OpenKasugai-Controller Attachment (CR/CM Specification)

v1.1.0

■**Each CR/CM expression**

■(resource name)
(Resource Description)

	Name	Type	Req/Opt	Description
	Name	-	-	
metadata	Namespace	-	-	
	Regions	[]RegionInfo	Optional	
	Name	string	Required	
Spec	Type	string	Required	
	Regions	[]RegionInfo	Optional	
Status	NodeName	string	Required	

Name : Parameter name
Type : Parameter type
Req/Opt : Required or Optional
Description : Parameter description

← Parameters containing the structure, or its map, list
← RegionInfo element of structure (1)
← RegionInfo element of structure (2)

・The parameter group deviated by one step indicates that the parameter in the previous step is a structure and is an element of the structure.
・For ConfigMap, data parameter replaces Spec/Status
・Parameters in gray represent parameters that exist in CR/CM specification but are not used in the current implementation.

■ComputeResource

Custom resource containing information about the hard configuration and capacity management of each node. Indicates CPU, GPU, and FPGA information on the node.

Also has information about device capacity management

	Name	Type	Req/Opt	Description
	Name	-	-	"compute-"+ node name
metadata	Namespace	-	-	(specified by various resource controllers)
	Regions	[]RegionInfo	Optional	Contains information about the smallest space on the physical device on that node. As in FunctionTarget, the units described are divided regions on physical devices.
	Name	string	Required	Unique name given to the partitioned region on the physical device
	Type	string	Required	Region type
	DeviceFilePath	string	Required	Device file path of the device where this region is located
	DeviceUUID	*string	Optional	UUID of the device on which this region is located
	DeviceType	string	Required	Accelerator type
	DeviceIndex	int32	Required	Device Number
	Available	bool	Required	Availability of deployment destination
	Status	WBRegionStatus	Required	Deployment destination state Varies based on child bs write state for FPGA only. CPU/GPU is always fixed to "Ready." ・NotReady (No child bs) "NotReady " ・Preparing (Writing) Preparing ・Ready (with child bs) Ready ・Failed to prepare Error
	MaxFunctions	*int32	Optional	Maximum number of deployments (Number of functions = number of circuits, number of pods) When the FPGA bridge bs is in the unwritten/writing state, the following values are taken according to the correspondence state of automatic writing. (the same shall apply hereinafter) ・Enable child bs AutoWrite: nil ・child bs does not support automatic writing: 0
	CurrentFunctions	*int32	Optional	Current number of deployments in the deployment destination
	MaxCapacity	*int32	Optional	Deployment destination Maximum Capacity (fps)
	CurrentCapacity	*int32	Optional	Current load on the deployment destination
	MaxTimeSlicingSeconds	*int32	Optional	Maximum value when time-sharing is used
	CurrentTimeSlicingSeconds	*int32	Optional	Current value when time-sharing is used
	Functions	[] functionInfrastruct	Optional	Deployed Function Information
	FunctionIndex	int32	Required	Serial number of the deployed function
	PartitionName	string	Required	Identity of the deployed function
	FunctionName	string	Required	Name of the deployed Function
	Available	bool	Required	Availability of Deployed Functions
	MaxDataFlows	*int32	Optional	Maximum number of deployed Function DF (number of WBFunction). Depend on the number of channels in the circuit
	CurrentDataFlows	*int32	Optional	Number of DF deployed functions currently installed (number of WBFunction)
	MaxCapacity	*int32	Optional	Maximum processing power of deployed functions (fps)
	CurrentCapacity	*int32	Optional	Current load of the deployed function (fps)
	MaxTimeSlicingSeconds	*int32	Optional	Maximum value when time-sharing is used
	CurrentTimeSlicingSeconds	*int32	Optional	Current value when time-sharing is used
Spec	NodeName	string	Required	worker node name
	Regions	[]RegionInfo	Optional	Contains information about the smallest space on the physical device on that node. Suppress Parameters
Status	NodeName	string	Required	worker node name

■FunctionTarget

custom resource that has information about the candidate locations for functions built from ComputeResource

Body: It represents information in regions such as Lane, PR and GPU. Number of available circuits/pods and processing performance

Functions: Represents information about the circuits and pods deployed on the region.

Maximum number of DF circuits or pods that can be installed, and processing performance per circuit or pod

	Name	Type	Req/Opt	Description
	Name	-	-	Name the node device region Assumptions generated from ComputeResource's corresponding regionInfo data <Node>. <DeviceType>-<DeviceIndex>. <Region> etc.
metadata	Namespace	-	-	(specified by various resource controllers)
	ComputeResourceRef	WBNamespacedName	Required	ComputeResource Resource Name and Namespace
	Name	string	Required	
Spec	Namespace	string	Required	
	RegionName	string	Required	Unique name given to the partitioned region on the physical device
	RegionType	string	Required	Region type
	NodeName	string	Required	worker node name
	DeviceType	string	Required	Accelerator type
	DeviceIndex	int32	Required	Device Number
	Available	bool	Required	Availability of deployment destination (minimum space such as Lane or gpu)
	Status	WBRegionStatus	Required	State of the deployment destination (minimum space such as Lane or gpu) Varies based on child bs write state for FPGA only. CPU/GPU is always fixed to "Ready." ・NotReady (No child bs) "NotReady " ・Preparing (Writing) Preparing ・Ready (with child bs) Ready ・Failed to prepare Error
	MaxFunctions	*int32	Optional	Maximum number of deployments (Number of circuits, number of pods), maximum number of Functions When the FPGA bridge bs is in the unwritten/writing state, the following values are taken according to the correspondence state of automatic writing. (the same shall apply hereinafter) ・Enable child bs AutoWrite: nil ・child bs does not support automatic writing: 0
	CurrentFunctions	*int32	Optional	The current load of the deployment destination. Current Functions Count
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployment destination
	CurrentCapacity	*int32	Optional	Current load to deploy to (fps). Sum of all Functions.CurrentCapacity
	Functions	[] functionstruct	Optional	Information about deployed functions (circuits and pods)
	FunctionIndex	int32	Required	Serial number of the deployed function
	FunctionName	string	Required	Name of the deployed Function
	Available	bool	Required	Availability of Deployed Functions
	MaxDataFlows	*int32	Optional	Maximum number of deployed Function DF (number of WBFunction). It depends on the number of channels of the
	CurrentDataFlows	*int32	Optional	Number of DF deployed functions currently installed (number of WBFunction)
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployed Function. It depends on the circuit and pod implementation.
	CurrentCapacity	*int32	Optional	Current load of the deployed function
	MaxTimeSlicingSeconds	*int32	Optional	
Status	CurrentTimeSlicingSeconds	*int32	Optional	

■DataFlow

custom resource with information about the configuration (function and connection specifications) and requirements (expected load) of data flow you want to deploy

	Name	Type	Req/Opt	Description
	Name	-	-	Be set by the user Since the value of this parameter is part of WBFunction's metadata.name, the length must satisfy the character limit of WBFunction's metadata.name. [Reference] Character limit for the name of the function (FC Functions key value) prepared in the sample data and the name of DataFlow that uses the following function (value of this parameter). In the prepared sample data, the maximum length of the function name (filter-resize-high-infer-main) is 29 characters, so in this case, DataFlow name must be within 14 characters. ・Advanced CPU filter/resize: filter-resize-high-infer-main (29 characters) * DataFlow name must be 14 characters or less ・Lightweight CPU filter/resize: filter-resize-low-infer-main (28 characters) * DataFlow name must be 15 characters or less ・CPU decode: decode-main (11 characters) # DataFlow name must be no more than 32 characters. ・CPU copy branch: copy-branch-main (16 characters) * DataFlow name must be 27 characters or less ・CPU glue: glue-fdma-to-tcp-main (21 characters) # DataFlow name must be no more than 22 characters. ・GPU advanced inference: high-infer-main (15 characters) * DataFlow name must be 28 characters or less ・GPU advanced inference: low-infer-main (14 characters) * DataFlow name must be 29 characters or less
metadata	Namespace	-	-	Be set by the user
	FunctionChainRef	WBNamespacedName	Required	Name and Namespace of FunctionChain where the deployment will take place
	Name	string	Required	
	Namespace	string	Required	
	DryRun	*bool	Optional	Used to pre-validate deployments (future features)
	StartPoint	*StartEndPoint	Optional	Set Start Point
	IP	string	Required	IP address of the starting point If set, WBConnection params with From "wb-start-of-chain" will be set to a value with a key of "TargetIP"
	Port	int32	Required	Port number of the starting point If set, WBConnection params with From "wb-start-of-chain" will be set to a value with a key of "TargetPort"
	Protocol	corev1.Protocol	Required	Protocol at the starting point corev1.Protocol is a string of "TCP," "UDP," or "SCTP." You may define your own strings If set, WBConnection params with From "wb-start-of-chain" will be populated with key "Protocol"
	EndPoint	*StartEndPoint	Optional	Set End Point
	IP	string	Required	IP address of the endpoint If set, WBConnection params with To "wb-end-of-chain" will be set to a value with a key of "TargetIP"
	Port	int32	Required	Port number of the endpoint If set, WBConnection params with To "wb-end-of-chain" will be set to a value with a key of "TargetPort"
	Protocol	corev1.Protocol	Required	Protocol at the end point corev1.Protocol is a string of "TCP," "UDP," or "SCTP." You may define your own strings If set, WBConnection params with To "wb-end-of-chain" will be set to a value with a key of "Protocol"
	FunctionUserParameter	[]FunctionParamStruct	Optional	User defined parameters for each function # This parameter overwrite FunctionChain CustomParameter.
	FunctionKey	string	Required	key value in DataFlowStatus. FunctionChain .FunctionChainSpec.Functions
	UserParams	map[string]intstr.IntOrString	Required	map key is the user-defined parameter name. The value of map is the value of a user-defined integer/string parameter.
	ConnectionUserParameter	[]ConnectionParamStruct	Optional	User-defined parameters per connection *This parameter overwrite FunctionChain's CustomParameter.
	From	FromToFunctionInfo	Required	Source Function Information in Connection
	FunctionKey	string	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	To	FromToFunction	Required	Destination Function information in Connection
	FunctionKey	FromToFunctionInfo	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	UserParams	map[string]intstr.IntOrString	Required	map key is the user-defined parameter name. The value of map is the value of a user-defined integer/string parameter.
	FunctionTargetSelectors	[]FunctionTargetSelector	Optional	If the user specifies where the Function is deployed
	FunctionKey	string	Required	Function to which the user specifies to deploy. FunctionChain .FunctionChainSpec.Functions key value
	NodeName	*string	Optional	Name of the Node to deploy to
	DeviceType	*string	Optional	DeviceType of the deployment destination
	DeviceIndex	*int32	Optional	DeviceIndex of the deployment destination
	RegionName	*string	Optional	Minimum space on the device to which it is deployed
	FunctionIndex	*int32	Optional	If you want to reuse a deployed function, set the serial number of the deployed function on the smallest region of the destination device. If you are deploying a new function, do not set it.
	Requirements	*DataFlowRequirementsStruct	Optional	List the requirements that must be met at scheduler Requirements can be specified in units of entire function chain, one function, or one connection.
	All	*AllRequirementsInfo	Optional	List requirements for function chain as a whole (function chain requirements are assumed to be one factor maximum)
	Capacity	int32	Required	Each connection and the amount of load assumed by each connection (fps). (each connection and the amount of resources required for each connection)
	Functions		Optional	Describe the requirements for each function that makes up function chain
	FunctionKey	string	Required	key value in DataFlowStatus. FunctionChain .FunctionChainSpec.Functions
	Capacity	int32	Required	Estimated load by this function (fps) (Resource capacity required for this function (processing power consumed by this function))
	Connections	[]ConnectionRequirementsInfo	Optional	Describe the requirements for each connection that makes up function chain
	From	FromToFunctionInfo	Required	Source Function Information in Connection
	FunctionKey	string	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	To	FromToFunction	Required	Destination Function information in Connection
	FunctionKey	FromToFunctionInfo	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	Capacity	int32	Required	Estimated load from this connection (fps) (amount of resources required for this connection)
Spec	UserRequirement	*string	optional	Specifies metadata.name of UserRequirement ConfigMap to be referenced to obtain various configuration information for DataFlow scheduling
	Status	string	Required	The state of DataFlow, following five patterns (1)"(empty character): Initial status. Obtaining information necessary for scheduling (2)"Scheduling in progress" — Scheduling in progress (3)"WBFunction/WBConnection creation in progress": Creating deployment request (4)"WBFunction/WBConnection created" — Deployment request created. Checking the deployment status of each Function and Connection (5)"Deployed" — Deployed
	FunctionChain	*FunctionChain	Optional	Stores FunctionChain Spec, Status
	FunctionType	[]*FunctionType	Optional	Stores the Spec and Status of FunctionType that make up FunctionChain.
	ConnectionType	[]*ConnectionType	Optional	Stores the Spec and Status of the ConnectionType that constitutes FunctionChain.
	ScheduledFunctions	map[string]FunctionScheduleInfo	Optional	(set by WB scheduler controller) key matches the key value of DataFlowStatus. FunctionChain .FunctionChainSpec.Functions.
	NodeName	string	Required	Scheduled Node
	DeviceType	string	Required	Scheduled DeviceType
	DeviceIndex	int32	Required	Scheduled DeviceIndex
	RegionName	string	Required	Minimum space on scheduled deployment destination devices
	FunctionIndex	*int32	Optional	If you are deploying a new function, do not have this parameter. If you are reusing a deployed function, the deployed function on the smallest region of the destination device
	ScheduledConnections	[]ConnectionScheduleInfo	Optional	(set by WB scheduler controller)
	From	FromToFunctionScheduleInfo	Required	Source Function Information in Connection
	FunctionKey	string	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	Port	*int32	Optional	Output port number of data source Function (0 is specified when Function is 1 output)
	InterfaceType	*string	Optional	Interface type of output used by the source Function (From function) in this connection
	To	FromToFunctionScheduleInfo	Required	Destination Function information in Connection
	FunctionKey	string	Required	Destination Function in Connection. key of Dataflow.Status. FunctionChain .Spec.Functions is set
	Port	*int32	Optional	Output port number of the data destination Function (if Function is 1 output, specify 0)
	InterfaceType	*string	Optional	Input interface type used by the destination Function (To function) in this Connection
	ConnectionMethod	string	Required	Specifies the From and To connection method (stores the value to be given to the Type of the WBConnectionSpec)
	ConnectionPath	[]WBConnectionPath	Optional	Path information between the From and To functions ・Scheduler sets the ID of the transit device/interface/network (ID of EntityInfo in TopologyInfo (CR)) on the route from the From deployment destination to the To deployment destination, and information on whether Incoming or Outgoing usage will increase or both. ・Stored in []WBConnectionPath in order of traversal ・This is the value given to the ConnectionPath of the WBConnectionSpec.
	EntityID	string	Required	ID of the via device/interface/network in the route (ID of EntityInfo in TopologyInfo (CR))
	UsedType	WBIOUsedType	Required	Incoming or Outgoing information about whether usage will increase or both, and no usage increase. One of the following four patterns of character strings is set. ・"Incoming" ・"Outgoing" ・"IncomingAndOutgoing" ・"(When EntityType is other than "interface" or "network")
	StartPoint	string	Optional	IP address and port number to which DataFlow data is to be input
Status	EndPoint	string	Optional	IP address and port number to retrieve data from DataFlow

