

## Schema [qis.instance.xsd](#)

schema location: [C:\inetpub\wwwroot\qisxml\schema\qis.instance.xsd](#)  
attribute form default: **unqualified**  
element form default: **qualified**  
targetNamespace: **qis:instance:1\_0**

Elements      Complex types  
[QIS](#)            [QISType](#)

schema location: [C:\inetpub\wwwroot\qisxml\schema\qis.gate.xsd](#)  
attribute form default: **unqualified**  
element form default: **qualified**  
targetNamespace: **qis:gate:1\_0**

Elements      Complex types  
[GateLibrary](#)    [GateType](#)

schema location: [C:\inetpub\wwwroot\qisxml\schema\qis.circuit.xsd](#)  
attribute form default: **unqualified**  
element form default: **qualified**  
targetNamespace: **qis:circuit:1\_0**

Elements                      Complex types  
[Circuit](#)                      [CircuitLibraryType](#)  
[CircuitLibrary](#)            [CircuitType](#)  
[GateEquivalentCircuit](#)    [GateEquivalentCircuitType](#)  
                                 [MapType](#)  
                                 [OperationType](#)  
                                 [StepType](#)

schema location: [C:\inetpub\wwwroot\qisxml\schema\qis.program.xsd](#)  
attribute form default: **unqualified**  
element form default: **qualified**  
targetNamespace: **qis:program:1\_0**

Elements                      Complex types  
[Memory](#)                      [MemoryType](#)  
[Prepare](#)                      [PrepareType](#)  
[Program](#)                      [ProgramLibraryType](#)  
[ProgramLibrary](#)            [ProgramType](#)  
[Register](#)                      [RegisterType](#)

schema location: [C:\inetpub\wwwroot\qisxml\schema\qis.reusable.xsd](#)  
attribute form default: **unqualified**

element form default: **qualified**  
 targetNamespace: **qjs:reusable:1\_0**

- |                                 |                                    |
|---------------------------------|------------------------------------|
| Elements                        | Complex types                      |
| <a href="#">Identification</a>  | <a href="#">ComplexNumberType</a>  |
| <a href="#">Input</a>           | <a href="#">IdentificationType</a> |
| <a href="#">Output</a>          | <a href="#">InputType</a>          |
| <a href="#">ProprietaryData</a> | <a href="#">MatrixCellType</a>     |
| <a href="#">Qubit</a>           | <a href="#">MatrixType</a>         |
| <a href="#">Reference</a>       | <a href="#">OutputType</a>         |
| <a href="#">Transformation</a>  | <a href="#">QubitType</a>          |
|                                 | <a href="#">ReferenceType</a>      |
|                                 | <a href="#">TransformationType</a> |

element **QIS**

diagram	
namespace	qjs:instance:1_0
type	<a href="#">QISType</a>
properties	content complex
children	<a href="#">r:Identification</a> <a href="#">g:GateLibrary</a> <a href="#">GateLibraryRef</a> <a href="#">c:CircuitLibrary</a> <a href="#">CircuitLibraryRef</a> <a href="#">p:ProgramLibrary</a> <a href="#">ProgramLibraryRef</a>
annotation	documentation The root element
source	<pre> &lt;xs:element name="QIS" type="QISType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The root element&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

## complexType QISType

diagram	<p>The diagram shows the structure of the QISType complex type. It consists of a sequence of elements: first, an element of type r:Identification, followed by a choice of six elements. The choice is optional, indicated by the cardinality 0..∞. The elements in the choice are: g:GateLibrary, GateLibraryRef (a reference type), c:CircuitLibrary, CircuitLibraryRef (a reference type), p:ProgramLibrary, and ProgramLibraryRef (a reference type).</p>
namespace	qis:instance:1_0
children	<a href="#">r:Identification</a> <a href="#">g:GateLibrary</a> <a href="#">GateLibraryRef</a> <a href="#">c:CircuitLibrary</a> <a href="#">CircuitLibraryRef</a> <a href="#">p:ProgramLibrary</a> <a href="#">ProgramLibraryRef</a>
used by	element <a href="#">QIS</a>
source	<pre> &lt;xs:complexType name="QISType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification"/&gt;     &lt;xs:choice minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:element ref="g:GateLibrary"/&gt;       &lt;xs:element name="GateLibraryRef" type="r:ReferenceType"/&gt;       &lt;xs:element ref="c:CircuitLibrary"/&gt;       &lt;xs:element name="CircuitLibraryRef" type="r:ReferenceType"/&gt;       &lt;xs:element ref="p:ProgramLibrary"/&gt;       &lt;xs:element name="ProgramLibraryRef" type="r:ReferenceType"/&gt;     &lt;/xs:choice&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

element **QISType/GateLibraryRef**

diagram							
namespace	qis:instance:1_0						
type	<a href="#">r:ReferenceType</a>						
properties	isRef	0					
	content	complex					
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	
	<a href="#">URI</a>	xs:anyURI					
source	<code>&lt;xs:element name="GateLibraryRef" type="r:ReferenceType"/&gt;</code>						

element **QISType/CircuitLibraryRef**

diagram							
namespace	qis:instance:1_0						
type	<a href="#">r:ReferenceType</a>						

properties	isRef 0 content complex												
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	xs:anyURI				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	xs:anyURI												
source	<code>&lt;xs:element name="CircuitLibraryRef" type="r:ReferenceType" /&gt;</code>												

### element **QIS**Type/ProgramLibraryRef


diagram													
namespace	qis:instance:1_0												
type	<a href="#">r:ReferenceType</a>												
properties	isRef 0 content complex												
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	xs:anyURI				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	xs:anyURI												
source	<code>&lt;xs:element name="ProgramLibraryRef" type="r:ReferenceType" /&gt;</code>												

### element **g**:GateLibrary

diagram	
namespace	qis:gate:1_0
properties	content complex
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Gate</a>

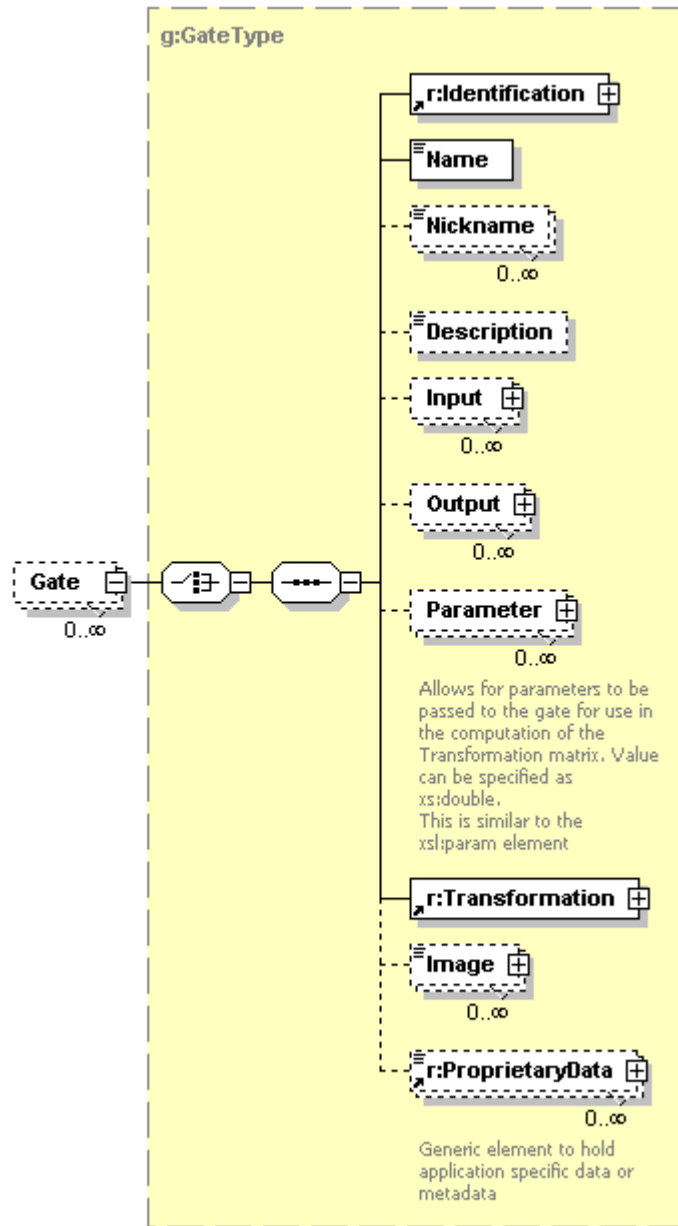
used by	complexType <a href="#">QIStype</a>
source	<pre> &lt;xs:element name="GateLibrary"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="r:Identification"/&gt;       &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element name="Gate" type="GateType" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

### element g:GateLibrary/Name

diagram	
namespace	qis:gate:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<pre>&lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;</pre>

element **g:GateLibrary/Gate**

diagram



namespace	qis:gate:1_0
type	<a href="#">g:GateType</a>
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Nickname</a> <a href="#">Description</a> <a href="#">Input</a> <a href="#">Output</a> <a href="#">Parameter</a> <a href="#">r:Transformation</a> <a href="#">Image</a> <a href="#">r:ProprietaryData</a>
source	<code>&lt;xs:element name="Gate" type="GateType" minOccurs="0" maxOccurs="unbounded" /&gt;</code>


complexType **g:GateType**

<p>diagram</p>	
<p>namespace</p>	<p>qis:gate:1_0</p>
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Nickname</a> <a href="#">Description</a> <a href="#">Input</a> <a href="#">Output</a> <a href="#">Parameter</a> <a href="#">r:Transformation</a> <a href="#">Image</a> <a href="#">r:ProprietaryData</a></p>
<p>used by</p>	<p>element <a href="#">g:GateLibrary/Gate</a></p>
<p>source</p>	<pre>&lt;xs:complexType name="GateType"&gt;   &lt;xs:choice&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="r:Identification"/&gt;       &lt;xs:element name="Name" type="xs:string"/&gt;       &lt;xs:element name="Nickname" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element name="Input" type="r:InputType" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xs:element name="Output" type="r:OutputType" minOccurs="0" maxOccurs="unbounded"/&gt;     </pre>

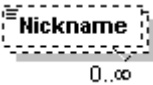


	<pre> &lt;xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Allows for parameters to be passed to the gate for use in the computation of the Transformation matrix. Value can be specified as xs:double. This is similar to the xsl:param element&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Name" type="xs:string"/&gt;       &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element name="Value" type="r:ComplexNumberType" minOccurs="0"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element ref="r:Transformation"/&gt; &lt;xs:element name="Image" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:anyURI"&gt;         &lt;xs:attribute name="format"/&gt;         &lt;xs:attribute name="width"/&gt;         &lt;xs:attribute name="height"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;xs:element ref="r:ProprietaryData" minOccurs="0" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt; &lt;/xs:choice&gt; &lt;/xs:complexType&gt; </pre>
--	---

element **g:GateType/Name**

diagram	
namespace	qis:gate:1_0
type	<b>xs:string</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="Name" type="xs:string"/&gt;</code>

element **g:GateType/Nickname**

diagram	
namespace	qis:gate:1_0

type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc unbounded content simple
source	<code>&lt;xs:element name="Nickname" type="xs:string" minOccurs="0" maxOccurs="unbounded"/&gt;</code>

**element g:GateType/Description**

diagram	
namespace	qis:gate:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>

**element g:GateType/Input**

diagram							
namespace	qis:gate:1_0						
type	<a href="#">r:InputType</a>						
properties	isRef 0 minOcc 0 maxOcc unbounded content complex						
children	<a href="#">Name</a> <a href="#">Description</a>						
attributes	Name	Type	Use	Default	Fixed	annotation documentation The input number. Must be lessof equal to gate size.	
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>					
source	<code>&lt;xs:element name="Input" type="r:InputType" minOccurs="0" maxOccurs="unbounded"/&gt;</code>						

element **g:GateType/Output**


diagram								
namespace	qis:gate:1_0							
type	<a href="#">r:OutputType</a>							
properties	isRef	0	minOcc	0	maxOcc	unbounded	content	complex
children	<a href="#">Name</a> <a href="#">Description</a>							
attributes	Name	Type	Use	Default	Fixed	annotation documentation		
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>				The input number. Must be lessof equalt to gate size.		
source	<pre>&lt;xs:element name="Output" type="r:OutputType" minOccurs="0" maxOccurs="unbounded" /&gt;</pre>							

element **g:GateType/Parameter**

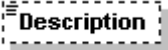
diagram	<p>Allows for parameters to be passed to the gate for use in the computation of the Transformation matrix. Value can be specified as xs:double. This is similar to the xs:param element</p>							
namespace	qis:gate:1_0							
properties	isRef	0	minOcc	0	maxOcc	unbounded	content	complex
children	<a href="#">Name</a> <a href="#">Description</a> <a href="#">Value</a>							

annotation	documentation Allows for parameters to be passed to the gate for use in the computation of the Transformation matrix. Value can be specified as xs:double. This is similar to the xsl:param element
source	<pre>&lt;xs:element name="Parameter" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Allows for parameters to be passed to the gate for use in the computation of the Transformation matrix. Value can be specified as xs:double. This is similar to the xsl:param element&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="Name" type="xs:string"/&gt;       &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element name="Value" type="r:ComplexNumberType" minOccurs="0"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

