diagram outputs or process from graph inputs which are connected to the output of this process from graph.
- Graph.
  - Name of the vision graph represented by this process from graph.
- Program
  - Name of the program diagram containing this process from graph.

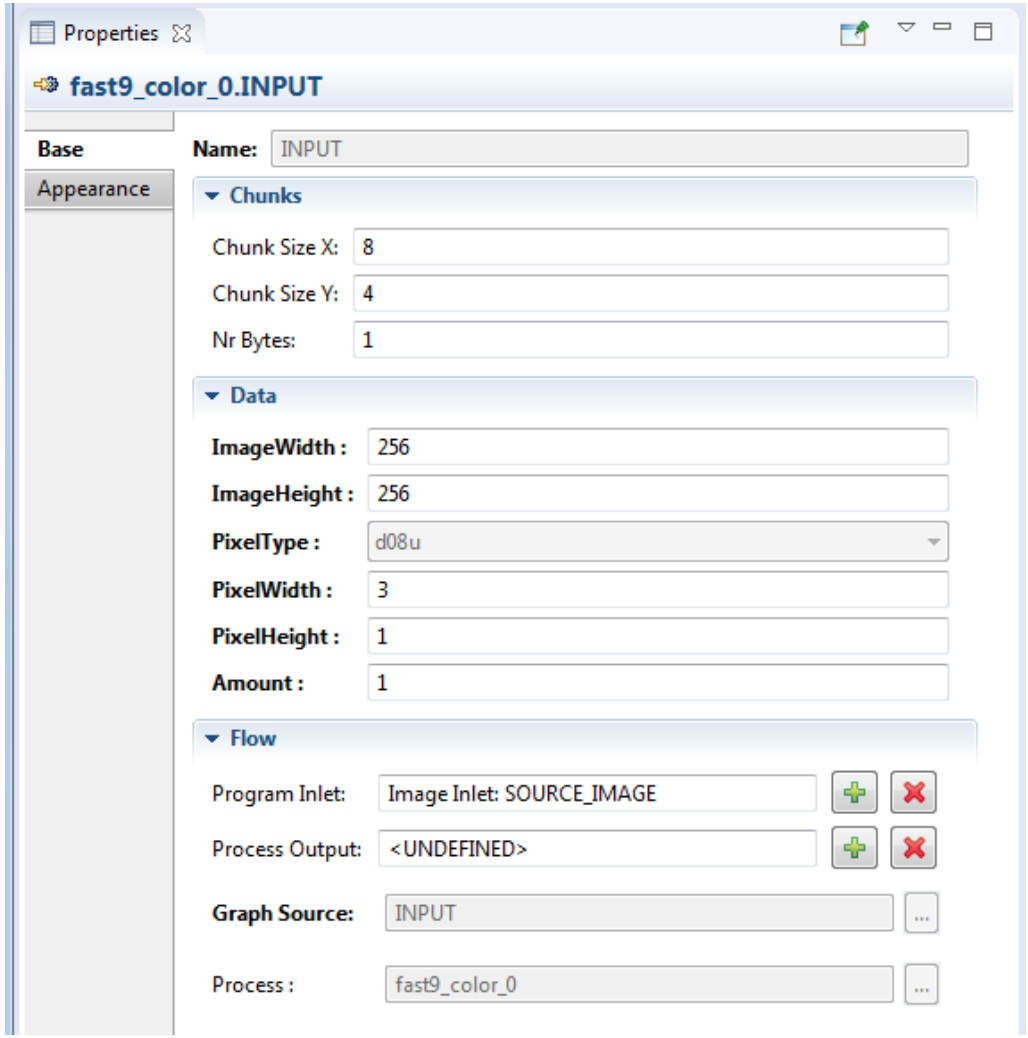


Program Diagram – Process from Graph Input



Properties:

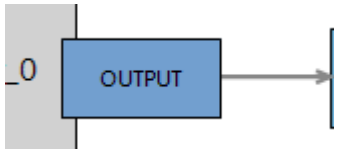
- Name.
  - Name of the process from graph input.
- Chunks. (optional)
  - Describes the 1-D or 2-D region of data to be processed
    - Chunk Size X
    - Chunk Size Y
    - Nr Bytes
- Data. (required)
  - Describes the input image data
    - ImageWidth
    - ImageHeight
    - PixelType
    - PixelWidth
    - PixelHeight
    - Amount
- Flow.
  - Details the connections between the port from the data source to the graph input
    - Program Inlet
    - Process Output
    - Graph Source
    - Process