■SchedulingData

custom resource with information on DataFlow's candidate fleet

	Name	Type	Req/Opt	Description
	Name	-	-	WB scheduler controller sets the same value as metadata.name in DataFlow
metadata	Namespace	-	-	WB scheduler controller sets the same value as metadata.namespace in DataFlow
	FilterPipeline	[]string	Required	Slice of the filter name to use
	Status	string	Required	The state of SchedulingData. The following three patterns. (1)"Filtering": Filtering in progress (2)"Finish": Finished filtering (3)"Failed" — Filtering operation failed.
	CurrentFilterIndex	*int32	Optional	Current Filter Number
	TypeCombinations	[]TypeCombinationStruct	Optional	Function DeviceType, ConnectionType, Score slices
	DeviceTypes	map[string]string	Optional	slice of DeviceType
	ConnectionTypes	ConnectionTypes	Optional	ConnectionType slices
	Score	*int64	Optional	Slicing Score
	TargetCombinations	[]TargetCombinationStruct	Optional	Slice of potential DataFlow locations
	ScheduledFunctions	map[string]FunctionScheduleInfo	Optional	(set by the filter specified in filterPipeline) Key is set to a key value of DataFlowStatus. FunctionChain .FunctionChainSpec.Functions
	NodeName	string	Required	NodeName of the candidate schedule destination of the Function
	DeviceType	string	Required	DeviceType of the candidate schedule destination of the Function
	DeviceIndex	int32	Required	DeviceIndex of the candidate schedule destination of the Function
	RegionName	string	Required	Minimum space on the device to which a candidate schedule a Function is deployed
	FunctionIndex	*int32	Optional	FunctionIndex of the candidate schedule destination of the Function
	ScheduledConenctions	[]ConnectionScheduleInfo	Optional	(set by the filter specified in filterPipeline)
	From	FromToFunctionScheduleInfo	Required	Source Function information in the connection of the schedule destination candidate
	FuntionKey	string	Required	Source Function in Connection. key value of Dataflow.Status. FunctionChain .Spec.Functions is set
	Port	*int32	Optional	Output port number of data source Function (0 is specified when Function is 1 output)
	InterfaceType	*string	Optional	Interface type of output used by the sender Function (From function) in this connection
	To	FromToFunctionScheduleInfo	Required	Destination Function information in the connection of the schedule destination candidate
	FuntionKey	string	Required	Destination Function in Connection. key value of Dataflow.Status. FunctionChain .Spec.Functions is set
	Port	*int32	Optional	Output port number of the data destination Function (if Function is 1 output, specify 0)
	InterfaceType	*string	Optional	Input interface type used by the destination Function (To function) in this Connection
Status	Socre	*int64	Optional	Score for DataFlow deployment candidates

■FunctionType

custom resource representing a Function available in function chain

	Name	Type	Req/Opt	Description
	Name	-	-	Be set by the user
metadata	Namespace	-	-	Optional by the user (specified by the administrator according to the function's template category)
	FunctionName	string	Required	The function name in the function catalog. Value to be set for FunctionName in FunctionChain
	FunctionInfoCMRef	WBNamespacedName	Required	
	Name	string	Required	Specifies the metadata.Name of FunctionInfo (ConfigMap) where the FunctionName function is defined
	Namespace	string	Required	Specifies the metadata.Namespace of FunctionInfo (ConfigMap) where the FunctionName function is defined
Spec	Version	string	Required	The version of the Function. Used to ensure uniqueness with Name+Version
	Status	string	Required	Function Availability (Ready/Not Ready/Error)
	RegionTypeCandidates	[]string	Optional	Candidate RegionType used when FunctionType is deployed
Status	RecommendConnection	[]string	Optional	Stores the recommended connection for each RegionType Stores the value from FunctionInfo.FunctionInfoRecommend.DeviceType in the format <DeviceType>--<interface>

■FunctionInfo (ConfigMap)

information equivalent to function catalog information

	Name	Type	Req/Opt	Description
metadata	Name	-	-	"funcinfo-"+ function name
	Namespace	-	-	Be set by the user (Assumptions specified by the administrator according to function catalog categories, etc.)
data	deployableItems	string	Required	A string value of an array whose elements are json objects consisting of the following key-values:
	name	string	Required	A name that refers to an element in the deployableItems array.
	regionType	string	Required	Deployable region type
*The data field value is of type map [string] string (ConfigMap specification).	inputInterfaceType	string	Required	Interface type of input available when deployed to the above <regionType>
	outputInterfaceType	string	Required	Interface type of output available when deployed to the above <regionType>. The following three interface type values can be set. ・"dev25gether" (Interface type used by functions running on the FPGA for external connection via the NIC of the FPGA) ・"host100gether"(Interface type used by functions running on GPUs for external connections via host NICs) ・"mem" (Interface type used by functions running on an FPGA, GPU, or CPU for connection via shared memory)"
	configName	string	Required	Name of information required for deployment when deploying to <regionType> above and using <inputInterfaceType> and <outputInterfaceType>
	specName	string	Required	Name of the function spec information when deployed to <regionType> above and using <inputInterfaceType> and <outputInterfaceType> above
	spec	string	Required	Specification information for the function A string value of an array whose elements are json objects consisting of the following key-values:
	name	int32	Required	A name that refers to an element of the spec array.
	minCore	int32	Required	The minimum value of the resource to use. Currently always assumes "1"
	maxCore	int32	Optional	The maximum value of the resource to use. Currently always assumes "1"
	maxDataFlowsBase	int32	Optional	Base maximum percentage DataFlow (maximum installed WBFunction). Depend on the number of channels in the circuit
	maxCapacityBase	int32	Optional	Base Max Processing Power (fps)
	maxInputNum	int32	Optional	Maximum number of function inputs
	maxOutputNum	int32	Optional	Maximum number of outputs of the function
	latencyBase	string	Optional	Base delay time, units (ms, us) available
	latencySizeScale	string	Optional	Change in delay time by DataSize
	latencyCoreScale	string	Optional	Variation of delay time with number of resources
	perfBase	string	Optional	Base throughput, units (Byte/sec, MiB/sec) available
	perfSizeBase	string	Optional	Throughput change with DataSize
	perfCoreBase	string	Optional	Throughput Variation with Number of Resources
	powerBase	string	Optional	Basic power consumption, units (mW, W) available
	powerSizeBase	string	Optional	Power consumption change by DataSize
	powerCoreBase	string	Optional	Power consumption variation with number of resources
	recommend	string	Optional	The recommended deployableItem (the destination region type & the set of available I/O interface types). Multiple settings allowed A string value of an array whose elements are json objects consisting of the following key-values.
	deployableItemName	string	Required	deployableItem name

■Strategy (ConfigMap)

A configmap that specifies the execution strategy of the filter in DataFlow scheduling

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Set the same value as DataFlow's metadata.namespace or "default"
data	referenceParameter	string	Optional	Specify metadata.Name of ConfigMap to reference for setting strategy
*The data	filterPipeline	[]string	Optional	Array of Filter names to use
field value	selectTop	int	Optional	Get up to < set value > th score of filtering results
is of type				where <N> is the index number of the filter specified for filterPipeline.
map	<N>.referenceParameter	string	Optional	For the <N> th filter in filterPipeline, specify the metadata.Name of the ConfigMap to reference for setting the strategy.
[string]				where <N> is the index number of the filter specified for filterPipeline.
string	<N>.selectTop	int	Optional	Get up to the < set value > th Score of the filter result of the <N> th filterPipeline Scorefilter
(ConfigMap				where <N> is the index number of the filter specified for filterPipeline.
specificatio	<N>.<parameterName>	T	Optional	Specify any user-defined value for the <N> th filter.