#### element **g:GateType/Parameter/Name**

diagram	
namespace	qis:gate:1_0
type	<b>xs:string</b>
properties	isRef 0 content simple
source	<pre>&lt;xs:element name="Name" type="xs:string"/&gt;</pre>

#### element **g:GateType/Parameter/Description**

diagram	
namespace	qis:gate:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<pre>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</pre>

element **g:GateType/Parameter/Value**

diagram	<p>Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>							
namespace	qis:gate:1_0							
type	<a href="#">r:ComplexNumberType</a>							
properties	isRef	0	minOcc	0	maxOcc	1	content	complex
children	<a href="#">Symbolic</a>							
attributes	Name	Type	Use	Default	Fixed	annotation documentation		
	<a href="#">r</a>	<b>xs:double</b>				Real component.		
	<a href="#">i</a>	<b>xs:double</b>				Imaginary component.		
source	<pre>&lt;xs:element name="Value" type="r:ComplexNumberType" minOccurs="0" /&gt;</pre>							

element **g:GateType/Image**

diagram							
namespace	qis:gate:1_0						
type	extension of <b>xs:anyURI</b>						
properties	isRef	0	minOcc	0			

	maxOcc content	unbounded complex					
attributes	Name <a href="#">format</a> <a href="#">width</a> <a href="#">height</a>	Type	Use	Default	Fixed	annotation	
source	<pre>&lt;xs:element name="Image" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:anyURI"&gt;         &lt;xs:attribute name="format"/&gt;         &lt;xs:attribute name="width"/&gt;         &lt;xs:attribute name="height"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>						

**attribute g:GateType/Image/@format**

properties	isRef 0
source	<pre>&lt;xs:attribute name="format"/&gt;</pre>

**attribute g:GateType/Image/@width**

properties	isRef 0
source	<pre>&lt;xs:attribute name="width"/&gt;</pre>

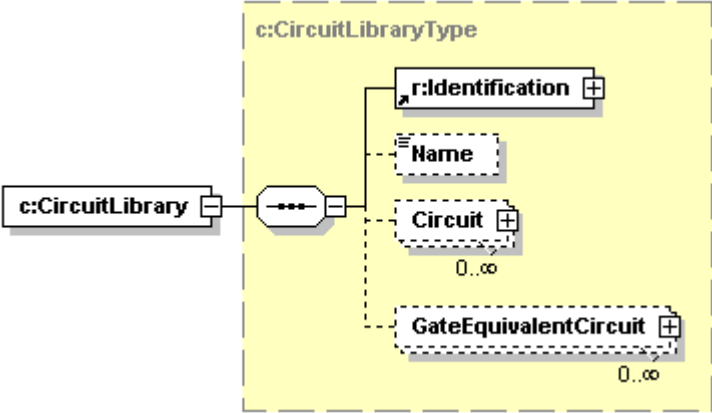
**attribute g:GateType/Image/@height**

properties	isRef 0
source	<pre>&lt;xs:attribute name="height"/&gt;</pre>

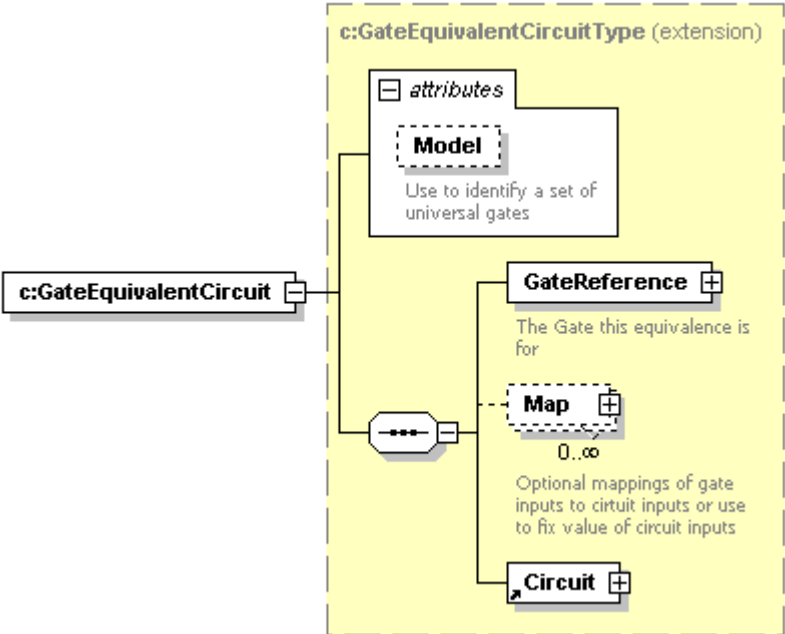
element **c:Circuit**

<p>diagram</p>	<p>The diagram illustrates the structure of the <b>c:CircuitType</b> complex type. It is composed of the following elements:</p> <ul style="list-style-type: none"> <li><b>Attributes:</b> A group containing the <b>size</b> attribute.</li> <li><b>Child Elements:</b> <ul style="list-style-type: none"> <li><b>r:Identification:</b> A required element containing <b>Name</b> and <b>Description</b> sub-elements.</li> <li><b>r:Input:</b> A required element with a cardinality of 0..∞, described as "Description of the gate/circuit input".</li> <li><b>r:Output:</b> A required element with a cardinality of 0..∞, described as "Description of the gate/circuit output".</li> <li><b>Step:</b> A required element with a cardinality of 1..∞.</li> <li><b>r:ProprietaryData:</b> A required element with a cardinality of 0..∞, described as "Generic element to hold application specific data or metadata".</li> </ul> </li> </ul>												
<p>namespace</p>	<p>qjs:circuit:1_0</p>												
<p>type</p>	<p><a href="#">c:CircuitType</a></p>												
<p>properties</p>	<p>content complex</p>												
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">r:Input</a> <a href="#">r:Output</a> <a href="#">Step</a> <a href="#">r:ProprietaryData</a></p>												
<p>used by</p>	<p>element <a href="#">p:ProgramType/Execute</a>          complexType <a href="#">c:GateEquivalentCircuitType</a></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">size</a></td> <td><b>xs:positiveInteger</b></td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">size</a>	<b>xs:positiveInteger</b>	required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">size</a>	<b>xs:positiveInteger</b>	required											
<p>source</p>	<pre>&lt;xs:element name="Circuit" type="CircuitType" /&gt;</pre>												

element **c:CircuitLibrary**

diagram	 <p>The diagram shows the structure of the <b>c:CircuitLibraryType</b>. It is a complex type containing the following elements:</p> <ul style="list-style-type: none"> <li><b>r:Identification</b> (required, 1 occurrence)</li> <li><b>Name</b> (optional, 1 occurrence)</li> <li><b>Circuit</b> (optional, 0..∞ occurrences)</li> <li><b>GateEquivalentCircuit</b> (optional, 0..∞ occurrences)</li> </ul>
namespace	qjs:circuit:1_0
type	<a href="#">c:CircuitLibraryType</a>
properties	content complex
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Circuit</a> <a href="#">GateEquivalentCircuit</a>
used by	complexType <a href="#">QISType</a>
source	<code>&lt;xs:element name="CircuitLibrary" type="CircuitLibraryType"/&gt;</code>

element **c:GateEquivalentCircuit**

diagram	 <p>The diagram shows the structure of the <b>c:GateEquivalentCircuitType (extension)</b>. It is a complex type containing the following elements:</p> <ul style="list-style-type: none"> <li><b>attributes</b> (optional):             <ul style="list-style-type: none"> <li><b>Model</b> (optional): Use to identify a set of universal gates</li> </ul> </li> <li><b>GateReference</b> (required, 1 occurrence): The Gate this equivalence is for</li> <li><b>Map</b> (optional, 0..∞ occurrences): Optional mappings of gate inputs to circuit inputs or use to fix value of circuit inputs</li> <li><b>Circuit</b> (required, 1 occurrence)</li> </ul>
namespace	qjs:circuit:1_0
type	extension of <a href="#">c:GateEquivalentCircuitType</a>
properties	content complex
children	<a href="#">GateReference</a> <a href="#">Map</a> <a href="#">c:Circuit</a>



attributes	Name <a href="#">Model</a>	Type	Use	Default	Fixed	annotation documentation Use to identify a set of universal gates
source	<pre>&lt;xs:element name="GateEquivalentCircuit"&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="GateEquivalentCircuitType"/&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>					

### complexType c:CircuitLibraryType

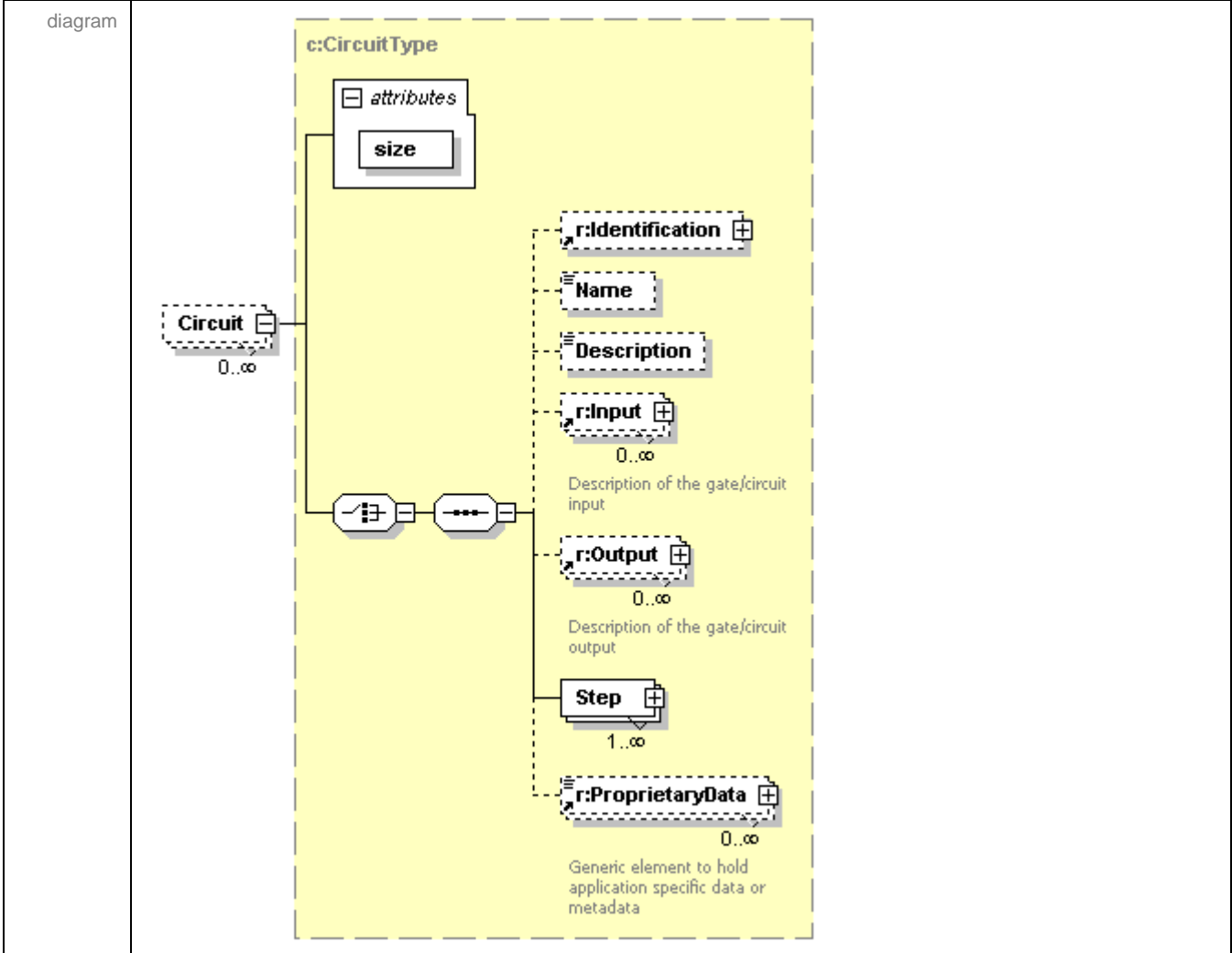
diagram	
namespace	qjs:circuit:1_0
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Circuit</a> <a href="#">GateEquivalentCircuit</a>
used by	element <a href="#">c:CircuitLibrary</a>
source	<pre>&lt;xs:complexType name="CircuitLibraryType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification"/&gt;     &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Circuit" type="CircuitType" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;xs:element name="GateEquivalentCircuit" type="GateEquivalentCircuitType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