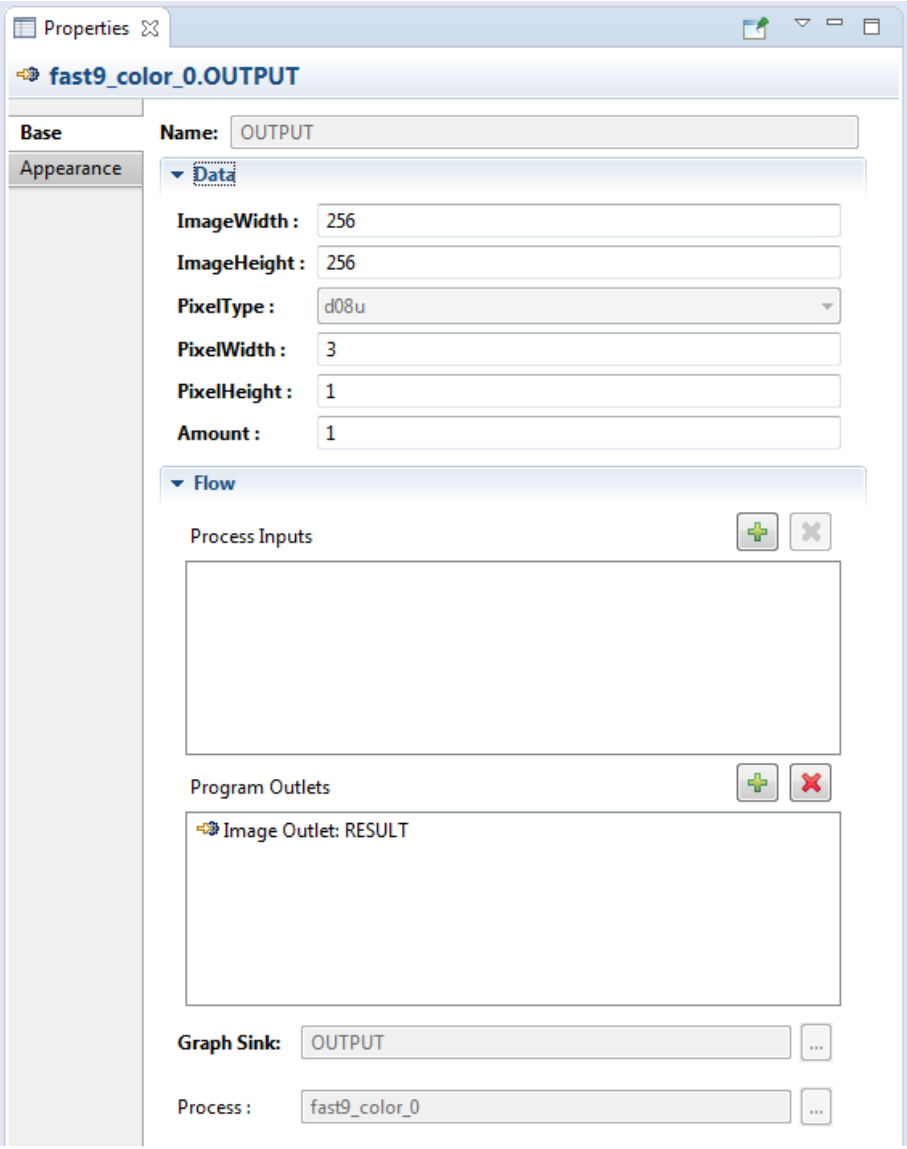


Program Diagram – Process from Graph Output

Properties:

- Name.
  - Name of the process from graph output.
- Data.
  - Describes the output image data
    - ImageWidth
    - ImageHeight
    - PixelType
    - PixelWidth
    - PixelHeight
    - Amount
- Flow.
  - Details the connections between the port from the data source to the graph input
    - Process Inputs
    - Program Outlets
    - Graph Sink
    - Process