■FunctionChain

Resource for representing Dataflow configuration. Combining FunctionType and ConnectionType

	Name	Type	Req/Opt	Description
	Name	-	-	Be set by the user
metadata	Namespace	-	-	Optional by the user (specified by the administrator according to the FC template category)
	FunctionTypeNamespace	string	Required	FunctionType Namespace
	ConnectionTypeNamespace	string	Required	ConnectionType Namespace
	Functions	map[string]FunctionStruct	Required	A map of the Functions that make up FunctionChain. key is the function identifier specified in Connections From or To (A string that is unique in this FunctionChain resource. The string to be used in each CR FunctionKey) Since the map key value (Function identifier) of this parameter is part of WBFunction's metadata.name, the number of characters must satisfy the character limit of WBFunction's metadata.name.
	FunctionName	string	Required	
	Version	string	Required	FunctionTypeSpecify the Name, Version defined in Spec. In the future, we want Version to support equal and not equal signs.
	CustomParameter	map[string]intstr.IntOrString	Optional	Define the settings to be given to the Function
	Connections	[]ConnectionStruct	Required	List of Connections that make up FunctionChain
	From	FromToFunction	Required	Source Function Information in Connection
	FunctionKey	string	Required	Identifier of the data source Function. Set Functions map key value
	Port	int32	Required	Output port identification number of the data transmission source Function (If Function is 1 output, specify 0)
	To	FromToFunction	Required	Destination Function information in Connection
	FunctionKey	string	Required	Identifier of the data destination Function. Set Functions map key value
	Port	int32	Required	Input port identification number of the data transmission destination Function (If Function is 1 input, specify 0)
	ConnectionTypeName	string	Required	Specify resource name or "auto" for ConnectionType. Currently always assumes "auto"
Spec	CustomParameter	map[string]intstr.IntOrString	Optional	Defines the settings to be given to Connection
Status	Status	string	Required	FunctionChain Availability (Ready/Not Ready/Error)

■UserRequirement (ConfigMap)

Strategy configuration maps used for DataFlow scheduling and configuration maps that specify filtering conditions for function/connection deployment destinations

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Set the same value as DataFlow's metadata.namespace or "default"
data *The data field value is of type map [string] string (ConfigMap specification).	strategy	string	Required	Specify metadata.name in Strategy's ConfigMap.
	scoreThreshold	map[string]string	Optional	Score threshold for each score type (currently unused)
	nodeGroups	[][]string	Optional	Specify a function to be deployed to the same Node. The outer slice value is the NodeName to deploy to. The values in the inner slice are key values from FunctionChain .FunctionChainSpec.Functions.
	requestNodeNames	map[string][]string	Optional	Specify the NodeName to which the function specified in the map key is to be deployed or not deployed. The key of map is the key value of FunctionChain .FunctionChainSpec.Functions that indicates the Function. The value of map is an array of NodeNames. ' - 'designates it as non-deployable.
	requestDeviceTypes	map[string][]string	Optional	Specify the DeviceType of the deployment destination/non-deployment destination of the function specified in the map key. The key of map is the key value of FunctionChain .FunctionChainSpec.Functions that indicates the Function.
	requestConnectionTypes	map[string][]string	Optional	The map key is the index number (specified as a string) in FunctionChain. The value of map is an array of ConnectionSideType. ' - 'is specified as an exclusion target.
	requestFunctionTargets	map[string][]string	Optional	Specifies FunctionTarget to which the function specified in the map key is to be deployed or not deployed. The key of map is the key value of FunctionChain .FunctionChainSpec.Functions that indicates the Function. The map value is an array of FunctionTarget. ' - 'designates it as non-deployable.
	requestConnectionTargets	map[string][]string	Optional	The map key is the index number (specified as a string) in FunctionChain. The value of map is an array of ConnectionTypeName. ' - 'is specified as an exclusion target.
	requestRegionNames	map[string][]string	Optional	Specify the RegionName of the deployment-destination or non-deployment-destination of the function specified in the map key. The key of map is the key value of FunctionChain .FunctionChainSpec.Functions that indicates the Function. The value of map is an array of RegionName. ' - 'designates it as non-deployable.
	requestFunctionIndexes	map[string][]string	Optional	Specifies the FunctionIndex to which the function specified in the map key is to be deployed or not deployed. The key of map is the key value of FunctionChain .FunctionChainSpec.Functions that indicates the Function. The value of map is an array of FunctionIndex. ' - 'designates it as non-deployable.
	functionTargetNameSpace	string	Optional	Specifies the metadata.namespace of FunctionTarget to be referenced when executing a Filter that uses device
	connectionTargetNameSpace	string	Optional	Specify metadata.namespace for ConnectionTarget
	topologyInfoName	string	Optional	Specify metadata.Name of the TopologyInfo to be referenced when executing a Filter that uses topology information
	topologyInfoNameSpace	string	Optional	Specifies the metadata.Namespace of the TopologyInfo to be referenced when executing a Filter that uses topology information

■WBFunction

custom resource for the Function to deploy. custom resource of FPGAFunction, GPUFunction, or CPUFunction is created based on the information in this custom resource.

	Name	Type	Req/Opt	Description
	Name	-	-	<metadata.name in DF>-wbfunction-<FC Functions key value > Character limit: Adjust the key value of metadata.name in DataFlow (DF) and Functions in FunctionChain (FC) so that the value of this parameter is within 55 characters.
metadata	Namespace	-	-	(specified by DF controller)
	DataFlowRef	WBNamespacedName	Required	Identify Original Dataflow
	Name	string	Required	
	Namespace	string	Required	
	NodeName	string	Required	
	NodeSelector	map[string]string	Optional	
	Affinity	*corev1.Affinity	Optional	
	DeviceType	string	Required	Destination Device Type
	DeviceIndex	int32	Required	Number of the destination device
	RegionName	string	Required	Deployment destination. Add the Region parameter for FunctionTarget
	FunctionIndex	*int32	Optional	Deployed functions on the destination. If this parameter is not present, a new deployment is requested. If this parameter is present, it means that the dataflow should be stored in the circuit or pod corresponding to the already deployed Functions with that FunctionIndex. See Dataflow.FunctionScheduleInfo
	FunctionName	string	Required	Function name
	ConfigName	string	Required	Config name required for deploy (name of ConfigMap in xxxfunc-config)
	InputInterface	map[string]string	Optional	The input interface type of Function is set. key is the input port number (interface identification number) as a character string.
	OutputInterface	map[string]string	Optional	The output interface type of Function is set. key is the output port number (interface identification number) as a character string
	Params	map[string]intstr.IntOrString	Optional	Integer/String Parameters
	PreviousWBFunctions	map[string]FromToWBFunction	Optional	Information on WBFunction in the first part. If there is no previous WBFunction (= wb-start-of-chain), it is uninstalled. key is the same as InputInterface and the input port number (interface identification number) is expressed as a character string.
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace from previous WBFunction
	Port	int32	Required	Output port number of the remote WBFunction that is connected to the input port number (key value) of the local WBFunction
	NextWBFunctions	map[string]FromToWBFunction	Optional	Information on WBFunction in the second part. Not set if there is no next WBFunction (= wb-end-of-chain) key is the same as OutputInterface, and the output port number (interface identification number) is expressed as a character string.
	WBFunctionRef	WBNamespacedName	Required	WBFunction resource name and namespace
	Port	int32	Required	Input port number of the remote WBFunction that is connected to the output port number (key value) of the local WBFunction
	MaxDataFlows	*int32	Optional	Maximum number of deployed Function DF (number of WBFunction). It depends on the number of channels of the circuit, etc.
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployed Function. It depends on the circuit and pod implementation.
	Requirements	*WBFunctionRequirementsInfo	Optional	The requirements that must be met at scheduler time have already been met, but the various resource controllers control it in declared value base via this parameter to know the value.
Spec	Capacity	int32	Required	Estimated load by this function (fps) (Resource capacity required for this function (processing power consumed by this function))
	DataFlowRef	WBNamespacedName	Required	Identify Original Dataflow
	Status	WBDeployStatus	Required	Deploy state. following seven patterns (1)": * Currently not used (2)"Failed": * Currently not used (3)"Allocated": * Currently not used (4)"Deployed" — Deployed (5)"Waiting" — Deploying (6)"Released": * Currently not used (7)"Terminating": * Currently not used
	NodeName	string	Required	Deploy to node name
	DeviceType	string	Required	Deploy to Device Type
	DeviceIndex	int32	Required	Number of the device to deploy to
	RegionName	string	Required	Deploy to
	FunctionIndex	int32	Required	The deployed Function to Deploy
	FunctionName	string	Required	Function name
	ConfigName	string	Required	Config name required for deploy (name of ConfigMap in xxxfunc-config)
	InputInterface	map[string]string	Optional	Select input interface of Function
	OutputInterface	map[string]string	Optional	Select output interface of Function
	Params	map[string]intstr.IntOrString	Optional	Integer/String Parameters
	PreviousWBFunctions	map[string]FromToWBFunction	Optional	Information on WBFunction in the first part. Not set if there is no previous WBFunction (= wb-start-of-chain). key is the same as InputInterface and the input port number (interface identification number) is expressed as a character string.
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace from previous WBFunction
	Port	int32	Required	Output port number of the remote WBFunction that is connected to the input port number (key value) of the local WBFunction
	NextWBFunctions	map[string]FromToWBFunction	Optional	Information on WBFunction in the second part. Not set if there is no next WBFunction (= wb-end-of-chain) key is the same as OutputInterface, and the output port number (interface identification number) is expressed as a character string.
	WBFunctionRef	WBNamespacedName	Required	WBFunction resource name and namespace
	Port	int32	Required	Input port number of the remote WBFunction that is connected to the output port number (key value) of the local WBFunction
	MaxDataFlows	*int32	Optional	Maximum number of deployed Function DF (number of WBFunction). It depends on the number of channels of the circuit, etc.
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployed Function. It depends on the circuit and pod implementation.
	SatisfiedRequirements	WBFunctionRequirementsInfo	Optional	The requirements that must be met at scheduler time have already been met, but the various resource controllers control it in declared value base via this parameter to know the value.
Status	Capacity	int32	Required	Estimated load by this function (fps) (Resource capacity required for this function (processing power consumed by this function))

■WBConnection

custom resource for the Connection to deploy. custom resource of either EthernetConnection or PCIeConnection is made based on the information of this custom resource.