### element c:CircuitLibraryType/Name

diagram	
namespace	qjs:circuit:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple

source	<xs:element name="Name" type="xs:string" minOccurs="0"/>
--------	--

element **c:CircuitLibraryType/Circuit**



namespace	qjs:circuit:1_0												
type	<a href="#">c:CircuitType</a>												
properties	isRef 0 minOcc 0 maxOcc unbounded content complex												
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">r:Input</a> <a href="#">r:Output</a> <a href="#">Step</a> <a href="#">r:ProprietaryData</a>												
used by	element <a href="#">p:ProgramType/Execute</a> complexType <a href="#">c:GateEquivalentCircuitType</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">size</a></td> <td>xs:positiveInteger</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">size</a>	xs:positiveInteger	required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">size</a>	xs:positiveInteger	required											
source	<xs:element name="Circuit" type="CircuitType" minOccurs="0" maxOccurs="unbounded"/>												

element **c:CircuitLibraryType/GateEquivalentCircuit**

diagram													
namespace	qis:circuit:1_0												
type	<a href="#">c:GateEquivalentCircuitType</a>												
properties	isRef 0 minOcc 0 maxOcc unbounded content complex												
children	<a href="#">GateReference</a> <a href="#">Map</a> <a href="#">c:Circuit</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">Model</a></td> <td></td> <td></td> <td></td> <td></td> <td>Use to identify a set of universal gates</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">Model</a>					Use to identify a set of universal gates
Name	Type	Use	Default	Fixed	annotation documentation								
<a href="#">Model</a>					Use to identify a set of universal gates								
source	<pre>&lt;xs:element name="GateEquivalentCircuit" type="GateEquivalentCircuitType" minOccurs="0" maxOccurs="unbounded"/&gt;</pre>												


complexType **c:CircuitType**

<p>diagram</p>													
<p>namespace</p>	<p>qis:circuit:1_0</p>												
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">r:Input</a> <a href="#">r:Output</a> <a href="#">Step</a> <a href="#">r:ProprietaryData</a></p>												
<p>used by</p>	<p>elements <a href="#">c:Circuit</a> <a href="#">c:CircuitLibraryType/Circuit</a></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">size</a></td> <td>xs:positiveInteger</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">size</a>	xs:positiveInteger	required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">size</a>	xs:positiveInteger	required											
<p>source</p>	<pre>&lt;xs:complexType name="CircuitType"&gt;   &lt;xs:choice&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="r:Identification" minOccurs="0"/&gt;       &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;       &lt;xs:element ref="r:Input" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xs:element ref="r:Output" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xs:element name="Step" type="StepType" maxOccurs="unbounded"/&gt;       &lt;xs:element ref="r:ProprietaryData" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:choice&gt;   &lt;xs:attribute name="size" type="xs:positiveInteger" use="required"/&gt; &lt;/xs:complexType&gt;</pre>												


**attribute c:CircuitType/@size**

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
source	<code>&lt;xs:attribute name="size" type="xs:positiveInteger" use="required"/&gt;</code>

**element c:CircuitType/Name**

diagram	
namespace	qis:circuit:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;</code>

**element c:CircuitType/Description**

diagram	
namespace	qis:circuit:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>

element **c:StepType/Step**

diagram									
namespace	qjs:circuit:1_0								
type	<a href="#">c:StepType</a>								
properties	<table border="0"> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>1</td></tr> <tr><td>maxOcc</td><td>unbounded</td></tr> <tr><td>content</td><td>complex</td></tr> </table>	isRef	0	minOcc	1	maxOcc	unbounded	content	complex
isRef	0								
minOcc	1								
maxOcc	unbounded								
content	complex								
children	<a href="#">Description</a> <a href="#">Operation</a> <a href="#">r:ProprietaryData</a>								
source	<code>&lt;xs:element name="Step" type="StepType" maxOccurs="unbounded" /&gt;</code>								

complexType **c:GateEquivalentCircuitType**

diagram	
namespace	qjs:circuit:1_0
children	<a href="#">GateReference</a> <a href="#">Map</a> <a href="#">c:Circuit</a>
used by	elements <a href="#">c:GateEquivalentCircuit</a> <a href="#">c:CircuitLibraryType/GateEquivalentCircuit</a>

attributes	Name	Type	Use	Default	Fixed	annotation documentation
	<a href="#">Model</a>					Use to identify a set of universal gates
source	<pre> &lt;xs:complexType name="GateEquivalentCircuitType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="GateReference" type="r:ReferenceType"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The Gate this equivalence is for&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element name="Map" type="MapType" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Optional mappings of gate inputs to circuit inputs or use to fix value of circuit inputs&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element ref="Circuit"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="Model"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Use to identify a set of universal gates&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>					

**attribute c:GateEquivalentCircuitType/@Model**

properties	isRef 0
annotation	documentation Use to identify a set of universal gates
source	<pre> &lt;xs:attribute name="Model"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Use to identify a set of universal gates&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

element **c:GateEquivalentCircuitType/GateReference**

<p>diagram</p>													
<p>namespace</p>	<p>qjs:circuit:1_0</p>												
<p>type</p>	<p><a href="#">r:ReferenceType</a></p>												
<p>properties</p>	<p>isRef 0 content complex</p>												
<p>children</p>	<p><a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	xs:anyURI				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	xs:anyURI												
<p>annotation</p>	<p>documentation The Gate this equivalence is for</p>												
<p>source</p>	<pre>&lt;xs:element name="GateReference" type="r:ReferenceType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The Gate this equivalence is for&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>												

element **c:GateEquivalentCircuitType/Map**

<p>diagram</p>	
----------------	--



namespace	qjs:circuit:1_0					
type	<a href="#">c:MapType</a>					
properties	isRef	0				
	minOcc	0				
	maxOcc	unbounded				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	annotation documentation
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>				Use to map a qubit to this input
	<a href="#">input</a>	<b>xs:positiveInteger</b>	required			
	<a href="#">value</a>	<b>xs:boolean</b>				documentation Use to set input to a specific value
annotation	documentation Optional mappings of gate inputs to circuit inputs or use to fix value of circuit inputs					
source	<pre>&lt;xs:element name="Map" type="MapType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Optional mappings of gate inputs to circuit inputs or use to fix value of circuit inputs&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>					

### complexType c:MapType

diagram						
namespace	qjs:circuit:1_0					
used by	elements	<a href="#">c:GateEquivalentCircuitType/Map</a> <a href="#">c:OperationType/Map</a>				
attributes	Name	Type	Use	Default	Fixed	annotation documentation
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>				Use to map a qubit to this input
	<a href="#">input</a>	<b>xs:positiveInteger</b>	required			
	<a href="#">value</a>	<b>xs:boolean</b>				documentation Use to set input to a specific value
source	<pre>&lt;xs:complexType name="MapType"&gt;   &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Use to map a qubit to this</pre>					

	<pre> input&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;xs:attribute name="input" type="xs:positiveInteger" use="required"/&gt; &lt;xs:attribute name="value" type="xs:boolean"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Use to set input to a specific value&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--	---

#### attribute c:MapType/@qubit

type	<b>xs:positiveInteger</b>
properties	isRef 0
annotation	documentation Use to map a qubit to this input
source	<pre> &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Use to map a qubit to this input&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

#### attribute c:MapType/@input

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
source	<pre> &lt;xs:attribute name="input" type="xs:positiveInteger" use="required"/&gt; </pre>

#### attribute c:MapType/@value

type	<b>xs:boolean</b>
properties	isRef 0
annotation	documentation Use to set input to a specific value
source	<pre> &lt;xs:attribute name="value" type="xs:boolean"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Use to set input to a specific value&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

complexType **c:OperationType**

<p>diagram</p>	<p><b>attributes</b></p> <p><b>reverse</b> If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)</p> <p><b>Map</b> 1..∞ Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?]</p> <p><b>GateRef</b></p> <p><b>CircuitRef</b></p> <p><b>Measurement</b> Multiple Map elements can be associated with a single measurement operation.</p> <p><b>r:ProprietaryData</b> 0..∞ Generic element to hold application specific data or metadata</p>												
<p>namespace</p>	<p>qjs:circuit:1_0</p>												
<p>children</p>	<p><a href="#">Map</a> <a href="#">GateRef</a> <a href="#">CircuitRef</a> <a href="#">Measurement</a> <a href="#">r:ProprietaryData</a></p>												
<p>used by</p>	<p>element <a href="#">c:StepType/Operation</a></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">reverse</a></td> <td></td> <td></td> <td></td> <td></td> <td>If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">reverse</a>					If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)
Name	Type	Use	Default	Fixed	annotation documentation								
<a href="#">reverse</a>					If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)								
<p>source</p>	<pre>&lt;xs:complexType name="OperationType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Map" type="MapType" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?]&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;</pre>												

	<pre> &lt;xs:choice&gt;   &lt;xs:element name="GateRef" type="r:ReferenceType"/&gt;   &lt;xs:element name="CircuitRef" type="r:ReferenceType"/&gt;   &lt;xs:element name="Measurement"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Multiple Map elements can be associated with a single measurement operation.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element ref="r:ProprietaryData" minOccurs="0" maxOccurs="unbounded"/&gt; &lt;/xs:sequence&gt; &lt;xs:attribute name="reverse"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--	---

**attribute c:OperationType/@reverse**

properties	isRef 0
annotation	documentation If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)
source	<pre> &lt;xs:attribute name="reverse"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

**element c:OperationType/Map**

diagram	<p> <b>Map</b>  1..∞  Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?] </p> <p> <b>c:MapType</b>  attributes  <b>qubit</b>  Use to map a qubit to this input  <b>input</b>  <b>value</b>  Use to set input to a specific value </p>
---------	--

namespace	qjs:circuit:1_0					
type	<a href="#">c:MapType</a>					
properties	isRef	0				
	minOcc	1				
	maxOcc	unbounded				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	annotation
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>				documentation Use to map a qubit to this input
	<a href="#">input</a>	<b>xs:positiveInteger</b>	required			
	<a href="#">value</a>	<b>xs:boolean</b>				documentation Use to set input to a specific value
annotation	documentation Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?]					
source	<pre>&lt;xs:element name="Map" type="MapType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?]&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>					

### element c:OperationType/GateRef

diagram						
namespace	qjs:circuit:1_0					
type	<a href="#">r:ReferenceType</a>					
properties	isRef	0				
	content	complex				
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>					
attributes	Name	Type	Use	Default	Fixed	annotation
	<a href="#">URI</a>	<b>xs:anyURI</b>				
source	<pre>&lt;xs:element name="GateRef" type="r:ReferenceType"/&gt;</pre>					

element **c:OperationType/CircuitRef**

diagram							
namespace	qjs:circuit:1_0						
type	<a href="#">r:ReferenceType</a>						
properties	isRef	0					
	content	complex					
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	
	<a href="#">URI</a>	xs:anyURI					
source	<code>&lt;xs:element name="CircuitRef" type="r:ReferenceType"/&gt;</code>						

element **c:OperationType/Measurement**

diagram							
namespace	qjs:circuit:1_0						
properties	isRef	0					
annotation	documentation	Multiple Map elements can be associated with a single measurement operation.					
source	<pre> &lt;xs:element name="Measurement"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Multiple Map elements can be associated with a single measurement operation.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>						