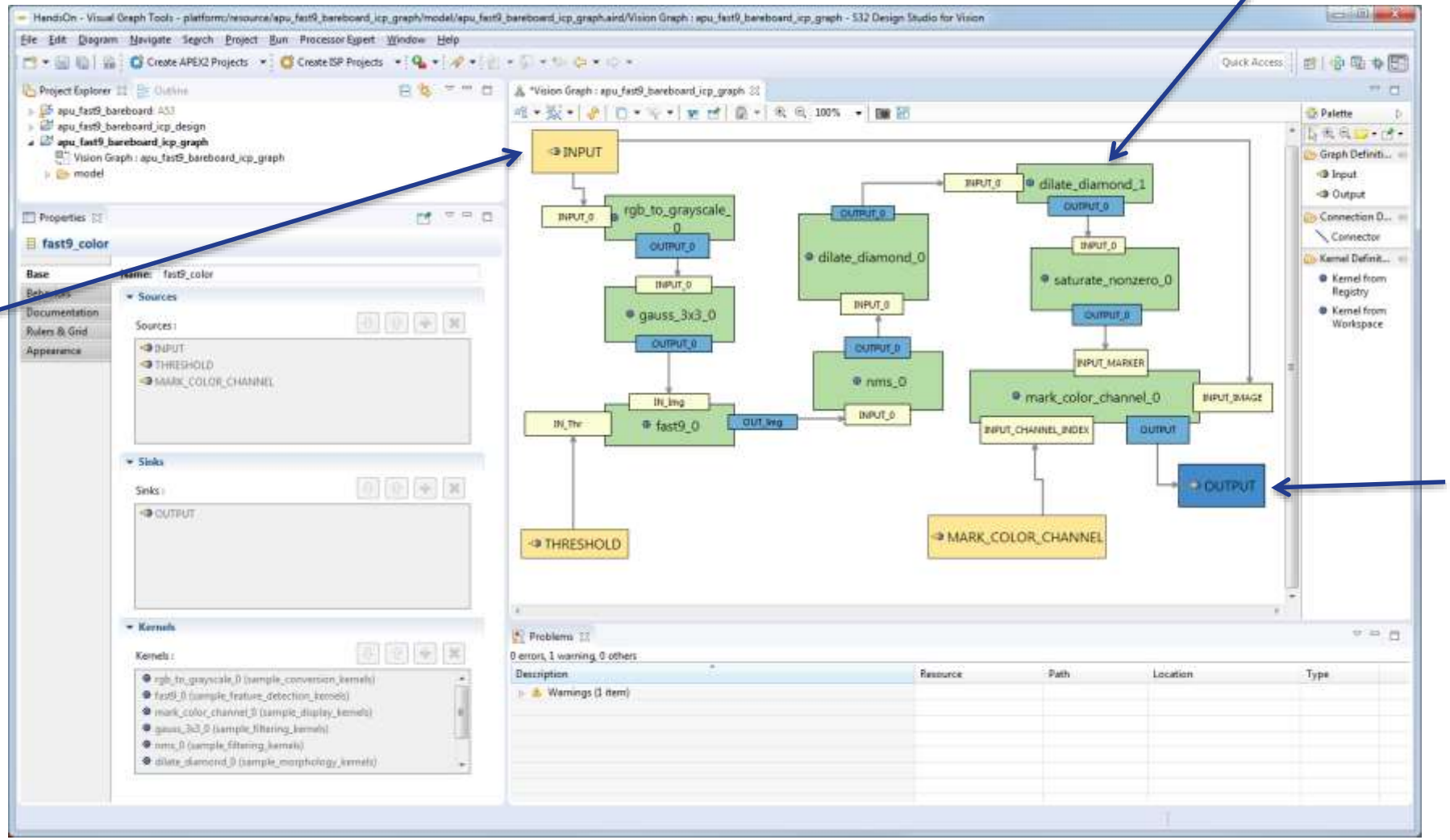




APEX Graph Tool – Vision Graph

Kernel

Input



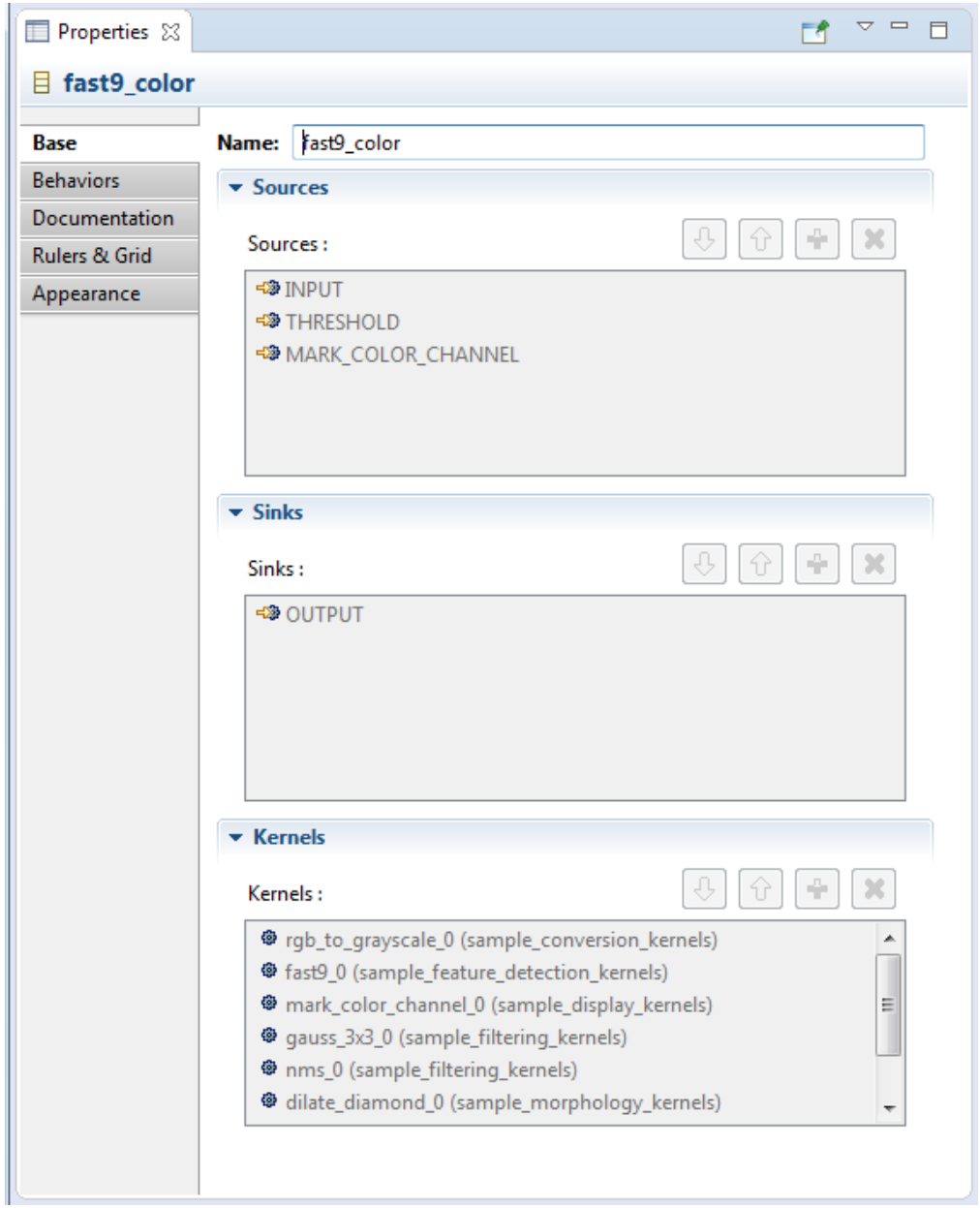
Output

Vision Graph – Graph Diagram

The most basic element of a graph is the graph itself. You can select it by clicking into the empty white canvas.

Properties:

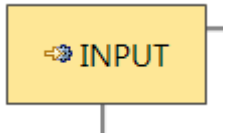
- Name.
  - Name of the graph as it will appear in Program Flow diagram and generated code. The only modifiable property.
- Sources.