	Name	Type	Req/Opt	Description
metadata	Name	-	-	<metadata.name in DF>-wbconnection-<FC Connections From>-<FC Connections To> (Based on the review of the first edition)
	Namespace	-	-	(specified by DF controller)
Spec	DataFlowRef	WBNamespacedName	Required	Identify Original Dataflow
	Name	string	Required	
	Namespace	string	Required	
	ConnectionMethod	string	Required	How From and To are connected. If auto, the connection is chosen as determined by the deploy destination on both ends. Note that each ConnectionController does not support auto, and the ConnectionWebhook replaces the Type. Currently, it stores information indicating whether it is an outer connection or an inner connection ("host-100gether," "host-mem").
	ConnectionPath	[]WBConnectionPath	Optional	Path information between FromFunction and ToFunction (The value of ConnectionPath parameter of ScheduledConnections in DataFlow is stored as is.)
	EntityID	string	Required	Path information between FromFunction and ToFunction (The value of the EntityID parameter for each element of ConnectionPath in DataFlow's ScheduledConnections is stored as is.)
	UsedType	WBIOUsedType	Required	Path information between FromFunction and ToFunction (The value of the UsedType parameter for each element of ConnectionPath in DataFlow's ScheduledConnections is stored as is.)
	From	FromToWBFunction	Required	Connection source WBFunction information
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the connection source WBFunction. In the case of a DataFlow starting point, it is a reserved word to indicate an external connection.
	Port	int32	Required	Output port number (interface identification number) of the connection source WBFunction
	To	FromToWBFunction	Required	Deployment destination WBFunction information
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of WBFunction to connect to. In the case of an DataFlow endpoint, a reserved word is used to indicate an external connection.
	Port	int32	Required	Input port number (interface identification number) of the connection destination WBFunction
	Params	map[string]intstr.IntOrString	Optional	Integer/String Parameters
	Requirements	*WBConnectionRequirementsInf	Optional	The topology information controller uses this parameter to update the topology information device interface usage and network usage in declared value base.
	Capacity	int32	Required	Indicates the assumed load on the device interface and network for the connection between FromFunction and ToFunction. This term, the unit of assumed load is treated as fps.
Status	DataFlowRef	WBNamespacedName	Required	Identify Original Dataflow
	Status	WBDeployStatus	Required	Deploy state for the entire connection. following seven patterns (1)": * Currently not used (2)"Failed": * Currently not used (3)"Allocated": * Currently not used (4)"Deployed" — Deployed (5)"Waiting" — Deploying (6)"Released": * Currently not used (7)"Terminating": * Currently not used
	ConnectionMethod	string	Required	Type of connection deployed
	ConnectionPath	[]WBConnectionPath	Optional	Path information between FromFunction and ToFunction for deployed connections (The value of ConnectionPath parameter of ScheduledConnections in DataFlow is stored as is.)
	EntityID	string	Required	Path information between FromFunction and ToFunction for deployed connections (The value of the EntityID parameter for each element of ConnectionPath in DataFlow's ScheduledConnections is stored as is.)
	UsedType	WBIOUsedType	Required	Path information between FromFunction and ToFunction for deployed connections (The value of the UsedType parameter for each element of ConnectionPath in DataFlow's ScheduledConnections is stored as is.)
	From	FromToWBFunction	Required	
	WBFunctionRef	WBNamespacedName	Required	Deployment source Function resource name
	Port	int32	Required	Output port number of the connection source Function
	To	FromToWBFunction	Required	
	WBFunctionRef	WBNamespacedName	Required	Connection destination Function resource name
	Port	int32	Required	Input port number of the connection destination Function
	Params	map[string]intstr.IntOrString	Optional	Integer/String Parameters
	SatisfiedRequirements	*WBConnectionRequirementsInf	Optional	The topology information controller uses this parameter to update the topology information device interface usage and network usage in declared value base.
	Capacity	int32	Required	Indicates the assumed load on the device interface and network for the connection between FromFunction and ToFunction. This term, the unit of assumed load is treated as fps.
	IOs	map[string]WBConnectionIO	Optional	I/O information used during deploy
	Status	string/WBDeployStatus	Required	Deploy state of this I/O
	IoType	string/WBIOType	Required	Direction of I/O used (Incoming/Outgoing from Device perspective)
	Node	string	Required	Name of the Node to which I/O will be deployed
	DeviceType	string	Required	Deployment destination I/O DeviceType
	DeviceIndex	int	Required	Deployment destination I/O Device number
	IoName	string	Required	I/O name to use
	Port	int	Required	I/O port number to use
	IntParams	map[string]int	Optional	integer parameter
	StrParams	map[string]string	Optional	string parameter
	Connections	[]WBConnectionEdge	Optional	SubConnection when the connection between Functions is broken down according to the granularity of resource management
	Status	string/WBDeployStatus	Required	Deploy state of this SubConnection
	From	WBNamespacedName	Required	Connection source name (Function, I/O)
	To	WBNamespacedName	Required	Connection destination name (Function, I/O)
	IntParams	map[string]int	Optional	integer parameter
	StrParams	map[string]string	Optional	string parameter

■GPUFunction

custom resource with information about the Function to deploy to the GPU

There is advanced inference/lightweight inference using Gstremer as a sample processing module. The connection method assumes PCIe connection from the FPGA and TCP connection from the CPU.

Converted from WBFunction and auto-generated

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
Spec	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction .spec.DataFlowRef)
	FunctionName	string	Required	Name of the Function to execute (Equivalent to the parent CR WBFunction.spec.FunctionName)
	NodeName	string	Required	Destination node name (Equivalent to the parent CR WBFunction.spec.NodeName)
	DeviceType	string	Required	Destination Device Type (Equivalent to the parent CR WBFunction.spec.DeviceType)
	AcceleratorIDs	[]AccIDInfo	Required	The identity of the destination device (considering the possibility of assigning more than one device to a GPUFunction in the future)
	PartitionName	*string	Optional	Information identifying the Function to which the accelerator is to be assigned (container name in the case of GPUFunction)
	ID	string	Required	Identifier of the accelerator to assign to the function (in the case of GPUFunction, the UUID of the GPU)
	RegionName	string	Required	Distinguished name of the deployment region to which it is deployed
	FunctionIndex	*int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.spec.RegionName) (for GPUFunction, the Id of the deployed Pod) If this parameter is not present, it means that a new deployment is requested. If this parameter is present, it means that the dataflow should be stored in the circuit or pod corresponding to the already deployed Functions with that FunctionIndex.
	Envs	[]EnvsInfo	Optional	For setting parameters for the processing module. Copied to the pod's containers.env (for each container)
	PartitionName	string	Required	Information identifying the function to pass this argument to (container name for GPUFunction)
	EachEnv	[]EnvsData	Required	List with information for each environment variable
	EnvKey	string	Required	Key values of environment variables
	EnvValue	string	Required	Value value of the environment variable
	RequestMemorySize	*int32	Optional	Minimum memory size required by the container to boot for this GPUFunction [Gib] (Not currently used. The value specified in GPUFunc configuration information is used.)
	SharedMemory	*SharedMemorySpec	Optional	Configuration Information Required to Perform a PCIe Connection over Shared Memory
	FilePrefix	string	Required	Information to identify the PCIe connection on the dpdk side
	CommandQueueID	string	Required	Identity of the CommandQueue used for data transfer
	SharedMemoryMiB	int32	Required	Required size of the shared memory used for data transfer on the PCIe connection [MegaByte] (Not currently used. The value is fixed inside the processing module.)
	Protocol	*string	Optional	Receiving communication protocol (required if data is received (source is present))
	ConfigName	string	Required	Config name required for Deploy (name of ConfigMap in gpufunc-config-xxx) (Equivalent to the parent CR WBFunction.spec.ConfigName)
	PreviousFunctions	map[string]FromToWBFunction	Optional	FunctionCR information for each resource system in the previous section. If there is no previous Function (= wb-start-of-chain), this parameter is not set. key is the same as InputInterface and the input port number (interface identification number) is expressed as a character string. (Equivalent to parent WBFunction.spec.PreviousWBFunctions)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Output port number of the other Function connected to the input port number (key value) of the current
	NextFunctions	map[string]FromToWBFunction	Optional	FunctionCR information of various resource systems in the second part. If there is no next Function (= wb-end-of-chain), it is not set. key is the same as OutputInterface, and the output port number (interface identification number) is expressed as a character string. (Equivalent to parent WBFunction.spec.NextWBFunctions)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Input port number of the other Function connected to the output port number (key value) of the current
	Params	map[string]intstr.IntOrString	Optional	Integer/String parameters (Equivalent to the parent CR WBFunction.spec.Params)
Status	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction .status.DataFlowRef)
	FunctionName	string	Required	Function name (Equivalent to the parent CR WBFunction.status.FunctionName)
	ImageURI	string	Required	Name of the container image of the container to be started by the GPUFunction
	SharedMemory	*SharedMemorySpec	Optional	Shared memory information set for GPUFunction (only when PCIe is connected)
	FilePrefix	string	Required	Identity of the CommandQueue used for data transfer
	CommandQueueID	string	Required	Information to identify the PCIe connection on the dpdk side
	sharedMemoryMiB	int32	Required	Required size of the shared memory used for data transfer on the PCIe connection [MegaByte] (Currently unused.). The value is fixed inside the processing module.)
	RxProtocol	*string	Optional	Receiving communication protocol (listed if data is received (source is present))
	TxProtocol	*string	Optional	Sender's communication protocol (listed if data is sent (destination is present))
	ConfigName	string	Required	Config name required for Deploy (name of ConfigMap in gpufunc-config-xxx) (Equivalent to the parent CR WBFunction.status.ConfigName)
	VirtualNetworkDeviceDriverType	string	Optional	CNI Plug-ins for 2nd NICs on Pod
	AdditionalNetwork	string	Optional	Whether to create a 2nd NIC on Pod
	FunctionIndex	*int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.status.RegionName)
	PodName	*string	Optional	CR name of Pod you created
	StartTime	metav1.Time	Required	Creation time
	Status	string	Required	The state of GPUFunction. Have the following two values ・Running: successful creation ・Pending: Creating *Currently I don't use Pending, I just run it after Pod creation is complete.
	IPAddress	*string	Optional	IP address (currently unused)
	AcceleratorStatuses	[]AccStatusesByContainer	Optional	State of the device that deployed this GPUFunction. It is recorded for each Function (for each container in the case of GPUFunc).
	PartitionName	*string	Optional	Information identifying the Function for which status is set
	Statuses	[]AccStatuses	Optional	Records status for each Accelerator assigned to GPUFunction
	AcceleratorID	*string	Optional	Device UUID
	Status	*string	Optional	Device status. Three types (deployed | deploying | error) are assumed.

■FPGAFunction

custom resource with information about the Function to be deployed on FPGA (modularization) for FJ

Sample processing includes filter and resize for advanced and lightweight inference. Ethernet and PCIe connections are assumed.