## complexType c:StepType

diagram	
namespace	qis:circuit:1_0
children	<a href="#">Description</a> <a href="#">Operation</a> <a href="#">r:ProprietaryData</a>
used by	element <a href="#">c:CircuitType/Step</a>
source	<pre>&lt;xs:complexType name="StepType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Operation" type="OperationType" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;An operation is a Gate with information mapping qubits or a fixed 0/1 value with the Gat or Circuit inputs&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element ref="r:ProprietaryData" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## element c:StepType/Description

diagram									
namespace	qis:circuit:1_0								
type	xs:string								
properties	<table> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>simple</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	simple
isRef	0								
minOcc	0								
maxOcc	1								
content	simple								
source	<pre>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</pre>								

element **c:StepType/Operation**

<p>diagram</p>	<p><b>c:OperationType</b></p> <p><b>attributes</b></p> <p><b>reverse</b> If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)</p> <p><b>Operation</b> 1..∞ An operation is a Gate with information mapping qubits or a fixed 0/1 value with the Gat or Circuit inputs</p> <p><b>Map</b> 1..∞ Maps registry qubits to gate inputs [COULD BE OPTIONAL? IF NOT SPECIFIED, USE NATURAL ORDER MAPPING?]</p> <p><b>GateRef</b></p> <p><b>CircuitRef</b></p> <p><b>Measurement</b> Multiple Map elements can be associated with a single measurement operation.</p> <p><b>r:ProprietaryData</b> 0..∞ Generic element to hold application specific data or metadata</p>												
<p>namespace</p>	<p>qjs:circuit:1_0</p>												
<p>type</p>	<p><a href="#">c:OperationType</a></p>												
<p>properties</p>	<p>isRef 0 minOcc 1 maxOcc unbounded content complex</p>												
<p>children</p>	<p><a href="#">Map</a> <a href="#">GateRef</a> <a href="#">CircuitRef</a> <a href="#">Measurement</a> <a href="#">r:ProprietaryData</a></p>												
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">reverse</a></td> <td></td> <td></td> <td></td> <td></td> <td>documentation If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">reverse</a>					documentation If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)
Name	Type	Use	Default	Fixed	annotation								
<a href="#">reverse</a>					documentation If true, indicates that the unitary transformation for this operation should be reversed (dagger/ complex conjugate transform)								
<p>annotation</p>	<p>documentation An operation is a Gate with information mapping qubits or a fixed 0/1 value with the Gat or Circuit inputs</p>												



source	<pre> &lt;xs:element name="Operation" type="OperationType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An operation is a Gate with information mapping qubits or a fixed 0/1 value with the Gat or Circuit inputs&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>
--------	--

**element p:Memory**

diagram							
namespace	qjs:program:1_0						
type	<a href="#">p:MemoryType</a>						
properties	content complex						
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">p:Prepare</a> <a href="#">Qubit</a>						
used by	complexType <a href="#">p:ProgramType</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	documentation
	<a href="#">size</a>	<b>xs:positiveInteger</b>	required				Number of qubits
source	<pre>&lt;xs:element name="Memory" type="MemoryType" /&gt;</pre>						

element **p:Prepare**

diagram						
namespace	qis:program:1_0					
type	<a href="#">p:PrepareType</a>					
properties	content complex					
children	<a href="#">QubitSet</a>					
used by	complexTypees <a href="#">p:MemoryType</a> <a href="#">p:RegisterType</a>					
attributes	Name <a href="#">reset</a>	Type <b>xs:boolean</b>	Use	Default	Fixed	annotation documentation Resets all qubits to zero
source	<code>&lt;xs:element name="Prepare" type="PrepareType" /&gt;</code>					

element **p:Program**

<p>diagram</p>	
<p>namespace</p>	<p>qis:program:1_0</p>
<p>type</p>	<p><a href="#">p:ProgramType</a></p>
<p>properties</p>	<p>content complex</p>
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">p:Memory</a> <a href="#">p:Register</a> <a href="#">Execute</a> <a href="#">Measure</a></p>
<p>used by</p>	<p>element <a href="#">p:ProgramType/Execute</a></p>
<p>source</p>	<p><code>&lt;xs:element name="Program" type="ProgramType" /&gt;</code></p>

### element **p:ProgramLibrary**

diagram	
namespace	qis:program:1_0
type	<a href="#">p:ProgramLibraryType</a>
properties	content complex
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Program</a>
used by	complexType <a href="#">QISType</a>
source	<code>&lt;xs:element name="ProgramLibrary" type="ProgramLibraryType" /&gt;</code>

### element **p:Register**

diagram	
---------	--

namespace	qis:program:1_0						
type	<a href="#">p:RegisterType</a>						
properties	content complex						
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">MemoryReference</a> <a href="#">QubitIndex</a> <a href="#">QubitRange</a> <a href="#">RegisterReference</a> <a href="#">p:Prepare</a>						
used by	elements	<a href="#">p:ProgramType/Execute</a> <a href="#">p:ProgramType/Measure</a>					
	complexType	<a href="#">p:ProgramType</a>					
attributes	Name	Type	Use	Default	Fixed	annotation	
	<a href="#">size</a>	<b>xs:positiveInteger</b>	required				
source	<code>&lt;xs:element name="Register" type="RegisterType" /&gt;</code>						

### complexType p:MemoryType


diagram							
namespace	qis:program:1_0						
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">p:Prepare</a> <a href="#">Qubit</a>						
used by	element	<a href="#">p:Memory</a>					
attributes	Name	Type	Use	Default	Fixed	annotation	documentation
	<a href="#">size</a>	<b>xs:positiveInteger</b>	required			Number of qubits	
source	<pre> &lt;xs:complexType name="MemoryType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification" minOccurs="0"/&gt;     &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element ref="Prepare" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Prepares the memory&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element name="Qubit" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Can be use to actually store memory values. </pre>						

	<p>Note that this element extends the r:Qubit type by adding an index attribute.</p> <pre> &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="r:QubitType"&gt;         &lt;xs:attribute name="index" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:sequence&gt; &lt;xs:attribute name="size" type="xs:positiveInteger" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Number of qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--	--

#### attribute p:MemoryType/@size

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
annotation	documentation Number of qubits
source	<pre> &lt;xs:attribute name="size" type="xs:positiveInteger" use="required"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Number of qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

#### element p:MemoryType/Name

diagram	
namespace	qis:program:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<pre> &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt; </pre>

element **p:MemoryType/Qubit**

diagram	<p>Can be use to actually store memory values. Note that this element extends the r:Qubit type by adding an index attribute.</p>												
namespace	qis:program:1_0												
type	extension of <a href="#">r:QubitType</a>												
properties	<table border="0"> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>unbounded</td></tr> <tr><td>content</td><td>complex</td></tr> </table>	isRef	0	minOcc	0	maxOcc	unbounded	content	complex				
isRef	0												
minOcc	0												
maxOcc	unbounded												
content	complex												
children	<a href="#">Zero</a> <a href="#">One</a>												
attributes	<table border="0"> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> <tr> <td><a href="#">index</a></td> <td></td> <td>required</td> <td></td> <td></td> <td></td> </tr> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">index</a>		required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">index</a>		required											
annotation	documentation Can be use to actually store memory values. Note that this element extends the r:Qubit type by adding an index attribute.												
source	<pre>&lt;xs:element name="Qubit" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Can be use to actually store memory values. Note that this element extends the r:Qubit type by adding an index attribute.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="r:QubitType"&gt;         &lt;xs:attribute name="index" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>												

attribute **p:MemoryType/Qubit/@index**

properties	<table border="0"> <tr><td>isRef</td><td>0</td></tr> <tr><td>use</td><td>required</td></tr> </table>	isRef	0	use	required
isRef	0				
use	required				
source	<pre>&lt;xs:attribute name="index" use="required"/&gt;</pre>				

complexType **p:PrepareType**

diagram													
namespace	qis:program:1_0												
children	<a href="#">QubitSet</a>												
used by	element <a href="#">p:Prepare</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">reset</a></td> <td><b>xs:boolean</b></td> <td></td> <td></td> <td></td> <td>Resets all qubits to zero</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">reset</a>	<b>xs:boolean</b>				Resets all qubits to zero
Name	Type	Use	Default	Fixed	annotation documentation								
<a href="#">reset</a>	<b>xs:boolean</b>				Resets all qubits to zero								
source	<pre> &lt;xs:complexType name="PrepareType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="QubitSet" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Set the value of qubit(s)&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:sequence&gt;           &lt;xs:choice maxOccurs="unbounded"&gt;             &lt;xs:element name="QubitIndex" type="xs:positiveInteger"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;A single qubit location&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="QubitRange"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;A range of qubits&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;xs:element name="StartQubit" type="xs:positiveInteger"/&gt;                   &lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;                 &lt;/xs:sequence&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:choice&gt;           &lt;xs:element name="Value" type="r:ComplexNumberType"/&gt;         &lt;/xs:sequence&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="reset" type="xs:boolean"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Resets all qubits to zero&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; </pre>												



	<pre> &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--	--

**attribute p:PrepareType/@reset**


type	<b>xs:boolean</b>
properties	isRef 0
annotation	documentation Resets all qubits to zero
source	<pre> &lt;xs:attribute name="reset" type="xs:boolean"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Resets all qubits to zero&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

**element p:PrepareType/QubitSet**

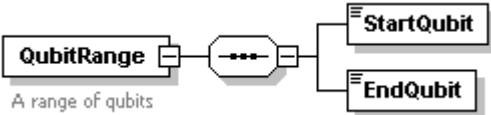
diagram	<p>The diagram illustrates the structure of the <b>QubitSet</b> element. It is a sequence of 1 to infinity instances. Each instance contains a choice of 1 to infinity elements: <b>QubitIndex</b> (1 to infinity) and <b>Value</b> (1 to infinity). The <b>QubitIndex</b> element itself contains a choice of 1 to infinity elements: <b>QubitIndex</b> (1 to infinity) and <b>QubitRange</b> (1 to infinity). The <b>QubitIndex</b> element is described as 'A single qubit location', and the <b>QubitRange</b> element is described as 'A range of qubits'. The <b>QubitSet</b> element is described as 'Set the value of qubit(s)'.</p>
namespace	qis:program:1_0
properties	isRef 0 minOcc 1 maxOcc unbounded content complex
children	<a href="#">QubitIndex</a> <a href="#">QubitRange</a> <a href="#">Value</a>
annotation	documentation Set the value of qubit(s)
source	<pre> &lt;xs:element name="QubitSet" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Set the value of qubit(s)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:choice maxOccurs="unbounded"&gt;         &lt;xs:element name="QubitIndex" type="xs:positiveInteger"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A single qubit location&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="QubitRange"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A range of qubits&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;       &lt;/xs:choice&gt;       &lt;xs:element name="Value" type="xs:complex"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

	<pre>                 &lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;             &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;     &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;xs:element name="Value" type="r:ComplexNumberType"/&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>
--	---


**element p:PrepareType/QubitSet/QubitIndex**

diagram	 <p>A single qubit location</p>
namespace	qis:program:1_0
type	<b>xs:positiveInteger</b>
properties	isRef 0 content simple
annotation	documentation A single qubit location
source	<pre> &lt;xs:element name="QubitIndex" type="xs:positiveInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single qubit location&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

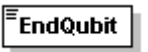
**element p:PrepareType/QubitSet/QubitRange**

diagram	 <p>A range of qubits</p>
namespace	qis:program:1_0
properties	isRef 0 content complex
children	<a href="#">StartQubit</a> <a href="#">EndQubit</a>
annotation	documentation A range of qubits
source	<pre> &lt;xs:element name="QubitRange"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A range of qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="StartQubit" type="xs:positiveInteger"/&gt;       &lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

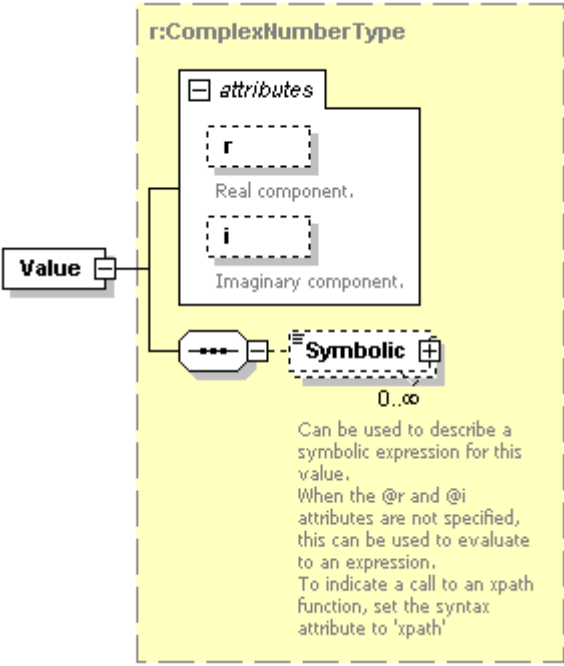
**element p:PrepareType/QubitSet/QubitRange/StartQubit**

diagram	
namespace	qis:program:1_0
type	xs:positiveInteger
properties	isRef 0 content simple
source	<xs:element name="StartQubit" type="xs:positiveInteger"/>