  - List of the sources added to the diagram. Not a modifiable property from this panel.(make changes in diagram)
- Sinks.
  - List of the sinks added to the diagram. Not a modifiable property from this panel.(make changes in diagram)
- Kernels.
  - List of the kernels and name of folder in VDSK containing the source code, in (), contained within this graph diagram. Not a modifiable property from this panel.(make changes in diagram)

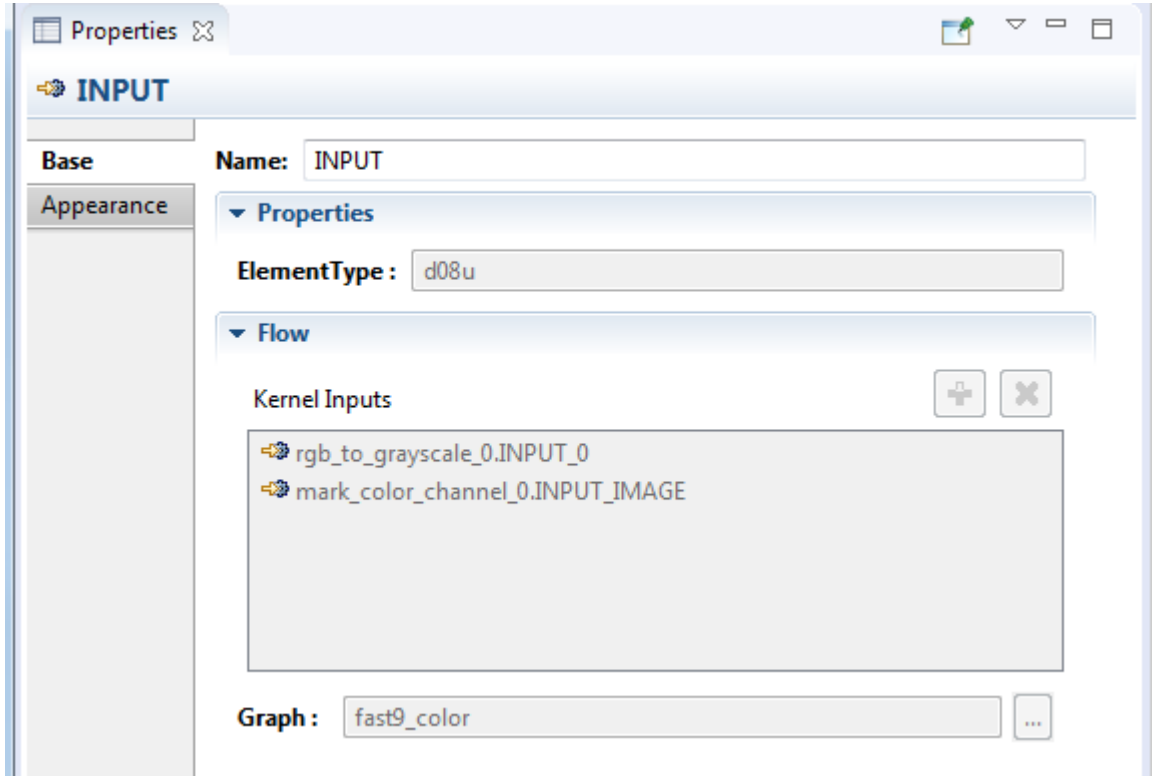


Vision Graph – Graph Input

Properties:

- Name.
  - Name of the input block in the graph diagram. The only modifiable property. Each must be unique.
- ElementType.
  - The datatype of the input. Not a modifiable property from this panel.
- Kernel Inputs.
  - List of the kernels block inputs this graph input is connected to. Not a modifiable property from this panel.(make changes in diagram)
- Graph.
  - Name of the current graph. Not a modifiable property from this panel.

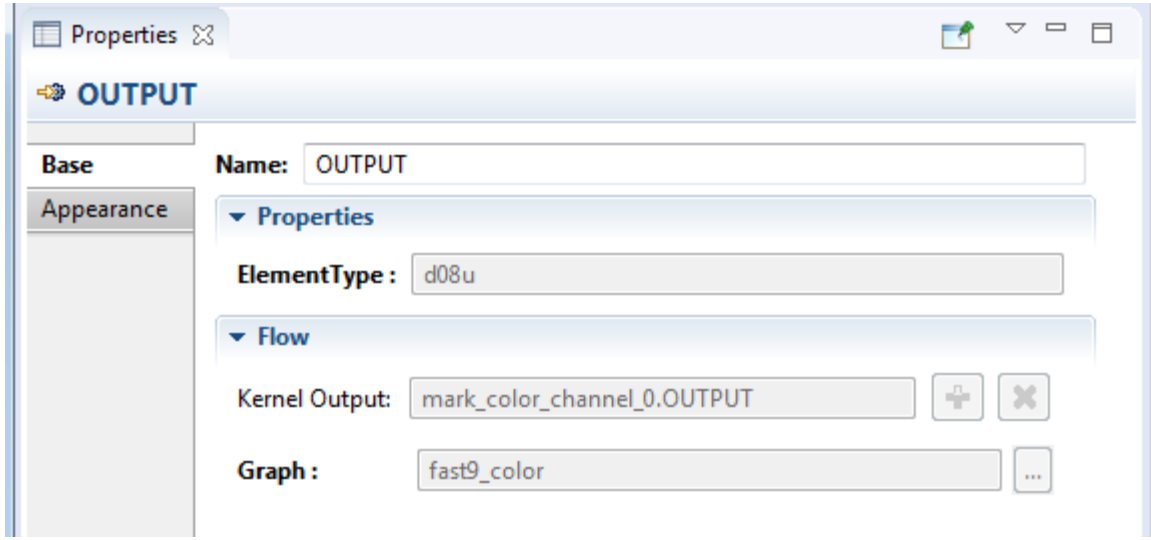




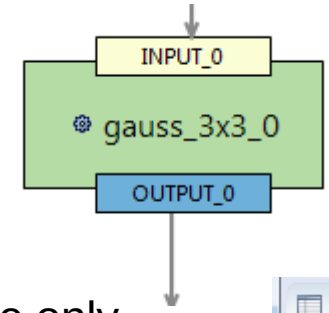
Vision Graph – Graph Output

Properties:

- Name.
  - Name of the output block in the graph diagram. The only modifiable property. Each must be unique.
- ElementType.
  - The datatype of the output. Not a modifiable property from this panel.
- Kernel Output.
  - The kernel block output connected to this graph output. Not a modifiable property from this panel.(make changes in diagram)
- Graph.
  - Name of the current graph. Not a modifiable property from this panel.