Converted from WBFunction and auto-generated

	Name	Type	Req/Opt	
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
Spec	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction .spec.DataFlowRef)
	FunctionName	string	Required	Name of the Function to execute (Equivalent to the parent CR WBFunction.spec.FunctionName)
	NodeName	string	Required	Destination node name (Equivalent to the parent CR WBFunction.spec.NodeName)
	DeviceType	string	Required	Destination Device Type (Equivalent to the parent CR WBFunction.spec.DeviceType)
	AcceleratorIDs	[]AccIDInfo	Required	The identity of the deployed device (considering the possibility of assigning more than one device to an FPGAFunction in the future)
	PartitionName	*string	Optional	Information (FunkKernelId in the case of FPGAFunction) identifying the Function to which the accelerator is to be assigned
	ID	string	Required	Accelerator identification to assign to the Function (FPGA device file path for FPGAFunction)
	RegionName	string	Required	Distinguished name (Equivalent to the parent CR WBFunction.spec.RegionName) of the deployment region to which you want to deploy
	FunctionIndex	*int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.spec.RegionName) (In the case of FPGAFunction, the FuncCHId of the destination circuit) If this parameter is not present, it means that a new deployment is requested. If this parameter is present, it means that the dataflow should be stored in the circuit or pod corresponding to the already deployed Functions with that FunctionIndex.
	Envs	[]EnvsInfo	Optional	For setting processing module parameters (not used for FPGAFunction). The parameters of the processing module are specified in the Config information (fpgafunc-config-xxx).)
	PartitionName	string	Required	-
	EachEnv	[]EnvsData	Required	-
	EnvKey	string	Required	-
	EnvValue	string	Required	-
	ConfigName	string	Requried	Config name required for deploy (name of ConfigMap in fpgafunc-config-xxx) (Equivalent to WBFunction .spec.ConfigName, which is the parent CR)
	SharedMemory	*SharedMemorySpec	Optional	Configuration Information Required to Perform a PCIe Connection over Shared Memory
	FilePrefix	string	Required	Information to identify the PCIe connection on the dpdk side
	CommandQueueID	string	Required	Identity of the CommandQueue used for data transfer
	SharedMemoryMiB	int32	Required	Required size of shared memory used for data transfer on PCIe connections [MegaByte] (not used in the
	FunctionKernelID	*int32	Optional	Of the child bs modules written to the FPGA: ID of the Function module whose FunctionChannel (FuncCH) is to be assigned to this FPGAFunction. (The Function module is the module in the child bs that is responsible for executing the processing module.)
	FunctionChannelID	*int32	Optional	ID of the FuncCH assigned to FPGAFunction (FunCH is a virtual resource allocated to each DataFlow (FPGAFunc) in order to share an FPGA circuit with multiple FPGAFunc.). managed by the Function module)
	PtuKernelID	*int32	Optional	Of child bs modules written to the FPGA: ID of the PTU module to be used by FPGAFunction (The Ptu module is a module in the child bs that is the NW termination of Ethernet communication)
	FrameworkKernelID	*int32	Optional	Chain Control Module ID(Modules.Chain.ID equivalent of ChildBs)
	Rx	RxTxData	Optional	Receiving network information to be assigned to FPGAFunction
	Protocol	string	Required	Receiving communication protocol
	IPAddress	*string	Optional	Receiving IP address (for Ethernet connections. Not required for PCIe connections)
	Port	*int32	Optional	Receiving port number (for Ethernet connection. Not required for PCIe connections)
	SubnetAddress	*string	Optional	Receiving subnet address (for Ethernet connections. Not required for PCIe connections)
	GatewayAddress	*string	Optional	Receiving gateway address (for Ethernet connections. Not required for PCIe connections)
	DMAChannelID	*int32	Optional	The ID of the receiving DMA channel (required for PCIe connections). Not required for Ethernet
	LLDMAConnectorID	*int32	Optional	The connector ID of the receiving LLDMA for DMA transfer (required for PCIe connection). Not required for Ethernet connections)
	Tx	RxTxData	Optional	Sender network information to be assigned to FPGAFunction
	Protocol	string	Required	sender's communication protocol
	IPAddress	*string	Optional	Sender IP address (for Ethernet connections. Not required for PCIe connections)
	Port	*int32	Optional	Transmitting port number (for Ethernet connection. Not required for PCIe connections)
	SubnetAddress	*string	Optional	The sender's subnet address (for Ethernet connections. Not required for PCIe connections)
	GatewayAddress	*string	Optional	The gateway address of the sender (for Ethernet connections. Not required for PCIe connections)
	DMAChannelID	*int32	Optional	The ID of the sending DMA channel (required for PCIe connections). Not required for Ethernet connections)
	LLDMAConnectorID	*int32	Optional	The connector ID of the sending LLDMA for DMA transfer (required for PCIe connection). Not required for Ethernet connections)
Status	PreviousFunctions	map[string]FromToWBFunc	Optional	FunctionCR information of the previous Function. If there is no previous Function (= wb-start-of-chain), this parameter is not set. Each key of this map is string of the input port number (interface identification number) same as InputInterface. (Equivalent to WBFunction.spec.PreviousWBFunctions. WBFunction is the parent CR of this Function.)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Output port number of the other Function connected to the input port number (key value) of the current Function
	NextFunctions	map[string]FromToWBFunc	Optional	FunctionCR information of the next Function. If there is no next Function (= wb-end-of-chain), it is not set. Each key of this map is string of the input port number (interface identification number) same as InputInterface. (Equivalent to WBFunction.spec.PreviousWBFunctions. WBFunction is the parent CR of this Function.)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Input port number of the other Function that is connected to the output port number (key value) of the current Function
	StartTime	metav1.Time	Requried	Creation time
	Status	string	Requried	The state of FPGAFunction. Have the following two values ・Running: successful creation ・Pending: Creating *I don't use Pending at the moment, I set it to Running after FPGAFunction creation process.
	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction.status.DataFlowRef)
	FunctionName	string	Required	Function name (Equivalent to WBFunction .status.FunctionName which will be the parent CR)
	FunctionIndex	int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.status.RegionName)
	ParentBitstreamName	string	Required	Parent bitstream name (Filename of the .mcs file) written to FPGAFunction destination
	ChildBitstreamName	string	Required	Child bit stream name (Filename of the.bit file) written in the deployment destination of the FPGAFunction
	SharedMemory	*SharedMemorySpec	Optional	Shared memory information set for FPGAFunction (only when PCIe is connected)
	FilePrefix	string	Required	Information to identify the PCIe connection on the dpdk side
	CommandQueueID	string	Required	Identity of the CommandQueue used for data transfer
	sharedMemoryMiB	int32	Required	Required size of shared memory used for data transfer on PCIe connections [MegaByte] (not used in the
	FunctionKernelID	int32	Required	ID of the Function module that has the FunctionChannelID (FuncCHId) to be assigned to this FPGAFunction
	FunctionChannelID	int32	Required	ID of the FuncCH assigned to FPGAFunction
	PtuKernelID	int32	Required	ID of the PTU module to be used by FPGAFunction
	FrameworkKernelID	int32	Required	Chain Control Module ID(Modules.Chain.ID equivalent of ChildBs)
	Rx	RxTxData	Required	Receiving network information assigned to FPGAFunction
	Protocol	string	Required	Receiving communication protocol
	IPAddress	*string	Optional	Receiving IP address (for Ethernet connections. Not required for PCIe connections)
	Port	*int32	Optional	Receiving port number (for Ethernet connection. Not required for PCIe connections)
	SubnetAddress	*string	Optional	Receiving subnet address (for Ethernet connections. Not required for PCIe connections)
	GatewayAddress	*string	Optional	Receiving gateway address (for Ethernet connections. Not required for PCIe connections)
	DMAChannelID	*int32	Optional	The ID of the receiving DMA channel (required for PCIe connections). Not required for Ethernet
	LLDMAConnectorID	*int32	Optional	The connector ID of the receiving LLDMA for DMA transfer (required for PCIe connection). Not required for Ethernet connections)
	Tx	RxTxData	Required	Sender's network information assigned to FPGAFunction
	Protocol	string	Required	sender's communication protocol
	IPAddress	*string	Optional	Sender IP address (for Ethernet connections. Not required for PCIe connections)
	Port	*int32	Optional	Transmitting port number (for Ethernet connection. Not required for PCIe connections)
	SubnetAddress	*string	Optional	The sender's subnet address (for Ethernet connections. Not required for PCIe connections)
	GatewayAddress	*string	Optional	The gateway address of the sender (for Ethernet connections. Not required for PCIe connections)
	DMAChannelID	*int32	Optional	The ID of the sending DMA channel (required for PCIe connections). Not required for Ethernet connections)
	LLDMAConnectorID	*int32	Optional	The connector ID of the sending LLDMA for DMA transfer (required for PCIe connection). Not required for Ethernet connections)
	AcceleratorStatuses	[]AccStatusesByDevice	Optional	State of the device that deployed this FPGAFunction. Each Function (or FuncCHId in the case of FPGAFunc).
	PartitionName	*string	Optional	Information identifying the Function that is the target of status (FuncCHId in the case of FPGAFunc)
	Statuses	[]AccStatuses	Optional	Records status for each Accelerator assigned to FPGAFunction
	AcceleratorID	*string	Optional	Device UUID
	Status	*string	Optional	Device status. Three types (deployed | deploying | error) are assumed.

■ChildBs

custom resource with information about the child bitstream (child bs) writing to the FPGA using a .bit file
It has all the information written in this child bs (including the parameters set for each module).
It also has all the resources in the FPGA after writing child bs.
Also has resource capacity information after child bs is written
Automatically generated by FPGAFunction controller when a child bs write is required upon receipt of a FPGAFunction deployment request

	Name	Req/Opt	Type	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
	OwnerReferences	-	-	Reference information for the k8s resource associated with the ChildBs. This contains information about the FPGA resource that is the parent CR.
	APIVersion	-	-	API version of the FPGA resource that is the parent CR
	Kind	-	-	"FPGA" fixed because it is the Kind of FPGA resource that is the parent CR
	Name	-	-	Name of the FPGA resource that is the parent CR
	Uid	-	-	UUID of the FPGA resource that is the parent CR
Spec	Regions	[]ChildBsRegion	Required	A list of each region present on the ChildBs. It has information of each region as a list. (Currently, each lane corresponds to each region.)
	Modules	*ChildBsModule	Optional	A group of FPGA modules constituting the region. Contains information about the following modules, if any.
	Ptu	*ChildBsPtu	Optional	Information of the PTU module responsible for Ethernet communication to be written for the region
	Cids	*string	Optional	Connection ID available to PTU modules under this region ("random" indicates that the value is determined during operation)
	ID	*int32	Optional	ID of the PTU module under this region (Currently the same as the region ID (= Lane number). 1 PTU module per lane)
	Parameters	*map[string]intstr.IntOrString	Optional	Parameters to set for the PTU module. A map where key is the parameter name and value is the value.
	LLDMA	*ChildBsLLDMA	Optional	Information of the LLDMA module responsible for DMA communication to be written for the region
	Cids	*string	Optional	Connection ID that can be taken by the LLDMA module under this region
	ID	*int32	Optional	ID of the LLDMA module under the region (Currently the same as the region ID (= Lane number). One LLDMA module per lane is assumed)
	Chain	*ChildBsChain	Optional	Information of the Chain module, which is responsible for associating I/O and Function modules, to be written for this region
	ID	*int32	Optional	ID of the Chain module under the region (Currently the same as the region ID (= Lane number). One Chain module per lane is assumed)
	Identifier	*string	Optional	Module identifier of the Chain module (with an identifier determined for each module type)
	Type	*string	Optional	String representing the module type of the Chain module
	Version	*string	Optional	The version of the Chain module (ChildBs implementation time).
	Directtrans	*ChildBsDirecttrans	Optional	Information of the Directtrans module responsible for direct transfer written under the region
	ID	*int32	Optional	ID of the Directtrans module under this region (Currently the same as the region ID (= Lane number). One Directtrans module per lane is assumed)
	Identifier	*string	Optional	Module identifier of the Directtrans module
	Type	*string	Optional	A string that indicates the module type of the Directtrans module.
	Version	*string	Optional	Directtrans module version (ChildBs implementation time)
	Conversion	*ChildBsConversion	Optional	Information of the Conversion module responsible for the conversion process to be written for the region
	ID	*int32	Optional	ID of the Conversion module under this region (Currently the same as the region ID (= Lane number). One Conversion module per lane is assumed)
	Module	*[]ConversionModule	Optional	List of modules that the Conversion module writing for this region can take
	Identifier	*string	Optional	Module identifier of the Conversion module
	Type	*string	Optional	String representing the module type of the Conversion module
	Version	*string	Optional	Conversion Module version (ChildBs implementation time)
	Functions	*[]ChildBsFunctions	Optional	A list containing information about each Function module responsible for executing the processing module to be written for this region.
	ID	*int32	Optional	ID of the Function module under this region (For filter/resize FPGAs, this is the same as the region ID (= Lane number). One Function module per lane is assumed)
	FunctionName	*string	Optional	Function name under this region
	Module	*[]FunctionsModule	Optional	List of Function modules to be written for this region
	FunctionChannelIDs	*string	Optional	Function channel ID that the function module can take
	Identifier	*string	Optional	Module identifier of the Function module for the processing module (The module identifier of a Function module differs depending on the type of processing module.)
	Type	*string	Optional	Character string indicating the module type of the Function module for the processing module
	Version	*string	Optional	Function module version for the processing module (ChildBs implementation time)
	Parameters	*map[string]intstr.IntOrString	Optional	Parameters set for the Function module. A map where key is the parameter name and value is the value.
	IntraResourceMgmtMap	*map[string] FunctionsIntraResourceMgmtMap	Optional	Resource information in the FPGA managed by the Function module where key is FunctionChannelID (FuncCHID) and value is a map of the resources in the FPGA to be prepared for that FuncCHID. (For each FuncCHID, the resource in the FPGA to be paid to the FPGAFunc is determined in combination with its Id.)
	Available	*bool	Optional	Whether the FuncCHID entry is available. (false if used. true if unused (= usable))
	FunctionCRName	*string	Optional	Information of CR, FPGAFunction to which the FuncCHID is assigned (Initially, nil. If FuncCHID is assigned to FPGAFunction, provide information about that FPGAFunction.)
	Rx	*RxTxSpec	Optional	Receiving side network information to be allocated to FPGAFunction in set with the FuncCHID
	Protocol	*map[string]ChildBsDetails	Optional	Details of the network information to give as the receiver. A map where key is the protocol name and value is the detail information.
	Port	*int32	Optional	Port number to be given as the receiver (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not required for DMA protocol)
	DMAChannelID	*int32	Optional	ID of the DMA channel to be given as the receiver (For PCIe connections (protocol is DMA). Ethernet connection (not required for TCP/RTP protocol)
	LLDMAConnectorID	*int32	Optional	ID of the connector for DMA transfer on the LLDMA side to be given as the receiver (For PCIe connections (protocol is DMA). Ethernet connection (not required for TCP/RTP protocol)
	Tx	*RxTxSpec	Optional	Transmitter's network information to be allocated to the FPGAFunction in set with the FuncCHID
	Protocol	*map[string]ChildBsDetails	Optional	Details of network information to give as sender. A map where key is the protocol name and value is the detail information.
	Port	*int32	Optional	Port number to be given as the sender (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not required for DMA protocol)
	DMAChannelID	*int32	Optional	ID of the DMA channel to be given as the sender (For PCIe connections (protocol is DMA). Ethernet connection (not required for TCP/RTP protocol)
	LLDMAConnectorID	*int32	Optional	Connector Id for DMA transfer on LLDMA side to be given as sender (For PCIe connections (protocol is DMA). Ethernet connection (not required for TCP/RTP protocol)
	DeploySpec	FunctionsDeploySpec	Required	Resource capacity information of the processing module written in the function module
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployed Function of the processing module. It depends on the circuit implementation.
	MaxDataFlows	*int32	Optional	Maximum number of DFs installed in the processing module (number of WBFunctions). It depends on the number of channels of the circuit, etc.
	MaxFunctions	*int32	Optional	Maximum number of processing modules (Functions) that can be written in this region (Number of Functions = Number of circuits)
	MaxCapacity	*int32	Optional	Maximum processing power (fps) for the entire region
	Name	*string	Optional	Domain name of the domain (currently Lane number)
	ChildBitstreamID	*string	Optional	ID of the Bitstream from which the ChildBs resource is based (Bitstream ID to be written in .bit file)
	ChildBitstreamFile	*string	Optional	Config name written to the ChildBs resource
	Regions	[]ChildBsRegion	Required	A list of each region present on the ChildBs. Have a list of information for each region (Currently, each lane corresponds to each region.)
	Modules	*ChildBsModule	Optional	A group of FPGA modules constituting the region. Contains information about the following modules, if any.
	Ptu	*ChildBsPtu	Optional	Information of the PTU module responsible for Ethernet communication written in the region
	Cids	*string	Optional	Connection ID that can be taken by the PTU module under this region ("random" indicates that the value is determined at runtime)
	ID	*int32	Optional	ID of the PTU module under this region (Currently the same as the region ID (= Lane number). 1 PTU module per lane)
	Parameters	*map[string]intstr.IntOrString	Optional	Parameters to set for the PTU module. A map where key is the parameter name and value is the value.
	LLDMA	*ChildBsLLDMA	Optional	Information of the LLDMA module responsible for DMA communication written in the region
	Cids	*string	Optional	Connection ID that can be taken by the LLDMA module under this region
	ID	*int32	Optional	ID of the LLDMA module under the region (Currently the same as the region ID (= Lane number). One LLDMA module per lane is assumed)
	Chain	*ChildBsChain	Optional	Information of the Chain module, which is responsible for associating I/O and Function modules, written in this region