**element p:PrepareType/QubitSet/QubitRange/EndQubit**

diagram	
namespace	qis:program:1_0
type	xs:positiveInteger
properties	isRef 0 content simple
source	<xs:element name="EndQubit" type="xs:positiveInteger"/>

**element p:PrepareType/QubitSet/Value**

diagram	 <p><b>r:ComplexNumberType</b></p> <p><b>attributes</b></p> <p><b>r</b> Real component.</p> <p><b>i</b> Imaginary component.</p> <p><b>Symbolic</b> 0..∞</p> <p>Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>
namespace	qis:program:1_0
type	<a href="#">r:ComplexNumberType</a>
properties	isRef 0 content complex

children	<a href="#">Symbolic</a>					
attributes	Name	Type	Use	Default	Fixed	annotation documentation Real component. documentation Imaginary component.
	<a href="#">r</a>	<b>xs:double</b>				
	<a href="#">i</a>	<b>xs:double</b>				
source	<code>&lt;xs:element name="Value" type="r:ComplexNumberType"/&gt;</code>					

### complexType **p:ProgramLibraryType**

diagram	
namespace	qis:program:1_0
children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Program</a>
used by	element <a href="#">p:ProgramLibrary</a>
source	<pre> &lt;xs:complexType name="ProgramLibraryType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification"/&gt;     &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Program" type="ProgramType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

### element **p:ProgramLibraryType/Name**

diagram	
namespace	qis:program:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;</code>

element **p:ProgramLibraryType/Program**

<p>diagram</p>	
<p>namespace</p>	<p>qis:program:1_0</p>
<p>type</p>	<p><a href="#">p:ProgramType</a></p>
<p>properties</p>	<p>isRef 0 minOcc 0 maxOcc unbounded content complex</p>
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">p:Memory</a> <a href="#">p:Register</a> <a href="#">Execute</a> <a href="#">Measure</a></p>
<p>used by</p>	<p>element <a href="#">p:ProgramType/Execute</a></p>
<p>source</p>	<pre>&lt;xs:element name="Program" type="ProgramType" minOccurs="0" maxOccurs="unbounded" /&gt;</pre>