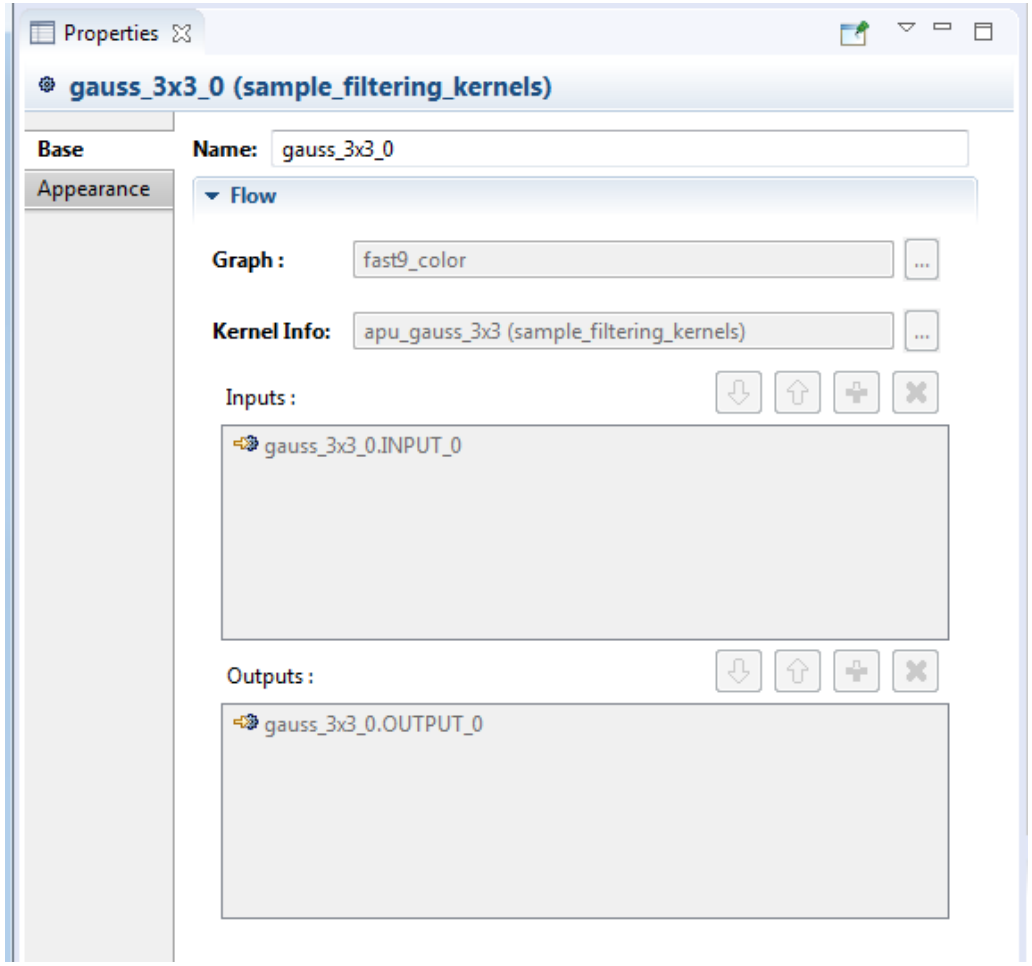


Vision Graph – Kernel



Properties:

- Name.
  - Name of the kernel block in the graph diagram. The only modifiable property. Each must be unique.
- Graph.
  - Name of this graph. Not a modifiable property from this panel.(make changes in diagram)
- Kernel Info.
  - Name of this kernel and name of folder in VDSK containing the source code, in (). Not a modifiable property from this panel.(make changes in diagram)
- Inputs.
  - List of the inputs added to the diagram. Not a modifiable property from this panel.(make changes in diagram)
- Outputs.
  - List of the outputs added to the diagram. Not a modifiable property from this panel.(make changes in diagram)



07.

APEX Graph Demo

08.

Q&A

Any Questions?

Please Fill Out Your Surveys.

Thank you for your time.

SECURE CONNECTIONS
FOR A SMARTER WORLD

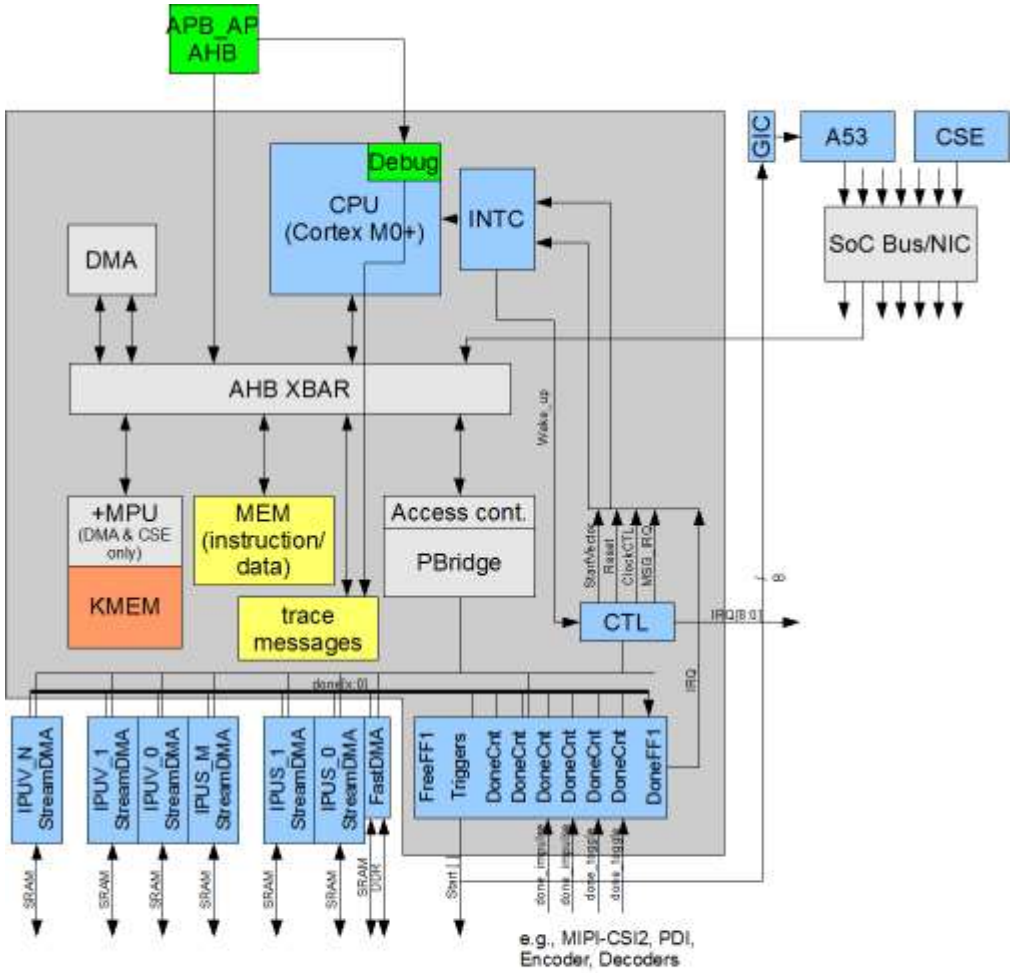
NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2017 NXP B.V.