Status

ID	*int32	Optional	ID of the Chain module under the region (Currently the same as the region ID (= Lane number). One Chain module per lane is assumed)
Identifier	*string	Optional	Module identifier of the Chain module (with an identifier determined for each module type)
Type	*string	Optional	String representing the module type of the Chain module
Version	*string	Optional	The version of the Chain module (ChildBs implementation time).
Directtrans	*ChildBsDirecttrans	Optional	Information of the Directtrans module that is responsible for direct transfer under the region written in the
ID	*int32	Optional	ID of the Directtrans module under this region (Currently the same as the region ID (= Lane number). One Directtrans module per lane is assumed)
Identifier	*string	Optional	Module identifier of the Directtrans module
Type	*string	Optional	A string that indicates the module type of the Directtrans module.
Version	*string	Optional	Directtrans module version (ChildBs implementation time)
Conversion	*ChildBsConversion	Optional	Information of the Conversion module responsible for conversion processing written in the region
ID	*int32	Optional	ID of the Conversion module under this region (Currently the same as the region ID (= Lane number). One Conversion module per lane is assumed)
Module	*[]ConversionModule	Optional	List of modules that Conversion modules under this region can take
Identifier	*string	Optional	Module identifier of the Conversion module
Type	*string	Optional	String representing the module type of the Conversion module
Version	*string	Optional	Conversion Module version (ChildBs implementation time)
Functions	*[]ChildBsFunctions	Optional	A list containing information about each Function module that is responsible for executing the processing module written in the region.
ID	*int32	Optional	ID of the Function module under the region (same as the region ID (= Lane number) for filter/resize FPGAs). One Function module per lane is assumed)
FunctionName	*string	Optional	Function name under this region
Module	*[]FunctionsModule	Optional	List of Function modules written in this region
FunctionChannelIDs	*string	Optional	Function channel ID that the function module can take
Identifier	*string	Optional	Module identifier of the Function module for the processing module (The module identifier of a Function module differs depending on the type of processing module.)
Type	*string	Optional	Character string indicating the module type of the Function module for the processing module
Version	*string	Optional	Function module version for the processing module (ChildBs implementation time)
Parameters	*map[string]intstr.IntOrString	Optional	Parameters set for the Function module. A map where key is the parameter name and value is the value.
IntraResourceMgmtMap	*map[string] FunctionsIntraResourceMgmtMap	Optional	Resource information in the FPGA managed by the function module. where key is FunctionChannelID (FuncCHID) and value is a map of the resources in the FPGA to be prepared for that FuncCHID. (For each FuncCHID, the resource in the FPGA to be paid to the FPGAFunc is determined in combination with its Id.)
Available	*bool	Optional	Whether the FuncCHID entry is available. (false if used. true if unused (= usable))
FunctionCRName	*string	Optional	Information of CR of FPGAFunction to which the entry was given (Initially, nil. If a FuncCHID is assigned to an FPGAFunc, enter the information of that FPGAFunc.)
Rx	*RxTxSpec	Optional	Receiving side network information to be allocated to FPGAFunction in set with the FuncCHID
Protocol	*map[string]ChildBsDetails	Optional	Details of the network information to give as the receiver. A map where key is the protocol name and value is the detail information.
Port	*int32	Optional	Port number to be given as the receiver (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not used with DMA protocol)
DMAChannelID	*int32	Optional	ID of the DMA channel to be given as the receiver (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
LLDMAConnectorID	*int32	Optional	Connector Id for DMA transfer on LLDMA side to be given as receiver (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
Tx	*RxTxSpec	Optional	Transmitter's network information to be allocated to the FPGAFunction in set with the FuncCHID
Protocol	*map[string]ChildBsDetails	Optional	Details of network information to give as sender. A map where key is the protocol name and value is the detail information.
Port	*int32	Optional	Port number to be given as the sender (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not used with DMA protocol)
DMAChannelID	*int32	Optional	ID of the DMA channel to be given as the sender (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
LLDMAConnectorID	*int32	Optional	ID of the connector for DMA transfer on the LLDMA side to be given as the sender (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
DeploySpec	FunctionsDeploySpec	Required	Resource capacity information of the processing module written in the function module
MaxCapacity	*int32	Optional	The maximum processing power (fps) of the deployed Function of the processing module. It depends on the circuit implementation.
MaxDataFlows	*int32	Optional	Maximum number of DFs installed in the processing module (number of WBFunctions). It depends on the number of channels of the circuit, etc.
MaxFunctions	*int32	Optional	Maximum number of processing modules (Functions) that can be written in this region (Number of Functions = Number of circuits)
MaxCapacity	*int32	Optional	Maximum processing power (fps) for the entire region
Name	*string	Optional	Domain name of the domain (currently Lane number)
Status	ChildBitstreamStatus	Required	The state of the ChildBs resource. Have the following four values ・"NotReady" — Before preparing (before writing child bs) ・"Preparing" — Preparing (child bs writing) ・"Ready" — Enabled (after child bs write is complete) ・"Error": Failed to prepare (if child bs write failed)
State	ChildBitstreamState	Required	The write status of child bs corresponding to the ChildBs. Has four values, no more than eleven values ・"StoppingModule": A module other than ptu is stopped. ・"NotStopNetworkModule": The ptu module has not been stopped. ・"StoppingNetworkModule": Stopping ptu module ・"NotWriteBitstreamFile": Bitstream file not written ・"Reconfiguring": FPGA reset complete ・"WritingBitstreamFile": Writing bitstream file ・"ConfiguringParameters": Setting parameters ・"NoConfigureNetwork": The network information has not been set. ・"ConfiguringNetwork": Configuring network information ・"Ready" — child bs write complete ・"Error": child bs write failure
ChildBitstreamID	*string	Optional	ID of the Bitstream from which the ChildBs resource is based
ChildBitstreamFile	*string	Optional	Config name written to the ChildBs resource

■FPGA

custom resource with information on FPGA devices

Automatically generated when infrastructure information collection management is executed and updated when child bs is written

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
spec	ChildBitstreamID	*string	Optional	id of the child bitstream written to this FPGA device in the.bit file
	DeviceIndex	int32	Required	Serial number on the installed server (as set in the FPGA driver library)
	DeviceFilePath	string	Required	DeviceFilePath for this FPGA device on the installed server
	DeviceUUID	string	Required	FPGA-ID for this FPGA device (as set in the FPGA driver library)
	NodeName	string	Required	Hostname of the server that contains this FPGA device
	ParentBitstreamID	string	Required	ID of the parent bitstream written to this FPGA device in the .mcs file
	PCIDomain	int32	Required	Domain number of the PCI in which this FPGA device is inserted
	PCIBus	int32	Required	Bus number of the PCI that this FPGA device is plugged into
	PCIDevice	int32	Required	Device number of the PCI in which this FPGA device is inserted
	PCIFunction	int32	Required	Function number of the PCI in which this FPGA device is inserted
	Vendor	string	Required	Vendor information for this FPGA device
status	ChildBitstreamID	*string	Optional	id of the child bitstream written to this FPGA device in the.bit file
	ChildBitstreamCRName	*string	Optional	Information about the ChildBs resource that will be the child CR of this FPGA device
	DeviceFilePath	string	Required	Serial number on the installed server (as set in the FPGA driver library)
	DeviceIndex	int32	Required	DeviceFilePath for this FPGA device on the installed server
	DeviceUUID	string	Required	FPGA-ID for this FPGA device (as set in the FPGA driver library)
	NodeName	string	Required	Hostname of the server that contains this FPGA device
	ParentBitstreamID	string	Required	ID of the parent bitstream written to this FPGA device in the .mcs file
	PCIDomain	int32	Required	Domain number of the PCI in which this FPGA device is inserted
	PCIBus	int32	Required	Bus number of the PCI that this FPGA device is plugged into
	PCIDevice	int32	Required	Device number of the PCI in which this FPGA device is inserted
	PCIFunction	int32	Required	Function number of the PCI in which this FPGA device is inserted
	Status	FPGAStatus	Required	The state of this FPGA device. Have the following four values ・"NotReady" — Before preparing (before writing child bs) ・"Preparing" — Preparing (child bs writing) ・"Ready" — Enabled (after child bs write is complete) ・"Error": Failed to prepare (if child bs write failed)
	Vendor	string	Required	Vendor information for this FPGA device

■FPGAReconfiguration

New from v1.1.0

custom resource with information passed from the FPGAReconfigurationTool to FPGAFunction controller when doing FPGA Bitstream write, FPGA reset, or Child Bitstream reset with the FPGAReConfiguration
Automatically generated by the FPGAReconfigurationTool when the tool is executed, and deleted by the tool after manual writing, FPGA reset, and child bs reset.

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
Spec	NodeName	string	Required	Hostname of target server that contains target FPGA device
	DeviceFilePath	string	Required	DeviceFilePath for target FPGA device on target server
	FPGAResetFlag	*bool	Optional	If true, FPGA reset request occurs. If both FPGAResetFlag and ChildBsResetFlag are false, a manual write request is issued.
	ChildBsResetFlag	*bool	Optional	If true, a Child Bitstream reset request occurs. If both FPGAResetFlag and ChildBsResetFlag are false, a manual write request is issued.
	ConfigNames	[]FPGAConfigNames	Optional	An array of information about the configuration for FPGAFunc needed for manual write requests. - FPGAReconfigurationTool automatically sets only LaneIndex: 0 when FPGA reset is requested - Child Bitstream reset request does not require this information - During FPGA Bitstream write request, this information is set by setting an array that matches Lane configuration of the FPGA in the argument when running FPGAReconfigurationTool
	LaneIndex	int32	Required	Information identifying the lane of interest
	ConfigName	string	Required	ConfigMap name of the FPGAFunc configuration (fpgafunc-config-xxx) required for writing
Status	Status	string	Required	Results of FPGA Bitstream write, FPGA reset, and child bs reset processing to this FPGA device.