complexType **p:ProgramType**

<p>diagram</p>	<p><b>r:Identification</b> </p> <p><b>Name</b> </p> <p><b>Description</b> </p> <p><b>Memory</b> </p> <p>Program memory pool used be Registry (inspired by QCL)</p> <p><b>Register</b> </p> <p>0..∞</p> <p>A global register that can be referred from Execute elements. It is highly recommended for such register to contain an r:Indetiofication element (but it could also be referred to by idnex value). This is a bit equivalent of a global variable.</p> <p><b>Execute</b> </p> <p>Circuit to execute</p> <p><b>Measure</b> </p> <p>Can be used to cause a measurement of a specific set of qubit in the memory. This allows to ((1) run a circuit, measure, and then run another circuit, (2) when this is the last element, to measure and return only a subset of the memory. In case (1), this should cause a "returned" value to be sent to the classical world</p> <p>1..∞</p> <p>On or more circuit followed by optional measurement (if not specified, assumes the whole memeory is measured and returned)</p>
<p>namespace</p>	<p>qis:program:1_0</p>
<p>children</p>	<p><a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">Description</a> <a href="#">p:Memory</a> <a href="#">p:Register</a> <a href="#">Execute</a> <a href="#">Measure</a></p>
<p>used by</p>	<p>elements <a href="#">p:Program</a> <a href="#">p:ProgramLibraryType/Program</a></p>
<p>source</p>	<pre>&lt;xs:complexType name="ProgramType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification" minOccurs="0"/&gt;     &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element ref="Memory"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Program memory pool used be Registry (inspired by QCL)&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element ref="Register" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;</pre>

```

    <xs:documentation>A global register that can be referred from
Execute elements. It is highly recommended for such register to contain
an r:Indetiofication element (but it could also be referred to by idnex
value). This is a bit equivalent of a global variable.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:choice maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>On or more circuit followed by optional
measurement (if not specified, assumes the whole memeory is measured and
returned)</xs:documentation>
  </xs:annotation>
  <xs:element name="Execute">
    <xs:annotation>
      <xs:documentation>Circuit to execute</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element ref="Register">
            <xs:annotation>
              <xs:documentation>A subset of qubits from the main
program memory to use for this circuit. If not specified, assume that the
whole memory is used. This is a bit equivalent ot a local
variable.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="RegisterRef" type="r:ReferenceType"/>
        </xs:choice>
        <xs:choice>
          <xs:annotation>
            <xs:documentation>Provides reference to the Circuit or an
inline Circuit</xs:documentation>
          </xs:annotation>
          <xs:element name="CircuitRef" type="r:ReferenceType">
            <xs:annotation>
              <xs:documentation>Reference to a circuit in a
library</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="c:Circuit"/>
          <xs:element ref="Program">
            <xs:annotation>
              <xs:documentation>EXPERIMENTAL: calls a subprogram
Need to think how this might work memory wise. May need to wrap in a
SubProgram element and add a register (like Exeecute).</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="ProgramRef" type="r:ReferenceType">
            <xs:annotation>
              <xs:documentation>Reference to a Program in a
ProgramLibrary</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

	<pre> &lt;/xs:element&gt; &lt;xs:element name="Measure"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Can be used to cause a measurement of a specific set of qubit in the memory This allows to ((1) run a circuit, measure, and then run another circuit, (2) when this is the last element, to measure and return only a subset of the memory. In case (1), this should cause a "returned" value to be sent to the classical world&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Register"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; &lt;/xs:choice&gt; &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>
--	---

**element p:ProgramType/Name**

diagram	
namespace	qis:program:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;</code>

**element p:ProgramType/Description**

diagram	
namespace	qis:program:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>



element **p:ProgramType/Execute**

<p>diagram</p>	
<p>namespace</p>	<p>qis:program:1_0</p>
<p>properties</p>	<p>isRef 0 content complex</p>
<p>children</p>	<p><a href="#">p:Register</a> <a href="#">RegisterRef</a> <a href="#">CircuitRef</a> <a href="#">c:Circuit</a> <a href="#">p:Program</a> <a href="#">ProgramRef</a></p>
<p>annotation</p>	<p>documentation Circuit to execute</p>
<p>source</p>	<pre> &lt;xs:element name="Execute"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Circuit to execute&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:choice minOccurs="0"&gt;         &lt;xs:element ref="Register"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A subset of qubits from the main program memory to use for this circuit. If not specified, assume that the whole memory is used. This is a bit equivalent of a local variable.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="RegisterRef" type="r:ReferenceType"/&gt;       &lt;/xs:choice&gt;       &lt;xs:choice&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Provides reference to the Circuit or an </pre>



	<a href="#">URI</a> <b>xs:anyURI</b>
source	<code>&lt;xs:element name="RegisterRef" type="r:ReferenceType"/&gt;</code>

**element p:ProgramType/Execute/CircuitRef**

diagram	<p>The diagram illustrates the structure of the <code>r:ReferenceType</code> element. It is shown as a yellow dashed box containing an <code>attributes</code> section with a <code>URI</code> attribute and a content section with four child elements: <code>ID</code> (Object identifier), <code>LibraryID</code> (Library identifier), <code>AgencyID</code>, and <code>Version</code>. A separate box labeled <code>CircuitRef</code> with the description "Reference to a circuit in a library" is connected to the <code>r:ReferenceType</code> box.</p>												
namespace	qis:program:1_0												
type	<a href="#">r:ReferenceType</a>												
properties	isRef 0 content complex												
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td><b>xs:anyURI</b></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	<b>xs:anyURI</b>				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	<b>xs:anyURI</b>												
annotation	documentation Reference to a circuit in a library												
source	<pre>&lt;xs:element name="CircuitRef" type="r:ReferenceType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Reference to a circuit in a library&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>												

element **p:ProgramType/Execute/ProgramRef**

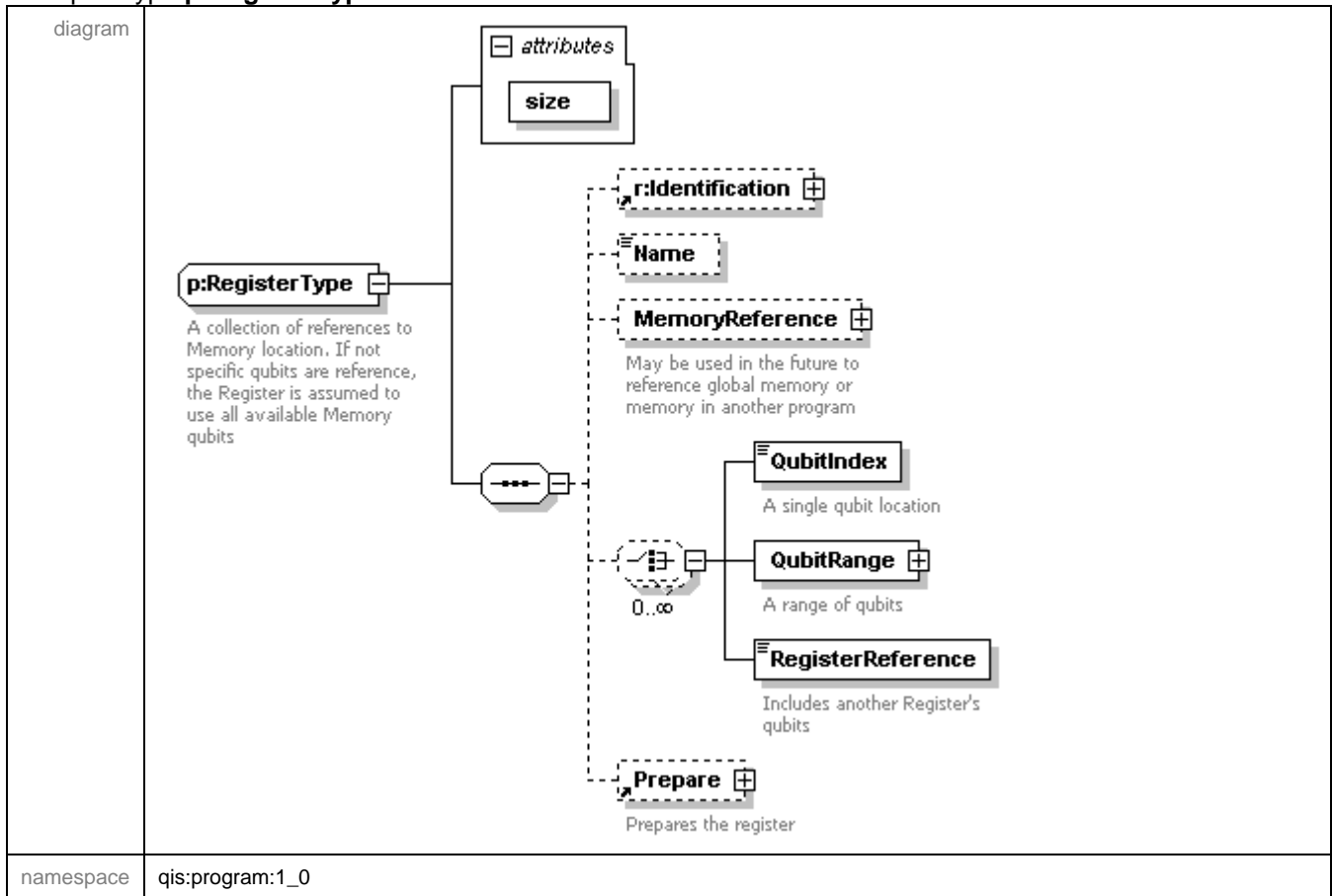
diagram							
namespace	qis:program:1_0						
type	<a href="#">r:ReferenceType</a>						
properties	isRef	0	content	complex			
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	
	<a href="#">URI</a>	<b>xs:anyURI</b>					
annotation	documentation Reference to a Program in a ProgramLibrary						
source	<pre>&lt;xs:element name="ProgramRef" type="r:ReferenceType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Reference to a Program in a ProgramLibrary&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>						

element **p:ProgramType/Measure**

diagram	<p>Can be used to cause a measurement of a specific set of qubit in the memory. This allows to (1) run a circuit, measure, and then run another circuit, (2) when this is the last element, to measure and return only a subset of the memory. In case (1), this should cause a "returned" value to be sent to the classical world</p>						
namespace	qis:program:1_0						
properties	isRef	0	content	complex			

children	<a href="#">p:Register</a>
annotation	documentation Can be used to cause a measurement of a specific set of qubit in the memory This allows to ((1) run a circuit, measure, and then run another circuit, (2) when this is the last element, to measure and return only a subset of the memory. In case (1), this should cause a "returned" value to be sent to the classical world
source	<pre> &lt;xs:element name="Measure"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Can be used to cause a measurement of a specific set of qubit in the memory This allows to ((1) run a circuit, measure, and then run another circuit, (2) when this is the last element, to measure and return only a subset of the memory. In case (1), this should cause a "returned" value to be sent to the classical world&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="Register"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

### complexType p:RegisterType



children	<a href="#">r:Identification</a> <a href="#">Name</a> <a href="#">MemoryReference</a> <a href="#">QubitIndex</a> <a href="#">QubitRange</a> <a href="#">RegisterReference</a> <a href="#">p:Prepare</a>					
used by	element <a href="#">p:Register</a>					
attributes	Name <a href="#">size</a>	Type <b>xs:positiveInteger</b>	Use required	Default	Fixed	annotation
annotation	documentation A collection of references to Memory location. If not specific qubits are reference, the Register is assumed to use all available Memory qubits					
source	<pre> &lt;xs:complexType name="RegisterType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A collection of references to Memory location. If not specific qubits are reference, the Register is assumed to use all available Memory qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="r:Identification" minOccurs="0"/&gt;     &lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;     &lt;xs:element name="MemoryReference" type="r:ReferenceType" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;May be used in the future to reference global memory or memory in another program&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:choice minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:element name="QubitIndex" type="xs:positiveInteger"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A single qubit location&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="QubitRange"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A range of qubits&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="StartQubit" type="xs:positiveInteger"/&gt;             &lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="RegisterReference"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Includes another Register's qubits&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;     &lt;xs:element ref="Prepare" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Prepares the register&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="size" type="xs:positiveInteger" use="required"/&gt; &lt;/xs:complexType&gt; </pre>					

attribute **p:RegisterType/@size**

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
source	<code>&lt;xs:attribute name="size" type="xs:positiveInteger" use="required"/&gt;</code>

element **p:RegisterType/Name**


diagram	
namespace	qis:program:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Name" type="xs:string" minOccurs="0"/&gt;</code>

element **p:RegisterType/MemoryReference**

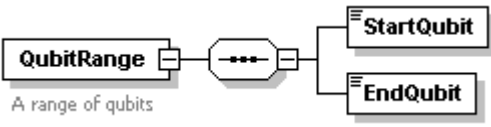
diagram						
namespace	qis:program:1_0					
type	<a href="#">r:ReferenceType</a>					
properties	isRef 0 minOcc 0 maxOcc 1 content complex					
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>					
attributes	Name	Type	Use	Default	Fixed	annotation
	<a href="#">URI</a>	<b>xs:anyURI</b>				

annotation	documentation May be used in the future to reference global memory or memory in another program
source	<pre>&lt;xs:element name="MemoryReference" type="r:ReferenceType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;May be used in the future to reference global memory or memory in another program&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**element p:RegisterType/QubitIndex**

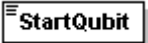
diagram	
namespace	qis:program:1_0
type	<b>xs:positiveInteger</b>
properties	isRef 0 content simple
annotation	documentation A single qubit location
source	<pre>&lt;xs:element name="QubitIndex" type="xs:positiveInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single qubit location&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**element p:RegisterType/QubitRange**


diagram	
namespace	qis:program:1_0
properties	isRef 0 content complex
children	<a href="#">StartQubit</a> <a href="#">EndQubit</a>
annotation	documentation A range of qubits
source	<pre>&lt;xs:element name="QubitRange"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A range of qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="StartQubit" type="xs:positiveInteger"/&gt;       &lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>



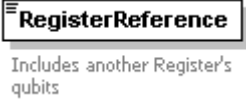
### element p:RegisterType/QubitRange/StartQubit

diagram	
namespace	qis:program:1_0
type	<b>xs:positiveInteger</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="StartQubit" type="xs:positiveInteger"/&gt;</code>

### element p:RegisterType/QubitRange/EndQubit

diagram	
namespace	qis:program:1_0
type	<b>xs:positiveInteger</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="EndQubit" type="xs:positiveInteger"/&gt;</code>

### element p:RegisterType/RegisterReference

diagram	 <p>Includes another Register's qubits</p>
namespace	qis:program:1_0
properties	isRef 0
annotation	documentation Includes another Register's qubits
