Summary - File Read

Name	file_read
Worker Type	Application
Version	v1.5
Release Date	4/2019
Component Library	ocpi.core
Workers	file_read.rcc file_read.hdl
Tested Platforms	isim, xsim, modelsim, xilinx13_3, xilinx13_4, centos6, centos7

Revision History

Revision	Description of Change	Date
v1.4	Initial release.	9/2019
v1.5	Changed all readable properties to readback and changed default value of MessageSize from 4096 to 0. Also updated the end of file descriptions with smart wrapper updates.	4/2019

Functionality

The File_Read component injects file-based data into an application. It is used by specifying an instance of the File_Read component, and connecting its output port to an input port of the component which will process the data first. The name of the file to be read is specified in a property.

Operating Modes

Data Streaming Mode

In data streaming mode, the contents of the file becomes the payloads of a stream of messages, each carrying a fixed number of bytes of file data and all with the same opcode. The length and opcode of all output messages are specified as properties. This means that this mode only lends itself to protocols that have a single opcode or the intent is to only send data on one of the opcodes of a multi-opcode protocol.

If the number of bytes in the file is not an even multiple of the message size the remaining bytes are sent in a final shorter message. The granularity of messages can also be specified. This forces the message size to be a multiple of this value, and forces truncation of the final message to be a multiple of this value.

Messaging Mode

In messaging mode, the contents of the file are interpreted as a sequence of defined messages, with an 8-byte header in the file itself preceding the payload for each message. This header contains the length and opcode of the message, with the data contents of the message following the header. The length can be zero, meaning that a message will be sent with the indicated opcode, and the length of the message will be zero.

The first 32-bit word of the header is interpreted as the message length in bytes, little-endian. The next 8-bit byte is the opcode of the message, followed by 3 padding bytes. If the end of the file is encountered while reading a message header, or while reading the header-specified length of the message payload, an error will be reported and the component will terminate. The following Messaging File Field Layout shows the layout of a file:

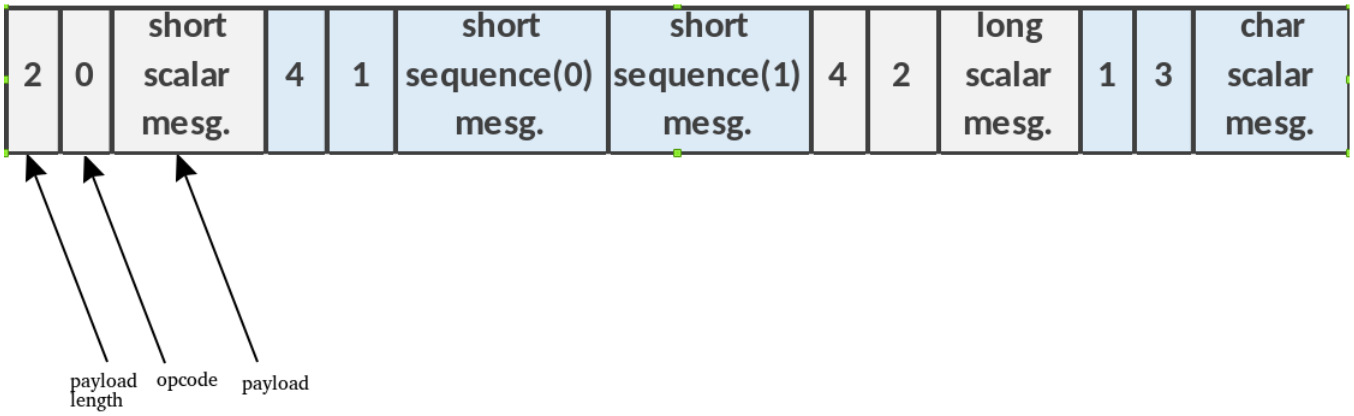

Figure 1: Messaging File Field Layout

No Protocol

The port on the component has no protocol specified. This means that the data file must be formatted to match the protocol of the input port of the connected worker. For message mode this means only using opcodes and payloads in the file that correspond to the protocol of the connected component. In data streaming mode the file structure needs to correspond to the opcode that is set by the *opcode* property.

Message Size/Buffers

The protocol of the connected component needs to be taken into account when selecting the message size (bytes) to prevent buffer overflow problems. The component has no information about what component it is connected to (by design). The application designer needs to set a valid value. For example iqstream protocol uses a sequence of 2048 16-bit I/Q pairs which means anything over 8192 (2048 * 4 bytes per pair) is invalid.

End of File Handling

When the File_Read component reaches the end of its input file, it will do one of three things:

- send a end of file notification
- enter the "done" state with no further action, when the *suppressEOF* property is true
- restart reading at the beginning of the file, when the *repeat* property is true

Worker Implementation Details

file_read.hdl

This worker will only run on simulator platforms. This includes isim, xsim, and modelsim and will not run on or be built for any other hardware platforms. This is because it conatins code that cannot be realized into RTL.

file_read.rcc

This worker in implemented in the C language version of the RCC model. Most new workers are implemented in the C++ language version of the RCC model.

Source Dependencies

file_read.rcc

- file_read.c
- file_read.h

file_read.hdl

- file_read.vhd

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
fileName	string length:1024	-	-	Initial	-	-	The name of the file whose contents are sent out as raw data to the output port.
messagesInFile	bool	-	-	Initial	-	false	The flag that is used to turn on and off message mode. See section on Message Mode.
opcode	uchar	-	-	Initial	-	0	The Opcode that all the data in the datafile is sent in streaming mode. The opcode of the ZLM at the end of the file in messaging mode.
messageSize	ulong	-	-	Initial	-	0	The size of the messages in bytes that are created on the output port. The connected worker's buffer needs to be big enough to take the data buffer that is being passed to the worker. See Output Port section.
granularity	ulong	-	-	Initial	-	1	The final message at the end of a file will truncated to be a multiple of this property's value in bytes.
repeat	bool	-	-	Writeable	-	-	The flag used to repeat the data file over and over.
bytesRead	uLongLong	-	-	Volatile	-	-	The number of bytes read from file. Useful for debugging data flow issues.
messagesWritten	uLongLong	-	-	Volatile	-	-	The number of messages read from file. Good property for debugging data flow issues.
suppressEOF	bool	-	-	Initial	-	-	The flag used to enable or disable the Zero Length message that is propagated at the end of the file.
badMessage	bool	-	-	Volatile	-	-	This flag is set by the worker if the worker has a problem getting data from file. An example of this happening is a bad filename.

Worker Properties

file_read.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Spec Property	fileName	string	-	-	Readback	-	-	added Readback
Spec Property	suppressEOF	bool	-	-	Readback	-	-	added Readback
Property	CWD_MAX_LENGTH	ulong	-	-	Paramater	-	512	parameter for max string length for the cwd property
Property	cwd	string length:CWD_MAX_LENGTH	-	-	Volatile	-	-	the current working directory of the application (this is required for the hdl version of this worker and cannot be determined automatically)

file_read.rcc

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Spec Property	messageSize	ulong	-	-	Volatile	-	4096	added Volatile

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
out	true	None	False	-	Data streamed from file

Worker Interfaces

file_read.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	out	32	-	Data streamed from file

Test and Verification

All test benches use this worker as part of the verification process. A unit-test does not exist for this component.