■CPUFunction

custom resource with information about the Function to deploy to the CPU
Sample processing modules include decoding, filter/resize, copy branch, and gleu (dma➡tcp).
Converted from WBFunction and auto-generated

	Nar	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
Spec	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction.spec.DataFlowRef)
	FunctionName	string	Required	Name of the Function to execute (Equivalent to the parent CR WBFunction.spec.FunctionName)
	NodeName	string	Required	Destination node name (Equivalent to the parent CR WBFunction.spec.NodeName)
	DeviceType	string	Required	Destination Device Type (Equivalent to the parent CR WBFunction.spec.DeviceType)
	AcceleratorIDs	[]AcclDInfo	Required	The identity of the destination device (considering the possibility of assigning more than one device to a GPUFunction in the future)
	PartitionName	*string	Optional	Information identifying the Function to which the accelerator is to be assigned (container name in the case of CPUFunction)
	ID	string	Required	Accelerator identification to assign to the Function (In the case of CPUFunction, "NodeName"+"-"+ "UUID of the CPU" * Specify the UUID of the CPU by yourself)
	RegionName	string	Required	Distinguished name of the deployment region to which you want to deploy
	FunctionIndex	*int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.spec.RegionName) (in the case of a CPUFunction, the Id of the deployed Pod) If this parameter is not present, it means that a new deployment is requested. If this parameter is present, it means that the dataflow should be stored in the circuit or pod corresponding to the already deployed Functions with that FunctionIndex.
	Envs	[]EnvsInfo	Optional	For setting parameters for the processing module. copied to containers.env on Pod (list for each container)
	PartitionName	string	Required	Information identifying the function to pass this argument to (container name for CPUFunction)
	EachEnv	[]EnvsData	Required	List with information for each environment variable
	EnvKey	string	Required	Key values of environment variables
	EnvValue	string	Required	Value value of the environment variable
	RequestMemorySize	*int32	Optional	Minimum memory size required by the container to boot for this CPUFunction [Gib] (Not currently used. Config for CPUFunc (ConfigMap of cpufunc-config-xxx)
	SharedMemory	*SharedMemorySpec	Optional	Configuration Information Required to Perform a PCIe Connection over Shared Memory
	FilePrefix	string	Required	Information to identify the PCIe connection on the dpdk side
	CommandQueueID	string	Required	Identity of the CommandQueue used for data transfer
	SharedMemoryMiB	int32	Required	Required size of the shared memory used for data transfer on the PCIe connection [MegaByte] (Not currently used. The value is fixed inside the processing module.)
	Protocol	*string	Optional	Receiving communication protocol (required if data is received (source is present))
	ConfigName	string	Required	Config name required for Deploy (name of ConfigMap in gpufunc-config-xxx) (Equivalent to the parent CR WBFunction.spec.ConfigName)
	PreviousFunctions	map[string]FromToWBFunction	Optional	FunctionCR information for each resource system in the previous section. If there is no previous Function (= wb-start-of-chain), this parameter is not set. key is the same as InputInterface and the input port number (interface identification number) is expressed as a character string. (Equivalent to parent WBFunction.spec.PreviousWBFunctions)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Output port number of the other Function connected to the input port number (key value) of the current Function
	NextFunctions	map[string]FromToWBFunction	Optional	FunctionCR information of various resource systems in the second part. If there is no next Function (= wb-end-of-chain), it is not set. key is the same as OutputInterface, and the output port number (interface identification number) is expressed as a character string. (Equivalent to parent WBFunction.spec.NextWBFunctions)
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the corresponding resource function
	Port	int32	Required	Input port number of the other Function connected to the output port number (key value) of the current Function
	Params	map[string]intstr.IntOrString	Optional	Integer/String parameters (Equivalent to the parent CR WBFunction.spec.Params)
Status	DataFlowRef	WBNamespaecedName	Required	Identify original Dataflow (Equivalent to the parent CR WBFunction.status.DataFlowRef)
	FunctionName	string	Required	Function name (Equivalent to the parent CR WBFunction.status.FunctionName)
	ImageURI	string	Required	Container image name of the container to be started in this CPUFunction
	SharedMemory	*SharedMemorySpec	Optional	Shared memory information set for CPUFunction (only when PCIe is connected)
	FilePrefix	string	Required	Identity of the CommandQueue used for data transfer
	CommandQueueID	string	Required	Information to identify the PCIe connection on the dpdk side
	sharedMemoryMiB	int32	Required	Required size of the shared memory used for data transfer on the PCIe connection [MegaByte] (Currently unused.). The value is fixed inside the processing module.)
	RxProtocol	*string	Optional	Receiving communication protocol (listed if data is received (source is present))
	TxProtocol	*string	Optional	Sender's communication protocol (listed if data is sent (destination is present))
	ConfigName	string	Required	Config name required for deploy (name of ConfigMap in cpufunc-config-xxx) (Equivalent to the parent CR WBFunction.status.ConfigName)
	VirtualNetworkDeviceDriverType	string	Optional	CNI Plug-ins for 2nd NICs on Pod
	AdditionalNetwork	*bool	Optional	Whether to create a 2nd NIC on Pod
	FunctionIndex	int32 *int32	Optional	Deployed functions on the destination (Equivalent to the parent CR WBFunction.status.RegionName)
	PodName	*string	Optional	CR name of Pod you created
	StartTime	metav1.Time	Requried	Creation time
	Status	string	Required	The state of CPUFunction. Have the following two values ・Running: successful creation ・Pending: Creating *I don't use Pending at the moment, I set it to Running after Pod creation is complete.
	IPAddress	*string	Optional	IP address (currently unused)
	AcceleratorStatuses	[]AccStatusesByContainer	Optional	State of the device that deployed this CPUFunction. It is recorded for each Function (for each container in the case of GPUFunc).
	PartitionName	*string	Optional	Information identifying the Function for which status is set
	Statuses	[]AccStatuses	Optional	Records status for each Accelerator assigned to CPUFunction
	AcceleratorID	*string	Optional	Device UUID
	Status	*string	Optional	Device status. Three types (deployed | deploying | error) are assumed.

■DeviceInfo

custom resource with information exchanged between WBFunction controller (WF controller) and DeviceInfo controller (DM controller)

WBFunction CRC deletes it when WBFunction CRC finishes processing (various CR creation processing), so it does not exist when DF deployment is completed.

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Set arbitrarily by the user.
	Namespace	-	-	Set arbitrarily by the user.
Spec	Request	WBFuncRequest	Required	Request to reserve or free up deployment space for a device Contains Spec information for WBFunctionCR.
	RequestType	string	Required	Type of processing request to DeviceManager.
	DeviceType	string	Required	Accelerator type.
	DeviceIndex	int32	Required	Device number.
	RegionName	string	Required	A unique name given to a partitioned region on a physical device.
	NodeName	string	Required	Host name.
	FunctionIndex	*int32	Optional	Serial number of the Function to be deployed or deployed.
	FunctionName	string	Required	Name of the Function to be deployed or deployed.
	MaxDataFlows	*int32	Optional	The maximum number of installed DF (WBFunction) for a scheduled or deployed function. It depends on the number of channels of the circuit, etc.
	MaxCapacity	*int32	Optional	The maximum processing power (fps) of the scheduled or deployed Function.
	Capacity	*int32	Optional	The load of a Function that will or has been deployed.
Status	Response	WBFuncResponse	Optional	Result of processing a request to allocate or release a device deployment region.
	Status	string	Required	The processing result for the request.
	FunctionIndex	*int32	Optional	Serial number of the Function that reserved or released the deployment region.
	DeviceUUID	string	Optional	The UUID of the device that reserved the deployment space. (Stores information only when a request is made to allocate a deployment region)
	DeviceFilePath	string	Optional	Device file path with allocated deployment space (Stores information only when the device is an FPGA and the allocation of the deployment region is

■EthernetConnection

custom resource with information about Ethernet connections

Currently, only FPGA (decode) →FPGA (filter/resize) communication is applicable.

Converted from WBConnection and auto-generated

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
Spec	DataFlowRef	WBNamespaecedName	Required	Identification of DataFlow from which EthernetConnection is based
	From	EthernetFunctionSpec	Requried	Function CR of the sender of EthernetConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the sending Function.
	To	EthernetFunctionSpec	Requried	Function CR on the destination side of EthernetConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the destination Function.
Status	DataFlowRef	WBNamespaecedName	Required	Identification of DataFlow from which EthernetConnection is based
	From	EthernetFunctionStatus	Requried	Function CR of the sender of EthernetConnection
	WBFunctionRef	WBNamespacedName	Requried	
	Status	string	Requried	The deployment state on the SrcFunc side. Have the following three values ・OK : Deployed ・INIT: ・NG: Not deployed *INIT is no longer used.
	To	EthernetFunctionStatus	Requried	Function CR on the destination side of EthernetConnection
	WBFunctionRef	WBNamespacedName	Requried	
	Status	string	Requried	The deployment state on the DstFunc side. Values and usage status are the same as From.Status above.
	StartTime	metav1.Time	Required	creation time of EthernetConnection
	Status	string	Requried	The state of EthernetConnection. Have the following two values ・Running: successful creation ・Pending: Creating *Currently I don't use Pending, I just run it after EthernetConnection creation process is complete.

■PCIeConnection

custom resource with information about PCIe connections over shared memory

Currently, it covers FPGA (filter/resize) →GPU (advanced or lightweight inference) communication, CPU (decode) →FPGA (filter/resize), and FPGA (decode) →FPGA (filter/resize).

Converted from WBConnection and auto-generated

	Name	Type	Req/Opt	Description
metadata	Name	-	-	Be set by the user
	Namespace	-	-	Be set by the user
	DataFlowRef	WBNamespaecedName	Required	Identification of DataFlow from which PCIeConnection is based
	From	PCIeFunctionSpec	Requried	Function CR of the sender of PCIeConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the sending Function.
	To	PCIeFunctionSpec	Requried	Function CR on the destination side of PCIeConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the destination Function.
Status	DataFlowRef	WBNamespaecedName	Required	Identification of DataFlow from which PCIeConnection is based
	From	PCIeFunctionStatus	Requried	Function CR of the sender of PCIeConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the sending Function.
	Status	string	Requried	The deployment state on the SrcFunc side. Have the following six values ・OK : Deployed ・INIT: ・NG: Not deployed ・PODDELETING: Deleting POD ・PODDELETED: POD deletion complete ・STOPPED: Deleted *INIT is no longer used.
	To	PCIeFunctionStatus	Requried	Function CR on the destination side of PCIeConnection
	WBFunctionRef	WBNamespacedName	Required	Resource name and namespace of the destination Function.
	Status	string	Requried	The deployment state on the DstFunc side. Values and usage status are the same as From.Status above.
	SharedMemory	SharedMemoryStatus	Optional	Shared memory allocation status
	Status	string	Optional	State of shared memory used by PCIe over shared memory (Allocated | Allocating | Error) (currently unused)
	StarttTime	metav1.Time	Required	creation time of PCIeConnection
	Status	string	Requried	The state of PCIeConnection. Have the following four values ・Running: successful creation ・Pending: Creating ・Terminating: Deleting ・Released: Deleted *Currently I don't use Pending, I just run it after EthernetConnection creation process is complete.

■Compatible information for FunctionKind identification

Information to identify which Function CR (GPUFunction/FPGAFunction) to convert from WBFunction.

Since this information is environment-independent, the following sample data can basically be used as it is.