source	<code>&lt;xs:element name="RegisterReference"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Includes another Register's qubits&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</code>

### element r:Identification

diagram	
namespace	qis:reusable:1_0
type	<a href="#">r:IdentificationType</a>
properties	content complex
children	<a href="#">ID</a> <a href="#">Agency</a> <a href="#">Version</a> <a href="#">VersionDescription</a>
used by	element <a href="#">g:GateLibrary</a> complexTypes <a href="#">c:CircuitLibraryType</a> <a href="#">c:CircuitType</a> <a href="#">g:GateType</a> <a href="#">p:MemoryType</a> <a href="#">p:ProgramLibraryType</a> <a href="#">p:ProgramType</a> <a href="#">p:RegisterType</a> <a href="#">QISType</a>
source	<code>&lt;xs:element name="Identification" type="IdentificationType" /&gt;</code>

### element r:Input

diagram													
namespace	qis:reusable:1_0												
type	<a href="#">r:InputType</a>												
properties	content complex												
children	<a href="#">Name</a> <a href="#">Description</a>												
used by	complexType <a href="#">c:CircuitType</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">qubit</a></td> <td><code>xs:positiveInteger</code></td> <td></td> <td></td> <td></td> <td>The input number. Must be lessof equalt to gate size.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">qubit</a>	<code>xs:positiveInteger</code>				The input number. Must be lessof equalt to gate size.
Name	Type	Use	Default	Fixed	annotation documentation								
<a href="#">qubit</a>	<code>xs:positiveInteger</code>				The input number. Must be lessof equalt to gate size.								
annotation	documentation Description of the gate/circuit input												

source	<pre>&lt;xs:element name="Input" type="InputType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Description of the gate/circuit input&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>
--------	---

### element r:Output

diagram							
namespace	qis:reusable:1_0						
type	<a href="#">r:OutputType</a>						
properties	content complex						
children	<a href="#">Name</a> <a href="#">Description</a>						
used by	complexType <a href="#">c:CircuitType</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	documentation
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>					The input number. Must be lessof equalt to gate size.
annotation	documentation Description of the gate/circuit output						
source	<pre>&lt;xs:element name="Output" type="OutputType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Description of the gate/circuit output&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>						

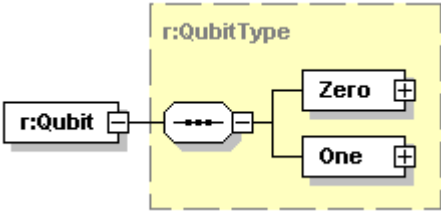
## element r:ProprietaryData

diagram													
namespace	qis:reusable:1_0												
type	extension of <b>xs:anyType</b>												
properties	content complex												
used by	complexTypes <a href="#">c:CircuitType</a> <a href="#">c:OperationType</a> <a href="#">c:StepType</a> <a href="#">g:GateType</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">type</a></td> <td></td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">type</a>		required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">type</a>		required											
annotation	documentation Generic element to hold application specific data or metadata												
source	<pre> &lt;xs:element name="ProprietaryData"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Generic element to hold application specific data or metadata &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="xs:anyType"&gt;         &lt;xs:attribute name="type" use="required"/&gt;       &lt;/xs:extension&gt;     &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;           </pre>												

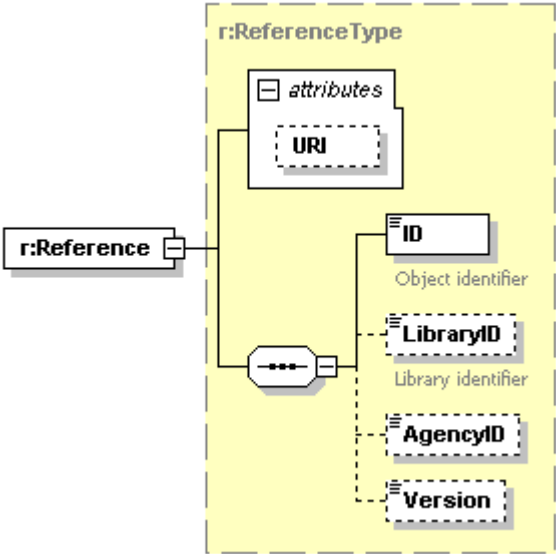
## attribute r:ProprietaryData/@type

properties	isRef 0 use required
source	<pre> &lt;xs:attribute name="type" use="required"/&gt;           </pre>

**element r:Qubit**

diagram	 <p>The diagram shows an <b>r:Qubit</b> element connected to a dashed box labeled <b>r:QubitType</b>. Inside this box, the <b>r:Qubit</b> element is connected to a container element (represented by a circle with three dots) which then branches into two child elements: <b>Zero</b> and <b>One</b>.</p>
namespace	qis:reusable:1_0
type	<a href="#">r:QubitType</a>
properties	content complex
children	<a href="#">Zero</a> <a href="#">One</a>
source	<code>&lt;xs:element name="Qubit" type="QubitType" /&gt;</code>

**element r:Reference**

diagram	 <p>The diagram shows an <b>r:Reference</b> element connected to a dashed box labeled <b>r:ReferenceType</b>. Inside this box, the <b>r:Reference</b> element is connected to an <b>attributes</b> container (represented by a folder icon) which contains a <b>URI</b> attribute. Below this, the <b>r:Reference</b> element is connected to a container element (represented by a circle with three dots) which then branches into four child elements: <b>ID</b> (Object identifier), <b>LibraryID</b> (Library identifier), <b>AgencyID</b>, and <b>Version</b>.</p>												
namespace	qis:reusable:1_0												
type	<a href="#">r:ReferenceType</a>												
properties	content complex												
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	xs:anyURI				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	xs:anyURI												
source	<code>&lt;xs:element name="Reference" type="ReferenceType" /&gt;</code>												

element **r:Transformation**

diagram							
namespace	qjs:reusable:1_0						
type	<a href="#">r:TransformationType</a>						
properties	content complex						
children	<a href="#">Multiplier</a> <a href="#">Cell</a>						
used by	complexType <a href="#">g:GateType</a>						
attributes	Name	Type	Use	Default	Fixed	annotation	
	<a href="#">size</a>	xs:int	required				
source	<code>&lt;xs:element name="Transformation" type="TransformationType" /&gt;</code>						

complexType **r:ComplexNumberType**

diagram	<p>Can be used to describe a symbolic expression for this value.          When the @r and @i attributes are not specified, this can be used to evaluate to an expression.          To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>						
namespace	qjs:reusable:1_0						
children	<a href="#">Symbolic</a>						
used by	elements	<a href="#">r:TransformationType/Multiplier</a> <a href="#">r:QubitType/One</a> <a href="#">g:GateType/Parameter/Value</a>					
	complexType	<a href="#">p:PrepareType/QubitSet/Value</a> <a href="#">r:QubitType/Zero</a> <a href="#">r:MatrixCellType</a>					
attributes	Name	Type	Use	Default	Fixed	annotation	

	<b>r</b>	<b>xs:double</b>	documentation Real component.
	<b>i</b>	<b>xs:double</b>	documentation Imaginary component.

source	<pre> &lt;xs:complexType name="ComplexNumberType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Symbolic" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:simpleContent&gt;           &lt;xs:extension base="xs:string"&gt;             &lt;xs:attribute name="syntax"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The language/syntax used for this symbolic representation (TeX, OpenOffice, XPath, etc.)&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:attribute&gt;           &lt;/xs:extension&gt;         &lt;/xs:simpleContent&gt;       &lt;/xs:complexType&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="r" type="xs:double"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Real component.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="i" type="xs:double"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Imaginary component.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--------	--

#### attribute r:ComplexNumberType/@r

type	<b>xs:double</b>
properties	isRef 0
annotation	documentation Real component.
source	<pre> &lt;xs:attribute name="r" type="xs:double"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Real component.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

attribute **r:ComplexNumberType/@i**

type	<b>xs:double</b>
properties	isRef 0
annotation	documentation Imaginary component.
source	<pre>&lt;xs:attribute name="i" type="xs:double"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Imaginary component.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

element **r:ComplexNumberType/Symbolic**

diagram	<p>Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>							
namespace	qjs:reusable:1_0							
type	extension of <b>xs:string</b>							
properties	isRef	0	minOcc	0	maxOcc	unbounded	content	complex
attributes	Name	Type	Use	Default	Fixed	annotation documentation The language/syntax used for this symbolic representation (TeX, OpenOffice, XPath, etc.)		
annotation	documentation Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'							
source	<pre>&lt;xs:element name="Symbolic" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Can be used to describe a symbolic expression for this value.     When the @r and @i attributes are not specified, this can be used to evaluate to an expression.     To indicate a call to an xpath function, set the syntax attribute to</pre>							



	<pre>'xpath'&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;xs:complexType&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="xs:string"&gt;       &lt;xs:attribute name="syntax"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The language/syntax used for this symbolic representation (TeX, OpenOffice, XPath, etc.)&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>
--	--

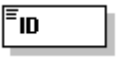
**attribute r:ComplexNumberType/Symbolic/@syntax**

properties	isRef 0
annotation	documentation The language/syntax used for this symbolic representation (TeX, OpenOffice, XPath, etc.)
source	<pre>&lt;xs:attribute name="syntax"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The language/syntax used for this symbolic representation (TeX, OpenOffice, XPath, etc.)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>


**complexType r:IdentificationType**

diagram	<pre> classDiagram     class r:IdentificationType     class ID     class Agency     class Version     class VersionDescription     r:IdentificationType -- ID     r:IdentificationType -- Agency     r:IdentificationType -- Version     r:IdentificationType -- VersionDescription   </pre>
namespace	qis:reusable:1_0
children	<a href="#">ID</a> <a href="#">Agency</a> <a href="#">Version</a> <a href="#">VersionDescription</a>
used by	element <a href="#">r:Identification</a>
source	<pre>&lt;xs:complexType name="IdentificationType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="ID" type="xs:NCName" /&gt;     &lt;xs:element name="Agency" type="xs:NCName" minOccurs="0" /&gt;     &lt;xs:element name="Version" type="xs:string" minOccurs="0" /&gt;     &lt;xs:element name="VersionDescription" type="xs:string" minOccurs="0" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>


### element r:IdentificationType/ID

diagram	
namespace	qis:reusable:1_0
type	<b>xs:NCName</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="ID" type="xs:NCName"/&gt;</code>


### element r:IdentificationType/Agency

diagram	
namespace	qis:reusable:1_0
type	<b>xs:NCName</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Agency" type="xs:NCName" minOccurs="0"/&gt;</code>

### element r:IdentificationType/Version

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Version" type="xs:string" minOccurs="0"/&gt;</code>

### element r:IdentificationType/VersionDescription

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="VersionDescription" type="xs:string" minOccurs="0"/&gt;</code>


### complexType r:InputType

diagram							
namespace	qis:reusable:1_0						
children	<a href="#">Name</a> <a href="#">Description</a>						
used by	elements <a href="#">g:GateType/Input r:Input</a>						
attributes	Name	Type	Use	Default	Fixed	annotation documentation The input number. Must be lessof equal to gate size.	
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>					
source	<pre> &lt;xs:complexType name="InputType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Name" type="xs:string"/&gt;     &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The input number. Must be lessof equal to gate size.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>						

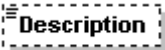
### attribute r:InputType/@qubit

type	<b>xs:positiveInteger</b>
properties	isRef 0
annotation	documentation The input number. Must be lessof equal to gate size.
source	<pre> &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The input number. Must be lessof equal to gate size.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>

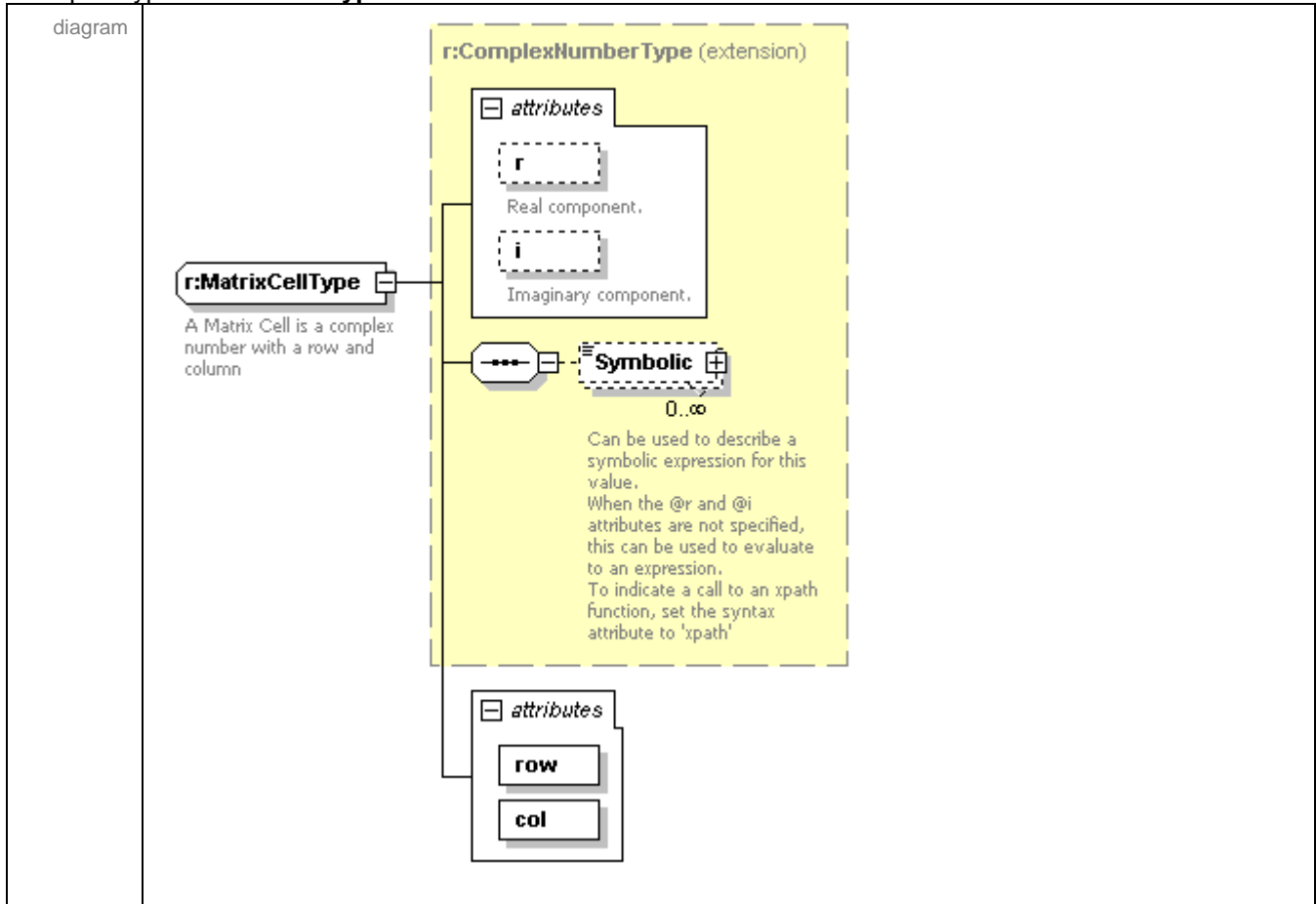
### element r:InputType/Name

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="Name" type="xs:string"/&gt;</code>

### element r:InputType/Description

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>

complexType r:MatrixCellType



namespace	qis:reusable:1_0					
type	extension of <a href="#">r:ComplexNumberType</a>					
properties	base ComplexNumberType					
children	<a href="#">Symbolic</a>					
used by	elements <a href="#">r:MatrixType/Cell</a> <a href="#">r:TransformationType/Cell</a>					
attributes	Name	Type	Use	Default	Fixed	annotation documentation
	<a href="#">r</a>	xs:double				Real component.
	<a href="#">i</a>	xs:double				Imaginary component.
	<a href="#">row</a>	xs:positiveInteger	required			
	<a href="#">col</a>	xs:positiveInteger	required			
annotation	documentation A Matrix Cell is a complex number with a row and column					
source	<pre>&lt;xs:complexType name="MatrixCellType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A Matrix Cell is a complex number with a row and column&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ComplexNumberType"&gt;</pre>					

	<pre> &lt;xs:attribute name="row" type="xs:positiveInteger" use="required"/&gt; &lt;xs:attribute name="col" type="xs:positiveInteger" use="required"/&gt; &lt;/xs:extension&gt; &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt; </pre>