09.

Backup Slides

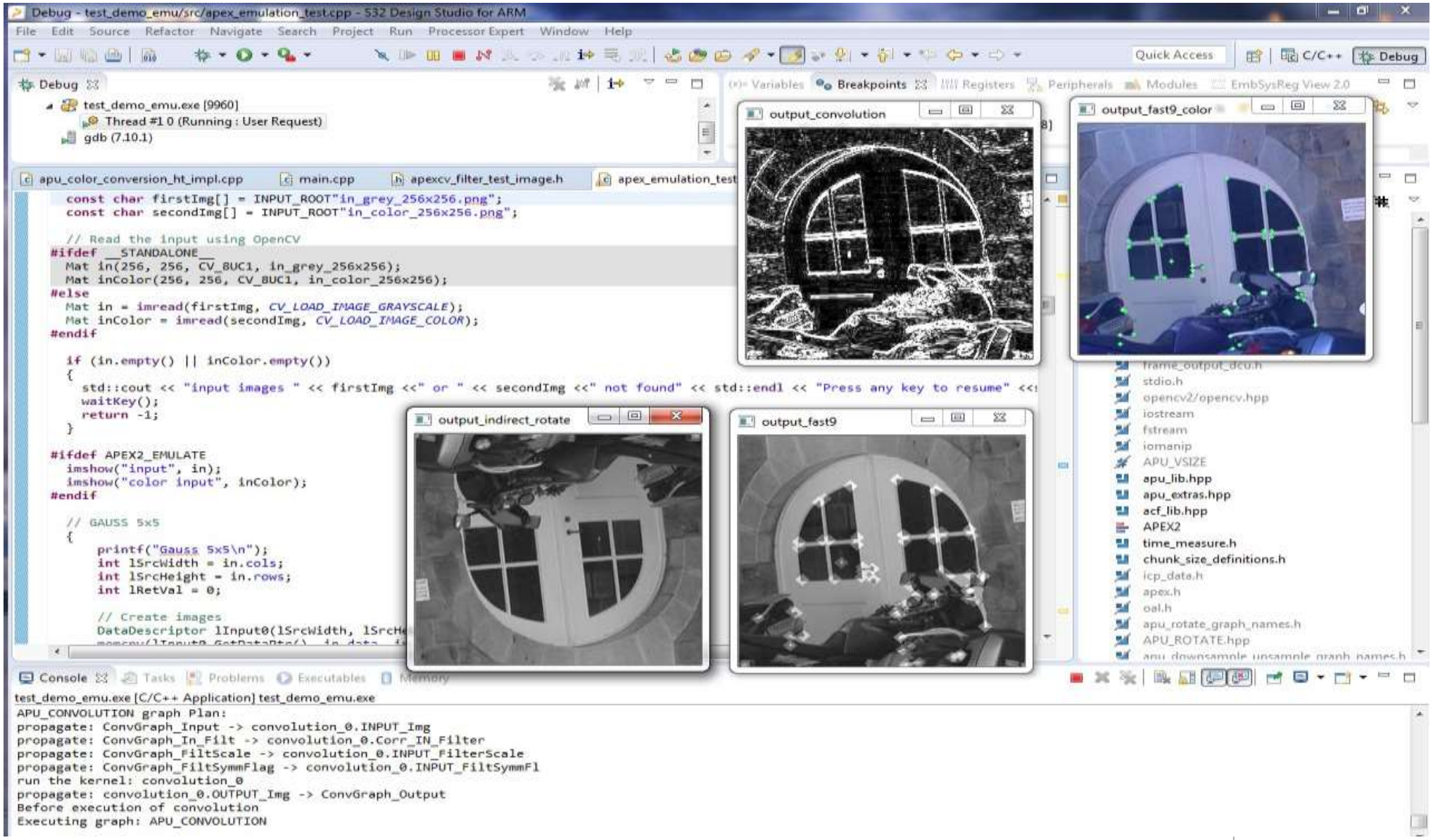
The Sequencer

I / E
T



- Cortex-M0+ mini platform
- Mapped into Host address space
- Peripherals can also be controlled by Host CPU
  - IPUs, and FastDMA
- Servicing the Debug Unit for the IPUs