However, you need to create the file and apply it as ConfigMap.

Name		Type	Req/Opt	Description
FunctionCRKinds		[]FunctionKindMap		
	DeviceType	string	Required	Type of the destination Device. WBFunction .spec.DeviceType itself.
	FunctionCRKind	string	Required	Function CR type. Currently, the following three types ・GPUFunction ・FPGAFunction ・CPUFunction

■Correspondence information for ConnectionKind identification

Information to identify which Connection CR (EthernetConnection/PCIeConnection) to convert from WBConnection.
Since this information is environment-independent, the following sample data can basically be used as it is.
However, you need to create the file and apply it as ConfigMap.

Name		Type	Req/Opt	Description
ConnectionCRKinds		[]ConnectionKindMap	Required	
	ConnectionMethod	string	Required	It represents the From and To connection method and corresponds to WBConnection.spec.ConnectionMethod. Currently, there are two types "host-100gether" (Ethernet connection) "host-mem" (PCIe connection)
	ConnectionCRKind	string	Required	Type of the Connection CR corresponding to each type of ConnectionMethod. Currently, there are two types ・EthernetConnecton: "hodt-100gether" support ・PCIeConnection: Support for "host-mem"

■infrastructure configuration information

Information to define the hard configuration of each node. Define deployment region information for each device (GPU, FPGA).
It currently covers information about installed devices (GPU, FPGA, CPU, memory).
Automatically generated by the infrastructure information collection management department

	Name	Type	Req/Opt	Description
Spec	Devices	[]DeviceInfo	Required	List of device information for devices installed on the node (Define the number of devices (Currently FPGA, GPU, CPU and memory) installed in the node)
	DeviceFilePath	*string	Optional	Device File Path (Information to identify which device is physical. For the time being, only FPGA can be used. For GPUs, it is not necessary because the device file path is not used to physically identify the GPU.)
	NodeName	string	Required	Host name
	DeviceUUID	*string	Optional	Globally unique identifier of the device
	DeviceType	string	Required	Accelerator type
	DeviceIndex	int32	Required	Serial number of the device

■Deployment Region Information

Information to define the deployable space provided on each node. Define deployment region information for each device (GPU, FPGA).

It currently covers information about installed devices (GPU, FPGA, CPU).

Automatically generated by the infrastructure information collection management department

Name	Type	Req/Opt	Description
Devices	[]DeviceRegionInfo	Required	List of region information created on the devices installed on the node (Define by the number of devices installed in the node (currently FPGAs, GPUs, and CPUs))
NodeName	string	Required	Host name
DeviceFilePath	*string	Optional	Device File Path (information to identify which FPGA device is physically)
DeviceUUID	*string	Optional	Globally unique identifier of the device (information to determine which GPU device is physical)
FunctionTargets	[]regionIndevice	Required	List with each region information as an element
RegionType	string	Required	region type of the region
RegionName	string	Required	identification of the region
MaxFunctions	*int32	Required	Maximum number of processing modules (Functions) that can be written in this region (Number of Functions = Number of
MaxCapacity	*int32	Required	Maximum processing power (fps) for the entire region
Functions	[]simplefunctioninfrastruct	Optional※	Information about functions already deployed in the region
FunctionIndex	*int32	Optional※	Serial number of the function
PartitionName	string	Required	Physical information that identifies where the function is actually deployed on the infrastructure
FunctionName	string	Required	Name of the function
MaxDataFlows	int32	Required	Maximum number of installed DF for the function (number of WBFunction)
MaxCapacity	int32	Required	Maximum processing power of the function (fps)

*Required if the device is an FPGA

■fixed region information

Information used to identify the RegionType of each region in Lane fixed method

Since this information is environment-dependent, a file must be created for each worker node in advance according to the environment.

The following is an image of how to use it.

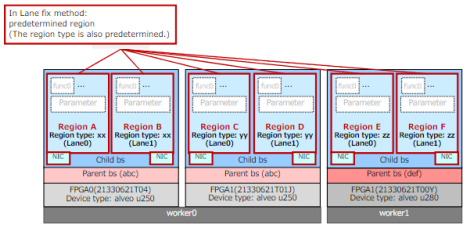
Prepared by someone (such as an infrastructure service administrator)

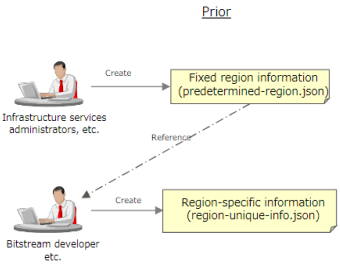
A person (such as a bitstream developer) creates an Region-specific information from this file.

Name		Type	Req/Opt	Description
PreDeterminedRegionInfos		[]PreDeterminedRegionInfo	Required	List of fixed regions
	NodeName	string	Required	Host name of the server on which the target region resides
	DeviceUUID	int32	Required	UUID of the device on which the region of interest resides
	SubDeviceSpecRef	string	Required	Information identifying the target region (lane number for FPGAs, device type (equivalent to DeviceType) for GPUs, "cpu" for CPUs)
	RegionType	string	Required	Region type of the target region (FPGA: "Device type" - "parent bs" - "Number of lanes" - "Number of nics") Equivalent to DeviceType for CPU/GPU)

■Remarks

image of how to use





■Type 1. Node & device information

List of nodes and information about the devices installed on each node

It currently covers information about installed devices (GPU, FPGA, CPU, memory).

Automatically generated by the infrastructure information collection and management department.

	Name	Type	Req/Opt	Description
Spec	Devices	[]DeviceInfo	Required	List of device information for devices installed on the node (Define the number of devices (Currently FPGA, GPU, CPU and memory) installed in the node)
	nodeName	string	Required	Device File Path (Information to identify which device is physical. For the time being, only FPGA can be used. For GPUs, it is not necessary because the device file path is not used to physically identify the GPU.)
	deviceFilePath	*string	Optional	Host name
	deviceUUID	*string	Optional	Globally unique identifier of the device
	deviceType	string	Required	Accelerator type
	deviceIndex	int32	Required	Serial number of the device

■Type 2. Deployment information within the device

Information to define the deployable space provided on each node. Define deployment region information for each device (GPU, FPGA, CPU).

It currently covers information about installed devices (GPU, FPGA, CPU).

Automatically generated by the infrastructure information collection and management department.

Name	Type	Req/Opt	Description	Remarks
Devices	[]deviceRegioninfo	Required	List of region information created on the devices installed on the node (Define by the number of devices installed in the node (currently FPGAs, GPUs, and CPUs))	Define by the devices installed in the node (currently FPGAs and GPUs)
nodeName	string	Required	Host name	
deviceFilePath	*string	Optional	Device File Path (information to identify which FPGA device is physically)	Information to identify which device the device is physically
deviceUUID	*string	Optional	Globally unique identifier of the device (information to determine which GPU device is physical)	Note: Only for GPUs for now. (+ FPGA will handle UUID acquisition in the future) *The first "GPU-" in the UUID must be lowercase "gpu-". Use this value as metadata.name in FunctionTarget, so do not use uppercase characters according to the k8s
subDeviceSpecRef	string	Required	Identification information to identify the region deployed on this device	Type 3. Reference information for pulling the corresponding region information from the region-specific information. Type 3 also has parameters of the same name.
functionTargets	[]RegionInDevice	Required	List with each region information as an element	Is it better to use CR's FunctionTarget data structure for each element of the list?
regionName	string	Required	identification of the region	In the second half ・FPGA: Id of Lane (=FrameworkKernelId (=PtuKernelId)) ・GPU: 0 (fixed 0 is fine because GPU does not divide space)
functions	[]simplefunctioninfrastruct	Optional※	Information about functions already deployed in the region	Information about previously written circuits. *Also indicate the value of the number of Pod to be deployed (number of elements in functions below) in the region of GPU (as of March 2023).
functionName	string	Required	Name of the function	
functionIndex	int32	Required	Serial number of the deployed function	
frameworkKernelID	int32	Required	The Id of the kernel for chain control (FrameworkKernel).	
partitionName	string	Required	Physical information that identifies where deployed functions are actually deployed on the infrastructure	・FPGA:FunctionKernelId ・GPU — The UUID or ID(0, 1, ...) of the destination GPU for MPS, or the MIG instance ID for MIG ・CPU: NUMA Node, core information, etc. (if available) *For GPU/CPU, may include the identity of the pod (name or UUID)

*Required if the device is an FPGA

■Type 3. Region-specific information

Information specific to the bitstream/container image, such as FPGA child bitstream and Pod at GPU/CPU time.

Fixed information regardless of environment (destination system). (Once created, it can be deployed horizontally to any environment)

It's pre-created by hand (assuming you're an app developer developing bitstreams and containers).

Name	Type	Req/Opt	Description
subDeviceSpecRef	string	Required	Identification information to identify the region
functionTargets	[]RegionInDevice	Required	List using the region information as an element
regionName	string	Required	identification of the region
regionType	string	Required	region type of the region
maxFunctions	int32	Required	Maximum number of processing modules (Functions) that can be written in this region (Number of Functions = Number of
maxCapacity	int32	Required	Maximum processing power (fps) for the entire region

■Type 4-1. Func Specific Information - Common Attributes

Information about the attributes common to all functions that are deployed and executed in the deployment region.

Fixed information regardless of environment (destination system). (Once created, it can be deployed horizontally to any environment)

It's pre-created by hand (assuming you're an app developer developing bitstreams and containers).

O Circuit deployment destination information

Name		Type	Req/Opt	Description
Item		map[string][]FPGACatalog	Required	key is the entry number (character string) of the issue information. List whose value is FPGACatalog (json format)
	functionID	int32	Required	Identifier of the function (circuit/container image) (not currently used)
	functionName	string	Required	such Function name
	maxDataFlows	int32	Required	Maximum number of DF (WBFunc) that can be deployed to the function
	maxCapacity	int32	Required	Maximum processing power of the function

■Type 4-2. Func-specific - dedicated (filter/resize)

Information about the dedicated attributes required for FPGA decoding and FPGA filter/resize among the functions deployed and executed on the deployment region.

Fixed information regardless of environment (destination system). (Once created, it can be deployed horizontally to any environment)

It's pre-created by hand (assuming you're an app developer developing bitstreams).

Name	Type	Req/Opt	Description
functionKernels	map[string][]FunctionDetail	Required	List of resources in the FPGA provided by each lane of child bs for filter/resize
partitionName	string	Required	Physical information that identifies where the function is actually deployed on the infrastructure
functionChannelIDList	[]int32	Required	List of FunctionChannelID (FuncCHID) provided by this Function
functionChannelIDs	FunctionDetail	Required	Details of the resources in the FPGA associated with each FuncCHId (For each FuncCHID, the resource in the FPGA to be paid to the FPGAFunc is determined in combination with its Id.)
functionChannelID	int32	Required	ID of the FuncCH
rx	FPGACatalogmapRxTx	Required	Receiving side network information provided to FPGAFunction in set with the FuncCHID
protocol	map[string] FPGAConnectionCatalogDetails	Required	communication protocol of interest
port	*int32	Optional	Port number to be given as the receiver (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not used with DMA protocol)
dmaChannelID	*int32	Optional	ID of the DMA channel to be given as the receiver (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
fdmaConnectorID	*int32	Optional	ID of the connector for DMA transfer on the LLDMA side to be given as the receiver (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
tx	map[string] FPGAConnectionCatalogDetails	Required	Transmitter's network information given to the FPGAFunction in set with the FuncCHID
protocol	string	Required	communication protocol of interest
port	*int32	Optional	Port number to be given as the sender (For Ethernet connections (protocol is TCP/RTP). PCIe connection (not used with DMA protocol)
dmaChannelID	*int32	Optional	ID of the DMA channel to be given as the sender (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)
fdmaConnectorID	*int32	Optional	ID of the connector for DMA transfer on the LLDMA side to be given as the sender (For PCIe connections (protocol is DMA). Ethernet connection (not used with TCP/RTP protocol)