--	---

#### attribute r:MatrixCellType/@row

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
source	<code>&lt;xs:attribute name="row" type="xs:positiveInteger" use="required"/&gt;</code>

#### attribute r:MatrixCellType/@col

type	<b>xs:positiveInteger</b>
properties	isRef 0 use required
source	<code>&lt;xs:attribute name="col" type="xs:positiveInteger" use="required"/&gt;</code>

#### complexType r:MatrixType

diagram							
namespace	qis:reusable:1_0						
children	<a href="#">Cell</a>						
attributes	Name	Type	Use	Default	Fixed	Use	annotation
	<a href="#">rows</a>	<b>xs:integer</b>					
	<a href="#">cols</a>	<b>xs:integer</b>					
source	<pre> &lt;xs:complexType name="MatrixType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Cell" type="MatrixCellType" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="rows" type="xs:integer"/&gt;   &lt;xs:attribute name="cols" type="xs:integer"/&gt; &lt;/xs:complexType&gt; </pre>						

#### attribute r:MatrixType/@rows

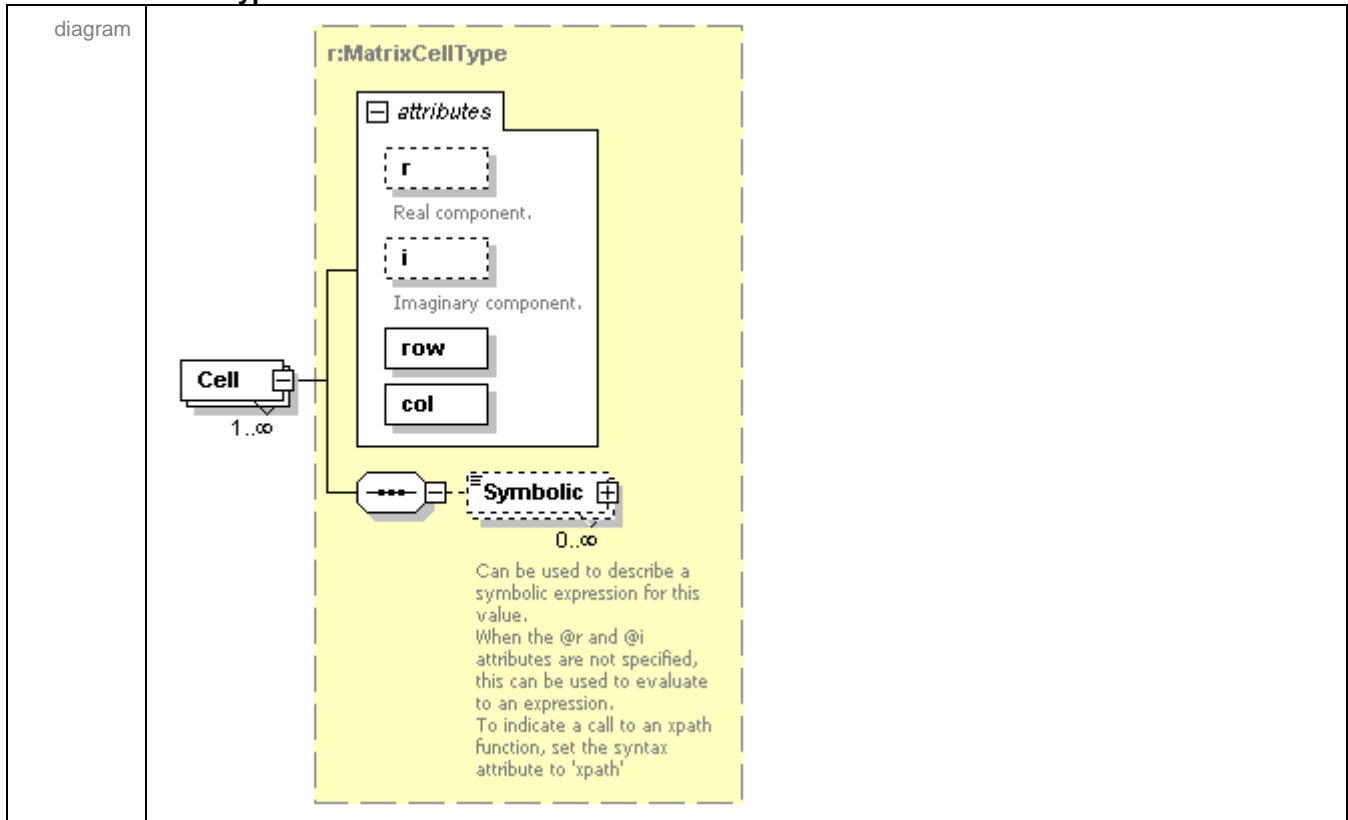
type	<b>xs:integer</b>
properties	isRef 0

source	<code>&lt;xs:attribute name="rows" type="xs:integer"/&gt;</code>
--------	--

### attribute r:MatrixType/@cols

type	<code>xs:integer</code>
properties	isRef 0
source	<code>&lt;xs:attribute name="cols" type="xs:integer"/&gt;</code>

### element r:MatrixType/Cell



namespace	qis:reusable:1_0
-----------	------------------

type	<a href="#">r:MatrixCellType</a>
------	----------------------------------

properties	isRef 0 minOcc 1 maxOcc unbounded content complex
------------	--

children	<a href="#">Symbolic</a>
----------	--------------------------

attributes	Name	Type	Use	Default	Fixed	annotation documentation
	<a href="#">r</a>	<code>xs:double</code>				Real component.
	<a href="#">i</a>	<code>xs:double</code>				Imaginary component.
	<a href="#">row</a>	<code>xs:positiveInteger</code>	required			
	<a href="#">col</a>	<code>xs:positiveInteger</code>	required			

source	<code>&lt;xs:element name="Cell" type="MatrixCellType" maxOccurs="unbounded"/&gt;</code>
--------	--

### complexType r:OutputType


diagram							
namespace	qis:reusable:1_0						
children	<a href="#">Name</a> <a href="#">Description</a>						
used by	elements <a href="#">g:GateType/Output r:Output</a>						
attributes	Name	Type	Use	Default	Fixed	annotation documentation The input number. Must be lessof equalt to gate size.	
	<a href="#">qubit</a>	<b>xs:positiveInteger</b>					
source	<pre> &lt;xs:complexType name="OutputType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Name" type="xs:string"/&gt;     &lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The input number. Must be lessof equalt to gate size.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>						

### attribute r:OutputType/@qubit

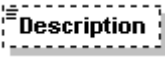
type	<b>xs:positiveInteger</b>
properties	isRef 0
annotation	documentation The input number. Must be lessof equalt to gate size.
source	<pre> &lt;xs:attribute name="qubit" type="xs:positiveInteger"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The input number. Must be lessof equalt to gate size.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; </pre>



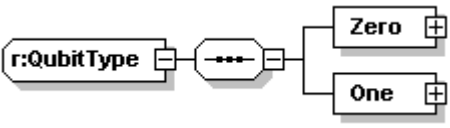
**element r:OutputType/Name**

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 content simple
source	<code>&lt;xs:element name="Name" type="xs:string"/&gt;</code>

**element r:OutputType/Description**

diagram	
namespace	qis:reusable:1_0
type	<b>xs:string</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
source	<code>&lt;xs:element name="Description" type="xs:string" minOccurs="0"/&gt;</code>

**complexType r:QubitType**

diagram	
namespace	qis:reusable:1_0
children	<a href="#">Zero</a> <a href="#">One</a>
used by	elements <a href="#">p:MemoryType/Qubit</a> <a href="#">r:Qubit</a>
source	<pre>&lt;xs:complexType name="QubitType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Zero" type="ComplexNumberType"/&gt;     &lt;xs:element name="One" type="ComplexNumberType"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

element **r:QubitType/Zero**

diagram	<p><b>r:ComplexNumberType</b></p> <p><b>attributes</b></p> <p><b>r</b> Real component.</p> <p><b>i</b> Imaginary component.</p> <p><b>Zero</b></p> <p><b>Symbolic</b> 0..∞</p> <p>Can be used to describe a symbolic expression for this value. When the @r and @i attributes are not specified, this can be used to evaluate to an expression. To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>																		
namespace	qis:reusable:1_0																		
type	<a href="#">r:ComplexNumberType</a>																		
properties	isRef 0 content complex																		
children	<a href="#">Symbolic</a>																		
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">r</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>documentation Real component.</td> </tr> <tr> <td><a href="#">i</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>documentation Imaginary component.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">r</a>	<b>xs:double</b>				documentation Real component.	<a href="#">i</a>	<b>xs:double</b>				documentation Imaginary component.
Name	Type	Use	Default	Fixed	annotation														
<a href="#">r</a>	<b>xs:double</b>				documentation Real component.														
<a href="#">i</a>	<b>xs:double</b>				documentation Imaginary component.														
source	<code>&lt;xs:element name="Zero" type="ComplexNumberType" /&gt;</code>																		

element r:QubitType/One

<p>diagram</p>																			
<p>namespace</p>	<p>qis:reusable:1_0</p>																		
<p>type</p>	<p><a href="#">r:ComplexNumberType</a></p>																		
<p>properties</p>	<table border="0"> <tr> <td>isRef</td> <td>0</td> </tr> <tr> <td>content</td> <td>complex</td> </tr> </table>	isRef	0	content	complex														
isRef	0																		
content	complex																		
<p>children</p>	<p><a href="#">Symbolic</a></p>																		
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">r</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>documentation Real component.</td> </tr> <tr> <td><a href="#">i</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>documentation Imaginary component.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">r</a>	<b>xs:double</b>				documentation Real component.	<a href="#">i</a>	<b>xs:double</b>				documentation Imaginary component.
Name	Type	Use	Default	Fixed	annotation														
<a href="#">r</a>	<b>xs:double</b>				documentation Real component.														
<a href="#">i</a>	<b>xs:double</b>				documentation Imaginary component.														
<p>source</p>	<pre>&lt;xs:element name="One" type="ComplexNumberType" /&gt;</pre>																		


### complexType r:ReferenceType

diagram													
namespace	qis:reusable:1_0												
children	<a href="#">ID</a> <a href="#">LibraryID</a> <a href="#">AgencyID</a> <a href="#">Version</a>												
used by	elements <a href="#">QISType/CircuitLibraryRef c:OperationType/CircuitRef p:ProgramType/Execute/CircuitRef</a> <a href="#">QISType/GateLibraryRef c:OperationType/GateRef c:GateEquivalentCircuitType/GateReference</a> <a href="#">p:RegisterType/MemoryReference QISType/ProgramLibraryRef p:ProgramType/Execute/ProgramRef</a> <a href="#">r:Reference p:ProgramType/Execute/RegisterRef</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">URI</a></td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">URI</a>	xs:anyURI				
Name	Type	Use	Default	Fixed	annotation								
<a href="#">URI</a>	xs:anyURI												
source	<pre>&lt;xs:complexType name="ReferenceType"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="ID" type="xs:NCName"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Object identifier&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element name="LibraryID" type="xs:NCName" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Library identifier&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:element name="AgencyID" minOccurs="0"/&gt;     &lt;xs:element name="Version" minOccurs="0"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="URI" type="xs:anyURI"/&gt; &lt;/xs:complexType&gt;</pre>												


### attribute r:ReferenceType/@URI

type	xs:anyURI
properties	isRef 0
source	<pre>&lt;xs:attribute name="URI" type="xs:anyURI"/&gt;</pre>

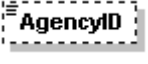
### element r:ReferenceType/ID

diagram	
namespace	qis:reusable:1_0
type	<b>xs:NCName</b>
properties	isRef 0 content simple
annotation	documentation Object identifier
source	<pre>&lt;xs:element name="ID" type="xs:NCName"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Object identifier&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### element r:ReferenceType/LibraryID

diagram	
namespace	qis:reusable:1_0
type	<b>xs:NCName</b>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
annotation	documentation Library identifier
source	<pre>&lt;xs:element name="LibraryID" type="xs:NCName" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Library identifier&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### element r:ReferenceType/AgencyID

diagram	
namespace	qis:reusable:1_0
properties	isRef 0 minOcc 0 maxOcc 1
source	<pre>&lt;xs:element name="AgencyID" minOccurs="0"/&gt;</pre>

### element r:ReferenceType/Version

diagram	
---------	---

namespace	qis:reusable:1_0
properties	isRef 0 minOcc 0 maxOcc 1
source	<xs:element name="Version" minOccurs="0"/>

### complexType r:TransformationType

diagram													
namespace	qis:reusable:1_0												
children	<a href="#">Multiplier</a> <a href="#">Cell</a>												
used by	element <a href="#">r:Transformation</a>												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td><a href="#">size</a></td> <td>xs:int</td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation	<a href="#">size</a>	xs:int	required			
Name	Type	Use	Default	Fixed	annotation								
<a href="#">size</a>	xs:int	required											
annotation	documentation A unitary Transformation is a square matrix of complex numbers.												
source	<pre>&lt;xs:complexType name="TransformationType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A unitary Transformation is a square matrix of complex numbers. &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Multiplier" type="ComplexNumberType" minOccurs="0"/&gt;     &lt;xs:element name="Cell" type="MatrixCellType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="size" type="xs:int" use="required"/&gt; &lt;/xs:complexType&gt;</pre>												

### attribute r:TransformationType/@size

type	xs:int
properties	isRef 0 use required
source	<xs:attribute name="size" type="xs:int" use="required"/>

element r:TransformationType/Multiplier

diagram	<p>Can be used to describe a symbolic expression for this value.          When the @r and @i attributes are not specified, this can be used to evaluate to an expression.          To indicate a call to an xpath function, set the syntax attribute to 'xpath'</p>																		
namespace	qis:reusable:1_0																		
type	<a href="#">r:ComplexNumberType</a>																		
properties	<table border="0"> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>complex</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	complex										
isRef	0																		
minOcc	0																		
maxOcc	1																		
content	complex																		
children	<a href="#">Symbolic</a>																		
attributes	<table border="0"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">r</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>Real component.</td> </tr> <tr> <td><a href="#">i</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>Imaginary component.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">r</a>	<b>xs:double</b>				Real component.	<a href="#">i</a>	<b>xs:double</b>				Imaginary component.
Name	Type	Use	Default	Fixed	annotation documentation														
<a href="#">r</a>	<b>xs:double</b>				Real component.														
<a href="#">i</a>	<b>xs:double</b>				Imaginary component.														
source	<code>&lt;xs:element name="Multiplier" type="ComplexNumberType" minOccurs="0" /&gt;</code>																		

element **r:TransformationType/Cell**

diagram																															
namespace	qjs:reusable:1_0																														
type	<a href="#">r:MatrixCellType</a>																														
properties	isRef 0 minOcc 0 maxOcc unbounded content complex																														
children	<a href="#">Symbolic</a>																														
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation documentation</th> </tr> </thead> <tbody> <tr> <td><a href="#">r</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>Real component.</td> </tr> <tr> <td><a href="#">i</a></td> <td><b>xs:double</b></td> <td></td> <td></td> <td></td> <td>Imaginary component.</td> </tr> <tr> <td><a href="#">row</a></td> <td><b>xs:positiveInteger</b></td> <td>required</td> <td></td> <td></td> <td></td> </tr> <tr> <td><a href="#">col</a></td> <td><b>xs:positiveInteger</b></td> <td>required</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation documentation	<a href="#">r</a>	<b>xs:double</b>				Real component.	<a href="#">i</a>	<b>xs:double</b>				Imaginary component.	<a href="#">row</a>	<b>xs:positiveInteger</b>	required				<a href="#">col</a>	<b>xs:positiveInteger</b>	required			
Name	Type	Use	Default	Fixed	annotation documentation																										
<a href="#">r</a>	<b>xs:double</b>				Real component.																										
<a href="#">i</a>	<b>xs:double</b>				Imaginary component.																										
<a href="#">row</a>	<b>xs:positiveInteger</b>	required																													
<a href="#">col</a>	<b>xs:positiveInteger</b>	required																													
source	<pre>&lt;xs:element name="Cell" type="MatrixCellType" minOccurs="0" maxOccurs="unbounded" /&gt;</pre>																														