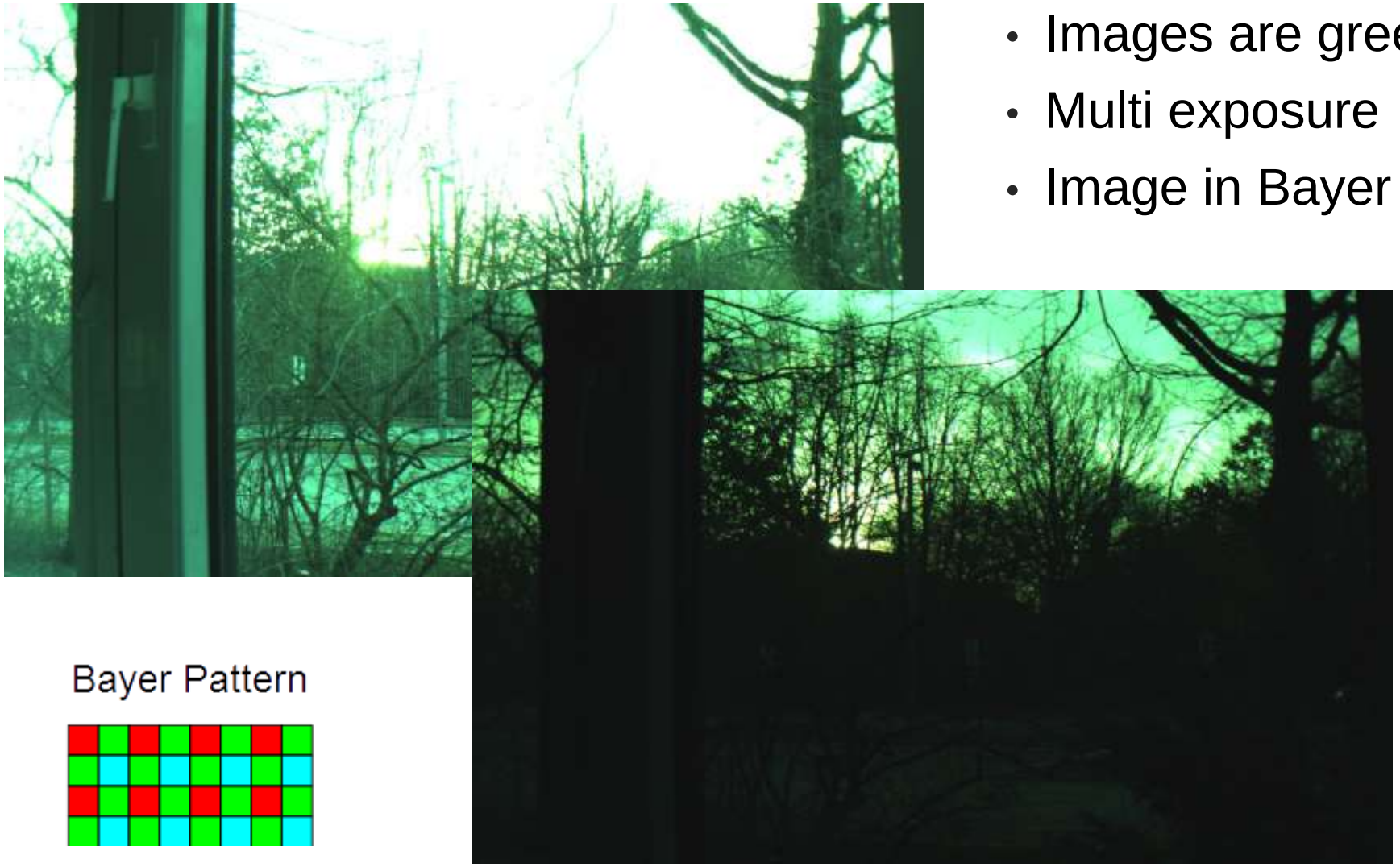
APEX Emulator Debugging – in S32DS for Vision



10.

Examples

ISP Function: Raw Sensor Image (Input to ISP)



Bayer Pattern

- Images are greenish
- Multi exposure
- Image in Bayer pattern

ISP function: Quality

ISP Function: local HDR + Tone Mapping

Examples: Image Signal Processing

Edge pixels (compressed)

- Detect head beams for head beam control

- Detect edges for lane detection

SECURE CONNECTIONS
FOR A SMARTER WORLD

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2017 NXP B.V.